Amazon Rekognition

Amazon Rekognition

Amazon Rekognition: Amazon Rekognition

Table of Contents

Welcome

Amazon Rekognition

This is the API Reference for Amazon Rekognition Image, Amazon Rekognition Custom Labels, Amazon Rekognition Stored Video, Amazon Rekognition Streaming Video. It provides descriptions of actions, data types, common parameters, and common errors.

Amazon Rekognition Image

- AssociateFaces
- CompareFaces
- CreateCollection
- CreateUser
- DeleteCollection
- DeleteFaces
- DeleteUser
- DescribeCollection
- DetectFaces
- DetectLabels
- DetectModerationLabels
- DetectProtectiveEquipment
- DetectText
- DisassociateFaces
- GetCelebrityInfo
- GetMediaAnalysisJob
- IndexFaces
- ListCollections
- ListMediaAnalysisJob
- ListFaces
- ListUsers
- RecognizeCelebrities

- SearchFaces
- SearchFacesByImage
- SearchUsers
- SearchUsersByImage
- StartMediaAnalysisJob

Amazon Rekognition Custom Labels

- CopyProjectVersion
- CreateDataset
- CreateProject
- CreateProjectVersion
- DeleteDataset
- DeleteProject
- DeleteProjectPolicy
- DeleteProjectVersion
- DescribeDataset
- DescribeProjects
- DescribeProjectVersions
- DetectCustomLabels
- DistributeDatasetEntries
- ListDatasetEntries
- ListDatasetLabels
- ListProjectPolicies
- PutProjectPolicy
- StartProjectVersion
- StopProjectVersion
- UpdateDatasetEntries

Amazon Rekognition Video Stored Video

- GetCelebrityRecognition

- GetContentModeration
- GetFaceDetection
- GetFaceSearch
- GetLabelDetection
- GetPersonTracking
- GetSegmentDetection
- GetTextDetection
- StartCelebrityRecognition
- StartContentModeration
- StartFaceDetection
- StartFaceSearch
- StartLabelDetection
- StartPersonTracking
- StartSegmentDetection
- StartTextDetection

Amazon Rekognition Video Streaming Video

- CreateStreamProcessor
- DeleteStreamProcessor
- DescribeStreamProcessor
- ListStreamProcessors
- StartStreamProcessor
- StopStreamProcessor
- UpdateStreamProcessor

Amazon Rekognition Streaming

A real-time video processing service based on Rekognition. This section documents the API operations for Rekognition Face Liveness.

Amazon Rekognition Face Liveness

- CreateFaceLivenessSession
- GetFaceLivenessSessionResults
- StartFaceLivenessSession

Actions

The following actions are supported by Amazon Rekognition:

- AssociateFaces
- CompareFaces
- CopyProjectVersion
- CreateCollection
- CreateDataset
- CreateFaceLivenessSession
- CreateProject
- CreateProjectVersion
- CreateStreamProcessor
- CreateUser
- DeleteCollection
- DeleteDataset
- DeleteFaces
- DeleteProject
- DeleteProjectPolicy
- DeleteProjectVersion
- DeleteStreamProcessor
- DeleteUser
- DescribeCollection
- DescribeDataset
- DescribeProjects
- DescribeProjectVersions
- DescribeStreamProcessor
- DetectCustomLabels
- DetectFaces
- DetectLabels
- DetectModerationLabels

- DetectProtectiveEquipment
- DetectText
- DisassociateFaces
- DistributeDatasetEntries
- GetCelebrityInfo
- GetCelebrityRecognition
- GetContentModeration
- GetFaceDetection
- GetFaceLivenessSessionResults
- GetFaceSearch
- GetLabelDetection
- GetMediaAnalysisJob
- GetPersonTracking
- GetSegmentDetection
- GetTextDetection
- IndexFaces
- ListCollections
- ListDatasetEntries
- ListDatasetLabels
- ListFaces
- ListMediaAnalysisJobs
- ListProjectPolicies
- ListStreamProcessors
- ListTagsForResource
- ListUsers
- PutProjectPolicy
- RecognizeCelebrities
- SearchFaces
- SearchFacesByImage
- SearchUsers

- SearchUsersByImage
- StartCelebrityRecognition
- StartContentModeration
- StartFaceDetection
- StartFaceSearch
- StartLabelDetection
- StartMediaAnalysisJob
- StartPersonTracking
- StartProjectVersion
- StartSegmentDetection
- StartStreamProcessor
- StartTextDetection
- StopProjectVersion
- StopStreamProcessor
- TagResource
- UntagResource
- UpdateDatasetEntries
- UpdateStreamProcessor

The following actions are supported by Amazon Rekognition Streaming:

- StartFaceLivenessSession

Amazon Rekognition

The following actions are supported by Amazon Rekognition:

- AssociateFaces
- CompareFaces
- CopyProjectVersion
- CreateCollection
- CreateDataset

- CreateFaceLivenessSession
- CreateProject
- CreateProjectVersion
- CreateStreamProcessor
- CreateUser
- DeleteCollection
- DeleteDataset
- DeleteFaces
- DeleteProject
- DeleteProjectPolicy
- DeleteProjectVersion
- DeleteStreamProcessor
- DeleteUser
- DescribeCollection
- DescribeDataset
- DescribeProjects
- DescribeProjectVersions
- DescribeStreamProcessor
- DetectCustomLabels
- DetectFaces
- DetectLabels
- DetectModerationLabels
- DetectProtectiveEquipment
- DetectText
- DisassociateFaces
- DistributeDatasetEntries
- GetCelebrityInfo
- GetCelebrityRecognition
- GetContentModeration
- GetFaceDetection

- GetFaceLivenessSessionResults
- GetFaceSearch
- GetLabelDetection
- GetMediaAnalysisJob
- GetPersonTracking
- GetSegmentDetection
- GetTextDetection
- IndexFaces
- ListCollections
- ListDatasetEntries
- ListDatasetLabels
- ListFaces
- ListMediaAnalysisJobs
- ListProjectPolicies
- ListStreamProcessors
- ListTagsForResource
- ListUsers
- PutProjectPolicy
- RecognizeCelebrities
- SearchFaces
- SearchFacesByImage
- SearchUsers
- SearchUsersByImage
- StartCelebrityRecognition
- StartContentModeration
- StartFaceDetection
- StartFaceSearch
- StartLabelDetection
- StartMediaAnalysisJob
- StartPersonTracking

- StartProjectVersion
- StartSegmentDetection
- StartStreamProcessor
- StartTextDetection
- StopProjectVersion
- StopStreamProcessor
- TagResource
- UntagResource
- UpdateDatasetEntries
- UpdateStreamProcessor

AssociateFaces

Service: Amazon Rekognition

Associates one or more faces with an existing UserID. Takes an array of `FaceIds`. Each `FaceId` that are present in the `FaceIds` list is associated with the provided UserID. The number of FaceIds that can be used as input in a single request is limited to 100.

Note that the total number of faces that can be associated with a single `UserID` is also limited to 100. Once a `UserID` has 100 faces associated with it, no additional faces can be added. If more API calls are made after the limit is reached, a `ServiceQuotaExceededException` will result.

The `UserMatchThreshold` parameter specifies the minimum user match confidence required for the face to be associated with a UserID that has at least one `FaceID` already associated. This ensures that the `FaceIds` are associated with the right UserID. The value ranges from 0-100 and default value is 75.

If successful, an array of `AssociatedFace` objects containing the associated `FaceIds` is returned. If a given face is already associated with the given `UserID`, it will be ignored and will not be returned in the response. If a given face is already associated to a different `UserID`, isn't found in the collection, doesn't meet the `UserMatchThreshold`, or there are already 100 faces associated with the `UserID`, it will be returned as part of an array of `UnsuccessfulFaceAssociations`.

The `UserStatus` reflects the status of an operation which updates a UserID representation with a list of given faces. The `UserStatus` can be:

- ACTIVE - All associations or disassociations of FaceID(s) for a UserID are complete.
- CREATED - A UserID has been created, but has no FaceID(s) associated with it.
- UPDATING - A UserID is being updated and there are current associations or disassociations of FaceID(s) taking place.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "CollectionId": "string",
   "FaceIds": [ "string" ],
   "UserId": "string",
   "UserMatchThreshold": number
```

```
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the request to `AssociateFaces`. If you use the same token with multiple `AssociateFaces` requests, the same response is returned. Use ClientRequestToken to prevent the same request from being processed more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

CollectionId

The ID of an existing collection containing the UserID.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

FaceIds

An array of FaceIDs to associate with the UserID.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: Yes

UserId

The ID for the existing UserID.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: Yes

UserMatchThreshold

An optional value specifying the minimum confidence in the UserID match to return. The default value is 75.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Response Syntax

```
{
   "AssociatedFaces": [
      {
         "FaceId": "string"
      }
   ],
   "UnsuccessfulFaceAssociations": [
      {
         "Confidence": number,
         "FaceId": "string",
         "Reasons": [ "string" ],
         "UserId": "string"
      }
   ],
   "UserStatus": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AssociatedFaces

An array of AssociatedFace objects containing FaceIDs that have been successfully associated with the UserID. Returned if the AssociateFaces action is successful.

Type: Array of AssociatedFace objects

Array Members: Minimum number of 0 items. Maximum number of 100 items.

UnsuccessfulFaceAssociations

An array of UnsuccessfulAssociation objects containing FaceIDs that are not successfully associated along with the reasons. Returned if the AssociateFaces action is successful.

Type: Array of UnsuccessfulFaceAssociation objects

Array Members: Minimum number of 0 items. Maximum number of 500 items.

UserStatus

The status of an update made to a UserID. Reflects if the UserID has been updated for every requested change.

Type: String

Valid Values: `ACTIVE | UPDATING | CREATING | CREATED`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ConflictException

A User with the same Id already exists within the collection, or the update or deletion of the User caused an inconsistent state. **

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CompareFaces

Service: Amazon Rekognition

Compares a face in the *source* input image with each of the 100 largest faces detected in the *target* input image.

If the source image contains multiple faces, the service detects the largest face and compares it with each face detected in the target image.

> **Note**
>
> CompareFaces uses machine learning algorithms, which are probabilistic. A false negative is an incorrect prediction that a face in the target image has a low similarity confidence score when compared to the face in the source image. To reduce the probability of false negatives, we recommend that you compare the target image against multiple source images. If you plan to use `CompareFaces` to make a decision that impacts an individual's rights, privacy, or access to services, we recommend that you pass the result to a human for review and further validation before taking action.

You pass the input and target images either as base64-encoded image bytes or as references to images in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes isn't supported. The image must be formatted as a PNG or JPEG file.

In response, the operation returns an array of face matches ordered by similarity score in descending order. For each face match, the response provides a bounding box of the face, facial landmarks, pose details (pitch, roll, and yaw), quality (brightness and sharpness), and confidence value (indicating the level of confidence that the bounding box contains a face). The response also provides a similarity score, which indicates how closely the faces match.

> **Note**
>
> By default, only faces with a similarity score of greater than or equal to 80% are returned in the response. You can change this value by specifying the `SimilarityThreshold` parameter.

`CompareFaces` also returns an array of faces that don't match the source image. For each face, it returns a bounding box, confidence value, landmarks, pose details, and quality. The response also

returns information about the face in the source image, including the bounding box of the face and confidence value.

The `QualityFilter` input parameter allows you to filter out detected faces that don't meet a required quality bar. The quality bar is based on a variety of common use cases. Use `QualityFilter` to set the quality bar by specifying `LOW`, `MEDIUM`, or `HIGH`. If you do not want to filter detected faces, specify `NONE`. The default value is `NONE`.

If the image doesn't contain Exif metadata, `CompareFaces` returns orientation information for the source and target images. Use these values to display the images with the correct image orientation.

If no faces are detected in the source or target images, `CompareFaces` returns an `InvalidParameterException` error.

> **Note**
>
> This is a stateless API operation. That is, data returned by this operation doesn't persist.

For an example, see Comparing faces in images.

This operation requires permissions to perform the `rekognition:CompareFaces` action.

Request Syntax

```
{
   "QualityFilter": "string",
   "SimilarityThreshold": number,
   "SourceImage": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "TargetImage": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
```

```
            "Name": "string",
            "Version": "string"
        }
    }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

QualityFilter

A filter that specifies a quality bar for how much filtering is done to identify faces. Filtered faces aren't compared. If you specify `AUTO`, Amazon Rekognition chooses the quality bar. If you specify `LOW`, `MEDIUM`, or `HIGH`, filtering removes all faces that don’t meet the chosen quality bar. The quality bar is based on a variety of common use cases. Low-quality detections can occur for a number of reasons. Some examples are an object that's misidentified as a face, a face that's too blurry, or a face with a pose that's too extreme to use. If you specify `NONE`, no filtering is performed. The default value is `NONE`.

To use quality filtering, the collection you are using must be associated with version 3 of the face model or higher.

Type: String

Valid Values: `NONE | AUTO | LOW | MEDIUM | HIGH`

Required: No

SimilarityThreshold

The minimum level of confidence in the face matches that a match must meet to be included in the `FaceMatches` array.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

SourceImage

The input image as base64-encoded bytes or an S3 object. If you use the AWS CLI to call Amazon Rekognition operations, passing base64-encoded image bytes is not supported.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

TargetImage

The target image as base64-encoded bytes or an S3 object. If you use the AWS CLI to call Amazon Rekognition operations, passing base64-encoded image bytes is not supported.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

Response Syntax

```
{
   "FaceMatches": [
      {
         "Face": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "Emotions": [
               {
                  "Confidence": number,
                  "Type": "string"
               }
            ],
            "Landmarks": [
```

```
                {
                    "Type": "string",
                    "X": number,
                    "Y": number
                }
            ],
            "Pose": {
                "Pitch": number,
                "Roll": number,
                "Yaw": number
            },
            "Quality": {
                "Brightness": number,
                "Sharpness": number
            },
            "Smile": {
                "Confidence": number,
                "Value": boolean
            }
        },
        "Similarity": number
    }
],
"SourceImageFace": {
    "BoundingBox": {
        "Height": number,
        "Left": number,
        "Top": number,
        "Width": number
    },
    "Confidence": number
},
"SourceImageOrientationCorrection": "string",
"TargetImageOrientationCorrection": "string",
"UnmatchedFaces": [
    {
        "BoundingBox": {
            "Height": number,
            "Left": number,
            "Top": number,
            "Width": number
        },
        "Confidence": number,
        "Emotions": [
```

```
            {
               "Confidence": number,
               "Type": "string"
            }
         ],
         "Landmarks": [
            {
               "Type": "string",
               "X": number,
               "Y": number
            }
         ],
         "Pose": {
            "Pitch": number,
            "Roll": number,
            "Yaw": number
         },
         "Quality": {
            "Brightness": number,
            "Sharpness": number
         },
         "Smile": {
            "Confidence": number,
            "Value": boolean
         }
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaceMatches

An array of faces in the target image that match the source image face. Each `CompareFacesMatch` object provides the bounding box, the confidence level that the bounding box contains a face, and the similarity score for the face in the bounding box and the face in the source image.

Type: Array of CompareFacesMatch objects

SourceImageFace

The face in the source image that was used for comparison.

Type: ComparedSourceImageFace object

SourceImageOrientationCorrection

The value of `SourceImageOrientationCorrection` is always null.

If the input image is in .jpeg format, it might contain exchangeable image file format (Exif) metadata that includes the image's orientation. Amazon Rekognition uses this orientation information to perform image correction. The bounding box coordinates are translated to represent object locations after the orientation information in the Exif metadata is used to correct the image orientation. Images in .png format don't contain Exif metadata.

Amazon Rekognition doesn’t perform image correction for images in .png format and .jpeg images without orientation information in the image Exif metadata. The bounding box coordinates aren't translated and represent the object locations before the image is rotated.

Type: String

Valid Values: `ROTATE_0 | ROTATE_90 | ROTATE_180 | ROTATE_270`

TargetImageOrientationCorrection

The value of `TargetImageOrientationCorrection` is always null.

If the input image is in .jpeg format, it might contain exchangeable image file format (Exif) metadata that includes the image's orientation. Amazon Rekognition uses this orientation information to perform image correction. The bounding box coordinates are translated to represent object locations after the orientation information in the Exif metadata is used to correct the image orientation. Images in .png format don't contain Exif metadata.

Amazon Rekognition doesn’t perform image correction for images in .png format and .jpeg images without orientation information in the image Exif metadata. The bounding box coordinates aren't translated and represent the object locations before the image is rotated.

Type: String

Valid Values: `ROTATE_0 | ROTATE_90 | ROTATE_180 | ROTATE_270`

UnmatchedFaces

An array of faces in the target image that did not match the source image face.

Type: Array of ComparedFace objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CopyProjectVersion

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Copies a version of an Amazon Rekognition Custom Labels model from a source project to a destination project. The source and destination projects can be in different AWS accounts but must be in the same AWS Region. You can't copy a model to another AWS service.

To copy a model version to a different AWS account, you need to create a resource-based policy known as a *project policy*. You attach the project policy to the source project by calling PutProjectPolicy. The project policy gives permission to copy the model version from a trusting AWS account to a trusted account.

For more information about creating and attaching a project policy document, see Attaching a project policy (SDK).

If you are copying a model version to a project in the same AWS account, you don't need to create a project policy.

> **Note**
>
> Copying project versions is supported only for Custom Labels models.
> To copy a model, the destination project, source project, and source model version must already exist.

Copying a model version takes a while to complete. To get the current status, call DescribeProjectVersions and check the value of `Status` in the ProjectVersionDescription object. The copy operation has finished when the value of `Status` is `COPYING_COMPLETED`.

This operation requires permissions to perform the `rekognition:CopyProjectVersion` action.

Request Syntax

```
{
```

```
    "DestinationProjectArn": "string",
    "KmsKeyId": "string",
    "OutputConfig": {
       "S3Bucket": "string",
       "S3KeyPrefix": "string"
    },
    "SourceProjectArn": "string",
    "SourceProjectVersionArn": "string",
    "Tags": {
       "string" : "string"
    },
    "VersionName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DestinationProjectArn

The ARN of the project in the trusted AWS account that you want to copy the model version to.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

KmsKeyId

The identifier for your AWS Key Management Service key (AWS KMS key). You can supply the Amazon Resource Name (ARN) of your KMS key, the ID of your KMS key, an alias for your KMS key, or an alias ARN. The key is used to encrypt training results and manifest files written to the output Amazon S3 bucket (`OutputConfig`).

If you choose to use your own KMS key, you need the following permissions on the KMS key.

- kms:CreateGrant
- kms:DescribeKey

- kms:GenerateDataKey
- kms:Decrypt

If you don't specify a value for `KmsKeyId`, images copied into the service are encrypted using a key that AWS owns and manages.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

Required: No

OutputConfig

The S3 bucket and folder location where the training output for the source model version is placed.

Type: OutputConfig object

Required: Yes

SourceProjectArn

The ARN of the source project in the trusting AWS account.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

SourceProjectVersionArn

The ARN of the model version in the source project that you want to copy to a destination project.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Tags

The key-value tags to assign to the model version.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Key Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

VersionName

A name for the version of the model that's copied to the destination project.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Response Syntax

```
{
   "ProjectVersionArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ProjectVersionArn

The ARN of the copied model version in the destination project.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateCollection

Service: Amazon Rekognition

Creates a collection in an AWS Region. You can add faces to the collection using the IndexFaces operation.

For example, you might create collections, one for each of your application users. A user can then index faces using the `IndexFaces` operation and persist results in a specific collection. Then, a user can search the collection for faces in the user-specific container.

When you create a collection, it is associated with the latest version of the face model version.

> **Note**
>
> Collection names are case-sensitive.

This operation requires permissions to perform the `rekognition:CreateCollection` action. If you want to tag your collection, you also require permission to perform the `rekognition:TagResource` operation.

Request Syntax

```
{
   "CollectionId": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

ID for the collection that you are creating.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Tags

A set of tags (key-value pairs) that you want to attach to the collection.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Key Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

Response Syntax

```
{
   "CollectionArn": "string",
   "FaceModelVersion": "string",
   "StatusCode": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CollectionArn

Amazon Resource Name (ARN) of the collection. You can use this to manage permissions on your resources.

Type: String

FaceModelVersion

Version number of the face detection model associated with the collection you are creating.

Type: String

StatusCode

HTTP status code indicating the result of the operation.

Type: Integer

Valid Range: Minimum value of 0.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceAlreadyExistsException

A resource with the specified ID already exists.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateDataset

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Creates a new Amazon Rekognition Custom Labels dataset. You can create a dataset by using an Amazon Sagemaker format manifest file or by copying an existing Amazon Rekognition Custom Labels dataset.

To create a training dataset for a project, specify `TRAIN` for the value of `DatasetType`. To create the test dataset for a project, specify `TEST` for the value of `DatasetType`.

The response from `CreateDataset` is the Amazon Resource Name (ARN) for the dataset. Creating a dataset takes a while to complete. Use DescribeDataset to check the current status. The dataset created successfully if the value of `Status` is `CREATE_COMPLETE`.

To check if any non-terminal errors occurred, call ListDatasetEntries and check for the presence of `errors` lists in the JSON Lines.

Dataset creation fails if a terminal error occurs (`Status` = `CREATE_FAILED`). Currently, you can't access the terminal error information.

For more information, see Creating datasets.

This operation requires permissions to perform the `rekognition:CreateDataset` action. If you want to copy an existing dataset, you also require permission to perform the `rekognition:ListDatasetEntries` action.

Request Syntax

```
{
   "DatasetSource": {
      "DatasetArn": "string",
      "GroundTruthManifest": {
         "S3Object": {
            "Bucket": "string",
            "Name": "string",
```

```
            "Version": "string"
         }
      }
   },
   "DatasetType": "string",
   "ProjectArn": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DatasetSource

The source files for the dataset. You can specify the ARN of an existing dataset or specify the Amazon S3 bucket location of an Amazon Sagemaker format manifest file. If you don't specify `datasetSource`, an empty dataset is created. To add labeled images to the dataset, You can use the console or call UpdateDatasetEntries.

Type: DatasetSource object

Required: No

DatasetType

The type of the dataset. Specify `TRAIN` to create a training dataset. Specify `TEST` to create a test dataset.

Type: String

Valid Values: `TRAIN | TEST`

Required: Yes

ProjectArn

The ARN of the Amazon Rekognition Custom Labels project to which you want to asssign the dataset.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Tags

A set of tags (key-value pairs) that you want to attach to the dataset.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Key Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

Response Syntax

```
{
   "DatasetArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DatasetArn

The ARN of the created Amazon Rekognition Custom Labels dataset.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceAlreadyExistsException

A resource with the specified ID already exists.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateFaceLivenessSession

Service: Amazon Rekognition

This API operation initiates a Face Liveness session. It returns a `SessionId`, which you can use to start streaming Face Liveness video and get the results for a Face Liveness session.

You can use the `OutputConfig` option in the Settings parameter to provide an Amazon S3 bucket location. The Amazon S3 bucket stores reference images and audit images. If no Amazon S3 bucket is defined, raw bytes are sent instead.

You can use `AuditImagesLimit` to limit the number of audit images returned when `GetFaceLivenessSessionResults` is called. This number is between 0 and 4. By default, it is set to 0. The limit is best effort and based on the duration of the selfie-video.

You can use the `ChallengePreferences` option in the `Settings` parameter to choose between the 'FaceMovementAndLightChallenge' or FaceMovementChallenge' settings. See the Shared Responsibility page for details on guidance for which setting to choose between these two settings depending on your use case and preferences. This parameter is optional and if no parameter is provided, then the 'FaceMovementAndLightChallenge' settings is applied by default.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "KmsKeyId": "string",
   "Settings": {
      "AuditImagesLimit": number,
      "ChallengePreferences": [
         {
            "Type": "string",
            "Versions": {
               "Maximum": "string",
               "Minimum": "string"
            }
         }
      ],
      "OutputConfig": {
         "S3Bucket": "string",
         "S3KeyPrefix": "string"
      }
   }
```

```
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token is used to recognize the Face Liveness request. If the same token is used with multiple `CreateFaceLivenessSession` requests, the same session is returned. This token is employed to avoid unintentionally creating the same session multiple times.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

KmsKeyId

The identifier for your AWS Key Management Service key (AWS KMS key). Used to encrypt audit images and reference images.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

Required: No

Settings

A session settings object. It contains settings for the operation to be performed. For Face Liveness, it accepts `OutputConfig` and `AuditImagesLimit`.

Type: CreateFaceLivenessSessionRequestSettings object

Required: No

Response Syntax

```
{
   "SessionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

SessionId

A unique 128-bit UUID identifying a Face Liveness session. A new sessionID must be used for every Face Liveness check. If a given sessionID is used for subsequent Face Liveness checks, the checks will fail. Additionally, a SessionId expires 3 minutes after it's sent, making all Liveness data associated with the session (e.g., sessionID, reference image, audit images, etc.) unavailable.

Type: String

Length Constraints: Fixed length of 36.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateProject

Service: Amazon Rekognition

Creates a new Amazon Rekognition project. A project is a group of resources (datasets, model versions) that you use to create and manage a Amazon Rekognition Custom Labels Model or custom adapter. You can specify a feature to create the project with, if no feature is specified then Custom Labels is used by default. For adapters, you can also choose whether or not to have the project auto update by using the AutoUpdate argument. This operation requires permissions to perform the `rekognition:CreateProject` action.

Request Syntax

```
{
   "AutoUpdate": "string",
   "Feature": "string",
   "ProjectName": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

AutoUpdate

Specifies whether automatic retraining should be attempted for the versions of the project. Automatic retraining is done as a best effort. Required argument for Content Moderation. Applicable only to adapters.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

Feature

Specifies feature that is being customized. If no value is provided CUSTOM_LABELS is used as a default.

Type: String

Valid Values: `CONTENT_MODERATION | CUSTOM_LABELS`

Required: No

ProjectName

The name of the project to create.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Tags

A set of tags (key-value pairs) that you want to attach to the project.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Key Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

Response Syntax

```
{
   "ProjectArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ProjectArn

The Amazon Resource Name (ARN) of the new project. You can use the ARN to configure IAM access to the project.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateProjectVersion

Service: Amazon Rekognition

Creates a new version of Amazon Rekognition project (like a Custom Labels model or a custom adapter) and begins training. Models and adapters are managed as part of a Rekognition project. The response from `CreateProjectVersion` is an Amazon Resource Name (ARN) for the project version.

The FeatureConfig operation argument allows you to configure specific model or adapter settings. You can provide a description to the project version by using the VersionDescription argment. Training can take a while to complete. You can get the current status by calling DescribeProjectVersions. Training completed successfully if the value of the `Status` field is `TRAINING_COMPLETED`. Once training has successfully completed, call DescribeProjectVersions to get the training results and evaluate the model.

This operation requires permissions to perform the `rekognition:CreateProjectVersion` action.

> **Note**
>
> *The following applies only to projects with Amazon Rekognition Custom Labels as the chosen feature:*
>
> For Custom Labels projects, training uses the training and test datasets associated with the project. For more information, see Creating training and test datasets. After evaluating the model, you start the model by calling StartProjectVersion. If training fails, see Debugging a failed model training. For more information, see Improving a trained Amazon Rekognition Custom Labels model.
>
> You can train a model in a project that doesn't have associated datasets by specifying manifest files in the `TrainingData` and `TestingData` fields.
>
> If you open the console after training a model with manifest files, Amazon Rekognition Custom Labels creates the datasets for you using the most recent manifest files. You can no longer train a model version for the project by specifying manifest files.
>
> Instead of training with a project without associated datasets, we recommend that you use the manifest files to create training and test datasets for the project.

Request Syntax

```
{
   "FeatureConfig": {
      "ContentModeration": {
         "ConfidenceThreshold": number
      }
   },
   "KmsKeyId": "string",
   "OutputConfig": {
      "S3Bucket": "string",
      "S3KeyPrefix": "string"
   },
   "ProjectArn": "string",
   "Tags": {
      "string" : "string"
   },
   "TestingData": {
      "Assets": [
         {
            "GroundTruthManifest": {
               "S3Object": {
                  "Bucket": "string",
                  "Name": "string",
                  "Version": "string"
               }
            }
         }
      ],
      "AutoCreate": boolean
   },
   "TrainingData": {
      "Assets": [
         {
            "GroundTruthManifest": {
               "S3Object": {
                  "Bucket": "string",
                  "Name": "string",
                  "Version": "string"
               }
            }
         }
      ]
```

```
    },
    "VersionDescription": "string",
    "VersionName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

FeatureConfig

Feature-specific configuration of the training job. If the job configuration does not match the feature type associated with the project, an InvalidParameterException is returned.

Type: CustomizationFeatureConfig object

Required: No

KmsKeyId

The identifier for your AWS Key Management Service key (AWS KMS key). You can supply the Amazon Resource Name (ARN) of your KMS key, the ID of your KMS key, an alias for your KMS key, or an alias ARN. The key is used to encrypt training images, test images, and manifest files copied into the service for the project version. Your source images are unaffected. The key is also used to encrypt training results and manifest files written to the output Amazon S3 bucket (`OutputConfig`).

If you choose to use your own KMS key, you need the following permissions on the KMS key.

- kms:CreateGrant
- kms:DescribeKey
- kms:GenerateDataKey
- kms:Decrypt

If you don't specify a value for `KmsKeyId`, images copied into the service are encrypted using a key that AWS owns and manages.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

Required: No

OutputConfig

The Amazon S3 bucket location to store the results of training. The bucket can be any S3 bucket in your AWS account. You need `s3:PutObject` permission on the bucket.

Type: OutputConfig object

Required: Yes

ProjectArn

The ARN of the Amazon Rekognition project that will manage the project version you want to train.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Tags

A set of tags (key-value pairs) that you want to attach to the project version.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Key Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

TestingData

Specifies an external manifest that the service uses to test the project version. If you specify `TestingData` you must also specify `TrainingData`. The project must not have any associated datasets.

Type: TestingData object

Required: No

TrainingData

Specifies an external manifest that the services uses to train the project version. If you specify `TrainingData` you must also specify `TestingData`. The project must not have any associated datasets.

Type: TrainingData object

Required: No

VersionDescription

A description applied to the project version being created.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9-_. ()':,;?]+`

Required: No

VersionName

A name for the version of the project version. This value must be unique.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Response Syntax

```
{
   "ProjectVersionArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ProjectVersionArn

The ARN of the model or the project version that was created. Use `DescribeProjectVersion` to get the current status of the training operation.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateStreamProcessor

Service: Amazon Rekognition

Creates an Amazon Rekognition stream processor that you can use to detect and recognize faces or to detect labels in a streaming video.

Amazon Rekognition Video is a consumer of live video from Amazon Kinesis Video Streams. There are two different settings for stream processors in Amazon Rekognition: detecting faces and detecting labels.

- If you are creating a stream processor for detecting faces, you provide as input a Kinesis video stream (`Input`) and a Kinesis data stream (`Output`) stream for receiving the output. You must use the `FaceSearch` option in `Settings`, specifying the collection that contains the faces you want to recognize. After you have finished analyzing a streaming video, use StopStreamProcessor to stop processing.
- If you are creating a stream processor to detect labels, you provide as input a Kinesis video stream (`Input`), Amazon S3 bucket information (`Output`), and an Amazon SNS topic ARN (`NotificationChannel`). You can also provide a KMS key ID to encrypt the data sent to your Amazon S3 bucket. You specify what you want to detect by using the `ConnectedHome` option in settings, and selecting one of the following: `PERSON`, `PET`, `PACKAGE`, `ALL` You can also specify where in the frame you want Amazon Rekognition to monitor with `RegionsOfInterest`. When you run the StartStreamProcessor operation on a label detection stream processor, you input start and stop information to determine the length of the processing time.

Use `Name` to assign an identifier for the stream processor. You use `Name` to manage the stream processor. For example, you can start processing the source video by calling StartStreamProcessor with the `Name` field.

This operation requires permissions to perform the `rekognition:CreateStreamProcessor` action. If you want to tag your stream processor, you also require permission to perform the `rekognition:TagResource` operation.

Request Syntax

```
{
   "DataSharingPreference": {
      "OptIn": boolean
   },
```

```
    "Input": {
       "KinesisVideoStream": {
          "Arn": "string"
       }
    },
    "KmsKeyId": "string",
    "Name": "string",
    "NotificationChannel": {
       "SNSTopicArn": "string"
    },
    "Output": {
       "KinesisDataStream": {
          "Arn": "string"
       },
       "S3Destination": {
          "Bucket": "string",
          "KeyPrefix": "string"
       }
    },
    "RegionsOfInterest": [
       {
          "BoundingBox": {
             "Height": number,
             "Left": number,
             "Top": number,
             "Width": number
          },
          "Polygon": [
             {
                "X": number,
                "Y": number
             }
          ]
       }
    ],
    "RoleArn": "string",
    "Settings": {
       "ConnectedHome": {
          "Labels": [ "string" ],
          "MinConfidence": number
       },
       "FaceSearch": {
          "CollectionId": "string",
          "FaceMatchThreshold": number
```

```
        }
    },
    "Tags": {
        "string" : "string"
    }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DataSharingPreference

Shows whether you are sharing data with Rekognition to improve model performance. You can choose this option at the account level or on a per-stream basis. Note that if you opt out at the account level this setting is ignored on individual streams.

Type: StreamProcessorDataSharingPreference object

Required: No

Input

Kinesis video stream stream that provides the source streaming video. If you are using the AWS CLI, the parameter name is `StreamProcessorInput`. This is required for both face search and label detection stream processors.

Type: StreamProcessorInput object

Required: Yes

KmsKeyId

The identifier for your AWS Key Management Service key (AWS KMS key). This is an optional parameter for label detection stream processors and should not be used to create a face search stream processor. You can supply the Amazon Resource Name (ARN) of your KMS key, the ID of your KMS key, an alias for your KMS key, or an alias ARN. The key is used to encrypt results and data published to your Amazon S3 bucket, which includes image frames and hero images. Your source images are unaffected.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

Required: No

Name

An identifier you assign to the stream processor. You can use `Name` to manage the stream processor. For example, you can get the current status of the stream processor by calling DescribeStreamProcessor. `Name` is idempotent. This is required for both face search and label detection stream processors.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

NotificationChannel

The Amazon Simple Notification Service topic to which Amazon Rekognition publishes the object detection results and completion status of a video analysis operation.

Amazon Rekognition publishes a notification the first time an object of interest or a person is detected in the video stream. For example, if Amazon Rekognition detects a person at second 2, a pet at second 4, and a person again at second 5, Amazon Rekognition sends 2 object class detected notifications, one for a person at second 2 and one for a pet at second 4.

Amazon Rekognition also publishes an an end-of-session notification with a summary when the stream processing session is complete.

Type: StreamProcessorNotificationChannel object

Required: No

Output

Kinesis data stream stream or Amazon S3 bucket location to which Amazon Rekognition Video puts the analysis results. If you are using the AWS CLI, the parameter name is `StreamProcessorOutput`. This must be a S3Destination of an Amazon S3 bucket that you

own for a label detection stream processor or a Kinesis data stream ARN for a face search stream processor.

Type: StreamProcessorOutput object

Required: Yes

RegionsOfInterest

Specifies locations in the frames where Amazon Rekognition checks for objects or people. You can specify up to 10 regions of interest, and each region has either a polygon or a bounding box. This is an optional parameter for label detection stream processors and should not be used to create a face search stream processor.

Type: Array of RegionOfInterest objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

RoleArn

The Amazon Resource Number (ARN) of the IAM role that allows access to the stream processor. The IAM role provides Rekognition read permissions for a Kinesis stream. It also provides write permissions to an Amazon S3 bucket and Amazon Simple Notification Service topic for a label detection stream processor. This is required for both face search and label detection stream processors.

Type: String

Pattern: `arn:aws:iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Required: Yes

Settings

Input parameters used in a streaming video analyzed by a stream processor. You can use `FaceSearch` to recognize faces in a streaming video, or you can use `ConnectedHome` to detect labels.

Type: StreamProcessorSettings object

Required: Yes

Tags

A set of tags (key-value pairs) that you want to attach to the stream processor.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Key Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: No

Response Syntax

```
{
   "StreamProcessorArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

StreamProcessorArn

Amazon Resource Number for the newly created stream processor.

Type: String

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:streamprocessor\/.+$)`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateUser

Service: Amazon Rekognition

Creates a new User within a collection specified by `CollectionId`. Takes `UserId` as a parameter, which is a user provided ID which should be unique within the collection. The provided `UserId` will alias the system generated UUID to make the `UserId` more user friendly.

Uses a `ClientToken`, an idempotency token that ensures a call to `CreateUser` completes only once. If the value is not supplied, the AWS SDK generates an idempotency token for the requests. This prevents retries after a network error results from making multiple `CreateUser` calls.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "CollectionId": "string",
   "UserId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the request to `CreateUser`. If you use the same token with multiple `CreateUser` requests, the same response is returned. Use ClientRequestToken to prevent the same request from being processed more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

CollectionId

The ID of an existing collection to which the new UserID needs to be created.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

UserId

ID for the UserID to be created. This ID needs to be unique within the collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ConflictException

A User with the same Id already exists within the collection, or the update or deletion of the User caused an inconsistent state. **

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteCollection

Service: Amazon Rekognition

Deletes the specified collection. Note that this operation removes all faces in the collection. For an example, see Deleting a collection.

This operation requires permissions to perform the `rekognition:DeleteCollection` action.

Request Syntax

```
{
   "CollectionId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

ID of the collection to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Response Syntax

```
{
   "StatusCode": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

StatusCode

HTTP status code that indicates the result of the operation.

Type: Integer

Valid Range: Minimum value of 0.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteDataset

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Deletes an existing Amazon Rekognition Custom Labels dataset. Deleting a dataset might take while. Use DescribeDataset to check the current status. The dataset is still deleting if the value of `Status` is `DELETE_IN_PROGRESS`. If you try to access the dataset after it is deleted, you get a `ResourceNotFoundException` exception.

You can't delete a dataset while it is creating (`Status` = `CREATE_IN_PROGRESS`) or if the dataset is updating (`Status` = `UPDATE_IN_PROGRESS`).

This operation requires permissions to perform the `rekognition:DeleteDataset` action.

Request Syntax

```
{
   "DatasetArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DatasetArn

The ARN of the Amazon Rekognition Custom Labels dataset that you want to delete.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteFaces

Service: Amazon Rekognition

Deletes faces from a collection. You specify a collection ID and an array of face IDs to remove from the collection.

This operation requires permissions to perform the `rekognition:DeleteFaces` action.

Request Syntax

```
{
   "CollectionId": "string",
   "FaceIds": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

Collection from which to remove the specific faces.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

FaceIds

An array of face IDs to delete.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4096 items.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: Yes

Response Syntax

```
{
   "DeletedFaces": [ "string" ],
   "UnsuccessfulFaceDeletions": [
      {
         "FaceId": "string",
         "Reasons": [ "string" ],
         "UserId": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DeletedFaces

An array of strings (face IDs) of the faces that were deleted.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4096 items.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

UnsuccessfulFaceDeletions

An array of any faces that weren't deleted.

Type: Array of UnsuccessfulFaceDeletion objects

Array Members: Minimum number of 0 items. Maximum number of 4096 items.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteProject

Service: Amazon Rekognition

Deletes a Amazon Rekognition project. To delete a project you must first delete all models or adapters associated with the project. To delete a model or adapter, see DeleteProjectVersion.

`DeleteProject` is an asynchronous operation. To check if the project is deleted, call DescribeProjects. The project is deleted when the project no longer appears in the response. Be aware that deleting a given project will also delete any `ProjectPolicies` associated with that project.

This operation requires permissions to perform the `rekognition:DeleteProject` action.

Request Syntax

```
{
   "ProjectArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ProjectArn

The Amazon Resource Name (ARN) of the project that you want to delete.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Response Syntax

```
{
```

```
    "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Status

The current status of the delete project operation.

Type: String

Valid Values: `CREATING | CREATED | DELETING`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteProjectPolicy

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Deletes an existing project policy.

To get a list of project policies attached to a project, call ListProjectPolicies. To attach a project policy to a project, call PutProjectPolicy.

This operation requires permissions to perform the `rekognition:DeleteProjectPolicy` action.

Request Syntax

```
{
   "PolicyName": "string",
   "PolicyRevisionId": "string",
   "ProjectArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

PolicyName

The name of the policy that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

PolicyRevisionId

The ID of the project policy revision that you want to delete.

Type: String

Length Constraints: Maximum length of 64.

Pattern: `[0-9A-Fa-f]+`

Required: No

ProjectArn

The Amazon Resource Name (ARN) of the project that the project policy you want to delete is attached to.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidPolicyRevisionIdException

The supplied revision id for the project policy is invalid.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteProjectVersion

Service: Amazon Rekognition

Deletes a Rekognition project model or project version, like a Amazon Rekognition Custom Labels model or a custom adapter.

You can't delete a project version if it is running or if it is training. To check the status of a project version, use the Status field returned from DescribeProjectVersions. To stop a project version call StopProjectVersion. If the project version is training, wait until it finishes.

This operation requires permissions to perform the `rekognition:DeleteProjectVersion` action.

Request Syntax

```
{
   "ProjectVersionArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ProjectVersionArn

The Amazon Resource Name (ARN) of the project version that you want to delete.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Response Syntax

```
{
```

```
    "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Status

The status of the deletion operation.

Type: String

Valid Values: `TRAINING_IN_PROGRESS | TRAINING_COMPLETED | TRAINING_FAILED | STARTING | RUNNING | FAILED | STOPPING | STOPPED | DELETING | COPYING_IN_PROGRESS | COPYING_COMPLETED | COPYING_FAILED | DEPRECATED | EXPIRED`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteStreamProcessor

Service: Amazon Rekognition

Deletes the stream processor identified by `Name`. You assign the value for `Name` when you create the stream processor with CreateStreamProcessor. You might not be able to use the same name for a stream processor for a few seconds after calling `DeleteStreamProcessor`.

Request Syntax

```
{
   "Name": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Name

The name of the stream processor you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteUser

Service: Amazon Rekognition

Deletes the specified UserID within the collection. Faces that are associated with the UserID are disassociated from the UserID before deleting the specified UserID. If the specified `Collection` or `UserID` is already deleted or not found, a `ResourceNotFoundException` will be thrown. If the action is successful with a 200 response, an empty HTTP body is returned.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "CollectionId": "string",
   "UserId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the request to `DeleteUser`. If you use the same token with multiple `DeleteUser` requests, the same response is returned. Use ClientRequestToken to prevent the same request from being processed more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

CollectionId

The ID of an existing collection from which the UserID needs to be deleted.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

UserId

ID for the UserID to be deleted.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ConflictException

A User with the same Id already exists within the collection, or the update or deletion of the User caused an inconsistent state. **

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeCollection

Service: Amazon Rekognition

Describes the specified collection. You can use `DescribeCollection` to get information, such as the number of faces indexed into a collection and the version of the model used by the collection for face detection.

For more information, see Describing a collection.

Request Syntax

```
{
   "CollectionId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

The ID of the collection to describe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Response Syntax

```
{
   "CollectionARN": "string",
   "CreationTimestamp": number,
   "FaceCount": number,
   "FaceModelVersion": "string",
   "UserCount": number
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CollectionARN

The Amazon Resource Name (ARN) of the collection.

Type: String

CreationTimestamp

The number of milliseconds since the Unix epoch time until the creation of the collection. The Unix epoch time is 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970.

Type: Timestamp

FaceCount

The number of faces that are indexed into the collection. To index faces into a collection, use IndexFaces.

Type: Long

Valid Range: Minimum value of 0.

FaceModelVersion

The version of the face model that's used by the collection for face detection.

For more information, see Model versioning.

Type: String

UserCount

The number of UserIDs assigned to the specified colleciton.

Type: Long

Valid Range: Minimum value of 0.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeDataset

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Describes an Amazon Rekognition Custom Labels dataset. You can get information such as the current status of a dataset and statistics about the images and labels in a dataset.

This operation requires permissions to perform the `rekognition:DescribeDataset` action.

Request Syntax

```
{
   "DatasetArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DatasetArn

The Amazon Resource Name (ARN) of the dataset that you want to describe.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Required: Yes

Response Syntax

```
{
```

```
    "DatasetDescription": {
       "CreationTimestamp": number,
       "DatasetStats": {
          "ErrorEntries": number,
          "LabeledEntries": number,
          "TotalEntries": number,
          "TotalLabels": number
       },
       "LastUpdatedTimestamp": number,
       "Status": "string",
       "StatusMessage": "string",
       "StatusMessageCode": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DatasetDescription

The description for the dataset.

Type: DatasetDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

DescribeProjects

Service: Amazon Rekognition

Gets information about your Rekognition projects.

This operation requires permissions to perform the `rekognition:DescribeProjects` action.

Request Syntax

```
{
   "Features": [ "string" ],
   "MaxResults": number,
   "NextToken": "string",
   "ProjectNames": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Features

Specifies the type of customization to filter projects by. If no value is specified, CUSTOM_LABELS is used as a default.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Valid Values: `CONTENT_MODERATION | CUSTOM_LABELS`

Required: No

MaxResults

The maximum number of results to return per paginated call. The largest value you can specify is 100. If you specify a value greater than 100, a ValidationException error occurs. The default value is 100.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken

If the previous response was incomplete (because there is more results to retrieve), Rekognition returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

ProjectNames

A list of the projects that you want Rekognition to describe. If you don't specify a value, the response includes descriptions for all the projects in your AWS account.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "ProjectDescriptions": [
      {
         "AutoUpdate": "string",
         "CreationTimestamp": number,
         "Datasets": [
            {
               "CreationTimestamp": number,
               "DatasetArn": "string",
               "DatasetType": "string",
```

```
                "Status": "string",
                "StatusMessage": "string",
                "StatusMessageCode": "string"
            }
        ],
        "Feature": "string",
        "ProjectArn": "string",
        "Status": "string"
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

If the previous response was incomplete (because there is more results to retrieve), Amazon Rekognition returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

ProjectDescriptions

A list of project descriptions. The list is sorted by the date and time the projects are created.

Type: Array of ProjectDescription objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeProjectVersions

Service: Amazon Rekognition

Lists and describes the versions of an Amazon Rekognition project. You can specify up to 10 model or adapter versions in `ProjectVersionArns`. If you don't specify a value, descriptions for all model/adapter versions in the project are returned.

This operation requires permissions to perform the `rekognition:DescribeProjectVersions` action.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ProjectArn": "string",
   "VersionNames": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxResults

The maximum number of results to return per paginated call. The largest value you can specify is 100. If you specify a value greater than 100, a ValidationException error occurs. The default value is 100.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken

If the previous response was incomplete (because there is more results to retrieve), Amazon Rekognition returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

ProjectArn

The Amazon Resource Name (ARN) of the project that contains the model/adapter you want to describe.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

VersionNames

A list of model or project version names that you want to describe. You can add up to 10 model or project version names to the list. If you don't specify a value, all project version descriptions are returned. A version name is part of a project version ARN. For example, `my-model.2020-01-21T09.10.15` is the version name in the following ARN. `arn:aws:rekognition:us-east-1:123456789012:project/getting-started/version/`*`my-model.2020-01-21T09.10.15`*`/1234567890123`.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "ProjectVersionDescriptions": [
```

```
{
   "BaseModelVersion": "string",
   "BillableTrainingTimeInSeconds": number,
   "CreationTimestamp": number,
   "EvaluationResult": {
      "F1Score": number,
      "Summary": {
         "S3Object": {
            "Bucket": "string",
            "Name": "string",
            "Version": "string"
         }
      }
   },
   "Feature": "string",
   "FeatureConfig": {
      "ContentModeration": {
         "ConfidenceThreshold": number
      }
   },
   "KmsKeyId": "string",
   "ManifestSummary": {
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "MaxInferenceUnits": number,
   "MinInferenceUnits": number,
   "OutputConfig": {
      "S3Bucket": "string",
      "S3KeyPrefix": "string"
   },
   "ProjectVersionArn": "string",
   "SourceProjectVersionArn": "string",
   "Status": "string",
   "StatusMessage": "string",
   "TestingDataResult": {
      "Input": {
         "Assets": [
            {
               "GroundTruthManifest": {
                  "S3Object": {
```

```
                        "Bucket": "string",
                        "Name": "string",
                        "Version": "string"
                    }
                }
            }
        ],
        "AutoCreate": boolean
    },
    "Output": {
        "Assets": [
            {
                "GroundTruthManifest": {
                    "S3Object": {
                        "Bucket": "string",
                        "Name": "string",
                        "Version": "string"
                    }
                }
            }
        ],
        "AutoCreate": boolean
    },
    "Validation": {
        "Assets": [
            {
                "GroundTruthManifest": {
                    "S3Object": {
                        "Bucket": "string",
                        "Name": "string",
                        "Version": "string"
                    }
                }
            }
        ]
    }
},
"TrainingDataResult": {
    "Input": {
        "Assets": [
            {
                "GroundTruthManifest": {
                    "S3Object": {
                        "Bucket": "string",
```

```
                            "Name": "string",
                            "Version": "string"
                         }
                      }
                   }
                ]
             },
             "Output": {
                "Assets": [
                   {
                      "GroundTruthManifest": {
                         "S3Object": {
                            "Bucket": "string",
                            "Name": "string",
                            "Version": "string"
                         }
                      }
                   }
                ]
             },
             "Validation": {
                "Assets": [
                   {
                      "GroundTruthManifest": {
                         "S3Object": {
                            "Bucket": "string",
                            "Name": "string",
                            "Version": "string"
                         }
                      }
                   }
                ]
             }
          },
          "TrainingEndTimestamp": number,
          "VersionDescription": "string"
       }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

If the previous response was incomplete (because there is more results to retrieve), Amazon Rekognition returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

ProjectVersionDescriptions

A list of project version descriptions. The list is sorted by the creation date and time of the project versions, latest to earliest.

Type: Array of ProjectVersionDescription objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeStreamProcessor

Service: Amazon Rekognition

Provides information about a stream processor created by CreateStreamProcessor. You can get information about the input and output streams, the input parameters for the face recognition being performed, and the current status of the stream processor.

Request Syntax

```
{
   "Name": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Name

Name of the stream processor for which you want information.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Response Syntax

```
{
   "CreationTimestamp": number,
   "DataSharingPreference": {
      "OptIn": boolean
   },
   "Input": {
      "KinesisVideoStream": {
         "Arn": "string"
      }
```

```
    },
    "KmsKeyId": "string",
    "LastUpdateTimestamp": number,
    "Name": "string",
    "NotificationChannel": {
       "SNSTopicArn": "string"
    },
    "Output": {
       "KinesisDataStream": {
          "Arn": "string"
       },
       "S3Destination": {
          "Bucket": "string",
          "KeyPrefix": "string"
       }
    },
    "RegionsOfInterest": [
       {
          "BoundingBox": {
             "Height": number,
             "Left": number,
             "Top": number,
             "Width": number
          },
          "Polygon": [
             {
                "X": number,
                "Y": number
             }
          ]
       }
    ],
    "RoleArn": "string",
    "Settings": {
       "ConnectedHome": {
          "Labels": [ "string" ],
          "MinConfidence": number
       },
       "FaceSearch": {
          "CollectionId": "string",
          "FaceMatchThreshold": number
       }
    },
    "Status": "string",
```

```
    "StatusMessage": "string",
    "StreamProcessorArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CreationTimestamp

Date and time the stream processor was created

Type: Timestamp

DataSharingPreference

Shows whether you are sharing data with Rekognition to improve model performance. You can choose this option at the account level or on a per-stream basis. Note that if you opt out at the account level this setting is ignored on individual streams.

Type: StreamProcessorDataSharingPreference object

Input

Kinesis video stream that provides the source streaming video.

Type: StreamProcessorInput object

KmsKeyId

The identifier for your AWS Key Management Service key (AWS KMS key). This is an optional parameter for label detection stream processors.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

LastUpdateTimestamp

The time, in Unix format, the stream processor was last updated. For example, when the stream processor moves from a running state to a failed state, or when the user starts or stops the stream processor.

Type: Timestamp

Name

Name of the stream processor.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

NotificationChannel

The Amazon Simple Notification Service topic to which Amazon Rekognition publishes the object detection results and completion status of a video analysis operation.

Amazon Rekognition publishes a notification the first time an object of interest or a person is detected in the video stream. For example, if Amazon Rekognition detects a person at second 2, a pet at second 4, and a person again at second 5, Amazon Rekognition sends 2 object class detected notifications, one for a person at second 2 and one for a pet at second 4.

Amazon Rekognition also publishes an an end-of-session notification with a summary when the stream processing session is complete.

Type: StreamProcessorNotificationChannel object

Output

Kinesis data stream to which Amazon Rekognition Video puts the analysis results.

Type: StreamProcessorOutput object

RegionsOfInterest

Specifies locations in the frames where Amazon Rekognition checks for objects or people. This is an optional parameter for label detection stream processors.

Type: Array of RegionOfInterest objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

RoleArn

ARN of the IAM role that allows access to the stream processor.

Type: String

Pattern: `arn:aws:iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Settings

Input parameters used in a streaming video analyzed by a stream processor. You can use `FaceSearch` to recognize faces in a streaming video, or you can use `ConnectedHome` to detect labels.

Type: StreamProcessorSettings object

Status

Current status of the stream processor.

Type: String

Valid Values: `STOPPED | STARTING | RUNNING | FAILED | STOPPING | UPDATING`

StatusMessage

Detailed status message about the stream processor.

Type: String

StreamProcessorArn

ARN of the stream processor.

Type: String

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:streamprocessor\/.+$)`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

DetectCustomLabels

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Detects custom labels in a supplied image by using an Amazon Rekognition Custom Labels model.

You specify which version of a model version to use by using the `ProjectVersionArn` input parameter.

You pass the input image as base64-encoded image bytes or as a reference to an image in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes is not supported. The image must be either a PNG or JPEG formatted file.

For each object that the model version detects on an image, the API returns a (`CustomLabel`) object in an array (`CustomLabels`). Each `CustomLabel` object provides the label name (`Name`), the level of confidence that the image contains the object (`Confidence`), and object location information, if it exists, for the label on the image (`Geometry`).

To filter labels that are returned, specify a value for `MinConfidence`. `DetectCustomLabelsLabels` only returns labels with a confidence that's higher than the specified value. The value of `MinConfidence` maps to the assumed threshold values created during training. For more information, see Assumed Threshold. Amazon Rekognition Custom Labels expresses an assumed threshold as a floating point value between 0-1. The range of `MinConfidence` normalizes the assumed threshold to a percentage value (0-100). Confidence responses from `DetectCustomLabels` are also returned as a percentage. You can use `MinConfidence` to change the precision and recall or your model. For more information, see Analyzing an image.

If you don't specify a value for `MinConfidence`, `DetectCustomLabels` returns labels based on the assumed threshold of each label.

This is a stateless API operation. That is, the operation does not persist any data.

This operation requires permissions to perform the `rekognition:DetectCustomLabels` action.

For more information, see Analyzing an image.

Request Syntax

```
{
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "MaxResults": number,
   "MinConfidence": number,
   "ProjectVersionArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Image

Provides the input image either as bytes or an S3 object.

You pass image bytes to an Amazon Rekognition API operation by using the `Bytes` property. For example, you would use the `Bytes` property to pass an image loaded from a local file system. Image bytes passed by using the `Bytes` property must be base64-encoded. Your code may not need to encode image bytes if you are using an AWS SDK to call Amazon Rekognition API operations.

For more information, see Analyzing an image loaded from a local file system .

You pass images stored in an S3 bucket to an Amazon Rekognition API operation by using the `S3Object` property. Images stored in an S3 bucket do not need to be base64-encoded.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes using the Bytes property is not supported. You must first upload the image to an Amazon S3 bucket and then call the operation using the S3Object property.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Type: Image object

Required: Yes

MaxResults

Maximum number of results you want the service to return in the response. The service returns the specified number of highest confidence labels ranked from highest confidence to lowest.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

MinConfidence

Specifies the minimum confidence level for the labels to return. `DetectCustomLabels` doesn't return any labels with a confidence value that's lower than this specified value. If you specify a value of 0, `DetectCustomLabels` returns all labels, regardless of the assumed threshold applied to each label. If you don't specify a value for `MinConfidence`, `DetectCustomLabels` returns labels based on the assumed threshold of each label.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

ProjectVersionArn

The ARN of the model version that you want to use. Only models associated with Custom Labels projects accepted by the operation. If a provided ARN refers to a model version associated with a project for a different feature type, then an InvalidParameterException is returned.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Response Syntax

```
{
   "CustomLabels": [
      {
         "Confidence": number,
         "Geometry": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Polygon": [
               {
                  "X": number,
                  "Y": number
               }
            ]
         },
         "Name": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CustomLabels

An array of custom labels detected in the input image.

Type: Array of CustomLabel objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will

raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ResourceNotReadyException

The requested resource isn't ready. For example, this exception occurs when you call `DetectCustomLabels` with a model version that isn't deployed.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DetectFaces

Service: Amazon Rekognition

Detects faces within an image that is provided as input.

`DetectFaces` detects the 100 largest faces in the image. For each face detected, the operation returns face details. These details include a bounding box of the face, a confidence value (that the bounding box contains a face), and a fixed set of attributes such as facial landmarks (for example, coordinates of eye and mouth), pose, presence of facial occlusion, and so on.

The face-detection algorithm is most effective on frontal faces. For non-frontal or obscured faces, the algorithm might not detect the faces or might detect faces with lower confidence.

You pass the input image either as base64-encoded image bytes or as a reference to an image in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes is not supported. The image must be either a PNG or JPEG formatted file.

> **Note**
>
> This is a stateless API operation. That is, the operation does not persist any data.

This operation requires permissions to perform the `rekognition:DetectFaces` action.

Request Syntax

```
{
   "Attributes": [ "string" ],
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Attributes

An array of facial attributes you want to be returned. A `DEFAULT` subset of facial attributes - `BoundingBox`, `Confidence`, `Pose`, `Quality`, and `Landmarks` - will always be returned. You can request for specific facial attributes (in addition to the default list) - by using `["DEFAULT", "FACE_OCCLUDED"]` or just `["FACE_OCCLUDED"]`. You can request for all facial attributes by using `["ALL"]`. Requesting more attributes may increase response time.

If you provide both, `["ALL", "DEFAULT"]`, the service uses a logical "AND" operator to determine which attributes to return (in this case, all attributes).

Note that while the FaceOccluded and EyeDirection attributes are supported when using `DetectFaces`, they aren't supported when analyzing videos with `StartFaceDetection` and `GetFaceDetection`.

Type: Array of strings

Valid Values: `DEFAULT | ALL | AGE_RANGE | BEARD | EMOTIONS | EYE_DIRECTION | EYEGLASSES | EYES_OPEN | GENDER | MOUTH_OPEN | MUSTACHE | FACE_OCCLUDED | SMILE | SUNGLASSES`

Required: No

Image

The input image as base64-encoded bytes or an S3 object. If you use the AWS CLI to call Amazon Rekognition operations, passing base64-encoded image bytes is not supported.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

Response Syntax

```
{
   "FaceDetails": [
      {
```

```
            "AgeRange": {
               "High": number,
               "Low": number
            },
            "Beard": {
               "Confidence": number,
               "Value": boolean
            },
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "Emotions": [
               {
                  "Confidence": number,
                  "Type": "string"
               }
            ],
            "EyeDirection": {
               "Confidence": number,
               "Pitch": number,
               "Yaw": number
            },
            "Eyeglasses": {
               "Confidence": number,
               "Value": boolean
            },
            "EyesOpen": {
               "Confidence": number,
               "Value": boolean
            },
            "FaceOccluded": {
               "Confidence": number,
               "Value": boolean
            },
            "Gender": {
               "Confidence": number,
               "Value": "string"
            },
            "Landmarks": [
               {
```

```
                "Type": "string",
                "X": number,
                "Y": number
            }
        ],
        "MouthOpen": {
            "Confidence": number,
            "Value": boolean
        },
        "Mustache": {
            "Confidence": number,
            "Value": boolean
        },
        "Pose": {
            "Pitch": number,
            "Roll": number,
            "Yaw": number
        },
        "Quality": {
            "Brightness": number,
            "Sharpness": number
        },
        "Smile": {
            "Confidence": number,
            "Value": boolean
        },
        "Sunglasses": {
            "Confidence": number,
            "Value": boolean
        }
    }
  ],
  "OrientationCorrection": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaceDetails

Details of each face found in the image.

Type: Array of FaceDetail objects

OrientationCorrection

The value of `OrientationCorrection` is always null.

If the input image is in .jpeg format, it might contain exchangeable image file format (Exif) metadata that includes the image's orientation. Amazon Rekognition uses this orientation information to perform image correction. The bounding box coordinates are translated to represent object locations after the orientation information in the Exif metadata is used to correct the image orientation. Images in .png format don't contain Exif metadata.

Amazon Rekognition doesn’t perform image correction for images in .png format and .jpeg images without orientation information in the image Exif metadata. The bounding box coordinates aren't translated and represent the object locations before the image is rotated.

Type: String

Valid Values: `ROTATE_0 | ROTATE_90 | ROTATE_180 | ROTATE_270`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DetectLabels

Service: Amazon Rekognition

Detects instances of real-world entities within an image (JPEG or PNG) provided as input. This includes objects like flower, tree, and table; events like wedding, graduation, and birthday party; and concepts like landscape, evening, and nature.

For an example, see Analyzing images stored in an Amazon S3 bucket.

You pass the input image as base64-encoded image bytes or as a reference to an image in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes is not supported. The image must be either a PNG or JPEG formatted file.

Optional Parameters

You can specify one or both of the `GENERAL_LABELS` and `IMAGE_PROPERTIES` feature types when calling the DetectLabels API. Including `GENERAL_LABELS` will ensure the response includes the labels detected in the input image, while including `IMAGE_PROPERTIES` will ensure the response includes information about the image quality and color.

When using `GENERAL_LABELS` and/or `IMAGE_PROPERTIES` you can provide filtering criteria to the Settings parameter. You can filter with sets of individual labels or with label categories. You can specify inclusive filters, exclusive filters, or a combination of inclusive and exclusive filters. For more information on filtering see Detecting Labels in an Image.

When getting labels, you can specify `MinConfidence` to control the confidence threshold for the labels returned. The default is 55%. You can also add the `MaxLabels` parameter to limit the number of labels returned. The default and upper limit is 1000 labels. These arguments are only valid when supplying GENERAL_LABELS as a feature type.

Response Elements

For each object, scene, and concept the API returns one or more labels. The API returns the following types of information about labels:

- Name - The name of the detected label.
- Confidence - The level of confidence in the label assigned to a detected object.
- Parents - The ancestor labels for a detected label. DetectLabels returns a hierarchical taxonomy of detected labels. For example, a detected car might be assigned the label car. The label car

has two parent labels: Vehicle (its parent) and Transportation (its grandparent). The response includes the all ancestors for a label, where every ancestor is a unique label. In the previous example, Car, Vehicle, and Transportation are returned as unique labels in the response.

- Aliases - Possible Aliases for the label.
- Categories - The label categories that the detected label belongs to.
- BoundingBox — Bounding boxes are described for all instances of detected common object labels, returned in an array of Instance objects. An Instance object contains a BoundingBox object, describing the location of the label on the input image. It also includes the confidence for the accuracy of the detected bounding box.

The API returns the following information regarding the image, as part of the ImageProperties structure:

- Quality - Information about the Sharpness, Brightness, and Contrast of the input image, scored between 0 to 100. Image quality is returned for the entire image, as well as the background and the foreground.
- Dominant Color - An array of the dominant colors in the image.
- Foreground - Information about the sharpness, brightness, and dominant colors of the input image's foreground.
- Background - Information about the sharpness, brightness, and dominant colors of the input image's background.

The list of returned labels will include at least one label for every detected object, along with information about that label. In the following example, suppose the input image has a lighthouse, the sea, and a rock. The response includes all three labels, one for each object, as well as the confidence in the label:

```
{Name: lighthouse, Confidence: 98.4629}

{Name: rock,Confidence: 79.2097}

{Name: sea,Confidence: 75.061}
```

The list of labels can include multiple labels for the same object. For example, if the input image shows a flower (for example, a tulip), the operation might return the following three labels.

```
{Name: flower,Confidence: 99.0562}
```

```
{Name: plant,Confidence: 99.0562}

{Name: tulip,Confidence: 99.0562}
```

In this example, the detection algorithm more precisely identifies the flower as a tulip.

> **Note**
>
> If the object detected is a person, the operation doesn't provide the same facial details that the DetectFaces operation provides.

This is a stateless API operation that doesn't return any data.

This operation requires permissions to perform the `rekognition:DetectLabels` action.

Request Syntax

```
{
   "Features": [ "string" ],
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "MaxLabels": number,
   "MinConfidence": number,
   "Settings": {
      "GeneralLabels": {
         "LabelCategoryExclusionFilters": [ "string" ],
         "LabelCategoryInclusionFilters": [ "string" ],
         "LabelExclusionFilters": [ "string" ],
         "LabelInclusionFilters": [ "string" ]
      },
      "ImageProperties": {
         "MaxDominantColors": number
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Features

A list of the types of analysis to perform. Specifying GENERAL_LABELS uses the label detection feature, while specifying IMAGE_PROPERTIES returns information regarding image color and quality. If no option is specified GENERAL_LABELS is used by default.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 2 items.

Valid Values: `GENERAL_LABELS | IMAGE_PROPERTIES`

Required: No

Image

The input image as base64-encoded bytes or an S3 object. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes is not supported. Images stored in an S3 Bucket do not need to be base64-encoded.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

MaxLabels

Maximum number of labels you want the service to return in the response. The service returns the specified number of highest confidence labels. Only valid when GENERAL_LABELS is specified as a feature type in the Feature input parameter.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

MinConfidence

Specifies the minimum confidence level for the labels to return. Amazon Rekognition doesn't return any labels with confidence lower than this specified value.

If `MinConfidence` is not specified, the operation returns labels with a confidence values greater than or equal to 55 percent. Only valid when GENERAL_LABELS is specified as a feature type in the Feature input parameter.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Settings

A list of the filters to be applied to returned detected labels and image properties. Specified filters can be inclusive, exclusive, or a combination of both. Filters can be used for individual labels or label categories. The exact label names or label categories must be supplied. For a full list of labels and label categories, see Detecting labels.

Type: DetectLabelsSettings object

Required: No

Response Syntax

```
{
   "ImageProperties": {
      "Background": {
         "DominantColors": [
            {
               "Blue": number,
               "CSSColor": "string",
               "Green": number,
               "HexCode": "string",
               "PixelPercent": number,
               "Red": number,
               "SimplifiedColor": "string"
            }
         ],
         "Quality": {
```

```
            "Brightness": number,
            "Contrast": number,
            "Sharpness": number
         }
      },
      "DominantColors": [
         {
            "Blue": number,
            "CSSColor": "string",
            "Green": number,
            "HexCode": "string",
            "PixelPercent": number,
            "Red": number,
            "SimplifiedColor": "string"
         }
      ],
      "Foreground": {
         "DominantColors": [
            {
               "Blue": number,
               "CSSColor": "string",
               "Green": number,
               "HexCode": "string",
               "PixelPercent": number,
               "Red": number,
               "SimplifiedColor": "string"
            }
         ],
         "Quality": {
            "Brightness": number,
            "Contrast": number,
            "Sharpness": number
         }
      },
      "Quality": {
         "Brightness": number,
         "Contrast": number,
         "Sharpness": number
      }
   },
   "LabelModelVersion": "string",
   "Labels": [
      {
         "Aliases": [
```

```
            {
               "Name": "string"
            }
         ],
         "Categories": [
            {
               "Name": "string"
            }
         ],
         "Confidence": number,
         "Instances": [
            {
               "BoundingBox": {
                  "Height": number,
                  "Left": number,
                  "Top": number,
                  "Width": number
               },
               "Confidence": number,
               "DominantColors": [
                  {
                     "Blue": number,
                     "CSSColor": "string",
                     "Green": number,
                     "HexCode": "string",
                     "PixelPercent": number,
                     "Red": number,
                     "SimplifiedColor": "string"
                  }
               ]
            }
         ],
         "Name": "string",
         "Parents": [
            {
               "Name": "string"
            }
         ]
      }
   ],
   "OrientationCorrection": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ImageProperties

Information about the properties of the input image, such as brightness, sharpness, contrast, and dominant colors.

Type: DetectLabelsImageProperties object

LabelModelVersion

Version number of the label detection model that was used to detect labels.

Type: String

Labels

An array of labels for the real-world objects detected.

Type: Array of Label objects

OrientationCorrection

The value of `OrientationCorrection` is always null.

If the input image is in .jpeg format, it might contain exchangeable image file format (Exif) metadata that includes the image's orientation. Amazon Rekognition uses this orientation information to perform image correction. The bounding box coordinates are translated to represent object locations after the orientation information in the Exif metadata is used to correct the image orientation. Images in .png format don't contain Exif metadata.

Amazon Rekognition doesn’t perform image correction for images in .png format and .jpeg images without orientation information in the image Exif metadata. The bounding box coordinates aren't translated and represent the object locations before the image is rotated.

Type: String

Valid Values: `ROTATE_0 | ROTATE_90 | ROTATE_180 | ROTATE_270`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DetectModerationLabels

Service: Amazon Rekognition

Detects unsafe content in a specified JPEG or PNG format image. Use `DetectModerationLabels` to moderate images depending on your requirements. For example, you might want to filter images that contain nudity, but not images containing suggestive content.

To filter images, use the labels returned by `DetectModerationLabels` to determine which types of content are appropriate.

For information about moderation labels, see Moderating content. For a list of moderation labels in Amazon Rekognition, see Using the image and video moderation APIs.

You pass the input image either as base64-encoded image bytes or as a reference to an image in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes is not supported. The image must be either a PNG or JPEG formatted file.

You can specify an adapter to use when retrieving label predictions by providing a `ProjectVersionArn` to the `ProjectVersion` argument.

Request Syntax

```
{
   "HumanLoopConfig": {
      "DataAttributes": {
         "ContentClassifiers": [ "string" ]
      },
      "FlowDefinitionArn": "string",
      "HumanLoopName": "string"
   },
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "MinConfidence": number,
   "ProjectVersion": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

HumanLoopConfig

Sets up the configuration for human evaluation, including the FlowDefinition the image will be sent to.

Type: HumanLoopConfig object

Required: No

Image

The input image as base64-encoded bytes or an S3 object. If you use the AWS CLI to call Amazon Rekognition operations, passing base64-encoded image bytes is not supported.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

MinConfidence

Specifies the minimum confidence level for the labels to return. Amazon Rekognition doesn't return any labels with a confidence level lower than this specified value.

If you don't specify `MinConfidence`, the operation returns labels with confidence values greater than or equal to 50 percent.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

ProjectVersion

Identifier for the custom adapter. Expects the ProjectVersionArn as a value. Use the CreateProject or CreateProjectVersion APIs to create a custom adapter.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: No

Response Syntax

```
{
   "ContentTypes": [
      {
         "Confidence": number,
         "Name": "string"
      }
   ],
   "HumanLoopActivationOutput": {
      "HumanLoopActivationConditionsEvaluationResults": "string",
      "HumanLoopActivationReasons": [ "string" ],
      "HumanLoopArn": "string"
   },
   "ModerationLabels": [
      {
         "Confidence": number,
         "Name": "string",
         "ParentName": "string",
         "TaxonomyLevel": number
      }
   ],
   "ModerationModelVersion": "string",
   "ProjectVersion": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ContentTypes

A list of predicted results for the type of content an image contains. For example, the image content might be from animation, sports, or a video game.

Type: Array of ContentType objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

HumanLoopActivationOutput

Shows the results of the human in the loop evaluation.

Type: HumanLoopActivationOutput object

ModerationLabels

Array of detected Moderation labels. For video operations, this includes the time, in milliseconds from the start of the video, they were detected.

Type: Array of ModerationLabel objects

ModerationModelVersion

Version number of the base moderation detection model that was used to detect unsafe content.

Type: String

ProjectVersion

Identifier of the custom adapter that was used during inference. If during inference the adapter was EXPIRED, then the parameter will not be returned, indicating that a base moderation detection project version was used.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

HumanLoopQuotaExceededException

The number of in-progress human reviews you have has exceeded the number allowed.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ResourceNotReadyException

The requested resource isn't ready. For example, this exception occurs when you call `DetectCustomLabels` with a model version that isn't deployed.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DetectProtectiveEquipment

Service: Amazon Rekognition

Detects Personal Protective Equipment (PPE) worn by people detected in an image. Amazon Rekognition can detect the following types of PPE.

- Face cover
- Hand cover
- Head cover

You pass the input image as base64-encoded image bytes or as a reference to an image in an Amazon S3 bucket. The image must be either a PNG or JPG formatted file.

`DetectProtectiveEquipment` detects PPE worn by up to 15 persons detected in an image.

For each person detected in the image the API returns an array of body parts (face, head, left-hand, right-hand). For each body part, an array of detected items of PPE is returned, including an indicator of whether or not the PPE covers the body part. The API returns the confidence it has in each detection (person, PPE, body part and body part coverage). It also returns a bounding box (BoundingBox) for each detected person and each detected item of PPE.

You can optionally request a summary of detected PPE items with the `SummarizationAttributes` input parameter. The summary provides the following information.

- The persons detected as wearing all of the types of PPE that you specify.
- The persons detected as not wearing all of the types PPE that you specify.
- The persons detected where PPE adornment could not be determined.

This is a stateless API operation. That is, the operation does not persist any data.

This operation requires permissions to perform the `rekognition:DetectProtectiveEquipment` action.

Request Syntax

```
{
   "Image": {
```

```
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "SummarizationAttributes": {
      "MinConfidence": number,
      "RequiredEquipmentTypes": [ "string" ]
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Image

The image in which you want to detect PPE on detected persons. The image can be passed as image bytes or you can reference an image stored in an Amazon S3 bucket.

Type: Image object

Required: Yes

SummarizationAttributes

An array of PPE types that you want to summarize.

Type: ProtectiveEquipmentSummarizationAttributes object

Required: No

Response Syntax

```
{
   "Persons": [
      {
         "BodyParts": [
```

```
            {
               "Confidence": number,
               "EquipmentDetections": [
                  {
                     "BoundingBox": {
                        "Height": number,
                        "Left": number,
                        "Top": number,
                        "Width": number
                     },
                     "Confidence": number,
                     "CoversBodyPart": {
                        "Confidence": number,
                        "Value": boolean
                     },
                     "Type": "string"
                  }
               ],
               "Name": "string"
            }
         ],
         "BoundingBox": {
            "Height": number,
            "Left": number,
            "Top": number,
            "Width": number
         },
         "Confidence": number,
         "Id": number
      }
   ],
   "ProtectiveEquipmentModelVersion": "string",
   "Summary": {
      "PersonsIndeterminate": [ number ],
      "PersonsWithoutRequiredEquipment": [ number ],
      "PersonsWithRequiredEquipment": [ number ]
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Persons

An array of persons detected in the image (including persons not wearing PPE).

Type: Array of ProtectiveEquipmentPerson objects

ProtectiveEquipmentModelVersion

The version number of the PPE detection model used to detect PPE in the image.

Type: String

Summary

Summary information for the types of PPE specified in the `SummarizationAttributes` input parameter.

Type: ProtectiveEquipmentSummary object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

DetectText

Service: Amazon Rekognition

Detects text in the input image and converts it into machine-readable text.

Pass the input image as base64-encoded image bytes or as a reference to an image in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, you must pass it as a reference to an image in an Amazon S3 bucket. For the AWS CLI, passing image bytes is not supported. The image must be either a .png or .jpeg formatted file.

The `DetectText` operation returns text in an array of TextDetection elements, `TextDetections`. Each `TextDetection` element provides information about a single word or line of text that was detected in the image.

A word is one or more script characters that are not separated by spaces. `DetectText` can detect up to 100 words in an image.

A line is a string of equally spaced words. A line isn't necessarily a complete sentence. For example, a driver's license number is detected as a line. A line ends when there is no aligned text after it. Also, a line ends when there is a large gap between words, relative to the length of the words. This means, depending on the gap between words, Amazon Rekognition may detect multiple lines in text aligned in the same direction. Periods don't represent the end of a line. If a sentence spans multiple lines, the `DetectText` operation returns multiple lines.

To determine whether a `TextDetection` element is a line of text or a word, use the `TextDetection` object `Type` field.

To be detected, text must be within +/- 90 degrees orientation of the horizontal axis.

For more information, see Detecting text.

Request Syntax

```
{
   "Filters": {
      "RegionsOfInterest": [
         {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
```

```
            },
            "Polygon": [
               {
                  "X": number,
                  "Y": number
               }
            ]
         }
      ],
      "WordFilter": {
         "MinBoundingBoxHeight": number,
         "MinBoundingBoxWidth": number,
         "MinConfidence": number
      }
   },
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Filters

Optional parameters that let you set the criteria that the text must meet to be included in your response.

Type: DetectTextFilters object

Required: No

Image

The input image as base64-encoded bytes or an Amazon S3 object. If you use the AWS CLI to call Amazon Rekognition operations, you can't pass image bytes.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

Response Syntax

```
{
   "TextDetections": [
      {
         "Confidence": number,
         "DetectedText": "string",
         "Geometry": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Polygon": [
               {
                  "X": number,
                  "Y": number
               }
            ]
         },
         "Id": number,
         "ParentId": number,
         "Type": "string"
      }
   ],
   "TextModelVersion": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TextDetections

An array of text that was detected in the input image.

Type: Array of TextDetection objects

TextModelVersion

The model version used to detect text.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DisassociateFaces

Service: Amazon Rekognition

Removes the association between a `Face` supplied in an array of `FaceIds` and the User. If the User is not present already, then a `ResourceNotFound` exception is thrown. If successful, an array of faces that are disassociated from the User is returned. If a given face is already disassociated from the given UserID, it will be ignored and not be returned in the response. If a given face is already associated with a different User or not found in the collection it will be returned as part of `UnsuccessfulDisassociations`. You can remove 1 - 100 face IDs from a user at one time.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "CollectionId": "string",
   "FaceIds": [ "string" ],
   "UserId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the request to `DisassociateFaces`. If you use the same token with multiple `DisassociateFaces` requests, the same response is returned. Use ClientRequestToken to prevent the same request from being processed more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

CollectionId

The ID of an existing collection containing the UserID.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

FaceIds

An array of face IDs to disassociate from the UserID.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: Yes

UserId

ID for the existing UserID.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: Yes

Response Syntax

```
{
   "DisassociatedFaces": [
      {
         "FaceId": "string"
      }
   ],
   "UnsuccessfulFaceDisassociations": [
      {
         "FaceId": "string",
         "Reasons": [ "string" ],
```

```
            "UserId": "string"
        }
    ],
    "UserStatus": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DisassociatedFaces

An array of DissociatedFace objects containing FaceIds that are successfully disassociated with the UserID is returned. Returned if the DisassociatedFaces action is successful.

Type: Array of DisassociatedFace objects

Array Members: Minimum number of 0 items. Maximum number of 100 items.

UnsuccessfulFaceDisassociations

An array of UnsuccessfulDisassociation objects containing FaceIds that are not successfully associated, along with the reasons for the failure to associate. Returned if the DisassociateFaces action is successful.

Type: Array of UnsuccessfulFaceDisassociation objects

Array Members: Minimum number of 0 items. Maximum number of 500 items.

UserStatus

The status of an update made to a User. Reflects if the User has been updated for every requested change.

Type: String

Valid Values: `ACTIVE | UPDATING | CREATING | CREATED`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ConflictException

A User with the same Id already exists within the collection, or the update or deletion of the User caused an inconsistent state. **

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DistributeDatasetEntries

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Distributes the entries (images) in a training dataset across the training dataset and the test dataset for a project. `DistributeDatasetEntries` moves 20% of the training dataset images to the test dataset. An entry is a JSON Line that describes an image.

You supply the Amazon Resource Names (ARN) of a project's training dataset and test dataset. The training dataset must contain the images that you want to split. The test dataset must be empty. The datasets must belong to the same project. To create training and test datasets for a project, call CreateDataset.

Distributing a dataset takes a while to complete. To check the status call `DescribeDataset`. The operation is complete when the `Status` field for the training dataset and the test dataset is `UPDATE_COMPLETE`. If the dataset split fails, the value of `Status` is `UPDATE_FAILED`.

This operation requires permissions to perform the `rekognition:DistributeDatasetEntries` action.

Request Syntax

```
{
   "Datasets": [
      {
         "Arn": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Datasets

The ARNS for the training dataset and test dataset that you want to use. The datasets must belong to the same project. The test dataset must be empty.

Type: Array of DistributeDataset objects

Array Members: Fixed number of 2 items.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ResourceNotReadyException

The requested resource isn't ready. For example, this exception occurs when you call `DetectCustomLabels` with a model version that isn't deployed.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetCelebrityInfo

Service: Amazon Rekognition

Gets the name and additional information about a celebrity based on their Amazon Rekognition ID. The additional information is returned as an array of URLs. If there is no additional information about the celebrity, this list is empty.

For more information, see Getting information about a celebrity.

This operation requires permissions to perform the `rekognition:GetCelebrityInfo` action.

Request Syntax

```
{
   "Id": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Id

The ID for the celebrity. You get the celebrity ID from a call to the RecognizeCelebrities operation, which recognizes celebrities in an image.

Type: String

Pattern: `[0-9A-Za-z]*`

Required: Yes

Response Syntax

```
{
   "KnownGender": {
      "Type": "string"
   },
   "Name": "string",
```

```
    "Urls": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

KnownGender

Retrieves the known gender for the celebrity.

Type: KnownGender object

Name

The name of the celebrity.

Type: String

Urls

An array of URLs pointing to additional celebrity information.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 255 items.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

GetCelebrityRecognition

Service: Amazon Rekognition

Gets the celebrity recognition results for a Amazon Rekognition Video analysis started by StartCelebrityRecognition.

Celebrity recognition in a video is an asynchronous operation. Analysis is started by a call to StartCelebrityRecognition which returns a job identifier (`JobId`).

When the celebrity recognition operation finishes, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic registered in the initial call to `StartCelebrityRecognition`. To get the results of the celebrity recognition analysis, first check that the status value published to the Amazon SNS topic is `SUCCEEDED`. If so, call `GetCelebrityDetection` and pass the job identifier (`JobId`) from the initial call to `StartCelebrityDetection`.

For more information, see Working with stored videos.

`GetCelebrityRecognition` returns detected celebrities and the time(s) they are detected in an array (`Celebrities`) of CelebrityRecognition objects. Each `CelebrityRecognition` contains information about the celebrity in a CelebrityDetail object and the time, `Timestamp`, the celebrity was detected. This CelebrityDetail object stores information about the detected celebrity's face attributes, a face bounding box, known gender, the celebrity's name, and a confidence estimate.

> **Note**
>
> `GetCelebrityRecognition` only returns the default facial attributes (`BoundingBox`, `Confidence`, `Landmarks`, `Pose`, and `Quality`). The `BoundingBox` field only applies to the detected face instance. The other facial attributes listed in the `Face` object of the following response syntax are not returned. For more information, see FaceDetail.

By default, the `Celebrities` array is sorted by time (milliseconds from the start of the video). You can also sort the array by celebrity by specifying the value `ID` in the `SortBy` input parameter.

The `CelebrityDetail` object includes the celebrity identifer and additional information urls. If you don't store the additional information urls, you can get them later by calling GetCelebrityInfo with the celebrity identifer.

No information is returned for faces not recognized as celebrities.

Use MaxResults parameter to limit the number of labels returned. If there are more results than specified in `MaxResults`, the value of `NextToken` in the operation response contains a pagination token for getting the next set of results. To get the next page of results, call `GetCelebrityDetection` and populate the `NextToken` request parameter with the token value returned from the previous call to `GetCelebrityRecognition`.

Request Syntax

```
{
   "JobId": "string",
   "MaxResults": number,
   "NextToken": "string",
   "SortBy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

JobId

Job identifier for the required celebrity recognition analysis. You can get the job identifer from a call to `StartCelebrityRecognition`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

MaxResults

Maximum number of results to return per paginated call. The largest value you can specify is 1000. If you specify a value greater than 1000, a maximum of 1000 results is returned. The default value is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the previous response was incomplete (because there is more recognized celebrities to retrieve), Amazon Rekognition Video returns a pagination token in the response. You can use this pagination token to retrieve the next set of celebrities.

Type: String

Length Constraints: Maximum length of 255.

Required: No

SortBy

Sort to use for celebrities returned in `Celebrities` field. Specify `ID` to sort by the celebrity identifier, specify `TIMESTAMP` to sort by the time the celebrity was recognized.

Type: String

Valid Values: `ID | TIMESTAMP`

Required: No

Response Syntax

```
{
   "Celebrities": [
      {
         "Celebrity": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "Face": {
               "AgeRange": {
                  "High": number,
                  "Low": number
               },
               "Beard": {
```

```
                "Confidence": number,
                "Value": boolean
            },
            "BoundingBox": {
                "Height": number,
                "Left": number,
                "Top": number,
                "Width": number
            },
            "Confidence": number,
            "Emotions": [
                {
                    "Confidence": number,
                    "Type": "string"
                }
            ],
            "EyeDirection": {
                "Confidence": number,
                "Pitch": number,
                "Yaw": number
            },
            "Eyeglasses": {
                "Confidence": number,
                "Value": boolean
            },
            "EyesOpen": {
                "Confidence": number,
                "Value": boolean
            },
            "FaceOccluded": {
                "Confidence": number,
                "Value": boolean
            },
            "Gender": {
                "Confidence": number,
                "Value": "string"
            },
            "Landmarks": [
                {
                    "Type": "string",
                    "X": number,
                    "Y": number
                }
            ],
```

```
                "MouthOpen": {
                   "Confidence": number,
                   "Value": boolean
                },
                "Mustache": {
                   "Confidence": number,
                   "Value": boolean
                },
                "Pose": {
                   "Pitch": number,
                   "Roll": number,
                   "Yaw": number
                },
                "Quality": {
                   "Brightness": number,
                   "Sharpness": number
                },
                "Smile": {
                   "Confidence": number,
                   "Value": boolean
                },
                "Sunglasses": {
                   "Confidence": number,
                   "Value": boolean
                }
             },
             "Id": "string",
             "KnownGender": {
                "Type": "string"
             },
             "Name": "string",
             "Urls": [ "string" ]
          },
          "Timestamp": number
       }
    ],
    "JobId": "string",
    "JobStatus": "string",
    "JobTag": "string",
    "NextToken": "string",
    "StatusMessage": "string",
    "Video": {
       "S3Object": {
          "Bucket": "string",
```

```
            "Name": "string",
            "Version": "string"
        }
    },
    "VideoMetadata": {
        "Codec": "string",
        "ColorRange": "string",
        "DurationMillis": number,
        "Format": "string",
        "FrameHeight": number,
        "FrameRate": number,
        "FrameWidth": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Celebrities

Array of celebrities recognized in the video.

Type: Array of CelebrityRecognition objects

JobId

Job identifier for the celebrity recognition operation for which you want to obtain results. The job identifer is returned by an initial call to StartCelebrityRecognition.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobStatus

The current status of the celebrity recognition job.

Type: String

Valid Values: `IN_PROGRESS | SUCCEEDED | FAILED`

JobTag

A job identifier specified in the call to StartCelebrityRecognition and returned in the job completion notification sent to your Amazon Simple Notification Service topic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

NextToken

If the response is truncated, Amazon Rekognition Video returns this token that you can use in the subsequent request to retrieve the next set of celebrities.

Type: String

Length Constraints: Maximum length of 255.

StatusMessage

If the job fails, `StatusMessage` provides a descriptive error message.

Type: String

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

VideoMetadata

Information about a video that Amazon Rekognition Video analyzed. `Videometadata` is returned in every page of paginated responses from a Amazon Rekognition Video operation.

Type: VideoMetadata object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetContentModeration

Service: Amazon Rekognition

Gets the inappropriate, unwanted, or offensive content analysis results for a Amazon Rekognition Video analysis started by StartContentModeration. For a list of moderation labels in Amazon Rekognition, see Using the image and video moderation APIs.

Amazon Rekognition Video inappropriate or offensive content detection in a stored video is an asynchronous operation. You start analysis by calling StartContentModeration which returns a job identifier (`JobId`). When analysis finishes, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic registered in the initial call to `StartContentModeration`. To get the results of the content analysis, first check that the status value published to the Amazon SNS topic is `SUCCEEDED`. If so, call `GetContentModeration` and pass the job identifier (`JobId`) from the initial call to `StartContentModeration`.

For more information, see Working with stored videos.

`GetContentModeration` returns detected inappropriate, unwanted, or offensive content moderation labels, and the time they are detected, in an array, `ModerationLabels`, of ContentModerationDetection objects.

By default, the moderated labels are returned sorted by time, in milliseconds from the start of the video. You can also sort them by moderated label by specifying `NAME` for the `SortBy` input parameter.

Since video analysis can return a large number of results, use the `MaxResults` parameter to limit the number of labels returned in a single call to `GetContentModeration`. If there are more results than specified in `MaxResults`, the value of `NextToken` in the operation response contains a pagination token for getting the next set of results. To get the next page of results, call `GetContentModeration` and populate the `NextToken` request parameter with the value of `NextToken` returned from the previous call to `GetContentModeration`.

For more information, see Moderating content.

Request Syntax

```
{
   "AggregateBy": "string",
   "JobId": "string",
   "MaxResults": number,
```

```
    "NextToken": "string",
    "SortBy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

AggregateBy

Defines how to aggregate results of the StartContentModeration request. Default aggregation option is TIMESTAMPS. SEGMENTS mode aggregates moderation labels over time.

Type: String

Valid Values: `TIMESTAMPS | SEGMENTS`

Required: No

JobId

The identifier for the inappropriate, unwanted, or offensive content moderation job. Use `JobId` to identify the job in a subsequent call to `GetContentModeration`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

MaxResults

Maximum number of results to return per paginated call. The largest value you can specify is 1000. If you specify a value greater than 1000, a maximum of 1000 results is returned. The default value is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the previous response was incomplete (because there is more data to retrieve), Amazon Rekognition returns a pagination token in the response. You can use this pagination token to retrieve the next set of content moderation labels.

Type: String

Length Constraints: Maximum length of 255.

Required: No

SortBy

Sort to use for elements in the `ModerationLabelDetections` array. Use `TIMESTAMP` to sort array elements by the time labels are detected. Use `NAME` to alphabetically group elements for a label together. Within each label group, the array element are sorted by detection confidence. The default sort is by `TIMESTAMP`.

Type: String

Valid Values: `NAME | TIMESTAMP`

Required: No

Response Syntax

```
{
   "GetRequestMetadata": {
      "AggregateBy": "string",
      "SortBy": "string"
   },
   "JobId": "string",
   "JobStatus": "string",
   "JobTag": "string",
   "ModerationLabels": [
      {
         "ContentTypes": [
            {
               "Confidence": number,
               "Name": "string"
            }
         ],
```

```
            "DurationMillis": number,
            "EndTimestampMillis": number,
            "ModerationLabel": {
               "Confidence": number,
               "Name": "string",
               "ParentName": "string",
               "TaxonomyLevel": number
            },
            "StartTimestampMillis": number,
            "Timestamp": number
         }
      ],
      "ModerationModelVersion": "string",
      "NextToken": "string",
      "StatusMessage": "string",
      "Video": {
         "S3Object": {
            "Bucket": "string",
            "Name": "string",
            "Version": "string"
         }
      },
      "VideoMetadata": {
         "Codec": "string",
         "ColorRange": "string",
         "DurationMillis": number,
         "Format": "string",
         "FrameHeight": number,
         "FrameRate": number,
         "FrameWidth": number
      }
   }
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GetRequestMetadata

Information about the paramters used when getting a response. Includes information on aggregation and sorting methods.

Type: GetContentModerationRequestMetadata object

JobId

Job identifier for the content moderation operation for which you want to obtain results. The job identifer is returned by an initial call to StartContentModeration.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobStatus

The current status of the content moderation analysis job.

Type: String

Valid Values: `IN_PROGRESS | SUCCEEDED | FAILED`

JobTag

A job identifier specified in the call to StartContentModeration and returned in the job completion notification sent to your Amazon Simple Notification Service topic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

ModerationLabels

The detected inappropriate, unwanted, or offensive content moderation labels and the time(s) they were detected.

Type: Array of ContentModerationDetection objects

ModerationModelVersion

Version number of the moderation detection model that was used to detect inappropriate, unwanted, or offensive content.

Type: String

NextToken

If the response is truncated, Amazon Rekognition Video returns this token that you can use in the subsequent request to retrieve the next set of content moderation labels.

Type: String

Length Constraints: Maximum length of 255.

StatusMessage

If the job fails, `StatusMessage` provides a descriptive error message.

Type: String

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

VideoMetadata

Information about a video that Amazon Rekognition analyzed. `Videometadata` is returned in every page of paginated responses from `GetContentModeration`.

Type: VideoMetadata object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetFaceDetection

Service: Amazon Rekognition

Gets face detection results for a Amazon Rekognition Video analysis started by StartFaceDetection.

Face detection with Amazon Rekognition Video is an asynchronous operation. You start face detection by calling StartFaceDetection which returns a job identifier (`JobId`). When the face detection operation finishes, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic registered in the initial call to `StartFaceDetection`. To get the results of the face detection operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call GetFaceDetection and pass the job identifier (`JobId`) from the initial call to `StartFaceDetection`.

`GetFaceDetection` returns an array of detected faces (`Faces`) sorted by the time the faces were detected.

Use MaxResults parameter to limit the number of labels returned. If there are more results than specified in `MaxResults`, the value of `NextToken` in the operation response contains a pagination token for getting the next set of results. To get the next page of results, call `GetFaceDetection` and populate the `NextToken` request parameter with the token value returned from the previous call to `GetFaceDetection`.

Note that for the `GetFaceDetection` operation, the returned values for `FaceOccluded` and `EyeDirection` will always be "null".

Request Syntax

```
{
   "JobId": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

JobId

Unique identifier for the face detection job. The `JobId` is returned from `StartFaceDetection`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

MaxResults

Maximum number of results to return per paginated call. The largest value you can specify is 1000. If you specify a value greater than 1000, a maximum of 1000 results is returned. The default value is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the previous response was incomplete (because there are more faces to retrieve), Amazon Rekognition Video returns a pagination token in the response. You can use this pagination token to retrieve the next set of faces.

Type: String

Length Constraints: Maximum length of 255.

Required: No

Response Syntax

```
{
   "Faces": [
      {
         "Face": {
            "AgeRange": {
               "High": number,
```

```
                "Low": number
            },
            "Beard": {
                "Confidence": number,
                "Value": boolean
            },
            "BoundingBox": {
                "Height": number,
                "Left": number,
                "Top": number,
                "Width": number
            },
            "Confidence": number,
            "Emotions": [
                {
                    "Confidence": number,
                    "Type": "string"
                }
            ],
            "EyeDirection": {
                "Confidence": number,
                "Pitch": number,
                "Yaw": number
            },
            "Eyeglasses": {
                "Confidence": number,
                "Value": boolean
            },
            "EyesOpen": {
                "Confidence": number,
                "Value": boolean
            },
            "FaceOccluded": {
                "Confidence": number,
                "Value": boolean
            },
            "Gender": {
                "Confidence": number,
                "Value": "string"
            },
            "Landmarks": [
                {
                    "Type": "string",
                    "X": number,
```

```
                "Y": number
            }
        ],
        "MouthOpen": {
            "Confidence": number,
            "Value": boolean
        },
        "Mustache": {
            "Confidence": number,
            "Value": boolean
        },
        "Pose": {
            "Pitch": number,
            "Roll": number,
            "Yaw": number
        },
        "Quality": {
            "Brightness": number,
            "Sharpness": number
        },
        "Smile": {
            "Confidence": number,
            "Value": boolean
        },
        "Sunglasses": {
            "Confidence": number,
            "Value": boolean
        }
    },
    "Timestamp": number
}
],
"JobId": "string",
"JobStatus": "string",
"JobTag": "string",
"NextToken": "string",
"StatusMessage": "string",
"Video": {
    "S3Object": {
        "Bucket": "string",
        "Name": "string",
        "Version": "string"
    }
},
```

```
    "VideoMetadata": {
       "Codec": "string",
       "ColorRange": "string",
       "DurationMillis": number,
       "Format": "string",
       "FrameHeight": number,
       "FrameRate": number,
       "FrameWidth": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Faces

An array of faces detected in the video. Each element contains a detected face's details and the time, in milliseconds from the start of the video, the face was detected.

Type: Array of FaceDetection objects

JobId

Job identifier for the face detection operation for which you want to obtain results. The job identifer is returned by an initial call to StartFaceDetection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobStatus

The current status of the face detection job.

Type: String

Valid Values: `IN_PROGRESS | SUCCEEDED | FAILED`

JobTag

A job identifier specified in the call to StartFaceDetection and returned in the job completion notification sent to your Amazon Simple Notification Service topic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

NextToken

If the response is truncated, Amazon Rekognition returns this token that you can use in the subsequent request to retrieve the next set of faces.

Type: String

Length Constraints: Maximum length of 255.

StatusMessage

If the job fails, `StatusMessage` provides a descriptive error message.

Type: String

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

VideoMetadata

Information about a video that Amazon Rekognition Video analyzed. `Videometadata` is returned in every page of paginated responses from a Amazon Rekognition video operation.

Type: VideoMetadata object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetFaceLivenessSessionResults

Service: Amazon Rekognition

Retrieves the results of a specific Face Liveness session. It requires the `sessionId` as input, which was created using `CreateFaceLivenessSession`. Returns the corresponding Face Liveness confidence score, a reference image that includes a face bounding box, and audit images that also contain face bounding boxes. The Face Liveness confidence score ranges from 0 to 100.

The number of audit images returned by `GetFaceLivenessSessionResults` is defined by the `AuditImagesLimit` paramater when calling `CreateFaceLivenessSession`. Reference images are always returned when possible.

Request Syntax

```
{
   "SessionId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

SessionId

A unique 128-bit UUID. This is used to uniquely identify the session and also acts as an idempotency token for all operations associated with the session.

Type: String

Length Constraints: Fixed length of 36.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: Yes

Response Syntax

```
{
```

```
    "AuditImages": [
       {
          "BoundingBox": {
             "Height": number,
             "Left": number,
             "Top": number,
             "Width": number
          },
          "Bytes": blob,
          "S3Object": {
             "Bucket": "string",
             "Name": "string",
             "Version": "string"
          }
       }
    ],
    "Challenge": {
       "Type": "string",
       "Version": "string"
    },
    "Confidence": number,
    "ReferenceImage": {
       "BoundingBox": {
          "Height": number,
          "Left": number,
          "Top": number,
          "Width": number
       },
       "Bytes": blob,
       "S3Object": {
          "Bucket": "string",
          "Name": "string",
          "Version": "string"
       }
    },
    "SessionId": "string",
    "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AuditImages

A set of images from the Face Liveness video that can be used for audit purposes. It includes a bounding box of the face and the Base64-encoded bytes that return an image. If the CreateFaceLivenessSession request included an OutputConfig argument, the image will be uploaded to an S3Object specified in the output configuration. If no Amazon S3 bucket is defined, raw bytes are sent instead.

Type: Array of AuditImage objects

Challenge

Contains information regarding the challenge type used for the Face Liveness check.

Type: Challenge object

Confidence

Probabalistic confidence score for if the person in the given video was live, represented as a float value between 0 to 100.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

ReferenceImage

A high-quality image from the Face Liveness video that can be used for face comparison or search. It includes a bounding box of the face and the Base64-encoded bytes that return an image. If the CreateFaceLivenessSession request included an OutputConfig argument, the image will be uploaded to an S3Object specified in the output configuration. In case the reference image is not returned, it's recommended to retry the Liveness check.

Type: AuditImage object

SessionId

The sessionId for which this request was called.

Type: String

Length Constraints: Fixed length of 36.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Status

Represents a status corresponding to the state of the session. Possible statuses are: CREATED, IN_PROGRESS, SUCCEEDED, FAILED, EXPIRED.

Type: String

Valid Values: `CREATED | IN_PROGRESS | SUCCEEDED | FAILED | EXPIRED`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

SessionNotFoundException

Occurs when a given sessionId is not found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetFaceSearch

Service: Amazon Rekognition

Gets the face search results for Amazon Rekognition Video face search started by StartFaceSearch. The search returns faces in a collection that match the faces of persons detected in a video. It also includes the time(s) that faces are matched in the video.

Face search in a video is an asynchronous operation. You start face search by calling to StartFaceSearch which returns a job identifier (`JobId`). When the search operation finishes, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic registered in the initial call to `StartFaceSearch`. To get the search results, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call `GetFaceSearch` and pass the job identifier (`JobId`) from the initial call to `StartFaceSearch`.

For more information, see Searching faces in a collection.

The search results are retured in an array, `Persons`, of PersonMatch objects. Each`PersonMatch` element contains details about the matching faces in the input collection, person information (facial attributes, bounding boxes, and person identifer) for the matched person, and the time the person was matched in the video.

> **Note**
>
> `GetFaceSearch` only returns the default facial attributes (`BoundingBox`, `Confidence`, `Landmarks`, `Pose`, and `Quality`). The other facial attributes listed in the `Face` object of the following response syntax are not returned. For more information, see FaceDetail.

By default, the `Persons` array is sorted by the time, in milliseconds from the start of the video, persons are matched. You can also sort by persons by specifying `INDEX` for the `SORTBY` input parameter.

Request Syntax

```
{
   "JobId": "string",
   "MaxResults": number,
   "NextToken": "string",
   "SortBy": "string"
```

```
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

JobId

The job identifer for the search request. You get the job identifier from an initial call to `StartFaceSearch`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

MaxResults

Maximum number of results to return per paginated call. The largest value you can specify is 1000. If you specify a value greater than 1000, a maximum of 1000 results is returned. The default value is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the previous response was incomplete (because there is more search results to retrieve), Amazon Rekognition Video returns a pagination token in the response. You can use this pagination token to retrieve the next set of search results.

Type: String

Length Constraints: Maximum length of 255.

Required: No

SortBy

Sort to use for grouping faces in the response. Use `TIMESTAMP` to group faces by the time that they are recognized. Use `INDEX` to sort by recognized faces.

Type: String

Valid Values: `INDEX | TIMESTAMP`

Required: No

Response Syntax

```
{
   "JobId": "string",
   "JobStatus": "string",
   "JobTag": "string",
   "NextToken": "string",
   "Persons": [
      {
         "FaceMatches": [
            {
               "Face": {
                  "BoundingBox": {
                     "Height": number,
                     "Left": number,
                     "Top": number,
                     "Width": number
                  },
                  "Confidence": number,
                  "ExternalImageId": "string",
                  "FaceId": "string",
                  "ImageId": "string",
                  "IndexFacesModelVersion": "string",
                  "UserId": "string"
               },
               "Similarity": number
            }
         ],
         "Person": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
```

```
                "Top": number,
                "Width": number
            },
            "Face": {
                "AgeRange": {
                    "High": number,
                    "Low": number
                },
                "Beard": {
                    "Confidence": number,
                    "Value": boolean
                },
                "BoundingBox": {
                    "Height": number,
                    "Left": number,
                    "Top": number,
                    "Width": number
                },
                "Confidence": number,
                "Emotions": [
                    {
                        "Confidence": number,
                        "Type": "string"
                    }
                ],
                "EyeDirection": {
                    "Confidence": number,
                    "Pitch": number,
                    "Yaw": number
                },
                "Eyeglasses": {
                    "Confidence": number,
                    "Value": boolean
                },
                "EyesOpen": {
                    "Confidence": number,
                    "Value": boolean
                },
                "FaceOccluded": {
                    "Confidence": number,
                    "Value": boolean
                },
                "Gender": {
                    "Confidence": number,
```

```
                "Value": "string"
            },
            "Landmarks": [
                {
                    "Type": "string",
                    "X": number,
                    "Y": number
                }
            ],
            "MouthOpen": {
                "Confidence": number,
                "Value": boolean
            },
            "Mustache": {
                "Confidence": number,
                "Value": boolean
            },
            "Pose": {
                "Pitch": number,
                "Roll": number,
                "Yaw": number
            },
            "Quality": {
                "Brightness": number,
                "Sharpness": number
            },
            "Smile": {
                "Confidence": number,
                "Value": boolean
            },
            "Sunglasses": {
                "Confidence": number,
                "Value": boolean
            }
        },
        "Index": number
    },
    "Timestamp": number
  }
],
"StatusMessage": "string",
"Video": {
  "S3Object": {
    "Bucket": "string",
```

```
            "Name": "string",
            "Version": "string"
        }
    },
    "VideoMetadata": {
        "Codec": "string",
        "ColorRange": "string",
        "DurationMillis": number,
        "Format": "string",
        "FrameHeight": number,
        "FrameRate": number,
        "FrameWidth": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

Job identifier for the face search operation for which you want to obtain results. The job identifer is returned by an initial call to StartFaceSearch.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobStatus

The current status of the face search job.

Type: String

Valid Values: `IN_PROGRESS | SUCCEEDED | FAILED`

JobTag

A job identifier specified in the call to StartFaceSearch and returned in the job completion notification sent to your Amazon Simple Notification Service topic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

NextToken

If the response is truncated, Amazon Rekognition Video returns this token that you can use in the subsequent request to retrieve the next set of search results.

Type: String

Length Constraints: Maximum length of 255.

Persons

An array of persons, PersonMatch, in the video whose face(s) match the face(s) in an Amazon Rekognition collection. It also includes time information for when persons are matched in the video. You specify the input collection in an initial call to `StartFaceSearch`. Each `Persons` element includes a time the person was matched, face match details (`FaceMatches`) for matching faces in the collection, and person information (`Person`) for the matched person.

Type: Array of PersonMatch objects

StatusMessage

If the job fails, `StatusMessage` provides a descriptive error message.

Type: String

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

VideoMetadata

Information about a video that Amazon Rekognition analyzed. `Videometadata` is returned in every page of paginated responses from a Amazon Rekognition Video operation.

Type: VideoMetadata object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetLabelDetection

Service: Amazon Rekognition

Gets the label detection results of a Amazon Rekognition Video analysis started by StartLabelDetection.

The label detection operation is started by a call to StartLabelDetection which returns a job identifier (`JobId`). When the label detection operation finishes, Amazon Rekognition publishes a completion status to the Amazon Simple Notification Service topic registered in the initial call to `StartlabelDetection`.

To get the results of the label detection operation, first check that the status value published to the Amazon SNS topic is `SUCCEEDED`. If so, call GetLabelDetection and pass the job identifier (`JobId`) from the initial call to `StartLabelDetection`.

`GetLabelDetection` returns an array of detected labels (`Labels`) sorted by the time the labels were detected. You can also sort by the label name by specifying `NAME` for the `SortBy` input parameter. If there is no `NAME` specified, the default sort is by timestamp.

You can select how results are aggregated by using the `AggregateBy` input parameter. The default aggregation method is `TIMESTAMPS`. You can also aggregate by `SEGMENTS`, which aggregates all instances of labels detected in a given segment.

The returned Labels array may include the following attributes:

- Name - The name of the detected label.
- Confidence - The level of confidence in the label assigned to a detected object.
- Parents - The ancestor labels for a detected label. GetLabelDetection returns a hierarchical taxonomy of detected labels. For example, a detected car might be assigned the label car. The label car has two parent labels: Vehicle (its parent) and Transportation (its grandparent). The response includes the all ancestors for a label, where every ancestor is a unique label. In the previous example, Car, Vehicle, and Transportation are returned as unique labels in the response.
- Aliases - Possible Aliases for the label.
- Categories - The label categories that the detected label belongs to.
- BoundingBox — Bounding boxes are described for all instances of detected common object labels, returned in an array of Instance objects. An Instance object contains a BoundingBox object, describing the location of the label on the input image. It also includes the confidence for the accuracy of the detected bounding box.

- Timestamp - Time, in milliseconds from the start of the video, that the label was detected. For aggregation by `SEGMENTS`, the `StartTimestampMillis`, `EndTimestampMillis`, and `DurationMillis` structures are what define a segment. Although the “Timestamp” structure is still returned with each label, its value is set to be the same as `StartTimestampMillis`.

Timestamp and Bounding box information are returned for detected Instances, only if aggregation is done by `TIMESTAMPS`. If aggregating by `SEGMENTS`, information about detected instances isn’t returned.

The version of the label model used for the detection is also returned.

Note `DominantColors` isn't returned for `Instances`, although it is shown as part of the response in the sample seen below.

Use `MaxResults` parameter to limit the number of labels returned. If there are more results than specified in `MaxResults`, the value of `NextToken` in the operation response contains a pagination token for getting the next set of results. To get the next page of results, call `GetlabelDetection` and populate the `NextToken` request parameter with the token value returned from the previous call to `GetLabelDetection`.

If you are retrieving results while using the Amazon Simple Notification Service, note that you will receive an "ERROR" notification if the job encounters an issue.

Request Syntax

```
{
   "AggregateBy": "string",
   "JobId": "string",
   "MaxResults": number,
   "NextToken": "string",
   "SortBy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

AggregateBy

Defines how to aggregate the returned results. Results can be aggregated by timestamps or segments.

Type: String

Valid Values: `TIMESTAMPS | SEGMENTS`

Required: No

JobId

Job identifier for the label detection operation for which you want results returned. You get the job identifer from an initial call to `StartlabelDetection`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

MaxResults

Maximum number of results to return per paginated call. The largest value you can specify is 1000. If you specify a value greater than 1000, a maximum of 1000 results is returned. The default value is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the previous response was incomplete (because there are more labels to retrieve), Amazon Rekognition Video returns a pagination token in the response. You can use this pagination token to retrieve the next set of labels.

Type: String

Length Constraints: Maximum length of 255.

Required: No

SortBy

Sort to use for elements in the `Labels` array. Use `TIMESTAMP` to sort array elements by the time labels are detected. Use `NAME` to alphabetically group elements for a label together. Within each label group, the array element are sorted by detection confidence. The default sort is by `TIMESTAMP`.

Type: String

Valid Values: `NAME | TIMESTAMP`

Required: No

Response Syntax

```
{
   "GetRequestMetadata": {
      "AggregateBy": "string",
      "SortBy": "string"
   },
   "JobId": "string",
   "JobStatus": "string",
   "JobTag": "string",
   "LabelModelVersion": "string",
   "Labels": [
      {
         "DurationMillis": number,
         "EndTimestampMillis": number,
         "Label": {
            "Aliases": [
               {
                  "Name": "string"
               }
            ],
            "Categories": [
               {
                  "Name": "string"
               }
            ],
            "Confidence": number,
            "Instances": [
               {
                  "BoundingBox": {
```

```
                    "Height": number,
                    "Left": number,
                    "Top": number,
                    "Width": number
                },
                "Confidence": number,
                "DominantColors": [
                    {
                        "Blue": number,
                        "CSSColor": "string",
                        "Green": number,
                        "HexCode": "string",
                        "PixelPercent": number,
                        "Red": number,
                        "SimplifiedColor": "string"
                    }
                ]
            }
        ],
        "Name": "string",
        "Parents": [
            {
                "Name": "string"
            }
        ]
    },
    "StartTimestampMillis": number,
    "Timestamp": number
  }
],
"NextToken": "string",
"StatusMessage": "string",
"Video": {
  "S3Object": {
    "Bucket": "string",
    "Name": "string",
    "Version": "string"
  }
},
"VideoMetadata": {
  "Codec": "string",
  "ColorRange": "string",
  "DurationMillis": number,
  "Format": "string",
```

```
        "FrameHeight": number,
        "FrameRate": number,
        "FrameWidth": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GetRequestMetadata

Information about the paramters used when getting a response. Includes information on aggregation and sorting methods.

Type: GetLabelDetectionRequestMetadata object

JobId

Job identifier for the label detection operation for which you want to obtain results. The job identifer is returned by an initial call to StartLabelDetection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobStatus

The current status of the label detection job.

Type: String

Valid Values: `IN_PROGRESS | SUCCEEDED | FAILED`

JobTag

A job identifier specified in the call to StartLabelDetection and returned in the job completion notification sent to your Amazon Simple Notification Service topic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

LabelModelVersion

Version number of the label detection model that was used to detect labels.

Type: String

Labels

An array of labels detected in the video. Each element contains the detected label and the time, in milliseconds from the start of the video, that the label was detected.

Type: Array of LabelDetection objects

NextToken

If the response is truncated, Amazon Rekognition Video returns this token that you can use in the subsequent request to retrieve the next set of labels.

Type: String

Length Constraints: Maximum length of 255.

StatusMessage

If the job fails, `StatusMessage` provides a descriptive error message.

Type: String

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

VideoMetadata

Information about a video that Amazon Rekognition Video analyzed. `Videometadata` is returned in every page of paginated responses from a Amazon Rekognition video operation.

Type: VideoMetadata object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetMediaAnalysisJob

Service: Amazon Rekognition

Retrieves the results for a given media analysis job. Takes a `JobId` returned by StartMediaAnalysisJob.

Request Syntax

```
{
   "JobId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

JobId

Unique identifier for the media analysis job for which you want to retrieve results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

Response Syntax

```
{
   "CompletionTimestamp": number,
   "CreationTimestamp": number,
   "FailureDetails": {
      "Code": "string",
      "Message": "string"
   },
   "Input": {
```

```
        "S3Object": {
            "Bucket": "string",
            "Name": "string",
            "Version": "string"
        }
    },
    "JobId": "string",
    "JobName": "string",
    "KmsKeyId": "string",
    "ManifestSummary": {
        "S3Object": {
            "Bucket": "string",
            "Name": "string",
            "Version": "string"
        }
    },
    "OperationsConfig": {
        "DetectModerationLabels": {
            "MinConfidence": number,
            "ProjectVersion": "string"
        }
    },
    "OutputConfig": {
        "S3Bucket": "string",
        "S3KeyPrefix": "string"
    },
    "Results": {
        "ModelVersions": {
            "Moderation": "string"
        },
        "S3Object": {
            "Bucket": "string",
            "Name": "string",
            "Version": "string"
        }
    },
    "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CompletionTimestamp

The Unix date and time when the job finished.

Type: Timestamp

CreationTimestamp

The Unix date and time when the job was started.

Type: Timestamp

FailureDetails

Details about the error that resulted in failure of the job.

Type: MediaAnalysisJobFailureDetails object

Input

Reference to the input manifest that was provided in the job creation request.

Type: MediaAnalysisInput object

JobId

The identifier for the media analysis job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobName

The name of the media analysis job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.\-]+`

KmsKeyId

KMS Key that was provided in the creation request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

ManifestSummary

The summary manifest provides statistics on input manifest and errors identified in the input manifest.

Type: MediaAnalysisManifestSummary object

OperationsConfig

Operation configurations that were provided during job creation.

Type: MediaAnalysisOperationsConfig object

OutputConfig

Output configuration that was provided in the creation request.

Type: MediaAnalysisOutputConfig object

Results

Output manifest that contains prediction results.

Type: MediaAnalysisResults object

Status

The current status of the media analysis job.

Type: String

Valid Values: `CREATED | QUEUED | IN_PROGRESS | SUCCEEDED | FAILED`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetPersonTracking

Service: Amazon Rekognition

> **Note**
>
> *End of support notice:* On October 31, 2025, AWS will discontinue support for Amazon Rekognition People Pathing. After October 31, 2025, you will no longer be able to use the Rekognition People Pathing capability. For more information, visit this blog post.

Gets the path tracking results of a Amazon Rekognition Video analysis started by StartPersonTracking.

The person path tracking operation is started by a call to `StartPersonTracking` which returns a job identifier (`JobId`). When the operation finishes, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic registered in the initial call to `StartPersonTracking`.

To get the results of the person path tracking operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call GetPersonTracking and pass the job identifier (`JobId`) from the initial call to `StartPersonTracking`.

`GetPersonTracking` returns an array, `Persons`, of tracked persons and the time(s) their paths were tracked in the video.

> **Note**
>
> `GetPersonTracking` only returns the default facial attributes (`BoundingBox`, `Confidence`, `Landmarks`, `Pose`, and `Quality`). The other facial attributes listed in the `Face` object of the following response syntax are not returned.
> For more information, see FaceDetail.

By default, the array is sorted by the time(s) a person's path is tracked in the video. You can sort by tracked persons by specifying `INDEX` for the `SortBy` input parameter.

Use the `MaxResults` parameter to limit the number of items returned. If there are more results than specified in `MaxResults`, the value of `NextToken` in the operation response contains

a pagination token for getting the next set of results. To get the next page of results, call `GetPersonTracking` and populate the `NextToken` request parameter with the token value returned from the previous call to `GetPersonTracking`.

Request Syntax

```
{
   "JobId": "string",
   "MaxResults": number,
   "NextToken": "string",
   "SortBy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

JobId

The identifier for a job that tracks persons in a video. You get the `JobId` from a call to `StartPersonTracking`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

MaxResults

Maximum number of results to return per paginated call. The largest value you can specify is 1000. If you specify a value greater than 1000, a maximum of 1000 results is returned. The default value is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the previous response was incomplete (because there are more persons to retrieve), Amazon Rekognition Video returns a pagination token in the response. You can use this pagination token to retrieve the next set of persons.

Type: String

Length Constraints: Maximum length of 255.

Required: No

SortBy

Sort to use for elements in the `Persons` array. Use `TIMESTAMP` to sort array elements by the time persons are detected. Use `INDEX` to sort by the tracked persons. If you sort by `INDEX`, the array elements for each person are sorted by detection confidence. The default sort is by `TIMESTAMP`.

Type: String

Valid Values: `INDEX | TIMESTAMP`

Required: No

Response Syntax

```
{
   "JobId": "string",
   "JobStatus": "string",
   "JobTag": "string",
   "NextToken": "string",
   "Persons": [
      {
         "Person": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Face": {
               "AgeRange": {
```

```
                "High": number,
                "Low": number
            },
            "Beard": {
                "Confidence": number,
                "Value": boolean
            },
            "BoundingBox": {
                "Height": number,
                "Left": number,
                "Top": number,
                "Width": number
            },
            "Confidence": number,
            "Emotions": [
                {
                    "Confidence": number,
                    "Type": "string"
                }
            ],
            "EyeDirection": {
                "Confidence": number,
                "Pitch": number,
                "Yaw": number
            },
            "Eyeglasses": {
                "Confidence": number,
                "Value": boolean
            },
            "EyesOpen": {
                "Confidence": number,
                "Value": boolean
            },
            "FaceOccluded": {
                "Confidence": number,
                "Value": boolean
            },
            "Gender": {
                "Confidence": number,
                "Value": "string"
            },
            "Landmarks": [
                {
                    "Type": "string",
```

```
                    "X": number,
                    "Y": number
                }
            ],
            "MouthOpen": {
                "Confidence": number,
                "Value": boolean
            },
            "Mustache": {
                "Confidence": number,
                "Value": boolean
            },
            "Pose": {
                "Pitch": number,
                "Roll": number,
                "Yaw": number
            },
            "Quality": {
                "Brightness": number,
                "Sharpness": number
            },
            "Smile": {
                "Confidence": number,
                "Value": boolean
            },
            "Sunglasses": {
                "Confidence": number,
                "Value": boolean
            }
        },
        "Index": number
    },
    "Timestamp": number
  }
],
"StatusMessage": "string",
"Video": {
  "S3Object": {
    "Bucket": "string",
    "Name": "string",
    "Version": "string"
  }
},
"VideoMetadata": {
```

```
        "Codec": "string",
        "ColorRange": "string",
        "DurationMillis": number,
        "Format": "string",
        "FrameHeight": number,
        "FrameRate": number,
        "FrameWidth": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

Job identifier for the person tracking operation for which you want to obtain results. The job identifer is returned by an initial call to StartPersonTracking.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobStatus

The current status of the person tracking job.

Type: String

Valid Values: `IN_PROGRESS | SUCCEEDED | FAILED`

JobTag

A job identifier specified in the call to StartCelebrityRecognition and returned in the job completion notification sent to your Amazon Simple Notification Service topic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

NextToken

If the response is truncated, Amazon Rekognition Video returns this token that you can use in the subsequent request to retrieve the next set of persons.

Type: String

Length Constraints: Maximum length of 255.

Persons

An array of the persons detected in the video and the time(s) their path was tracked throughout the video. An array element will exist for each time a person's path is tracked.

Type: Array of PersonDetection objects

StatusMessage

If the job fails, `StatusMessage` provides a descriptive error message.

Type: String

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

VideoMetadata

Information about a video that Amazon Rekognition Video analyzed. `Videometadata` is returned in every page of paginated responses from a Amazon Rekognition Video operation.

Type: VideoMetadata object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSegmentDetection

Service: Amazon Rekognition

Gets the segment detection results of a Amazon Rekognition Video analysis started by StartSegmentDetection.

Segment detection with Amazon Rekognition Video is an asynchronous operation. You start segment detection by calling StartSegmentDetection which returns a job identifier (`JobId`). When the segment detection operation finishes, Amazon Rekognition publishes a completion status to the Amazon Simple Notification Service topic registered in the initial call to `StartSegmentDetection`. To get the results of the segment detection operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. if so, call `GetSegmentDetection` and pass the job identifier (`JobId`) from the initial call of `StartSegmentDetection`.

`GetSegmentDetection` returns detected segments in an array (`Segments`) of SegmentDetection objects. `Segments` is sorted by the segment types specified in the `SegmentTypes` input parameter of `StartSegmentDetection`. Each element of the array includes the detected segment, the precentage confidence in the acuracy of the detected segment, the type of the segment, and the frame in which the segment was detected.

Use `SelectedSegmentTypes` to find out the type of segment detection requested in the call to `StartSegmentDetection`.

Use the `MaxResults` parameter to limit the number of segment detections returned. If there are more results than specified in `MaxResults`, the value of `NextToken` in the operation response contains a pagination token for getting the next set of results. To get the next page of results, call `GetSegmentDetection` and populate the `NextToken` request parameter with the token value returned from the previous call to `GetSegmentDetection`.

For more information, see Detecting video segments in stored video.

Request Syntax

```
{
   "JobId": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

JobId

Job identifier for the text detection operation for which you want results returned. You get the job identifer from an initial call to `StartSegmentDetection`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

MaxResults

Maximum number of results to return per paginated call. The largest value you can specify is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the response is truncated, Amazon Rekognition Video returns this token that you can use in the subsequent request to retrieve the next set of text.

Type: String

Length Constraints: Maximum length of 255.

Required: No

Response Syntax

```
{
   "AudioMetadata": [
```

```
        {
            "Codec": "string",
            "DurationMillis": number,
            "NumberOfChannels": number,
            "SampleRate": number
        }
    ],
    "JobId": "string",
    "JobStatus": "string",
    "JobTag": "string",
    "NextToken": "string",
    "Segments": [
        {
            "DurationFrames": number,
            "DurationMillis": number,
            "DurationSMPTE": "string",
            "EndFrameNumber": number,
            "EndTimecodeSMPTE": "string",
            "EndTimestampMillis": number,
            "ShotSegment": {
                "Confidence": number,
                "Index": number
            },
            "StartFrameNumber": number,
            "StartTimecodeSMPTE": "string",
            "StartTimestampMillis": number,
            "TechnicalCueSegment": {
                "Confidence": number,
                "Type": "string"
            },
            "Type": "string"
        }
    ],
    "SelectedSegmentTypes": [
        {
            "ModelVersion": "string",
            "Type": "string"
        }
    ],
    "StatusMessage": "string",
    "Video": {
        "S3Object": {
            "Bucket": "string",
            "Name": "string",
```

```
            "Version": "string"
        }
    },
    "VideoMetadata": [
        {
            "Codec": "string",
            "ColorRange": "string",
            "DurationMillis": number,
            "Format": "string",
            "FrameHeight": number,
            "FrameRate": number,
            "FrameWidth": number
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AudioMetadata

An array of AudioMetadata objects. There can be multiple audio streams. Each `AudioMetadata` object contains metadata for a single audio stream. Audio information in an `AudioMetadata` objects includes the audio codec, the number of audio channels, the duration of the audio stream, and the sample rate. Audio metadata is returned in each page of information returned by `GetSegmentDetection`.

Type: Array of AudioMetadata objects

JobId

Job identifier for the segment detection operation for which you want to obtain results. The job identifer is returned by an initial call to StartSegmentDetection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobStatus

Current status of the segment detection job.

Type: String

Valid Values: `IN_PROGRESS | SUCCEEDED | FAILED`

JobTag

A job identifier specified in the call to StartSegmentDetection and returned in the job completion notification sent to your Amazon Simple Notification Service topic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

NextToken

If the previous response was incomplete (because there are more labels to retrieve), Amazon Rekognition Video returns a pagination token in the response. You can use this pagination token to retrieve the next set of text.

Type: String

Length Constraints: Maximum length of 255.

Segments

An array of segments detected in a video. The array is sorted by the segment types (TECHNICAL_CUE or SHOT) specified in the `SegmentTypes` input parameter of `StartSegmentDetection`. Within each segment type the array is sorted by timestamp values.

Type: Array of SegmentDetection objects

SelectedSegmentTypes

An array containing the segment types requested in the call to `StartSegmentDetection`.

Type: Array of SegmentTypeInfo objects

StatusMessage

If the job fails, `StatusMessage` provides a descriptive error message.

Type: String

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

VideoMetadata

Currently, Amazon Rekognition Video returns a single VideoMetadata object in the `VideoMetadata` array. The object contains information about the video stream in the input file that Amazon Rekognition Video chose to analyze. The `VideoMetadata` object includes the video codec, video format and other information. Video metadata is returned in each page of information returned by `GetSegmentDetection`.

Type: Array of VideoMetadata objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetTextDetection

Service: Amazon Rekognition

Gets the text detection results of a Amazon Rekognition Video analysis started by StartTextDetection.

Text detection with Amazon Rekognition Video is an asynchronous operation. You start text detection by calling StartTextDetection which returns a job identifier (`JobId`) When the text detection operation finishes, Amazon Rekognition publishes a completion status to the Amazon Simple Notification Service topic registered in the initial call to `StartTextDetection`. To get the results of the text detection operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. if so, call `GetTextDetection` and pass the job identifier (`JobId`) from the initial call of `StartLabelDetection`.

`GetTextDetection` returns an array of detected text (`TextDetections`) sorted by the time the text was detected, up to 100 words per frame of video.

Each element of the array includes the detected text, the precentage confidence in the acuracy of the detected text, the time the text was detected, bounding box information for where the text was located, and unique identifiers for words and their lines.

Use MaxResults parameter to limit the number of text detections returned. If there are more results than specified in `MaxResults`, the value of `NextToken` in the operation response contains a pagination token for getting the next set of results. To get the next page of results, call `GetTextDetection` and populate the `NextToken` request parameter with the token value returned from the previous call to `GetTextDetection`.

Request Syntax

```
{
   "JobId": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

JobId

Job identifier for the text detection operation for which you want results returned. You get the job identifer from an initial call to `StartTextDetection`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

MaxResults

Maximum number of results to return per paginated call. The largest value you can specify is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the previous response was incomplete (because there are more labels to retrieve), Amazon Rekognition Video returns a pagination token in the response. You can use this pagination token to retrieve the next set of text.

Type: String

Length Constraints: Maximum length of 255.

Required: No

Response Syntax

```
{
   "JobId": "string",
   "JobStatus": "string",
   "JobTag": "string",
   "NextToken": "string",
   "StatusMessage": "string",
   "TextDetections": [
```

```
        {
            "TextDetection": {
                "Confidence": number,
                "DetectedText": "string",
                "Geometry": {
                    "BoundingBox": {
                        "Height": number,
                        "Left": number,
                        "Top": number,
                        "Width": number
                    },
                    "Polygon": [
                        {
                            "X": number,
                            "Y": number
                        }
                    ]
                },
                "Id": number,
                "ParentId": number,
                "Type": "string"
            },
            "Timestamp": number
        }
    ],
    "TextModelVersion": "string",
    "Video": {
        "S3Object": {
            "Bucket": "string",
            "Name": "string",
            "Version": "string"
        }
    },
    "VideoMetadata": {
        "Codec": "string",
        "ColorRange": "string",
        "DurationMillis": number,
        "Format": "string",
        "FrameHeight": number,
        "FrameRate": number,
        "FrameWidth": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

Job identifier for the text detection operation for which you want to obtain results. The job identifer is returned by an initial call to StartTextDetection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

JobStatus

Current status of the text detection job.

Type: String

Valid Values: `IN_PROGRESS | SUCCEEDED | FAILED`

JobTag

A job identifier specified in the call to StartTextDetection and returned in the job completion notification sent to your Amazon Simple Notification Service topic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

NextToken

If the response is truncated, Amazon Rekognition Video returns this token that you can use in the subsequent request to retrieve the next set of text.

Type: String

Length Constraints: Maximum length of 255.

StatusMessage

If the job fails, `StatusMessage` provides a descriptive error message.

Type: String

TextDetections

An array of text detected in the video. Each element contains the detected text, the time in milliseconds from the start of the video that the text was detected, and where it was detected on the screen.

Type: Array of TextDetectionResult objects

TextModelVersion

Version number of the text detection model that was used to detect text.

Type: String

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

VideoMetadata

Information about a video that Amazon Rekognition analyzed. `Videometadata` is returned in every page of paginated responses from a Amazon Rekognition video operation.

Type: VideoMetadata object

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

IndexFaces

Service: Amazon Rekognition

Detects faces in the input image and adds them to the specified collection.

Amazon Rekognition doesn't save the actual faces that are detected. Instead, the underlying detection algorithm first detects the faces in the input image. For each face, the algorithm extracts facial features into a feature vector, and stores it in the backend database. Amazon Rekognition uses feature vectors when it performs face match and search operations using the SearchFaces and SearchFacesByImage operations.

For more information, see Adding faces to a collection.

To get the number of faces in a collection, call DescribeCollection.

If you're using version 1.0 of the face detection model, `IndexFaces` indexes the 15 largest faces in the input image. Later versions of the face detection model index the 100 largest faces in the input image.

If you're using version 4 or later of the face model, image orientation information is not returned in the `OrientationCorrection` field.

To determine which version of the model you're using, call DescribeCollection and supply the collection ID. You can also get the model version from the value of `FaceModelVersion` in the response from `IndexFaces`

For more information, see Model versioning.

If you provide the optional `ExternalImageId` for the input image you provided, Amazon Rekognition associates this ID with all faces that it detects. When you call the ListFaces operation, the response returns the external ID. You can use this external image ID to create a client-side index to associate the faces with each image. You can then use the index to find all faces in an image.

You can specify the maximum number of faces to index with the `MaxFaces` input parameter. This is useful when you want to index the largest faces in an image and don't want to index smaller faces, such as those belonging to people standing in the background.

The `QualityFilter` input parameter allows you to filter out detected faces that don’t meet a required quality bar. The quality bar is based on a variety of common use cases. By default, `IndexFaces` chooses the quality bar that's used to filter faces. You can also explicitly choose the

quality bar. Use `QualityFilter`, to set the quality bar by specifying `LOW`, `MEDIUM`, or `HIGH`. If you do not want to filter detected faces, specify `NONE`.

> **Note**
>
> To use quality filtering, you need a collection associated with version 3 of the face model or higher. To get the version of the face model associated with a collection, call DescribeCollection.

Information about faces detected in an image, but not indexed, is returned in an array of UnindexedFace objects, `UnindexedFaces`. Faces aren't indexed for reasons such as:

- The number of faces detected exceeds the value of the `MaxFaces` request parameter.
- The face is too small compared to the image dimensions.
- The face is too blurry.
- The image is too dark.
- The face has an extreme pose.
- The face doesn’t have enough detail to be suitable for face search.

In response, the `IndexFaces` operation returns an array of metadata for all detected faces, `FaceRecords`. This includes:

- The bounding box, `BoundingBox`, of the detected face.
- A confidence value, `Confidence`, which indicates the confidence that the bounding box contains a face.
- A face ID, `FaceId`, assigned by the service for each face that's detected and stored.
- An image ID, `ImageId`, assigned by the service for the input image.

If you request `ALL` or specific facial attributes (e.g., `FACE_OCCLUDED`) by using the detectionAttributes parameter, Amazon Rekognition returns detailed facial attributes, such as facial landmarks (for example, location of eye and mouth), facial occlusion, and other facial attributes.

If you provide the same image, specify the same collection, and use the same external ID in the `IndexFaces` operation, Amazon Rekognition doesn't save duplicate face metadata.

The input image is passed either as base64-encoded image bytes, or as a reference to an image in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes isn't supported. The image must be formatted as a PNG or JPEG file.

This operation requires permissions to perform the `rekognition:IndexFaces` action.

Request Syntax

```
{
   "CollectionId": "string",
   "DetectionAttributes": [ "string" ],
   "ExternalImageId": "string",
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "MaxFaces": number,
   "QualityFilter": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

The ID of an existing collection to which you want to add the faces that are detected in the input images.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

DetectionAttributes

An array of facial attributes you want to be returned. A `DEFAULT` subset of facial attributes - `BoundingBox`, `Confidence`, `Pose`, `Quality`, and `Landmarks` - will always be returned. You can request for specific facial attributes (in addition to the default list) - by using `["DEFAULT", "FACE_OCCLUDED"]` or just `["FACE_OCCLUDED"]`. You can request for all facial attributes by using `["ALL"]`. Requesting more attributes may increase response time.

If you provide both, `["ALL", "DEFAULT"]`, the service uses a logical AND operator to determine which attributes to return (in this case, all attributes).

Type: Array of strings

Valid Values: `DEFAULT | ALL | AGE_RANGE | BEARD | EMOTIONS | EYE_DIRECTION | EYEGLASSES | EYES_OPEN | GENDER | MOUTH_OPEN | MUSTACHE | FACE_OCCLUDED | SMILE | SUNGLASSES`

Required: No

ExternalImageId

The ID you want to assign to all the faces detected in the image.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

Image

The input image as base64-encoded bytes or an S3 object. If you use the AWS CLI to call Amazon Rekognition operations, passing base64-encoded image bytes isn't supported.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

MaxFaces

The maximum number of faces to index. The value of `MaxFaces` must be greater than or equal to 1. `IndexFaces` returns no more than 100 detected faces in an image, even if you specify a larger value for `MaxFaces`.

If `IndexFaces` detects more faces than the value of `MaxFaces`, the faces with the lowest quality are filtered out first. If there are still more faces than the value of `MaxFaces`, the faces with the smallest bounding boxes are filtered out (up to the number that's needed to satisfy the value of `MaxFaces`). Information about the unindexed faces is available in the `UnindexedFaces` array.

The faces that are returned by `IndexFaces` are sorted by the largest face bounding box size to the smallest size, in descending order.

`MaxFaces` can be used with a collection associated with any version of the face model.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

QualityFilter

A filter that specifies a quality bar for how much filtering is done to identify faces. Filtered faces aren't indexed. If you specify `AUTO`, Amazon Rekognition chooses the quality bar. If you specify `LOW`, `MEDIUM`, or `HIGH`, filtering removes all faces that don’t meet the chosen quality bar. The default value is `AUTO`. The quality bar is based on a variety of common use cases. Low-quality detections can occur for a number of reasons. Some examples are an object that's misidentified as a face, a face that's too blurry, or a face with a pose that's too extreme to use. If you specify `NONE`, no filtering is performed.

To use quality filtering, the collection you are using must be associated with version 3 of the face model or higher.

Type: String

Valid Values: `NONE | AUTO | LOW | MEDIUM | HIGH`

Required: No

Response Syntax

```
{
   "FaceModelVersion": "string",
   "FaceRecords": [
      {
         "Face": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "ExternalImageId": "string",
            "FaceId": "string",
            "ImageId": "string",
            "IndexFacesModelVersion": "string",
            "UserId": "string"
         },
         "FaceDetail": {
            "AgeRange": {
               "High": number,
               "Low": number
            },
            "Beard": {
               "Confidence": number,
               "Value": boolean
            },
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "Emotions": [
               {
                  "Confidence": number,
                  "Type": "string"
               }
            ],
            "EyeDirection": {
```

```
                "Confidence": number,
                "Pitch": number,
                "Yaw": number
            },
            "Eyeglasses": {
                "Confidence": number,
                "Value": boolean
            },
            "EyesOpen": {
                "Confidence": number,
                "Value": boolean
            },
            "FaceOccluded": {
                "Confidence": number,
                "Value": boolean
            },
            "Gender": {
                "Confidence": number,
                "Value": "string"
            },
            "Landmarks": [
                {
                    "Type": "string",
                    "X": number,
                    "Y": number
                }
            ],
            "MouthOpen": {
                "Confidence": number,
                "Value": boolean
            },
            "Mustache": {
                "Confidence": number,
                "Value": boolean
            },
            "Pose": {
                "Pitch": number,
                "Roll": number,
                "Yaw": number
            },
            "Quality": {
                "Brightness": number,
                "Sharpness": number
            },
```

```
            "Smile": {
               "Confidence": number,
               "Value": boolean
            },
            "Sunglasses": {
               "Confidence": number,
               "Value": boolean
            }
         }
      }
   ],
   "OrientationCorrection": "string",
   "UnindexedFaces": [
      {
         "FaceDetail": {
            "AgeRange": {
               "High": number,
               "Low": number
            },
            "Beard": {
               "Confidence": number,
               "Value": boolean
            },
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "Emotions": [
               {
                  "Confidence": number,
                  "Type": "string"
               }
            ],
            "EyeDirection": {
               "Confidence": number,
               "Pitch": number,
               "Yaw": number
            },
            "Eyeglasses": {
               "Confidence": number,
               "Value": boolean
```

```
            },
            "EyesOpen": {
               "Confidence": number,
               "Value": boolean
            },
            "FaceOccluded": {
               "Confidence": number,
               "Value": boolean
            },
            "Gender": {
               "Confidence": number,
               "Value": "string"
            },
            "Landmarks": [
               {
                  "Type": "string",
                  "X": number,
                  "Y": number
               }
            ],
            "MouthOpen": {
               "Confidence": number,
               "Value": boolean
            },
            "Mustache": {
               "Confidence": number,
               "Value": boolean
            },
            "Pose": {
               "Pitch": number,
               "Roll": number,
               "Yaw": number
            },
            "Quality": {
               "Brightness": number,
               "Sharpness": number
            },
            "Smile": {
               "Confidence": number,
               "Value": boolean
            },
            "Sunglasses": {
               "Confidence": number,
               "Value": boolean
```

```
            }
         },
         "Reasons": [ "string" ]
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaceModelVersion

The version number of the face detection model that's associated with the input collection (`CollectionId`).

Type: String

FaceRecords

An array of faces detected and added to the collection. For more information, see Searching faces in a collection.

Type: Array of FaceRecord objects

OrientationCorrection

If your collection is associated with a face detection model that's later than version 3.0, the value of `OrientationCorrection` is always null and no orientation information is returned.

If your collection is associated with a face detection model that's version 3.0 or earlier, the following applies:

- If the input image is in .jpeg format, it might contain exchangeable image file format (Exif) metadata that includes the image's orientation. Amazon Rekognition uses this orientation information to perform image correction - the bounding box coordinates are translated to represent object locations after the orientation information in the Exif metadata is used to correct the image orientation. Images in .png format don't contain Exif metadata. The value of `OrientationCorrection` is null.
- If the image doesn't contain orientation information in its Exif metadata, Amazon Rekognition returns an estimated orientation (ROTATE_0, ROTATE_90, ROTATE_180,

ROTATE_270). Amazon Rekognition doesn't perform image correction for images. The bounding box coordinates aren't translated and represent the object locations before the image is rotated.

Bounding box information is returned in the `FaceRecords` array. You can get the version of the face detection model by calling DescribeCollection.

Type: String

Valid Values: `ROTATE_0 | ROTATE_90 | ROTATE_180 | ROTATE_270`

UnindexedFaces

An array of faces that were detected in the image but weren't indexed. They weren't indexed because the quality filter identified them as low quality, or the `MaxFaces` request parameter filtered them out. To use the quality filter, you specify the `QualityFilter` request parameter.

Type: Array of UnindexedFace objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListCollections

Service: Amazon Rekognition

Returns list of collection IDs in your account. If the result is truncated, the response also provides a `NextToken` that you can use in the subsequent request to fetch the next set of collection IDs.

For an example, see Listing collections.

This operation requires permissions to perform the `rekognition:ListCollections` action.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxResults

Maximum number of collection IDs to return.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 4096.

Required: No

NextToken

Pagination token from the previous response.

Type: String

Length Constraints: Maximum length of 255.

Required: No

Response Syntax

```
{
   "CollectionIds": [ "string" ],
   "FaceModelVersions": [ "string" ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CollectionIds

An array of collection IDs.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

FaceModelVersions

Version numbers of the face detection models associated with the collections in the array `CollectionIds`. For example, the value of `FaceModelVersions[2]` is the version number for the face detection model used by the collection in `CollectionId[2]`.

Type: Array of strings

NextToken

If the result is truncated, the response provides a `NextToken` that you can use in the subsequent request to fetch the next set of collection IDs.

Type: String

Length Constraints: Maximum length of 255.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListDatasetEntries

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Lists the entries (images) within a dataset. An entry is a JSON Line that contains the information for a single image, including the image location, assigned labels, and object location bounding boxes. For more information, see Creating a manifest file.

JSON Lines in the response include information about non-terminal errors found in the dataset. Non terminal errors are reported in `errors` lists within each JSON Line. The same information is reported in the training and testing validation result manifests that Amazon Rekognition Custom Labels creates during model training.

You can filter the response in variety of ways, such as choosing which labels to return and returning JSON Lines created after a specific date.

This operation requires permissions to perform the `rekognition:ListDatasetEntries` action.

Request Syntax

```
{
   "ContainsLabels": [ "string" ],
   "DatasetArn": "string",
   "HasErrors": boolean,
   "Labeled": boolean,
   "MaxResults": number,
   "NextToken": "string",
   "SourceRefContains": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ContainsLabels

Specifies a label filter for the response. The response includes an entry only if one or more of the labels in `ContainsLabels` exist in the entry.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `.{1,}`

Required: No

DatasetArn

The Amazon Resource Name (ARN) for the dataset that you want to use.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Required: Yes

HasErrors

Specifies an error filter for the response. Specify `True` to only include entries that have errors.

Type: Boolean

Required: No

Labeled

Specify `true` to get only the JSON Lines where the image is labeled. Specify `false` to get only the JSON Lines where the image isn't labeled. If you don't specify `Labeled`, `ListDatasetEntries` returns JSON Lines for labeled and unlabeled images.

Type: Boolean

Required: No

MaxResults

The maximum number of results to return per paginated call. The largest value you can specify is 100. If you specify a value greater than 100, a ValidationException error occurs. The default value is 100.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken

If the previous response was incomplete (because there is more results to retrieve), Amazon Rekognition Custom Labels returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

SourceRefContains

If specified, `ListDatasetEntries` only returns JSON Lines where the value of `SourceRefContains` is part of the `source-ref` field. The `source-ref` field contains the Amazon S3 location of the image. You can use `SouceRefContains` for tasks such as getting the JSON Line for a single image, or gettting JSON Lines for all images within a specific folder.

For more information, see Creating a manifest file.

For more information, see Creating a manifest file in the *Amazon Rekognition Custom Labels Developer Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `.*\S.*`

Required: No

Response Syntax

```
{
   "DatasetEntries": [ "string" ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DatasetEntries

A list of entries (images) in the dataset.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 100000.

Pattern: `^\{.*\}$`

NextToken

If the previous response was incomplete (because there is more results to retrieve), Amazon Rekognition Custom Labels returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ResourceNotReadyException

The requested resource isn't ready. For example, this exception occurs when you call `DetectCustomLabels` with a model version that isn't deployed.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListDatasetLabels

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Lists the labels in a dataset. Amazon Rekognition Custom Labels uses labels to describe images. For more information, see Labeling images.

This operation requires permissions to perform the `rekognition:ListDatasetLabels` action.

Request Syntax

```
{
   "DatasetArn": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DatasetArn

The Amazon Resource Name (ARN) of the dataset that you want to use.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Required: Yes

MaxResults

The maximum number of results to return per paginated call. The largest value you can specify is 100. If you specify a value greater than 100, a ValidationException error occurs. The default value is 100.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken

If the previous response was incomplete (because there is more results to retrieve), Amazon Rekognition Custom Labels returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

Response Syntax

```
{
   "DatasetLabelDescriptions": [
      {
         "LabelName": "string",
         "LabelStats": {
            "BoundingBoxCount": number,
            "EntryCount": number
         }
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DatasetLabelDescriptions

A list of the labels in the dataset.

Type: Array of DatasetLabelDescription objects

NextToken

If the previous response was incomplete (because there is more results to retrieve), Amazon Rekognition Custom Labels returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ResourceNotReadyException

The requested resource isn't ready. For example, this exception occurs when you call `DetectCustomLabels` with a model version that isn't deployed.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListFaces

Service: Amazon Rekognition

Returns metadata for faces in the specified collection. This metadata includes information such as the bounding box coordinates, the confidence (that the bounding box contains a face), and face ID. For an example, see Listing faces in a collection.

This operation requires permissions to perform the `rekognition:ListFaces` action.

Request Syntax

```
{
   "CollectionId": "string",
   "FaceIds": [ "string" ],
   "MaxResults": number,
   "NextToken": "string",
   "UserId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

ID of the collection from which to list the faces.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

FaceIds

An array of face IDs to filter results with when listing faces in a collection.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4096 items.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

MaxResults

Maximum number of faces to return.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 4096.

Required: No

NextToken

If the previous response was incomplete (because there is more data to retrieve), Amazon Rekognition returns a pagination token in the response. You can use this pagination token to retrieve the next set of faces.

Type: String

Length Constraints: Maximum length of 255.

Required: No

UserId

An array of user IDs to filter results with when listing faces in a collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

Response Syntax

```
{
   "FaceModelVersion": "string",
```

```
    "Faces": [
       {
          "BoundingBox": {
             "Height": number,
             "Left": number,
             "Top": number,
             "Width": number
          },
          "Confidence": number,
          "ExternalImageId": "string",
          "FaceId": "string",
          "ImageId": "string",
          "IndexFacesModelVersion": "string",
          "UserId": "string"
       }
    ],
    "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaceModelVersion

Version number of the face detection model associated with the input collection (`CollectionId`).

Type: String

Faces

An array of `Face` objects.

Type: Array of Face objects

NextToken

If the response is truncated, Amazon Rekognition returns this token that you can use in the subsequent request to retrieve the next set of faces.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListMediaAnalysisJobs

Service: Amazon Rekognition

Returns a list of media analysis jobs. Results are sorted by `CreationTimestamp` in descending order.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxResults

The maximum number of results to return per paginated call. The largest value user can specify is 100. If user specifies a value greater than 100, an `InvalidParameterException` error occurs. The default value is 100.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken

Pagination token, if the previous response was incomplete.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

Response Syntax

```
{
   "MediaAnalysisJobs": [
      {
         "CompletionTimestamp": number,
         "CreationTimestamp": number,
         "FailureDetails": {
            "Code": "string",
            "Message": "string"
         },
         "Input": {
            "S3Object": {
               "Bucket": "string",
               "Name": "string",
               "Version": "string"
            }
         },
         "JobId": "string",
         "JobName": "string",
         "KmsKeyId": "string",
         "ManifestSummary": {
            "S3Object": {
               "Bucket": "string",
               "Name": "string",
               "Version": "string"
            }
         },
         "OperationsConfig": {
            "DetectModerationLabels": {
               "MinConfidence": number,
               "ProjectVersion": "string"
            }
         },
         "OutputConfig": {
            "S3Bucket": "string",
            "S3KeyPrefix": "string"
         },
         "Results": {
            "ModelVersions": {
               "Moderation": "string"
            },
            "S3Object": {
```

```
                "Bucket": "string",
                "Name": "string",
                "Version": "string"
            }
        },
        "Status": "string"
    }
  ],
  "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

MediaAnalysisJobs

Contains a list of all media analysis jobs.

Type: Array of MediaAnalysisJobDescription objects

NextToken

Pagination token, if the previous response was incomplete.

Type: String

Length Constraints: Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

ListProjectPolicies

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Gets a list of the project policies attached to a project.

To attach a project policy to a project, call PutProjectPolicy. To remove a project policy from a project, call DeleteProjectPolicy.

This operation requires permissions to perform the `rekognition:ListProjectPolicies` action.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ProjectArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxResults

The maximum number of results to return per paginated call. The largest value you can specify is 5. If you specify a value greater than 5, a ValidationException error occurs. The default value is 5.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 5.

Required: No

NextToken

If the previous response was incomplete (because there is more results to retrieve), Amazon Rekognition Custom Labels returns a pagination token in the response. You can use this pagination token to retrieve the next set of results.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

ProjectArn

The ARN of the project for which you want to list the project policies.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Response Syntax

```
{
   "NextToken": "string",
   "ProjectPolicies": [
      {
         "CreationTimestamp": number,
         "LastUpdatedTimestamp": number,
         "PolicyDocument": "string",
         "PolicyName": "string",
         "PolicyRevisionId": "string",
         "ProjectArn": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

If the response is truncated, Amazon Rekognition returns this token that you can use in the subsequent request to retrieve the next set of project policies.

Type: String

Length Constraints: Maximum length of 1024.

ProjectPolicies

A list of project policies attached to the project.

Type: Array of ProjectPolicy objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

ListStreamProcessors

Service: Amazon Rekognition

Gets a list of stream processors that you have created with CreateStreamProcessor.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxResults

Maximum number of stream processors you want Amazon Rekognition Video to return in the response. The default is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

If the previous response was incomplete (because there are more stream processors to retrieve), Amazon Rekognition Video returns a pagination token in the response. You can use this pagination token to retrieve the next set of stream processors.

Type: String

Length Constraints: Maximum length of 255.

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "StreamProcessors": [
      {
         "Name": "string",
         "Status": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

If the response is truncated, Amazon Rekognition Video returns this token that you can use in the subsequent request to retrieve the next set of stream processors.

Type: String

Length Constraints: Maximum length of 255.

StreamProcessors

List of stream processors that you have created.

Type: Array of StreamProcessor objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsForResource

Service: Amazon Rekognition

Returns a list of tags in an Amazon Rekognition collection, stream processor, or Custom Labels model.

This operation requires permissions to perform the `rekognition:ListTagsForResource` action.

Request Syntax

```
{
   "ResourceArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ResourceArn

Amazon Resource Name (ARN) of the model, collection, or stream processor that contains the tags that you want a list of.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Required: Yes

Response Syntax

```
{
   "Tags": {
      "string" : "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Tags

A list of key-value tags assigned to the resource.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Key Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListUsers

Service: Amazon Rekognition

Returns metadata of the User such as `UserID` in the specified collection. Anonymous User (to reserve faces without any identity) is not returned as part of this request. The results are sorted by system generated primary key ID. If the response is truncated, `NextToken` is returned in the response that can be used in the subsequent request to retrieve the next set of identities.

Request Syntax

```
{
   "CollectionId": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

The ID of an existing collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

MaxResults

Maximum number of UsersID to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

Required: No

NextToken

Pagingation token to receive the next set of UsersID.

Type: String

Length Constraints: Maximum length of 255.

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "Users": [
      {
         "UserId": "string",
         "UserStatus": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

A pagination token to be used with the subsequent request if the response is truncated.

Type: String

Length Constraints: Maximum length of 255.

Users

List of UsersID associated with the specified collection.

Type: Array of User objects

Array Members: Maximum number of 500 items.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidPaginationTokenException

Pagination token in the request is not valid.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutProjectPolicy

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Attaches a project policy to a Amazon Rekognition Custom Labels project in a trusting AWS account. A project policy specifies that a trusted AWS account can copy a model version from a trusting AWS account to a project in the trusted AWS account. To copy a model version you use the CopyProjectVersion operation. Only applies to Custom Labels projects.

For more information about the format of a project policy document, see Attaching a project policy (SDK).

The response from `PutProjectPolicy` is a revision ID for the project policy. You can attach multiple project policies to a project. You can also update an existing project policy by specifying the policy revision ID of the existing policy.

To remove a project policy from a project, call DeleteProjectPolicy. To get a list of project policies attached to a project, call ListProjectPolicies.

You copy a model version by calling CopyProjectVersion.

This operation requires permissions to perform the `rekognition:PutProjectPolicy` action.

Request Syntax

```
{
   "PolicyDocument": "string",
   "PolicyName": "string",
   "PolicyRevisionId": "string",
   "ProjectArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

PolicyDocument

A resource policy to add to the model. The policy is a JSON structure that contains one or more statements that define the policy. The policy must follow the IAM syntax. For more information about the contents of a JSON policy document, see IAM JSON policy reference.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: `[\u0009\u000A\u000D\u0020-\u00FF]+`

Required: Yes

PolicyName

A name for the policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

PolicyRevisionId

The revision ID for the Project Policy. Each time you modify a policy, Amazon Rekognition Custom Labels generates and assigns a new `PolicyRevisionId` and then deletes the previous version of the policy.

Type: String

Length Constraints: Maximum length of 64.

Pattern: `[0-9A-Fa-f]+`

Required: No

ProjectArn

The Amazon Resource Name (ARN) of the project that the project policy is attached to.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Response Syntax

```
{
   "PolicyRevisionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

PolicyRevisionId

The ID of the project policy.

Type: String

Length Constraints: Maximum length of 64.

Pattern: `[0-9A-Fa-f]+`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidPolicyRevisionIdException

The supplied revision id for the project policy is invalid.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

MalformedPolicyDocumentException

The format of the project policy document that you supplied to `PutProjectPolicy` is incorrect.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceAlreadyExistsException

A resource with the specified ID already exists.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

RecognizeCelebrities

Service: Amazon Rekognition

Returns an array of celebrities recognized in the input image. For more information, see Recognizing celebrities.

`RecognizeCelebrities` returns the 64 largest faces in the image. It lists the recognized celebrities in the `CelebrityFaces` array and any unrecognized faces in the `UnrecognizedFaces` array. `RecognizeCelebrities` doesn't return celebrities whose faces aren't among the largest 64 faces in the image.

For each celebrity recognized, `RecognizeCelebrities` returns a `Celebrity` object. The `Celebrity` object contains the celebrity name, ID, URL links to additional information, match confidence, and a `ComparedFace` object that you can use to locate the celebrity's face on the image.

Amazon Rekognition doesn't retain information about which images a celebrity has been recognized in. Your application must store this information and use the `Celebrity` ID property as a unique identifier for the celebrity. If you don't store the celebrity name or additional information URLs returned by `RecognizeCelebrities`, you will need the ID to identify the celebrity in a call to the GetCelebrityInfo operation.

You pass the input image either as base64-encoded image bytes or as a reference to an image in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes is not supported. The image must be either a PNG or JPEG formatted file.

For an example, see Recognizing celebrities in an image.

This operation requires permissions to perform the `rekognition:RecognizeCelebrities` operation.

Request Syntax

```
{
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
```

```
    }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Image

The input image as base64-encoded bytes or an S3 object. If you use the AWS CLI to call Amazon Rekognition operations, passing base64-encoded image bytes is not supported.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

Response Syntax

```
{
   "CelebrityFaces": [
      {
         "Face": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "Emotions": [
               {
                  "Confidence": number,
                  "Type": "string"
               }
            ],
            "Landmarks": [
               {
                  "Type": "string",
```

```
                "X": number,
                "Y": number
            }
        ],
        "Pose": {
            "Pitch": number,
            "Roll": number,
            "Yaw": number
        },
        "Quality": {
            "Brightness": number,
            "Sharpness": number
        },
        "Smile": {
            "Confidence": number,
            "Value": boolean
        }
    },
    "Id": "string",
    "KnownGender": {
        "Type": "string"
    },
    "MatchConfidence": number,
    "Name": "string",
    "Urls": [ "string" ]
  }
],
"OrientationCorrection": "string",
"UnrecognizedFaces": [
  {
    "BoundingBox": {
        "Height": number,
        "Left": number,
        "Top": number,
        "Width": number
    },
    "Confidence": number,
    "Emotions": [
        {
            "Confidence": number,
            "Type": "string"
        }
    ],
    "Landmarks": [
```

```
            {
               "Type": "string",
               "X": number,
               "Y": number
            }
         ],
         "Pose": {
            "Pitch": number,
            "Roll": number,
            "Yaw": number
         },
         "Quality": {
            "Brightness": number,
            "Sharpness": number
         },
         "Smile": {
            "Confidence": number,
            "Value": boolean
         }
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

CelebrityFaces

Details about each celebrity found in the image. Amazon Rekognition can detect a maximum of 64 celebrities in an image. Each celebrity object includes the following attributes: `Face`, `Confidence`, `Emotions`, `Landmarks`, `Pose`, `Quality`, `Smile`, `Id`, `KnownGender`, `MatchConfidence`, `Name`, `Urls`.

Type: Array of Celebrity objects

OrientationCorrection

> **Note**
>
> Support for estimating image orientation using the the OrientationCorrection field has ceased as of August 2021. Any returned values for this field included in an API response will always be NULL.

The orientation of the input image (counterclockwise direction). If your application displays the image, you can use this value to correct the orientation. The bounding box coordinates returned in `CelebrityFaces` and `UnrecognizedFaces` represent face locations before the image orientation is corrected.

> **Note**
>
> If the input image is in .jpeg format, it might contain exchangeable image (Exif) metadata that includes the image's orientation. If so, and the Exif metadata for the input image populates the orientation field, the value of `OrientationCorrection` is null. The `CelebrityFaces` and `UnrecognizedFaces` bounding box coordinates represent face locations after Exif metadata is used to correct the image orientation. Images in .png format don't contain Exif metadata.

Type: String

Valid Values: `ROTATE_0 | ROTATE_90 | ROTATE_180 | ROTATE_270`

UnrecognizedFaces

Details about each unrecognized face in the image.

Type: Array of ComparedFace objects

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

SearchFaces

Service: Amazon Rekognition

For a given input face ID, searches for matching faces in the collection the face belongs to. You get a face ID when you add a face to the collection using the IndexFaces operation. The operation compares the features of the input face with faces in the specified collection.

> **Note**
>
> You can also search faces without indexing faces by using the `SearchFacesByImage` operation.

The operation response returns an array of faces that match, ordered by similarity score with the highest similarity first. More specifically, it is an array of metadata for each face match that is found. Along with the metadata, the response also includes a `confidence` value for each face match, indicating the confidence that the specific face matches the input face.

For an example, see Searching for a face using its face ID.

This operation requires permissions to perform the `rekognition:SearchFaces` action.

Request Syntax

```
{
   "CollectionId": "string",
   "FaceId": "string",
   "FaceMatchThreshold": number,
   "MaxFaces": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

ID of the collection the face belongs to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

FaceId

ID of a face to find matches for in the collection.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: Yes

FaceMatchThreshold

Optional value specifying the minimum confidence in the face match to return. For example, don't return any matches where confidence in matches is less than 70%. The default value is 80%.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

MaxFaces

Maximum number of faces to return. The operation returns the maximum number of faces with the highest confidence in the match.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 4096.

Required: No

Response Syntax

```
{
```

```
   "FaceMatches": [
      {
         "Face": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "ExternalImageId": "string",
            "FaceId": "string",
            "ImageId": "string",
            "IndexFacesModelVersion": "string",
            "UserId": "string"
         },
         "Similarity": number
      }
   ],
   "FaceModelVersion": "string",
   "SearchedFaceId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaceMatches

An array of faces that matched the input face, along with the confidence in the match.

Type: Array of FaceMatch objects

FaceModelVersion

Version number of the face detection model associated with the input collection (`CollectionId`).

Type: String

SearchedFaceId

ID of the face that was searched for matches in a collection.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

SearchFacesByImage

Service: Amazon Rekognition

For a given input image, first detects the largest face in the image, and then searches the specified collection for matching faces. The operation compares the features of the input face with faces in the specified collection.

> **Note**
>
> To search for all faces in an input image, you might first call the IndexFaces operation, and then use the face IDs returned in subsequent calls to the SearchFaces operation.
> You can also call the `DetectFaces` operation and use the bounding boxes in the response to make face crops, which then you can pass in to the `SearchFacesByImage` operation.

You pass the input image either as base64-encoded image bytes or as a reference to an image in an Amazon S3 bucket. If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes is not supported. The image must be either a PNG or JPEG formatted file.

The response returns an array of faces that match, ordered by similarity score with the highest similarity first. More specifically, it is an array of metadata for each face match found. Along with the metadata, the response also includes a `similarity` indicating how similar the face is to the input face. In the response, the operation also returns the bounding box (and a confidence level that the bounding box contains a face) of the face that Amazon Rekognition used for the input image.

If no faces are detected in the input image, `SearchFacesByImage` returns an `InvalidParameterException` error.

For an example, see Searching for a Face Using an Image.

The `QualityFilter` input parameter allows you to filter out detected faces that don't meet a required quality bar. The quality bar is based on a variety of common use cases. Use `QualityFilter` to set the quality bar for filtering by specifying `LOW`, `MEDIUM`, or `HIGH`. If you do not want to filter detected faces, specify `NONE`. The default value is `NONE`.

> **Note**
>
> To use quality filtering, you need a collection associated with version 3 of the face model or higher. To get the version of the face model associated with a collection, call DescribeCollection.

This operation requires permissions to perform the `rekognition:SearchFacesByImage` action.

Request Syntax

```
{
   "CollectionId": "string",
   "FaceMatchThreshold": number,
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "MaxFaces": number,
   "QualityFilter": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

ID of the collection to search.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

FaceMatchThreshold

(Optional) Specifies the minimum confidence in the face match to return. For example, don't return any matches where confidence in matches is less than 70%. The default value is 80%.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Image

The input image as base64-encoded bytes or an S3 object. If you use the AWS CLI to call Amazon Rekognition operations, passing base64-encoded image bytes is not supported.

If you are using an AWS SDK to call Amazon Rekognition, you might not need to base64-encode image bytes passed using the `Bytes` field. For more information, see Image specifications.

Type: Image object

Required: Yes

MaxFaces

Maximum number of faces to return. The operation returns the maximum number of faces with the highest confidence in the match.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 4096.

Required: No

QualityFilter

A filter that specifies a quality bar for how much filtering is done to identify faces. Filtered faces aren't searched for in the collection. If you specify `AUTO`, Amazon Rekognition chooses the quality bar. If you specify `LOW`, `MEDIUM`, or `HIGH`, filtering removes all faces that don’t meet the chosen quality bar. The quality bar is based on a variety of common use cases. Low-quality detections can occur for a number of reasons. Some examples are an object that's misidentified

as a face, a face that's too blurry, or a face with a pose that's too extreme to use. If you specify NONE, no filtering is performed. The default value is NONE.

To use quality filtering, the collection you are using must be associated with version 3 of the face model or higher.

Type: String

Valid Values: NONE | AUTO | LOW | MEDIUM | HIGH

Required: No

Response Syntax

```
{
   "FaceMatches": [
      {
         "Face": {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Confidence": number,
            "ExternalImageId": "string",
            "FaceId": "string",
            "ImageId": "string",
            "IndexFacesModelVersion": "string",
            "UserId": "string"
         },
         "Similarity": number
      }
   ],
   "FaceModelVersion": "string",
   "SearchedFaceBoundingBox": {
      "Height": number,
      "Left": number,
      "Top": number,
      "Width": number
   },
   "SearchedFaceConfidence": number
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaceMatches

An array of faces that match the input face, along with the confidence in the match.

Type: Array of FaceMatch objects

FaceModelVersion

Version number of the face detection model associated with the input collection (`CollectionId`).

Type: String

SearchedFaceBoundingBox

The bounding box around the face in the input image that Amazon Rekognition used for the search.

Type: BoundingBox object

SearchedFaceConfidence

The level of confidence that the `searchedFaceBoundingBox`, contains a face.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

SearchUsers

Service: Amazon Rekognition

Searches for UserIDs within a collection based on a `FaceId` or `UserId`. This API can be used to find the closest UserID (with a highest similarity) to associate a face. The request must be provided with either `FaceId` or `UserId`. The operation returns an array of UserID that match the `FaceId` or `UserId`, ordered by similarity score with the highest similarity first.

Request Syntax

```
{
   "CollectionId": "string",
   "FaceId": "string",
   "MaxUsers": number,
   "UserId": "string",
   "UserMatchThreshold": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

The ID of an existing collection containing the UserID, used with a UserId or FaceId. If a FaceId is provided, UserId isn't required to be present in the Collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

FaceId

ID for the existing face.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

MaxUsers

Maximum number of identities to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

Required: No

UserId

ID for the existing User.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

UserMatchThreshold

Optional value that specifies the minimum confidence in the matched UserID to return. Default value of 80.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Response Syntax

```
{
   "FaceModelVersion": "string",
   "SearchedFace": {
      "FaceId": "string"
   },
   "SearchedUser": {
```

```
        "UserId": "string"
    },
    "UserMatches": [
        {
            "Similarity": number,
            "User": {
                "UserId": "string",
                "UserStatus": "string"
            }
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaceModelVersion

Version number of the face detection model associated with the input CollectionId.

Type: String

SearchedFace

Contains the ID of a face that was used to search for matches in a collection.

Type: SearchedFace object

SearchedUser

Contains the ID of the UserID that was used to search for matches in a collection.

Type: SearchedUser object

UserMatches

An array of UserMatch objects that matched the input face along with the confidence in the match. Array will be empty if there are no matches.

Type: Array of UserMatch objects

Array Members: Maximum number of 500 items.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

SearchUsersByImage

Service: Amazon Rekognition

Searches for UserIDs using a supplied image. It first detects the largest face in the image, and then searches a specified collection for matching UserIDs.

The operation returns an array of UserIDs that match the face in the supplied image, ordered by similarity score with the highest similarity first. It also returns a bounding box for the face found in the input image.

Information about faces detected in the supplied image, but not used for the search, is returned in an array of `UnsearchedFace` objects. If no valid face is detected in the image, the response will contain an empty `UserMatches` list and no `SearchedFace` object.

Request Syntax

```
{
   "CollectionId": "string",
   "Image": {
      "Bytes": blob,
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "MaxUsers": number,
   "QualityFilter": "string",
   "UserMatchThreshold": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

CollectionId

The ID of an existing collection containing the UserID.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Image

Provides the input image either as bytes or an S3 object.

You pass image bytes to an Amazon Rekognition API operation by using the `Bytes` property. For example, you would use the `Bytes` property to pass an image loaded from a local file system. Image bytes passed by using the `Bytes` property must be base64-encoded. Your code may not need to encode image bytes if you are using an AWS SDK to call Amazon Rekognition API operations.

For more information, see Analyzing an image loaded from a local file system .

You pass images stored in an S3 bucket to an Amazon Rekognition API operation by using the `S3Object` property. Images stored in an S3 bucket do not need to be base64-encoded.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes using the Bytes property is not supported. You must first upload the image to an Amazon S3 bucket and then call the operation using the S3Object property.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Type: Image object

Required: Yes

MaxUsers

Maximum number of UserIDs to return.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

Required: No

QualityFilter

A filter that specifies a quality bar for how much filtering is done to identify faces. Filtered faces aren't searched for in the collection. The default value is NONE.

Type: String

Valid Values: `NONE | AUTO | LOW | MEDIUM | HIGH`

Required: No

UserMatchThreshold

Specifies the minimum confidence in the UserID match to return. Default value is 80.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Response Syntax

```
{
   "FaceModelVersion": "string",
   "SearchedFace": {
      "FaceDetail": {
         "AgeRange": {
            "High": number,
            "Low": number
         },
         "Beard": {
            "Confidence": number,
            "Value": boolean
         },
         "BoundingBox": {
            "Height": number,
            "Left": number,
            "Top": number,
            "Width": number
         },
         "Confidence": number,
         "Emotions": [
            {
```

```
                "Confidence": number,
                "Type": "string"
            }
        ],
        "EyeDirection": {
            "Confidence": number,
            "Pitch": number,
            "Yaw": number
        },
        "Eyeglasses": {
            "Confidence": number,
            "Value": boolean
        },
        "EyesOpen": {
            "Confidence": number,
            "Value": boolean
        },
        "FaceOccluded": {
            "Confidence": number,
            "Value": boolean
        },
        "Gender": {
            "Confidence": number,
            "Value": "string"
        },
        "Landmarks": [
            {
                "Type": "string",
                "X": number,
                "Y": number
            }
        ],
        "MouthOpen": {
            "Confidence": number,
            "Value": boolean
        },
        "Mustache": {
            "Confidence": number,
            "Value": boolean
        },
        "Pose": {
            "Pitch": number,
            "Roll": number,
            "Yaw": number
```

```
            },
            "Quality": {
               "Brightness": number,
               "Sharpness": number
            },
            "Smile": {
               "Confidence": number,
               "Value": boolean
            },
            "Sunglasses": {
               "Confidence": number,
               "Value": boolean
            }
         }
      },
      "UnsearchedFaces": [
         {
            "FaceDetails": {
               "AgeRange": {
                  "High": number,
                  "Low": number
               },
               "Beard": {
                  "Confidence": number,
                  "Value": boolean
               },
               "BoundingBox": {
                  "Height": number,
                  "Left": number,
                  "Top": number,
                  "Width": number
               },
               "Confidence": number,
               "Emotions": [
                  {
                     "Confidence": number,
                     "Type": "string"
                  }
               ],
               "EyeDirection": {
                  "Confidence": number,
                  "Pitch": number,
                  "Yaw": number
               },
```

```
            "Eyeglasses": {
               "Confidence": number,
               "Value": boolean
            },
            "EyesOpen": {
               "Confidence": number,
               "Value": boolean
            },
            "FaceOccluded": {
               "Confidence": number,
               "Value": boolean
            },
            "Gender": {
               "Confidence": number,
               "Value": "string"
            },
            "Landmarks": [
               {
                  "Type": "string",
                  "X": number,
                  "Y": number
               }
            ],
            "MouthOpen": {
               "Confidence": number,
               "Value": boolean
            },
            "Mustache": {
               "Confidence": number,
               "Value": boolean
            },
            "Pose": {
               "Pitch": number,
               "Roll": number,
               "Yaw": number
            },
            "Quality": {
               "Brightness": number,
               "Sharpness": number
            },
            "Smile": {
               "Confidence": number,
               "Value": boolean
            },
```

```
            "Sunglasses": {
               "Confidence": number,
               "Value": boolean
            }
         },
         "Reasons": [ "string" ]
      }
   ],
   "UserMatches": [
      {
         "Similarity": number,
         "User": {
            "UserId": "string",
            "UserStatus": "string"
         }
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

FaceModelVersion

Version number of the face detection model associated with the input collection CollectionId.

Type: String

SearchedFace

A list of FaceDetail objects containing the BoundingBox for the largest face in image, as well as the confidence in the bounding box, that was searched for matches. If no valid face is detected in the image the response will contain no SearchedFace object.

Type: SearchedFaceDetails object

UnsearchedFaces

List of UnsearchedFace objects. Contains the face details infered from the specified image but not used for search. Contains reasons that describe why a face wasn't used for Search.

Type: Array of UnsearchedFace objects

UserMatches

An array of UserID objects that matched the input face, along with the confidence in the match. The returned structure will be empty if there are no matches. Returned if the SearchUsersByImageResponse action is successful.

Type: Array of UserMatch objects

Array Members: Maximum number of 500 items.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

ImageTooLargeException

The input image size exceeds the allowed limit. If you are calling DetectProtectiveEquipment, the image size or resolution exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidImageFormatException

The provided image format is not supported.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

StartCelebrityRecognition

Service: Amazon Rekognition

Starts asynchronous recognition of celebrities in a stored video.

Amazon Rekognition Video can detect celebrities in a video must be stored in an Amazon S3 bucket. Use Video to specify the bucket name and the filename of the video. `StartCelebrityRecognition` returns a job identifier (`JobId`) which you use to get the results of the analysis. When celebrity recognition analysis is finished, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic that you specify in `NotificationChannel`. To get the results of the celebrity recognition analysis, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call GetCelebrityRecognition and pass the job identifier (`JobId`) from the initial call to `StartCelebrityRecognition`.

For more information, see Recognizing celebrities.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "JobTag": "string",
   "NotificationChannel": {
      "RoleArn": "string",
      "SNSTopicArn": "string"
   },
   "Video": {
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the start request. If you use the same token with multiple `StartCelebrityRecognition` requests, the same `JobId` is returned. Use `ClientRequestToken` to prevent the same job from being accidently started more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

JobTag

An identifier you specify that's returned in the completion notification that's published to your Amazon Simple Notification Service topic. For example, you can use `JobTag` to group related jobs and identify them in the completion notification.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

Required: No

NotificationChannel

The Amazon SNS topic ARN that you want Amazon Rekognition Video to publish the completion status of the celebrity recognition analysis to. The Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy.

Type: NotificationChannel object

Required: No

Video

The video in which you want to recognize celebrities. The video must be stored in an Amazon S3 bucket.

Type: Video object

Required: Yes

Response Syntax

```
{
   "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

The identifier for the celebrity recognition analysis job. Use `JobId` to identify the job in a subsequent call to `GetCelebrityRecognition`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

VideoTooLargeException

The file size or duration of the supplied media is too large. The maximum file size is 10GB. The maximum duration is 6 hours.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartContentModeration

Service: Amazon Rekognition

Starts asynchronous detection of inappropriate, unwanted, or offensive content in a stored video. For a list of moderation labels in Amazon Rekognition, see Using the image and video moderation APIs.

Amazon Rekognition Video can moderate content in a video stored in an Amazon S3 bucket. Use Video to specify the bucket name and the filename of the video. `StartContentModeration` returns a job identifier (`JobId`) which you use to get the results of the analysis. When content analysis is finished, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic that you specify in `NotificationChannel`.

To get the results of the content analysis, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call GetContentModeration and pass the job identifier (`JobId`) from the initial call to `StartContentModeration`.

For more information, see Moderating content.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "JobTag": "string",
   "MinConfidence": number,
   "NotificationChannel": {
      "RoleArn": "string",
      "SNSTopicArn": "string"
   },
   "Video": {
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the start request. If you use the same token with multiple `StartContentModeration` requests, the same `JobId` is returned. Use `ClientRequestToken` to prevent the same job from being accidently started more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

JobTag

An identifier you specify that's returned in the completion notification that's published to your Amazon Simple Notification Service topic. For example, you can use `JobTag` to group related jobs and identify them in the completion notification.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

Required: No

MinConfidence

Specifies the minimum confidence that Amazon Rekognition must have in order to return a moderated content label. Confidence represents how certain Amazon Rekognition is that the moderated content is correctly identified. 0 is the lowest confidence. 100 is the highest confidence. Amazon Rekognition doesn't return any moderated content labels with a confidence level lower than this specified value. If you don't specify `MinConfidence`, `GetContentModeration` returns labels with confidence values greater than or equal to 50 percent.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

NotificationChannel

The Amazon SNS topic ARN that you want Amazon Rekognition Video to publish the completion status of the content analysis to. The Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy to access the topic.

Type: NotificationChannel object

Required: No

Video

The video in which you want to detect inappropriate, unwanted, or offensive content. The video must be stored in an Amazon S3 bucket.

Type: Video object

Required: Yes

Response Syntax

```
{
   "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

The identifier for the content analysis job. Use `JobId` to identify the job in a subsequent call to `GetContentModeration`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

VideoTooLargeException

The file size or duration of the supplied media is too large. The maximum file size is 10GB. The maximum duration is 6 hours.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartFaceDetection

Service: Amazon Rekognition

Starts asynchronous detection of faces in a stored video.

Amazon Rekognition Video can detect faces in a video stored in an Amazon S3 bucket. Use Video to specify the bucket name and the filename of the video. `StartFaceDetection` returns a job identifier (`JobId`) that you use to get the results of the operation. When face detection is finished, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic that you specify in `NotificationChannel`. To get the results of the face detection operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call GetFaceDetection and pass the job identifier (`JobId`) from the initial call to `StartFaceDetection`.

For more information, see Detecting faces in a stored video.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "FaceAttributes": "string",
   "JobTag": "string",
   "NotificationChannel": {
      "RoleArn": "string",
      "SNSTopicArn": "string"
   },
   "Video": {
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the start request. If you use the same token with multiple `StartFaceDetection` requests, the same `JobId` is returned. Use `ClientRequestToken` to prevent the same job from being accidently started more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

FaceAttributes

The face attributes you want returned.

`DEFAULT` - The following subset of facial attributes are returned: BoundingBox, Confidence, Pose, Quality and Landmarks.

`ALL` - All facial attributes are returned.

Type: String

Valid Values: `DEFAULT | ALL`

Required: No

JobTag

An identifier you specify that's returned in the completion notification that's published to your Amazon Simple Notification Service topic. For example, you can use `JobTag` to group related jobs and identify them in the completion notification.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

Required: No

NotificationChannel

The ARN of the Amazon SNS topic to which you want Amazon Rekognition Video to publish the completion status of the face detection operation. The Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy.

Type: NotificationChannel object

Required: No

Video

The video in which you want to detect faces. The video must be stored in an Amazon S3 bucket.

Type: Video object

Required: Yes

Response Syntax

```
{
   "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

The identifier for the face detection job. Use `JobId` to identify the job in a subsequent call to `GetFaceDetection`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

VideoTooLargeException

The file size or duration of the supplied media is too large. The maximum file size is 10GB. The maximum duration is 6 hours.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartFaceSearch

Service: Amazon Rekognition

Starts the asynchronous search for faces in a collection that match the faces of persons detected in a stored video.

The video must be stored in an Amazon S3 bucket. Use Video to specify the bucket name and the filename of the video. `StartFaceSearch` returns a job identifier (`JobId`) which you use to get the search results once the search has completed. When searching is finished, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic that you specify in `NotificationChannel`. To get the search results, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call GetFaceSearch and pass the job identifier (`JobId`) from the initial call to `StartFaceSearch`. For more information, see Searching stored videos for faces.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "CollectionId": "string",
   "FaceMatchThreshold": number,
   "JobTag": "string",
   "NotificationChannel": {
      "RoleArn": "string",
      "SNSTopicArn": "string"
   },
   "Video": {
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the start request. If you use the same token with multiple `StartFaceSearch` requests, the same `JobId` is returned. Use `ClientRequestToken` to prevent the same job from being accidently started more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

CollectionId

ID of the collection that contains the faces you want to search for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

FaceMatchThreshold

The minimum confidence in the person match to return. For example, don't return any matches where confidence in matches is less than 70%. The default value is 80%.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

JobTag

An identifier you specify that's returned in the completion notification that's published to your Amazon Simple Notification Service topic. For example, you can use `JobTag` to group related jobs and identify them in the completion notification.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

Required: No

NotificationChannel

The ARN of the Amazon SNS topic to which you want Amazon Rekognition Video to publish the completion status of the search. The Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy to access the topic.

Type: NotificationChannel object

Required: No

Video

The video you want to search. The video must be stored in an Amazon S3 bucket.

Type: Video object

Required: Yes

Response Syntax

```
{
   "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

The identifier for the search job. Use `JobId` to identify the job in a subsequent call to `GetFaceSearch`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will

raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

VideoTooLargeException

The file size or duration of the supplied media is too large. The maximum file size is 10GB. The maximum duration is 6 hours.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartLabelDetection

Service: Amazon Rekognition

Starts asynchronous detection of labels in a stored video.

Amazon Rekognition Video can detect labels in a video. Labels are instances of real-world entities. This includes objects like flower, tree, and table; events like wedding, graduation, and birthday party; concepts like landscape, evening, and nature; and activities like a person getting out of a car or a person skiing.

The video must be stored in an Amazon S3 bucket. Use Video to specify the bucket name and the filename of the video. `StartLabelDetection` returns a job identifier (`JobId`) which you use to get the results of the operation. When label detection is finished, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic that you specify in `NotificationChannel`.

To get the results of the label detection operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call GetLabelDetection and pass the job identifier (`JobId`) from the initial call to `StartLabelDetection`.

Optional Parameters

`StartLabelDetection` has the `GENERAL_LABELS` Feature applied by default. This feature allows you to provide filtering criteria to the `Settings` parameter. You can filter with sets of individual labels or with label categories. You can specify inclusive filters, exclusive filters, or a combination of inclusive and exclusive filters. For more information on filtering, see Detecting labels in a video.

You can specify `MinConfidence` to control the confidence threshold for the labels returned. The default is 50.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "Features": [ "string" ],
   "JobTag": "string",
   "MinConfidence": number,
   "NotificationChannel": {
      "RoleArn": "string",
      "SNSTopicArn": "string"
```

```
    },
    "Settings": {
       "GeneralLabels": {
          "LabelCategoryExclusionFilters": [ "string" ],
          "LabelCategoryInclusionFilters": [ "string" ],
          "LabelExclusionFilters": [ "string" ],
          "LabelInclusionFilters": [ "string" ]
       }
    },
    "Video": {
       "S3Object": {
          "Bucket": "string",
          "Name": "string",
          "Version": "string"
       }
    }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the start request. If you use the same token with multiple `StartLabelDetection` requests, the same `JobId` is returned. Use `ClientRequestToken` to prevent the same job from being accidently started more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

Features

The features to return after video analysis. You can specify that GENERAL_LABELS are returned.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 1 item.

Valid Values: `GENERAL_LABELS`

Required: No

JobTag

An identifier you specify that's returned in the completion notification that's published to your Amazon Simple Notification Service topic. For example, you can use `JobTag` to group related jobs and identify them in the completion notification.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

Required: No

MinConfidence

Specifies the minimum confidence that Amazon Rekognition Video must have in order to return a detected label. Confidence represents how certain Amazon Rekognition is that a label is correctly identified.0 is the lowest confidence. 100 is the highest confidence. Amazon Rekognition Video doesn't return any labels with a confidence level lower than this specified value.

If you don't specify `MinConfidence`, the operation returns labels and bounding boxes (if detected) with confidence values greater than or equal to 50 percent.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

NotificationChannel

The Amazon SNS topic ARN you want Amazon Rekognition Video to publish the completion status of the label detection operation to. The Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy.

Type: NotificationChannel object

Required: No

Settings

The settings for a StartLabelDetection request.Contains the specified parameters for the label detection request of an asynchronous label analysis operation. Settings can include filters for GENERAL_LABELS.

Type: LabelDetectionSettings object

Required: No

Video

The video in which you want to detect labels. The video must be stored in an Amazon S3 bucket.

Type: Video object

Required: Yes

Response Syntax

```
{
   "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

The identifier for the label detection job. Use `JobId` to identify the job in a subsequent call to `GetLabelDetection`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

VideoTooLargeException

The file size or duration of the supplied media is too large. The maximum file size is 10GB. The maximum duration is 6 hours.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartMediaAnalysisJob

Service: Amazon Rekognition

Initiates a new media analysis job. Accepts a manifest file in an Amazon S3 bucket. The output is a manifest file and a summary of the manifest stored in the Amazon S3 bucket.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "Input": {
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   },
   "JobName": "string",
   "KmsKeyId": "string",
   "OperationsConfig": {
      "DetectModerationLabels": {
         "MinConfidence": number,
         "ProjectVersion": "string"
      }
   },
   "OutputConfig": {
      "S3Bucket": "string",
      "S3KeyPrefix": "string"
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotency token used to prevent the accidental creation of duplicate versions. If you use the same token with multiple `StartMediaAnalysisJobRequest` requests, the same response is

returned. Use `ClientRequestToken` to prevent the same request from being processed more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

Input

Input data to be analyzed by the job.

Type: MediaAnalysisInput object

Required: Yes

JobName

The name of the job. Does not have to be unique.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: No

KmsKeyId

The identifier of customer managed AWS KMS key (name or ARN). The key is used to encrypt images copied into the service. The key is also used to encrypt results and manifest files written to the output Amazon S3 bucket.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

Required: No

OperationsConfig

Configuration options for the media analysis job to be created.

Type: MediaAnalysisOperationsConfig object

Required: Yes

OutputConfig

The Amazon S3 bucket location to store the results.

Type: MediaAnalysisOutputConfig object

Required: Yes

Response Syntax

```
{
   "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

Identifier for the created job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidManifestException

Indicates that a provided manifest file is empty or larger than the allowed limit.

HTTP Status Code: 400

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ResourceNotReadyException

The requested resource isn't ready. For example, this exception occurs when you call `DetectCustomLabels` with a model version that isn't deployed.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

StartPersonTracking

Service: Amazon Rekognition

> **Note**
>
> *End of support notice:* On October 31, 2025, AWS will discontinue support for Amazon Rekognition People Pathing. After October 31, 2025, you will no longer be able to use the Rekognition People Pathing capability. For more information, visit this blog post.

Starts the asynchronous tracking of a person's path in a stored video.

Amazon Rekognition Video can track the path of people in a video stored in an Amazon S3 bucket. Use Video to specify the bucket name and the filename of the video. `StartPersonTracking` returns a job identifier (`JobId`) which you use to get the results of the operation. When label detection is finished, Amazon Rekognition publishes a completion status to the Amazon Simple Notification Service topic that you specify in `NotificationChannel`.

To get the results of the person detection operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. If so, call GetPersonTracking and pass the job identifier (`JobId`) from the initial call to `StartPersonTracking`.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "JobTag": "string",
   "NotificationChannel": {
      "RoleArn": "string",
      "SNSTopicArn": "string"
   },
   "Video": {
      "S3Object": {
         "Bucket": "string",
         "Name": "string",
         "Version": "string"
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the start request. If you use the same token with multiple `StartPersonTracking` requests, the same `JobId` is returned. Use `ClientRequestToken` to prevent the same job from being accidently started more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

JobTag

An identifier you specify that's returned in the completion notification that's published to your Amazon Simple Notification Service topic. For example, you can use `JobTag` to group related jobs and identify them in the completion notification.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

Required: No

NotificationChannel

The Amazon SNS topic ARN you want Amazon Rekognition Video to publish the completion status of the people detection operation to. The Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy.

Type: NotificationChannel object

Required: No

Video

The video in which you want to detect people. The video must be stored in an Amazon S3 bucket.

Type: Video object

Required: Yes

Response Syntax

```
{
   "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

The identifier for the person detection job. Use `JobId` to identify the job in a subsequent call to `GetPersonTracking`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

VideoTooLargeException

The file size or duration of the supplied media is too large. The maximum file size is 10GB. The maximum duration is 6 hours.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartProjectVersion

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Starts the running of the version of a model. Starting a model takes a while to complete. To check the current state of the model, use DescribeProjectVersions.

Once the model is running, you can detect custom labels in new images by calling DetectCustomLabels.

> **Note**
>
> You are charged for the amount of time that the model is running. To stop a running model, call StopProjectVersion.

For more information, see Running a trained Amazon Rekognition Custom Labels model.

This operation requires permissions to perform the `rekognition:StartProjectVersion` action.

Request Syntax

```
{
   "MaxInferenceUnits": number,
   "MinInferenceUnits": number,
   "ProjectVersionArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

MaxInferenceUnits

The maximum number of inference units to use for auto-scaling the model. If you don't specify a value, Amazon Rekognition Custom Labels doesn't auto-scale the model.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

MinInferenceUnits

The minimum number of inference units to use. A single inference unit represents 1 hour of processing.

For information about the number of transactions per second (TPS) that an inference unit can support, see Running a trained Amazon Rekognition Custom Labels model.

Use a higher number to increase the TPS throughput of your model. You are charged for the number of inference units that you use.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

ProjectVersionArn

The Amazon Resource Name(ARN) of the model version that you want to start.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Response Syntax

```
{
```

```
    "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Status

The current running status of the model.

Type: String

Valid Values: `TRAINING_IN_PROGRESS | TRAINING_COMPLETED | TRAINING_FAILED | STARTING | RUNNING | FAILED | STOPPING | STOPPED | DELETING | COPYING_IN_PROGRESS | COPYING_COMPLETED | COPYING_FAILED | DEPRECATED | EXPIRED`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2

- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartSegmentDetection

Service: Amazon Rekognition

Starts asynchronous detection of segment detection in a stored video.

Amazon Rekognition Video can detect segments in a video stored in an Amazon S3 bucket. Use Video to specify the bucket name and the filename of the video. `StartSegmentDetection` returns a job identifier (`JobId`) which you use to get the results of the operation. When segment detection is finished, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic that you specify in `NotificationChannel`.

You can use the `Filters` (StartSegmentDetectionFilters) input parameter to specify the minimum detection confidence returned in the response. Within `Filters`, use `ShotFilter` (StartShotDetectionFilter) to filter detected shots. Use `TechnicalCueFilter` (StartTechnicalCueDetectionFilter) to filter technical cues.

To get the results of the segment detection operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. if so, call GetSegmentDetection and pass the job identifier (`JobId`) from the initial call to `StartSegmentDetection`.

For more information, see Detecting video segments in stored video.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "Filters": {
      "ShotFilter": {
         "MinSegmentConfidence": number
      },
      "TechnicalCueFilter": {
         "BlackFrame": {
            "MaxPixelThreshold": number,
            "MinCoveragePercentage": number
         },
         "MinSegmentConfidence": number
      }
   },
   "JobTag": "string",
   "NotificationChannel": {
      "RoleArn": "string",
```

```
        "SNSTopicArn": "string"
    },
    "SegmentTypes": [ "string" ],
    "Video": {
        "S3Object": {
            "Bucket": "string",
            "Name": "string",
            "Version": "string"
        }
    }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the start request. If you use the same token with multiple `StartSegmentDetection` requests, the same `JobId` is returned. Use `ClientRequestToken` to prevent the same job from being accidently started more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

Filters

Filters for technical cue or shot detection.

Type: StartSegmentDetectionFilters object

Required: No

JobTag

An identifier you specify that's returned in the completion notification that's published to your Amazon Simple Notification Service topic. For example, you can use `JobTag` to group related jobs and identify them in the completion notification.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

Required: No

NotificationChannel

The ARN of the Amazon SNS topic to which you want Amazon Rekognition Video to publish the completion status of the segment detection operation. Note that the Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy to access the topic.

Type: NotificationChannel object

Required: No

SegmentTypes

An array of segment types to detect in the video. Valid values are TECHNICAL_CUE and SHOT.

Type: Array of strings

Array Members: Minimum number of 1 item.

Valid Values: `TECHNICAL_CUE | SHOT`

Required: Yes

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

Required: Yes

Response Syntax

```
{
```

```
    "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

Unique identifier for the segment detection job. The `JobId` is returned from `StartSegmentDetection`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

VideoTooLargeException

The file size or duration of the supplied media is too large. The maximum file size is 10GB. The maximum duration is 6 hours.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartStreamProcessor

Service: Amazon Rekognition

Starts processing a stream processor. You create a stream processor by calling CreateStreamProcessor. To tell `StartStreamProcessor` which stream processor to start, use the value of the `Name` field specified in the call to `CreateStreamProcessor`.

If you are using a label detection stream processor to detect labels, you need to provide a `Start selector` and a `Stop selector` to determine the length of the stream processing time.

Request Syntax

```
{
   "Name": "string",
   "StartSelector": {
      "KVSStreamStartSelector": {
         "FragmentNumber": "string",
         "ProducerTimestamp": number
      }
   },
   "StopSelector": {
      "MaxDurationInSeconds": number
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Name

The name of the stream processor to start processing.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

StartSelector

Specifies the starting point in the Kinesis stream to start processing. You can use the producer timestamp or the fragment number. If you use the producer timestamp, you must put the time in milliseconds. For more information about fragment numbers, see Fragment.

This is a required parameter for label detection stream processors and should not be used to start a face search stream processor.

Type: StreamProcessingStartSelector object

Required: No

StopSelector

Specifies when to stop processing the stream. You can specify a maximum amount of time to process the video.

This is a required parameter for label detection stream processors and should not be used to start a face search stream processor.

Type: StreamProcessingStopSelector object

Required: No

Response Syntax

```
{
   "SessionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

SessionId

A unique identifier for the stream processing session.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartTextDetection

Service: Amazon Rekognition

Starts asynchronous detection of text in a stored video.

Amazon Rekognition Video can detect text in a video stored in an Amazon S3 bucket. Use Video to specify the bucket name and the filename of the video. `StartTextDetection` returns a job identifier (`JobId`) which you use to get the results of the operation. When text detection is finished, Amazon Rekognition Video publishes a completion status to the Amazon Simple Notification Service topic that you specify in `NotificationChannel`.

To get the results of the text detection operation, first check that the status value published to the Amazon SNS topic is SUCCEEDED. if so, call GetTextDetection and pass the job identifier (`JobId`) from the initial call to `StartTextDetection`.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "Filters": {
      "RegionsOfInterest": [
         {
            "BoundingBox": {
               "Height": number,
               "Left": number,
               "Top": number,
               "Width": number
            },
            "Polygon": [
               {
                  "X": number,
                  "Y": number
               }
            ]
         }
      ],
      "WordFilter": {
         "MinBoundingBoxHeight": number,
         "MinBoundingBoxWidth": number,
         "MinConfidence": number
      }
   },
```

```
    "JobTag": "string",
    "NotificationChannel": {
       "RoleArn": "string",
       "SNSTopicArn": "string"
    },
    "Video": {
       "S3Object": {
          "Bucket": "string",
          "Name": "string",
          "Version": "string"
       }
    }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ClientRequestToken

Idempotent token used to identify the start request. If you use the same token with multiple `StartTextDetection` requests, the same `JobId` is returned. Use `ClientRequestToken` to prevent the same job from being accidentaly started more than once.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: No

Filters

Optional parameters that let you set criteria the text must meet to be included in your response.

Type: StartTextDetectionFilters object

Required: No

JobTag

An identifier returned in the completion status published by your Amazon Simple Notification Service topic. For example, you can use `JobTag` to group related jobs and identify them in the completion notification.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `[a-zA-Z0-9_.\-:+=\/]+`

Required: No

NotificationChannel

The Amazon Simple Notification Service topic to which Amazon Rekognition publishes the completion status of a video analysis operation. For more information, see Calling Amazon Rekognition Video operations. Note that the Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy to access the topic. For more information, see Giving access to multiple Amazon SNS topics.

Type: NotificationChannel object

Required: No

Video

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Type: Video object

Required: Yes

Response Syntax

```
{
   "JobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

JobId

Identifier for the text detection job. Use `JobId` to identify the job in a subsequent call to `GetTextDetection`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

IdempotentParameterMismatchException

A `ClientRequestToken` input parameter was reused with an operation, but at least one of the other input parameters is different from the previous call to the operation.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

InvalidS3ObjectException

Amazon Rekognition is unable to access the S3 object specified in the request.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

VideoTooLargeException

The file size or duration of the supplied media is too large. The maximum file size is 10GB. The maximum duration is 6 hours.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StopProjectVersion

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Stops a running model. The operation might take a while to complete. To check the current status, call DescribeProjectVersions. Only applies to Custom Labels projects.

This operation requires permissions to perform the `rekognition:StopProjectVersion` action.

Request Syntax

```
{
   "ProjectVersionArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ProjectVersionArn

The Amazon Resource Name (ARN) of the model version that you want to stop.

This operation requires permissions to perform the `rekognition:StopProjectVersion` action.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: Yes

Response Syntax

```
{
   "Status": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Status

The current status of the stop operation.

Type: String

Valid Values: `TRAINING_IN_PROGRESS | TRAINING_COMPLETED | TRAINING_FAILED | STARTING | RUNNING | FAILED | STOPPING | STOPPED | DELETING | COPYING_IN_PROGRESS | COPYING_COMPLETED | COPYING_FAILED | DEPRECATED | EXPIRED`

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

StopStreamProcessor

Service: Amazon Rekognition

Stops a running stream processor that was created by CreateStreamProcessor.

Request Syntax

```
{
   "Name": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Name

The name of a stream processor created by CreateStreamProcessor.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagResource

Service: Amazon Rekognition

Adds one or more key-value tags to an Amazon Rekognition collection, stream processor, or Custom Labels model. For more information, see Tagging AWS Resources.

This operation requires permissions to perform the `rekognition:TagResource` action.

Request Syntax

```
{
   "ResourceArn": "string",
   "Tags": {
      "string" : "string"
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ResourceArn

Amazon Resource Name (ARN) of the model, collection, or stream processor that you want to assign the tags to.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Required: Yes

Tags

The key-value tags to assign to the resource.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 200 items.

Key Length Constraints: Minimum length of 1. Maximum length of 128.

Key Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ServiceQuotaExceededException

The size of the resource exceeds the allowed limit. For more information, see Guidelines and quotas in Amazon Rekognition.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagResource

Service: Amazon Rekognition

Removes one or more tags from an Amazon Rekognition collection, stream processor, or Custom Labels model.

This operation requires permissions to perform the `rekognition:UntagResource` action.

Request Syntax

```
{
   "ResourceArn": "string",
   "TagKeys": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

ResourceArn

Amazon Resource Name (ARN) of the model, collection, or stream processor that you want to remove the tags from.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Required: Yes

TagKeys

A list of the tags that you want to remove.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 200 items.

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `^(?!aws:)[\p{L}\p{Z}\p{N}_.:/=+\-@]*$`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateDatasetEntries

Service: Amazon Rekognition

> **Note**
>
> This operation applies only to Amazon Rekognition Custom Labels.

Adds or updates one or more entries (images) in a dataset. An entry is a JSON Line which contains the information for a single image, including the image location, assigned labels, and object location bounding boxes. For more information, see Image-Level labels in manifest files and Object localization in manifest files.

If the `source-ref` field in the JSON line references an existing image, the existing image in the dataset is updated. If `source-ref` field doesn't reference an existing image, the image is added as a new image to the dataset.

You specify the changes that you want to make in the `Changes` input parameter. There isn't a limit to the number JSON Lines that you can change, but the size of `Changes` must be less than 5MB.

`UpdateDatasetEntries` returns immediatly, but the dataset update might take a while to complete. Use DescribeDataset to check the current status. The dataset updated successfully if the value of `Status` is `UPDATE_COMPLETE`.

To check if any non-terminal errors occured, call ListDatasetEntries and check for the presence of `errors` lists in the JSON Lines.

Dataset update fails if a terminal error occurs (`Status` = `UPDATE_FAILED`). Currently, you can't access the terminal error information from the Amazon Rekognition Custom Labels SDK.

This operation requires permissions to perform the `rekognition:UpdateDatasetEntries` action.

Request Syntax

```
{
   "Changes": {
      "GroundTruth": blob
   },
   "DatasetArn": "string"
```

```
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

Changes

The changes that you want to make to the dataset.

Type: DatasetChanges object

Required: Yes

DatasetArn

The Amazon Resource Name (ARN) of the dataset that you want to update.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

LimitExceededException

An Amazon Rekognition service limit was exceeded. For example, if you start too many jobs concurrently, subsequent calls to start operations (ex: `StartLabelDetection`) will raise a `LimitExceededException` exception (HTTP status code: 400) until the number of concurrently running jobs is below the Amazon Rekognition service limit.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateStreamProcessor

Service: Amazon Rekognition

Allows you to update a stream processor. You can change some settings and regions of interest and delete certain parameters.

Request Syntax

```
{
   "DataSharingPreferenceForUpdate": {
      "OptIn": boolean
   },
   "Name": "string",
   "ParametersToDelete": [ "string" ],
   "RegionsOfInterestForUpdate": [
      {
         "BoundingBox": {
            "Height": number,
            "Left": number,
            "Top": number,
            "Width": number
         },
         "Polygon": [
            {
               "X": number,
               "Y": number
            }
         ]
      }
   ],
   "SettingsForUpdate": {
      "ConnectedHomeForUpdate": {
         "Labels": [ "string" ],
         "MinConfidence": number
      }
   }
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters.

The request accepts the following data in JSON format.

DataSharingPreferenceForUpdate

Shows whether you are sharing data with Rekognition to improve model performance. You can choose this option at the account level or on a per-stream basis. Note that if you opt out at the account level this setting is ignored on individual streams.

Type: StreamProcessorDataSharingPreference object

Required: No

Name

Name of the stream processor that you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: Yes

ParametersToDelete

A list of parameters you want to delete from the stream processor.

Type: Array of strings

Valid Values: `ConnectedHomeMinConfidence | RegionsOfInterest`

Required: No

RegionsOfInterestForUpdate

Specifies locations in the frames where Amazon Rekognition checks for objects or people. This is an optional parameter for label detection stream processors.

Type: Array of RegionOfInterest objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

SettingsForUpdate

The stream processor settings that you want to update. Label detection settings can be updated to detect different labels with a different minimum confidence.

Type: StreamProcessorSettingsForUpdate object

Required: No

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

You are not authorized to perform the action.

HTTP Status Code: 400

InternalServerError

Amazon Rekognition experienced a service issue. Try your call again.

HTTP Status Code: 500

InvalidParameterException

Input parameter violated a constraint. Validate your parameter before calling the API operation again.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The number of requests exceeded your throughput limit. If you want to increase this limit, contact Amazon Rekognition.

HTTP Status Code: 400

ResourceInUseException

The specified resource is already being used.

HTTP Status Code: 400

ResourceNotFoundException

The resource specified in the request cannot be found.

HTTP Status Code: 400

ThrottlingException

Amazon Rekognition is temporarily unable to process the request. Try your call again.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Amazon Rekognition Streaming

The following actions are supported by Amazon Rekognition Streaming:

- StartFaceLivenessSession

StartFaceLivenessSession

Service: Amazon Rekognition Streaming

Starts a Face Liveness video stream and liveness detection process for a given session.

Requires `sessionId`, `ChallengeVersions`, `VideoWidth`, `VideoHeight` and a `RequestEventStream` as input. The event stream contains information about different events for the session, including the challenge information used for verification.

The maximum video size for Face Liveness is 10 MB. Face Liveness throws a `ValidationException` if the video does not match the necessary formatting and size parameters.

StartFaceLivenessSession supports the websockets and the AWS SDK for JavaScript.

Request Syntax

```
POST /start-face-liveness-session HTTP/1.1
x-amz-rekognition-streaming-liveness-session-id: SessionId
x-amz-rekognition-streaming-liveness-video-width: VideoWidth
x-amz-rekognition-streaming-liveness-video-height: VideoHeight
x-amz-rekognition-streaming-liveness-challenge-versions: ChallengeVersions
Content-type: application/json

{
   "ClientSessionInformationEvent": {
      "Challenge": { ... }
   },
   "VideoEvent": {
      "TimestampMillis": number,
      "VideoChunk": blob
   }
}
```

URI Request Parameters

The request uses the following URI parameters.

ChallengeVersions

String containing comma separated list of challenge versions supported by client.

Length Constraints: Minimum length of 15. Maximum length of 150.

Pattern: `^([a-zA-Z]{10,}_\d+\.\d+\.\d+){1}(,[a-zA-Z]{10,}_\d+\.\d+\.\d+)*`

Required: Yes

SessionId

A unique 128-bit UUID. Used to uniquely identify the session and also acta as an idempotency token for all operations associated with the session.

Length Constraints: Fixed length of 36.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: Yes

VideoHeight

The height of the video object included in a request to start a Face Liveness session.

Length Constraints: Minimum length of 3. Maximum length of 4.

Pattern: `^[0-9]{3,}$`

Required: Yes

VideoWidth

The width of the video object included in a request to start a Face Liveness session.

Length Constraints: Minimum length of 3. Maximum length of 4.

Pattern: `^[0-9]{3,}$`

Required: Yes

Request Body

The request accepts the following data in JSON format.

ClientSessionInformationEvent

Information event for the client session for a given Face Liveness request stream.

Type: ClientSessionInformationEvent object

Required: No

VideoEvent

Contains video events for a Face Liveness request stream.

Type: VideoEvent object

Required: No

Response Syntax

```
HTTP/1.1 200
x-amz-rekognition-streaming-liveness-session-id: SessionId
Content-type: application/json

{
   "DisconnectionEvent": {
      "TimestampMillis": number
   },
   "InternalServerException": {
   },
   "ServerSessionInformationEvent": {
      "SessionInformation": {
         "Challenge": { ... }
      }
   },
   "ServiceQuotaExceededException": {
   },
   "ServiceUnavailableException": {
   },
   "ThrottlingException": {
   },
   "ValidationException": {
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

SessionId

The ID that identifies a Face Liveness session.

Length Constraints: Fixed length of 36.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

The following data is returned in JSON format by the service.

DisconnectionEvent

Notification that disconnection event has occurred to a Face Liveness session.

Type: DisconnectionEvent object

InternalServerException

Notification that an InternalServerException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 500

ServerSessionInformationEvent

Details for a server session information event.

Type: ServerSessionInformationEvent object

ServiceQuotaExceededException

Notification that a ServiceQuotaExceededException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 429

ServiceUnavailableException

Notification that a ServiceUnavailableException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 503

ThrottlingException

Notification that a ThrottlingException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 429

ValidationException

Notification that a ValidationException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 400

Errors

For information about the errors that are common to all actions, see Common Errors.

AccessDeniedException

Occurs when the client isn't authorized to perform the action.

HTTP Status Code: 403

InternalServerException

Unexpected error during processing of request.

HTTP Status Code: 500

ServiceQuotaExceededException

Occurs when a request would cause a service quota to be exceeded.

HTTP Status Code: 429

ServiceUnavailableException

Service-wide throttling to recover from an operational event or service is not able to scale.

HTTP Status Code: 503

SessionNotFoundException

Occurs when the given `sessionId` is not found.

HTTP Status Code: 400

ThrottlingException

A request was denied due to request throttling. Occurs when too many requests were made by a user (exceeding their service quota), the service isn't able to scale, or a service-wide throttling was done to recover from an operational event.

HTTP Status Code: 429

ValidationException

The input fails to satisfy the constraints specified by the service. Potential reasons inlcude: video quality or size is invalid, video container format not supported, video does not have enough information - no person detected in video, request couldn't be parsed or is invalid, session has expired or is invalid, S3 bucket is invalid/in another AWS region, KMS Key is invalid.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

The following data types are supported by Amazon Rekognition:

- AgeRange
- Asset
- AssociatedFace
- AudioMetadata
- AuditImage
- Beard
- BlackFrame
- BoundingBox
- Celebrity
- CelebrityDetail
- CelebrityRecognition
- Challenge
- ChallengePreference
- ComparedFace
- ComparedSourceImageFace
- CompareFacesMatch
- ConnectedHomeSettings
- ConnectedHomeSettingsForUpdate
- ContentModerationDetection
- ContentType
- CoversBodyPart
- CreateFaceLivenessSessionRequestSettings
- CustomizationFeatureConfig
- CustomizationFeatureContentModerationConfig
- CustomLabel
- DatasetChanges
- DatasetDescription

- DatasetLabelDescription
- DatasetLabelStats
- DatasetMetadata
- DatasetSource
- DatasetStats
- DetectionFilter
- DetectLabelsImageBackground
- DetectLabelsImageForeground
- DetectLabelsImageProperties
- DetectLabelsImagePropertiesSettings
- DetectLabelsImageQuality
- DetectLabelsSettings
- DetectTextFilters
- DisassociatedFace
- DistributeDataset
- DominantColor
- Emotion
- EquipmentDetection
- EvaluationResult
- EyeDirection
- Eyeglasses
- EyeOpen
- Face
- FaceDetail
- FaceDetection
- FaceMatch
- FaceOccluded
- FaceRecord
- FaceSearchSettings
- Gender

- GeneralLabelsSettings
- Geometry
- GetContentModerationRequestMetadata
- GetLabelDetectionRequestMetadata
- GroundTruthManifest
- HumanLoopActivationOutput
- HumanLoopConfig
- HumanLoopDataAttributes
- Image
- ImageQuality
- Instance
- KinesisDataStream
- KinesisVideoStream
- KinesisVideoStreamStartSelector
- KnownGender
- Label
- LabelAlias
- LabelCategory
- LabelDetection
- LabelDetectionSettings
- Landmark
- LivenessOutputConfig
- MatchedUser
- MediaAnalysisDetectModerationLabelsConfig
- MediaAnalysisInput
- MediaAnalysisJobDescription
- MediaAnalysisJobFailureDetails
- MediaAnalysisManifestSummary
- MediaAnalysisModelVersions
- MediaAnalysisOperationsConfig

- MediaAnalysisOutputConfig
- MediaAnalysisResults
- ModerationLabel
- MouthOpen
- Mustache
- NotificationChannel
- OutputConfig
- Parent
- PersonDetail
- PersonDetection
- PersonMatch
- Point
- Pose
- ProjectDescription
- ProjectPolicy
- ProjectVersionDescription
- ProtectiveEquipmentBodyPart
- ProtectiveEquipmentPerson
- ProtectiveEquipmentSummarizationAttributes
- ProtectiveEquipmentSummary
- RegionOfInterest
- S3Destination
- S3Object
- SearchedFace
- SearchedFaceDetails
- SearchedUser
- SegmentDetection
- SegmentTypeInfo
- ShotSegment
- Smile

- StartSegmentDetectionFilters
- StartShotDetectionFilter
- StartTechnicalCueDetectionFilter
- StartTextDetectionFilters
- StreamProcessingStartSelector
- StreamProcessingStopSelector
- StreamProcessor
- StreamProcessorDataSharingPreference
- StreamProcessorInput
- StreamProcessorNotificationChannel
- StreamProcessorOutput
- StreamProcessorSettings
- StreamProcessorSettingsForUpdate
- Summary
- Sunglasses
- TechnicalCueSegment
- TestingData
- TestingDataResult
- TextDetection
- TextDetectionResult
- TrainingData
- TrainingDataResult
- UnindexedFace
- UnsearchedFace
- UnsuccessfulFaceAssociation
- UnsuccessfulFaceDeletion
- UnsuccessfulFaceDisassociation
- User
- UserMatch
- ValidationData

- Versions
- Video
- VideoMetadata

The following data types are supported by Amazon Rekognition Streaming:

- BoundingBox
- ChallengeConfig
- ClientChallenge
- ClientSessionInformationEvent
- ColorDisplayed
- ColorSequence
- DisconnectionEvent
- FaceMovementAndLightClientChallenge
- FaceMovementAndLightServerChallenge
- FreshnessColor
- InitialFace
- LivenessRequestStream
- LivenessResponseStream
- OvalParameters
- ServerChallenge
- ServerSessionInformationEvent
- SessionInformation
- TargetFace
- VideoEvent

Amazon Rekognition

The following data types are supported by Amazon Rekognition:

- AgeRange
- Asset

- AssociatedFace
- AudioMetadata
- AuditImage
- Beard
- BlackFrame
- BoundingBox
- Celebrity
- CelebrityDetail
- CelebrityRecognition
- Challenge
- ChallengePreference
- ComparedFace
- ComparedSourceImageFace
- CompareFacesMatch
- ConnectedHomeSettings
- ConnectedHomeSettingsForUpdate
- ContentModerationDetection
- ContentType
- CoversBodyPart
- CreateFaceLivenessSessionRequestSettings
- CustomizationFeatureConfig
- CustomizationFeatureContentModerationConfig
- CustomLabel
- DatasetChanges
- DatasetDescription
- DatasetLabelDescription
- DatasetLabelStats
- DatasetMetadata
- DatasetSource
- DatasetStats

- DetectionFilter
- DetectLabelsImageBackground
- DetectLabelsImageForeground
- DetectLabelsImageProperties
- DetectLabelsImagePropertiesSettings
- DetectLabelsImageQuality
- DetectLabelsSettings
- DetectTextFilters
- DisassociatedFace
- DistributeDataset
- DominantColor
- Emotion
- EquipmentDetection
- EvaluationResult
- EyeDirection
- Eyeglasses
- EyeOpen
- Face
- FaceDetail
- FaceDetection
- FaceMatch
- FaceOccluded
- FaceRecord
- FaceSearchSettings
- Gender
- GeneralLabelsSettings
- Geometry
- GetContentModerationRequestMetadata
- GetLabelDetectionRequestMetadata
- GroundTruthManifest

- HumanLoopActivationOutput
- HumanLoopConfig
- HumanLoopDataAttributes
- Image
- ImageQuality
- Instance
- KinesisDataStream
- KinesisVideoStream
- KinesisVideoStreamStartSelector
- KnownGender
- Label
- LabelAlias
- LabelCategory
- LabelDetection
- LabelDetectionSettings
- Landmark
- LivenessOutputConfig
- MatchedUser
- MediaAnalysisDetectModerationLabelsConfig
- MediaAnalysisInput
- MediaAnalysisJobDescription
- MediaAnalysisJobFailureDetails
- MediaAnalysisManifestSummary
- MediaAnalysisModelVersions
- MediaAnalysisOperationsConfig
- MediaAnalysisOutputConfig
- MediaAnalysisResults
- ModerationLabel
- MouthOpen
- Mustache

- NotificationChannel
- OutputConfig
- Parent
- PersonDetail
- PersonDetection
- PersonMatch
- Point
- Pose
- ProjectDescription
- ProjectPolicy
- ProjectVersionDescription
- ProtectiveEquipmentBodyPart
- ProtectiveEquipmentPerson
- ProtectiveEquipmentSummarizationAttributes
- ProtectiveEquipmentSummary
- RegionOfInterest
- S3Destination
- S3Object
- SearchedFace
- SearchedFaceDetails
- SearchedUser
- SegmentDetection
- SegmentTypeInfo
- ShotSegment
- Smile
- StartSegmentDetectionFilters
- StartShotDetectionFilter
- StartTechnicalCueDetectionFilter
- StartTextDetectionFilters
- StreamProcessingStartSelector

- StreamProcessingStopSelector
- StreamProcessor
- StreamProcessorDataSharingPreference
- StreamProcessorInput
- StreamProcessorNotificationChannel
- StreamProcessorOutput
- StreamProcessorSettings
- StreamProcessorSettingsForUpdate
- Summary
- Sunglasses
- TechnicalCueSegment
- TestingData
- TestingDataResult
- TextDetection
- TextDetectionResult
- TrainingData
- TrainingDataResult
- UnindexedFace
- UnsearchedFace
- UnsuccessfulFaceAssociation
- UnsuccessfulFaceDeletion
- UnsuccessfulFaceDisassociation
- User
- UserMatch
- ValidationData
- Versions
- Video
- VideoMetadata

AgeRange

Service: Amazon Rekognition

Structure containing the estimated age range, in years, for a face.

Amazon Rekognition estimates an age range for faces detected in the input image. Estimated age ranges can overlap. A face of a 5-year-old might have an estimated range of 4-6, while the face of a 6-year-old might have an estimated range of 4-8.

Contents

High

The highest estimated age.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

Low

The lowest estimated age.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Asset

Service: Amazon Rekognition

Assets are the images that you use to train and evaluate a model version. Assets can also contain validation information that you use to debug a failed model training.

Contents

GroundTruthManifest

The S3 bucket that contains an Amazon Sagemaker Ground Truth format manifest file.

Type: GroundTruthManifest object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AssociatedFace

Service: Amazon Rekognition

Provides face metadata for the faces that are associated to a specific UserID.

Contents

FaceId

Unique identifier assigned to the face.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AudioMetadata

Service: Amazon Rekognition

Metadata information about an audio stream. An array of `AudioMetadata` objects for the audio streams found in a stored video is returned by GetSegmentDetection.

Contents

Codec

The audio codec used to encode or decode the audio stream.

Type: String

Required: No

DurationMillis

The duration of the audio stream in milliseconds.

Type: Long

Valid Range: Minimum value of 0.

Required: No

NumberOfChannels

The number of audio channels in the segment.

Type: Long

Valid Range: Minimum value of 0.

Required: No

SampleRate

The sample rate for the audio stream.

Type: Long

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AuditImage

Service: Amazon Rekognition

An image that is picked from the Face Liveness video and returned for audit trail purposes, returned as Base64-encoded bytes.

Contents

BoundingBox

Identifies the bounding box around the label, face, text, object of interest, or personal protective equipment. The `left` (x-coordinate) and `top` (y-coordinate) are coordinates representing the top and left sides of the bounding box. Note that the upper-left corner of the image is the origin (0,0).

The `top` and `left` values returned are ratios of the overall image size. For example, if the input image is 700x200 pixels, and the top-left coordinate of the bounding box is 350x50 pixels, the API returns a `left` value of 0.5 (350/700) and a `top` value of 0.25 (50/200).

The `width` and `height` values represent the dimensions of the bounding box as a ratio of the overall image dimension. For example, if the input image is 700x200 pixels, and the bounding box width is 70 pixels, the width returned is 0.1.

> **Note**
>
> The bounding box coordinates can have negative values. For example, if Amazon Rekognition is able to detect a face that is at the image edge and is only partially visible, the service can return coordinates that are outside the image bounds and, depending on the image edge, you might get negative values or values greater than 1 for the `left` or `top` values.

Type: BoundingBox object

Required: No

Bytes

The Base64-encoded bytes representing an image selected from the Face Liveness video and returned for audit purposes.

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 1. Maximum length of 204800.

Required: No

S3Object

Provides the S3 bucket name and object name.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Type: S3Object object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Beard

Service: Amazon Rekognition

Indicates whether or not the face has a beard, and the confidence level in the determination.

Contents

Confidence

Level of confidence in the determination.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

Boolean value that indicates whether the face has beard or not.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BlackFrame

Service: Amazon Rekognition

A filter that allows you to control the black frame detection by specifying the black levels and pixel coverage of black pixels in a frame. As videos can come from multiple sources, formats, and time periods, they may contain different standards and varying noise levels for black frames that need to be accounted for. For more information, see StartSegmentDetection.

Contents

MaxPixelThreshold

A threshold used to determine the maximum luminance value for a pixel to be considered black. In a full color range video, luminance values range from 0-255. A pixel value of 0 is pure black, and the most strict filter. The maximum black pixel value is computed as follows: max_black_pixel_value = minimum_luminance + MaxPixelThreshold *luminance_range.

For example, for a full range video with BlackPixelThreshold = 0.1, max_black_pixel_value is 0 + 0.1 * (255-0) = 25.5.

The default value of MaxPixelThreshold is 0.2, which maps to a max_black_pixel_value of 51 for a full range video. You can lower this threshold to be more strict on black levels.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

MinCoveragePercentage

The minimum percentage of pixels in a frame that need to have a luminance below the max_black_pixel_value for a frame to be considered a black frame. Luminance is calculated using the BT.709 matrix.

The default value is 99, which means at least 99% of all pixels in the frame are black pixels as per the `MaxPixelThreshold` set. You can reduce this value to allow more noise on the black frame.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BoundingBox

Service: Amazon Rekognition

Identifies the bounding box around the label, face, text, object of interest, or personal protective equipment. The `left` (x-coordinate) and `top` (y-coordinate) are coordinates representing the top and left sides of the bounding box. Note that the upper-left corner of the image is the origin (0,0).

The `top` and `left` values returned are ratios of the overall image size. For example, if the input image is 700x200 pixels, and the top-left coordinate of the bounding box is 350x50 pixels, the API returns a `left` value of 0.5 (350/700) and a `top` value of 0.25 (50/200).

The `width` and `height` values represent the dimensions of the bounding box as a ratio of the overall image dimension. For example, if the input image is 700x200 pixels, and the bounding box width is 70 pixels, the width returned is 0.1.

> **Note**
>
> The bounding box coordinates can have negative values. For example, if Amazon Rekognition is able to detect a face that is at the image edge and is only partially visible, the service can return coordinates that are outside the image bounds and, depending on the image edge, you might get negative values or values greater than 1 for the `left` or `top` values.

Contents

Height

Height of the bounding box as a ratio of the overall image height.

Type: Float

Required: No

Left

Left coordinate of the bounding box as a ratio of overall image width.

Type: Float

Required: No

Top

Top coordinate of the bounding box as a ratio of overall image height.

Type: Float

Required: No

Width

Width of the bounding box as a ratio of the overall image width.

Type: Float

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Celebrity

Service: Amazon Rekognition

Provides information about a celebrity recognized by the RecognizeCelebrities operation.

Contents

Face

Provides information about the celebrity's face, such as its location on the image.

Type: ComparedFace object

Required: No

Id

A unique identifier for the celebrity.

Type: String

Pattern: `[0-9A-Za-z]*`

Required: No

KnownGender

The known gender identity for the celebrity that matches the provided ID. The known gender identity can be Male, Female, Nonbinary, or Unlisted.

Type: KnownGender object

Required: No

MatchConfidence

The confidence, in percentage, that Amazon Rekognition has that the recognized face is the celebrity.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Name

The name of the celebrity.

Type: String

Required: No

Urls

An array of URLs pointing to additional information about the celebrity. If there is no additional information about the celebrity, this list is empty.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 255 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CelebrityDetail

Service: Amazon Rekognition

Information about a recognized celebrity.

Contents

BoundingBox

Bounding box around the body of a celebrity.

Type: BoundingBox object

Required: No

Confidence

The confidence, in percentage, that Amazon Rekognition has that the recognized face is the celebrity.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Face

Face details for the recognized celebrity.

Type: FaceDetail object

Required: No

Id

The unique identifier for the celebrity.

Type: String

Pattern: `[0-9A-Za-z]*`

Required: No

KnownGender

Retrieves the known gender for the celebrity.

Type: KnownGender object

Required: No

Name

The name of the celebrity.

Type: String

Required: No

Urls

An array of URLs pointing to additional celebrity information.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 255 items.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CelebrityRecognition

Service: Amazon Rekognition

Information about a detected celebrity and the time the celebrity was detected in a stored video. For more information, see GetCelebrityRecognition.

Contents

Celebrity

Information about a recognized celebrity.

Type: CelebrityDetail object

Required: No

Timestamp

The time, in milliseconds from the start of the video, that the celebrity was recognized. Note that `Timestamp` is not guaranteed to be accurate to the individual frame where the celebrity first appears.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Challenge

Service: Amazon Rekognition

Describes the type and version of the challenge being used for the Face Liveness session.

Contents

Type

The type of the challenge being used for the Face Liveness session.

Type: String

Valid Values: `FaceMovementAndLightChallenge | FaceMovementChallenge`

Required: Yes

Version

The version of the challenge being used for the Face Liveness session.

Type: String

Length Constraints: Minimum length of 5. Maximum length of 11.

Pattern: `^(0|[1-9]\d{0,2})\.(0|[1-9]\d{0,2})\.(0|[1-9]\d{0,2})$`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ChallengePreference

Service: Amazon Rekognition

An ordered list of preferred challenge type and versions.

Contents

Type

The types of challenges that have been selected for the Face Liveness session.

Type: String

Valid Values: `FaceMovementAndLightChallenge | FaceMovementChallenge`

Required: Yes

Versions

The version of the challenges that have been selected for the Face Liveness session.

Type: Versions object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ComparedFace

Service: Amazon Rekognition

Provides face metadata for target image faces that are analyzed by `CompareFaces` and `RecognizeCelebrities`.

Contents

BoundingBox

Bounding box of the face.

Type: BoundingBox object

Required: No

Confidence

Level of confidence that what the bounding box contains is a face.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Emotions

The emotions that appear to be expressed on the face, and the confidence level in the determination. Valid values include "Happy", "Sad", "Angry", "Confused", "Disgusted", "Surprised", "Calm", "Unknown", and "Fear".

Type: Array of Emotion objects

Required: No

Landmarks

An array of facial landmarks.

Type: Array of Landmark objects

Required: No

Pose

Indicates the pose of the face as determined by its pitch, roll, and yaw.

Type: Pose object

Required: No

Quality

Identifies face image brightness and sharpness.

Type: ImageQuality object

Required: No

Smile

Indicates whether or not the face is smiling, and the confidence level in the determination.

Type: Smile object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ComparedSourceImageFace

Service: Amazon Rekognition

Type that describes the face Amazon Rekognition chose to compare with the faces in the target. This contains a bounding box for the selected face and confidence level that the bounding box contains a face. Note that Amazon Rekognition selects the largest face in the source image for this comparison.

Contents

BoundingBox

Bounding box of the face.

Type: BoundingBox object

Required: No

Confidence

Confidence level that the selected bounding box contains a face.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CompareFacesMatch

Service: Amazon Rekognition

Provides information about a face in a target image that matches the source image face analyzed by `CompareFaces`. The `Face` property contains the bounding box of the face in the target image. The `Similarity` property is the confidence that the source image face matches the face in the bounding box.

Contents

Face

Provides face metadata (bounding box and confidence that the bounding box actually contains a face).

Type: ComparedFace object

Required: No

Similarity

Level of confidence that the faces match.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ConnectedHomeSettings

Service: Amazon Rekognition

Label detection settings to use on a streaming video. Defining the settings is required in the request parameter for CreateStreamProcessor. Including this setting in the `CreateStreamProcessor` request enables you to use the stream processor for label detection. You can then select what you want the stream processor to detect, such as people or pets. When the stream processor has started, one notification is sent for each object class specified. For example, if packages and pets are selected, one SNS notification is published the first time a package is detected and one SNS notification is published the first time a pet is detected, as well as an end-of-session summary.

Contents

Labels

Specifies what you want to detect in the video, such as people, packages, or pets. The current valid labels you can include in this list are: "PERSON", "PET", "PACKAGE", and "ALL".

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 128 items.

Required: Yes

MinConfidence

The minimum confidence required to label an object in the video.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

ConnectedHomeSettingsForUpdate

Service: Amazon Rekognition

The label detection settings you want to use in your stream processor. This includes the labels you want the stream processor to detect and the minimum confidence level allowed to label objects.

Contents

Labels

Specifies what you want to detect in the video, such as people, packages, or pets. The current valid labels you can include in this list are: "PERSON", "PET", "PACKAGE", and "ALL".

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 128 items.

Required: No

MinConfidence

The minimum confidence required to label an object in the video.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContentModerationDetection

Service: Amazon Rekognition

Information about an inappropriate, unwanted, or offensive content label detection in a stored video.

Contents

ContentTypes

A list of predicted results for the type of content an image contains. For example, the image content might be from animation, sports, or a video game.

Type: Array of ContentType objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

DurationMillis

The time duration of a segment in milliseconds, I.e. time elapsed from StartTimestampMillis to EndTimestampMillis.

Type: Long

Valid Range: Minimum value of 0.

Required: No

EndTimestampMillis

The time in milliseconds defining the end of the timeline segment containing a continuously detected moderation label.

Type: Long

Valid Range: Minimum value of 0.

Required: No

ModerationLabel

The content moderation label detected by in the stored video.

Type: ModerationLabel object

Required: No

StartTimestampMillis

The time in milliseconds defining the start of the timeline segment containing a continuously detected moderation label.

Type: Long

Valid Range: Minimum value of 0.

Required: No

Timestamp

Time, in milliseconds from the beginning of the video, that the content moderation label was detected. Note that `Timestamp` is not guaranteed to be accurate to the individual frame where the moderated content first appears.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContentType

Service: Amazon Rekognition

Contains information regarding the confidence and name of a detected content type.

Contents

Confidence

The confidence level of the label given

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Name

The name of the label

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CoversBodyPart

Service: Amazon Rekognition

Information about an item of Personal Protective Equipment covering a corresponding body part. For more information, see DetectProtectiveEquipment.

Contents

Confidence

The confidence that Amazon Rekognition has in the value of `Value`.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

True if the PPE covers the corresponding body part, otherwise false.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CreateFaceLivenessSessionRequestSettings

Service: Amazon Rekognition

A session settings object. It contains settings for the operation to be performed. It accepts arguments for OutputConfig and AuditImagesLimit.

Contents

AuditImagesLimit

Number of audit images to be returned back. Takes an integer between 0-4. Any integer less than 0 will return 0, any integer above 4 will return 4 images in the response. By default, it is set to 0. The limit is best effort and is based on the actual duration of the selfie-video.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 4.

Required: No

ChallengePreferences

Indicates preferred challenge types and versions for the Face Liveness session to be created.

Type: Array of ChallengePreference objects

Required: No

OutputConfig

Can specify the location of an Amazon S3 bucket, where reference and audit images will be stored. Note that the Amazon S3 bucket must be located in the caller's AWS account and in the same region as the Face Liveness end-point. Additionally, the Amazon S3 object keys are auto-generated by the Face Liveness system. Requires that the caller has the `s3:PutObject` permission on the Amazon S3 bucket.

Type: LivenessOutputConfig object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CustomizationFeatureConfig

Service: Amazon Rekognition

Feature specific configuration for the training job. Configuration provided for the job must match the feature type parameter associated with project. If configuration and feature type do not match an InvalidParameterException is returned.

Contents

ContentModeration

Configuration options for Custom Moderation training.

Type: CustomizationFeatureContentModerationConfig object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CustomizationFeatureContentModerationConfig

Service: Amazon Rekognition

Configuration options for Content Moderation training.

Contents

ConfidenceThreshold

The confidence level you plan to use to identify if unsafe content is present during inference.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CustomLabel

Service: Amazon Rekognition

A custom label detected in an image by a call to DetectCustomLabels.

Contents

Confidence

The confidence that the model has in the detection of the custom label. The range is 0-100. A higher value indicates a higher confidence.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Geometry

The location of the detected object on the image that corresponds to the custom label. Includes an axis aligned coarse bounding box surrounding the object and a finer grain polygon for more accurate spatial information.

Type: Geometry object

Required: No

Name

The name of the custom label.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

DatasetChanges

Service: Amazon Rekognition

Describes updates or additions to a dataset. A Single update or addition is an entry (JSON Line) that provides information about a single image. To update an existing entry, you match the `source-ref` field of the update entry with the `source-ref` filed of the entry that you want to update. If the `source-ref` field doesn't match an existing entry, the entry is added to dataset as a new entry.

Contents

GroundTruth

A Base64-encoded binary data object containing one or JSON lines that either update the dataset or are additions to the dataset. You change a dataset by calling UpdateDatasetEntries. If you are using an AWS SDK to call `UpdateDatasetEntries`, you don't need to encode `Changes` as the SDK encodes the data for you.

For example JSON lines, see Image-Level labels in manifest files and Object localization in manifest files.

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 1. Maximum length of 5242880.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DatasetDescription

Service: Amazon Rekognition

A description for a dataset. For more information, see DescribeDataset.

The status fields `Status`, `StatusMessage`, and `StatusMessageCode` reflect the last operation on the dataset.

Contents

CreationTimestamp

The Unix timestamp for the time and date that the dataset was created.

Type: Timestamp

Required: No

DatasetStats

The status message code for the dataset.

Type: DatasetStats object

Required: No

LastUpdatedTimestamp

The Unix timestamp for the date and time that the dataset was last updated.

Type: Timestamp

Required: No

Status

The status of the dataset.

Type: String

Valid Values: `CREATE_IN_PROGRESS | CREATE_COMPLETE | CREATE_FAILED | UPDATE_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_FAILED | DELETE_IN_PROGRESS`

Required: No

StatusMessage

The status message for the dataset.

Type: String

Required: No

StatusMessageCode

The status message code for the dataset operation. If a service error occurs, try the API call again later. If a client error occurs, check the input parameters to the dataset API call that failed.

Type: String

Valid Values: `SUCCESS | SERVICE_ERROR | CLIENT_ERROR`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DatasetLabelDescription

Service: Amazon Rekognition

Describes a dataset label. For more information, see ListDatasetLabels.

Contents

LabelName

The name of the label.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `.{1,}`

Required: No

LabelStats

Statistics about the label.

Type: DatasetLabelStats object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DatasetLabelStats

Service: Amazon Rekognition

Statistics about a label used in a dataset. For more information, see DatasetLabelDescription.

Contents

BoundingBoxCount

The total number of images that have the label assigned to a bounding box.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

EntryCount

The total number of images that use the label.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DatasetMetadata

Service: Amazon Rekognition

Summary information for an Amazon Rekognition Custom Labels dataset. For more information, see ProjectDescription.

Contents

CreationTimestamp

The Unix timestamp for the date and time that the dataset was created.

Type: Timestamp

Required: No

DatasetArn

The Amazon Resource Name (ARN) for the dataset.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Required: No

DatasetType

The type of the dataset.

Type: String

Valid Values: `TRAIN | TEST`

Required: No

Status

The status for the dataset.

Type: String

Valid Values: `CREATE_IN_PROGRESS | CREATE_COMPLETE | CREATE_FAILED | UPDATE_IN_PROGRESS | UPDATE_COMPLETE | UPDATE_FAILED | DELETE_IN_PROGRESS`

Required: No

StatusMessage

The status message for the dataset.

Type: String

Required: No

StatusMessageCode

The status message code for the dataset operation. If a service error occurs, try the API call again later. If a client error occurs, check the input parameters to the dataset API call that failed.

Type: String

Valid Values: `SUCCESS | SERVICE_ERROR | CLIENT_ERROR`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DatasetSource

Service: Amazon Rekognition

The source that Amazon Rekognition Custom Labels uses to create a dataset. To use an Amazon Sagemaker format manifest file, specify the S3 bucket location in the `GroundTruthManifest` field. The S3 bucket must be in your AWS account. To create a copy of an existing dataset, specify the Amazon Resource Name (ARN) of an existing dataset in `DatasetArn`.

You need to specify a value for `DatasetArn` or `GroundTruthManifest`, but not both. if you supply both values, or if you don't specify any values, an InvalidParameterException exception occurs.

For more information, see CreateDataset.

Contents

DatasetArn

The ARN of an Amazon Rekognition Custom Labels dataset that you want to copy.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Required: No

GroundTruthManifest

The S3 bucket that contains an Amazon Sagemaker Ground Truth format manifest file.

Type: GroundTruthManifest object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DatasetStats

Service: Amazon Rekognition

Provides statistics about a dataset. For more information, see DescribeDataset.

Contents

ErrorEntries

The total number of entries that contain at least one error.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

LabeledEntries

The total number of images in the dataset that have labels.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

TotalEntries

The total number of images in the dataset.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

TotalLabels

The total number of labels declared in the dataset.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DetectionFilter

Service: Amazon Rekognition

A set of parameters that allow you to filter out certain results from your returned results.

Contents

MinBoundingBoxHeight

Sets the minimum height of the word bounding box. Words with bounding box heights lesser than this value will be excluded from the result. Value is relative to the video frame height.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

MinBoundingBoxWidth

Sets the minimum width of the word bounding box. Words with bounding boxes widths lesser than this value will be excluded from the result. Value is relative to the video frame width.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

MinConfidence

Sets the confidence of word detection. Words with detection confidence below this will be excluded from the result. Values should be between 0 and 100. The default MinConfidence is 80.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DetectLabelsImageBackground

Service: Amazon Rekognition

The background of the image with regard to image quality and dominant colors.

Contents

DominantColors

The dominant colors found in the background of an image, defined with RGB values, CSS color name, simplified color name, and PixelPercentage (the percentage of image pixels that have a particular color).

Type: Array of DominantColor objects

Required: No

Quality

The quality of the image background as defined by brightness and sharpness.

Type: DetectLabelsImageQuality object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DetectLabelsImageForeground

Service: Amazon Rekognition

The foreground of the image with regard to image quality and dominant colors.

Contents

DominantColors

The dominant colors found in the foreground of an image, defined with RGB values, CSS color name, simplified color name, and PixelPercentage (the percentage of image pixels that have a particular color).

Type: Array of DominantColor objects

Required: No

Quality

The quality of the image foreground as defined by brightness and sharpness.

Type: DetectLabelsImageQuality object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DetectLabelsImageProperties

Service: Amazon Rekognition

Information about the quality and dominant colors of an input image. Quality and color information is returned for the entire image, foreground, and background.

Contents

Background

Information about the properties of an image's background, including the background's quality and dominant colors, including the quality and dominant colors of the image.

Type: DetectLabelsImageBackground object

Required: No

DominantColors

Information about the dominant colors found in an image, described with RGB values, CSS color name, simplified color name, and PixelPercentage (the percentage of image pixels that have a particular color).

Type: Array of DominantColor objects

Required: No

Foreground

Information about the properties of an image's foreground, including the foreground's quality and dominant colors, including the quality and dominant colors of the image.

Type: DetectLabelsImageForeground object

Required: No

Quality

Information about the quality of the image foreground as defined by brightness, sharpness, and contrast. The higher the value the greater the brightness, sharpness, and contrast respectively.

Type: DetectLabelsImageQuality object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DetectLabelsImagePropertiesSettings

Service: Amazon Rekognition

Settings for the IMAGE_PROPERTIES feature type.

Contents

MaxDominantColors

The maximum number of dominant colors to return when detecting labels in an image. The default value is 10.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 20.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DetectLabelsImageQuality

Service: Amazon Rekognition

The quality of an image provided for label detection, with regard to brightness, sharpness, and contrast.

Contents

Brightness

The brightness of an image provided for label detection.

Type: Float

Required: No

Contrast

The contrast of an image provided for label detection.

Type: Float

Required: No

Sharpness

The sharpness of an image provided for label detection.

Type: Float

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DetectLabelsSettings

Service: Amazon Rekognition

Settings for the DetectLabels request. Settings can include filters for both GENERAL_LABELS and IMAGE_PROPERTIES. GENERAL_LABELS filters can be inclusive or exclusive and applied to individual labels or label categories. IMAGE_PROPERTIES filters allow specification of a maximum number of dominant colors.

Contents

GeneralLabels

Contains the specified filters for GENERAL_LABELS.

Type: GeneralLabelsSettings object

Required: No

ImageProperties

Contains the chosen number of maximum dominant colors in an image.

Type: DetectLabelsImagePropertiesSettings object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DetectTextFilters

Service: Amazon Rekognition

A set of optional parameters that you can use to set the criteria that the text must meet to be included in your response. `WordFilter` looks at a word's height, width, and minimum confidence. `RegionOfInterest` lets you set a specific region of the image to look for text in.

Contents

RegionsOfInterest

A Filter focusing on a certain area of the image. Uses a `BoundingBox` object to set the region of the image.

Type: Array of RegionOfInterest objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

WordFilter

A set of parameters that allow you to filter out certain results from your returned results.

Type: DetectionFilter object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DisassociatedFace

Service: Amazon Rekognition

Provides face metadata for the faces that are disassociated from a specific UserID.

Contents

FaceId

Unique identifier assigned to the face.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DistributeDataset

Service: Amazon Rekognition

A training dataset or a test dataset used in a dataset distribution operation. For more information, see DistributeDatasetEntries.

Contents

Arn

The Amazon Resource Name (ARN) of the dataset that you want to use.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/dataset\/(train|test)\/[0-9]+$)`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DominantColor

Service: Amazon Rekognition

A description of the dominant colors in an image.

Contents

Blue

The Blue RGB value for a dominant color.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

CSSColor

The CSS color name of a dominant color.

Type: String

Required: No

Green

The Green RGB value for a dominant color.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

HexCode

The Hex code equivalent of the RGB values for a dominant color.

Type: String

Required: No

PixelPercent

The percentage of image pixels that have a given dominant color.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Red

The Red RGB value for a dominant color.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

SimplifiedColor

One of 12 simplified color names applied to a dominant color.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Emotion

Service: Amazon Rekognition

The API returns a prediction of an emotion based on a person's facial expressions, along with the confidence level for the predicted emotion. It is not a determination of the person’s internal emotional state and should not be used in such a way. For example, a person pretending to have a sad face might not be sad emotionally. The API is not intended to be used, and you may not use it, in a manner that violates the EU Artificial Intelligence Act or any other applicable law.

Contents

Confidence

Level of confidence in the determination.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Type

Type of emotion detected.

Type: String

Valid Values: `HAPPY | SAD | ANGRY | CONFUSED | DISGUSTED | SURPRISED | CALM | UNKNOWN | FEAR`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EquipmentDetection

Service: Amazon Rekognition

Information about an item of Personal Protective Equipment (PPE) detected by DetectProtectiveEquipment. For more information, see DetectProtectiveEquipment.

Contents

BoundingBox

A bounding box surrounding the item of detected PPE.

Type: BoundingBox object

Required: No

Confidence

The confidence that Amazon Rekognition has that the bounding box (`BoundingBox`) contains an item of PPE.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

CoversBodyPart

Information about the body part covered by the detected PPE.

Type: CoversBodyPart object

Required: No

Type

The type of detected PPE.

Type: String

Valid Values: `FACE_COVER` | `HAND_COVER` | `HEAD_COVER`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EvaluationResult

Service: Amazon Rekognition

The evaluation results for the training of a model.

Contents

F1Score

The F1 score for the evaluation of all labels. The F1 score metric evaluates the overall precision and recall performance of the model as a single value. A higher value indicates better precision and recall performance. A lower score indicates that precision, recall, or both are performing poorly.

Type: Float

Required: No

Summary

The S3 bucket that contains the training summary.

Type: Summary object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EyeDirection

Service: Amazon Rekognition

Indicates the direction the eyes are gazing in (independent of the head pose) as determined by its pitch and yaw.

Contents

Confidence

The confidence that the service has in its predicted eye direction.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Pitch

Value representing eye direction on the pitch axis.

Type: Float

Valid Range: Minimum value of -180. Maximum value of 180.

Required: No

Yaw

Value representing eye direction on the yaw axis.

Type: Float

Valid Range: Minimum value of -180. Maximum value of 180.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

Eyeglasses

Service: Amazon Rekognition

Indicates whether or not the face is wearing eye glasses, and the confidence level in the determination.

Contents

Confidence

Level of confidence in the determination.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

Boolean value that indicates whether the face is wearing eye glasses or not.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EyeOpen

Service: Amazon Rekognition

Indicates whether or not the eyes on the face are open, and the confidence level in the determination.

Contents

Confidence

Level of confidence in the determination.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

Boolean value that indicates whether the eyes on the face are open.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Face

Service: Amazon Rekognition

Describes the face properties such as the bounding box, face ID, image ID of the input image, and external image ID that you assigned.

Contents

BoundingBox

Bounding box of the face.

Type: BoundingBox object

Required: No

Confidence

Confidence level that the bounding box contains a face (and not a different object such as a tree).

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

ExternalImageId

Identifier that you assign to all the faces in the input image.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

FaceId

Unique identifier that Amazon Rekognition assigns to the face.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

ImageId

Unique identifier that Amazon Rekognition assigns to the input image.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

IndexFacesModelVersion

The version of the face detect and storage model that was used when indexing the face vector.

Type: String

Pattern: `[0-9\.]+`

Required: No

UserId

Unique identifier assigned to the user.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FaceDetail

Service: Amazon Rekognition

Structure containing attributes of the face that the algorithm detected.

A `FaceDetail` object contains either the default facial attributes or all facial attributes. The default attributes are `BoundingBox`, `Confidence`, `Landmarks`, `Pose`, and `Quality`.

GetFaceDetection is the only Amazon Rekognition Video stored video operation that can return a `FaceDetail` object with all attributes. To specify which attributes to return, use the `FaceAttributes` input parameter for StartFaceDetection. The following Amazon Rekognition Video operations return only the default attributes. The corresponding Start operations don't have a `FaceAttributes` input parameter:

- GetCelebrityRecognition
- GetPersonTracking
- GetFaceSearch

The Amazon Rekognition Image DetectFaces and IndexFaces operations can return all facial attributes. To specify which attributes to return, use the `Attributes` input parameter for `DetectFaces`. For `IndexFaces`, use the `DetectAttributes` input parameter.

Contents

AgeRange

The estimated age range, in years, for the face. Low represents the lowest estimated age and High represents the highest estimated age.

Type: AgeRange object

Required: No

Beard

Indicates whether or not the face has a beard, and the confidence level in the determination.

Type: Beard object

Required: No

BoundingBox

Bounding box of the face. Default attribute.

Type: BoundingBox object

Required: No

Confidence

Confidence level that the bounding box contains a face (and not a different object such as a tree). Default attribute.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Emotions

The emotions that appear to be expressed on the face, and the confidence level in the determination. The API is only making a determination of the physical appearance of a person's face. It is not a determination of the person’s internal emotional state and should not be used in such a way. For example, a person pretending to have a sad face might not be sad emotionally.

Type: Array of Emotion objects

Required: No

EyeDirection

Indicates the direction the eyes are gazing in, as defined by pitch and yaw.

Type: EyeDirection object

Required: No

Eyeglasses

Indicates whether or not the face is wearing eye glasses, and the confidence level in the determination.

Type: Eyeglasses object

Required: No

EyesOpen

Indicates whether or not the eyes on the face are open, and the confidence level in the determination.

Type: EyeOpen object

Required: No

FaceOccluded

`FaceOccluded` should return "true" with a high confidence score if a detected face's eyes, nose, and mouth are partially captured or if they are covered by masks, dark sunglasses, cell phones, hands, or other objects. `FaceOccluded` should return "false" with a high confidence score if common occurrences that do not impact face verification are detected, such as eye glasses, lightly tinted sunglasses, strands of hair, and others.

Type: FaceOccluded object

Required: No

Gender

The predicted gender of a detected face.

Type: Gender object

Required: No

Landmarks

Indicates the location of landmarks on the face. Default attribute.

Type: Array of Landmark objects

Required: No

MouthOpen

Indicates whether or not the mouth on the face is open, and the confidence level in the determination.

Type: MouthOpen object

Required: No

Mustache

Indicates whether or not the face has a mustache, and the confidence level in the determination.

Type: Mustache object

Required: No

Pose

Indicates the pose of the face as determined by its pitch, roll, and yaw. Default attribute.

Type: Pose object

Required: No

Quality

Identifies image brightness and sharpness. Default attribute.

Type: ImageQuality object

Required: No

Smile

Indicates whether or not the face is smiling, and the confidence level in the determination.

Type: Smile object

Required: No

Sunglasses

Indicates whether or not the face is wearing sunglasses, and the confidence level in the determination.

Type: Sunglasses object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FaceDetection

Service: Amazon Rekognition

Information about a face detected in a video analysis request and the time the face was detected in the video.

Contents

Face

The face properties for the detected face.

Type: FaceDetail object

Required: No

Timestamp

Time, in milliseconds from the start of the video, that the face was detected. Note that `Timestamp` is not guaranteed to be accurate to the individual frame where the face first appears.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FaceMatch

Service: Amazon Rekognition

Provides face metadata. In addition, it also provides the confidence in the match of this face with the input face.

Contents

Face

Describes the face properties such as the bounding box, face ID, image ID of the source image, and external image ID that you assigned.

Type: Face object

Required: No

Similarity

Confidence in the match of this face with the input face.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FaceOccluded

Service: Amazon Rekognition

`FaceOccluded` should return "true" with a high confidence score if a detected face's eyes, nose, and mouth are partially captured or if they are covered by masks, dark sunglasses, cell phones, hands, or other objects. `FaceOccluded` should return "false" with a high confidence score if common occurrences that do not impact face verification are detected, such as eye glasses, lightly tinted sunglasses, strands of hair, and others.

You can use `FaceOccluded` to determine if an obstruction on a face negatively impacts using the image for face matching.

Contents

Confidence

The confidence that the service has detected the presence of a face occlusion.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

True if a detected face's eyes, nose, and mouth are partially captured or if they are covered by masks, dark sunglasses, cell phones, hands, or other objects. False if common occurrences that do not impact face verification are detected, such as eye glasses, lightly tinted sunglasses, strands of hair, and others.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

FaceRecord

Service: Amazon Rekognition

Object containing both the face metadata (stored in the backend database), and facial attributes that are detected but aren't stored in the database.

Contents

Face

Describes the face properties such as the bounding box, face ID, image ID of the input image, and external image ID that you assigned.

Type: Face object

Required: No

FaceDetail

Structure containing attributes of the face that the algorithm detected.

Type: FaceDetail object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FaceSearchSettings

Service: Amazon Rekognition

Input face recognition parameters for an Amazon Rekognition stream processor. Includes the collection to use for face recognition and the face attributes to detect. Defining the settings is required in the request parameter for CreateStreamProcessor.

Contents

CollectionId

The ID of a collection that contains faces that you want to search for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: No

FaceMatchThreshold

Minimum face match confidence score that must be met to return a result for a recognized face. The default is 80. 0 is the lowest confidence. 100 is the highest confidence. Values between 0 and 100 are accepted, and values lower than 80 are set to 80.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Gender

Service: Amazon Rekognition

The predicted gender of a detected face.

Amazon Rekognition makes gender binary (male/female) predictions based on the physical appearance of a face in a particular image. This kind of prediction is not designed to categorize a person's gender identity, and you shouldn't use Amazon Rekognition to make such a determination. For example, a male actor wearing a long-haired wig and earrings for a role might be predicted as female.

Using Amazon Rekognition to make gender binary predictions is best suited for use cases where aggregate gender distribution statistics need to be analyzed without identifying specific users. For example, the percentage of female users compared to male users on a social media platform.

We don't recommend using gender binary predictions to make decisions that impact an individual's rights, privacy, or access to services.

Contents

Confidence

Level of confidence in the prediction.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

The predicted gender of the face.

Type: String

Valid Values: `Male | Female`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GeneralLabelsSettings

Service: Amazon Rekognition

Contains filters for the object labels returned by DetectLabels. Filters can be inclusive, exclusive, or a combination of both and can be applied to individual labels or entire label categories. To see a list of label categories, see Detecting Labels.

Contents

LabelCategoryExclusionFilters

The label categories that should be excluded from the return from DetectLabels.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 0. Maximum length of 50.

Pattern: `[A-Za-z0-9àâèçñó'-_(). ]*`

Required: No

LabelCategoryInclusionFilters

The label categories that should be included in the return from DetectLabels.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 0. Maximum length of 50.

Pattern: `[A-Za-z0-9àâèçñó'-_(). ]*`

Required: No

LabelExclusionFilters

The labels that should be excluded from the return from DetectLabels.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 0. Maximum length of 50.

Pattern: `[A-Za-z0-9àâèçñó'-_(). ]*`

Required: No

LabelInclusionFilters

The labels that should be included in the return from DetectLabels.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.

Length Constraints: Minimum length of 0. Maximum length of 50.

Pattern: `[A-Za-z0-9àâèçñó'-_(). ]*`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Geometry

Service: Amazon Rekognition

Information about where an object (DetectCustomLabels) or text (DetectText) is located on an image.

Contents

BoundingBox

An axis-aligned coarse representation of the detected item's location on the image.

Type: BoundingBox object

Required: No

Polygon

Within the bounding box, a fine-grained polygon around the detected item.

Type: Array of Point objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetContentModerationRequestMetadata

Service: Amazon Rekognition

Contains metadata about a content moderation request, including the SortBy and AggregateBy options.

Contents

AggregateBy

The aggregation method chosen for a GetContentModeration request.

Type: String

Valid Values: `TIMESTAMPS | SEGMENTS`

Required: No

SortBy

The sorting method chosen for a GetContentModeration request.

Type: String

Valid Values: `NAME | TIMESTAMP`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GetLabelDetectionRequestMetadata

Service: Amazon Rekognition

Contains metadata about a label detection request, including the SortBy and AggregateBy options.

Contents

AggregateBy

The aggregation method chosen for a GetLabelDetection request.

Type: String

Valid Values: `TIMESTAMPS | SEGMENTS`

Required: No

SortBy

The sorting method chosen for a GetLabelDetection request.

Type: String

Valid Values: `NAME | TIMESTAMP`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GroundTruthManifest

Service: Amazon Rekognition

The S3 bucket that contains an Amazon Sagemaker Ground Truth format manifest file.

Contents

S3Object

Provides the S3 bucket name and object name.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Type: S3Object object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HumanLoopActivationOutput

Service: Amazon Rekognition

Shows the results of the human in the loop evaluation. If there is no HumanLoopArn, the input did not trigger human review.

Contents

HumanLoopActivationConditionsEvaluationResults

Shows the result of condition evaluations, including those conditions which activated a human review.

Type: String

Length Constraints: Maximum length of 10240.

Required: No

HumanLoopActivationReasons

Shows if and why human review was needed.

Type: Array of strings

Array Members: Minimum number of 1 item.

Required: No

HumanLoopArn

The Amazon Resource Name (ARN) of the HumanLoop created.

Type: String

Length Constraints: Maximum length of 256.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HumanLoopConfig

Service: Amazon Rekognition

Sets up the flow definition the image will be sent to if one of the conditions is met. You can also set certain attributes of the image before review.

Contents

FlowDefinitionArn

The Amazon Resource Name (ARN) of the flow definition. You can create a flow definition by using the Amazon Sagemaker CreateFlowDefinition Operation.

Type: String

Length Constraints: Maximum length of 256.

Required: Yes

HumanLoopName

The name of the human review used for this image. This should be kept unique within a region.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 63.

Pattern: `^[a-z0-9](-*[a-z0-9])*`

Required: Yes

DataAttributes

Sets attributes of the input data.

Type: HumanLoopDataAttributes object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HumanLoopDataAttributes

Service: Amazon Rekognition

Allows you to set attributes of the image. Currently, you can declare an image as free of personally identifiable information.

Contents

ContentClassifiers

Sets whether the input image is free of personally identifiable information.

Type: Array of strings

Array Members: Maximum number of 256 items.

Valid Values: `FreeOfPersonallyIdentifiableInformation | FreeOfAdultContent`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Image

Service: Amazon Rekognition

Provides the input image either as bytes or an S3 object.

You pass image bytes to an Amazon Rekognition API operation by using the `Bytes` property. For example, you would use the `Bytes` property to pass an image loaded from a local file system. Image bytes passed by using the `Bytes` property must be base64-encoded. Your code may not need to encode image bytes if you are using an AWS SDK to call Amazon Rekognition API operations.

For more information, see Analyzing an image loaded from a local file system .

You pass images stored in an S3 bucket to an Amazon Rekognition API operation by using the `S3Object` property. Images stored in an S3 bucket do not need to be base64-encoded.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

If you use the AWS CLI to call Amazon Rekognition operations, passing image bytes using the Bytes property is not supported. You must first upload the image to an Amazon S3 bucket and then call the operation using the S3Object property.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Contents

Bytes

Blob of image bytes up to 5 MBs. Note that the maximum image size you can pass to DetectCustomLabels is 4MB.

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 1. Maximum length of 5242880.

Required: No

S3Object

Identifies an S3 object as the image source.

Type: S3Object object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ImageQuality

Service: Amazon Rekognition

Identifies face image brightness and sharpness.

Contents

Brightness

Value representing brightness of the face. The service returns a value between 0 and 100 (inclusive). A higher value indicates a brighter face image.

Type: Float

Required: No

Sharpness

Value representing sharpness of the face. The service returns a value between 0 and 100 (inclusive). A higher value indicates a sharper face image.

Type: Float

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Instance

Service: Amazon Rekognition

An instance of a label returned by Amazon Rekognition Image (DetectLabels) or by Amazon Rekognition Video (GetLabelDetection).

Contents

BoundingBox

The position of the label instance on the image.

Type: BoundingBox object

Required: No

Confidence

The confidence that Amazon Rekognition has in the accuracy of the bounding box.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

DominantColors

The dominant colors found in an individual instance of a label.

Type: Array of DominantColor objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KinesisDataStream

Service: Amazon Rekognition

The Kinesis data stream Amazon Rekognition to which the analysis results of a Amazon Rekognition stream processor are streamed. For more information, see CreateStreamProcessor.

Contents

Arn

ARN of the output Amazon Kinesis Data Streams stream.

Type: String

Pattern: `(^arn:([a-z\d-]+):kinesis:([a-z\d-]+):\d{12}:.+$)`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KinesisVideoStream

Service: Amazon Rekognition

Kinesis video stream stream that provides the source streaming video for a Amazon Rekognition Video stream processor. For more information, see CreateStreamProcessor.

Contents

Arn

ARN of the Kinesis video stream stream that streams the source video.

Type: String

Pattern: `(^arn:([a-z\d-]+):kinesisvideo:([a-z\d-]+):\d{12}:.+$)`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KinesisVideoStreamStartSelector

Service: Amazon Rekognition

Specifies the starting point in a Kinesis stream to start processing. You can use the producer timestamp or the fragment number. One of either producer timestamp or fragment number is required. If you use the producer timestamp, you must put the time in milliseconds. For more information about fragment numbers, see Fragment.

Contents

FragmentNumber

The unique identifier of the fragment. This value monotonically increases based on the ingestion order.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `^[0-9]+$`

Required: No

ProducerTimestamp

The timestamp from the producer corresponding to the fragment, in milliseconds, expressed in unix time format.

Type: Long

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

KnownGender

Service: Amazon Rekognition

The known gender identity for the celebrity that matches the provided ID. The known gender identity can be Male, Female, Nonbinary, or Unlisted.

Contents

Type

A string value of the KnownGender info about the Celebrity.

Type: String

Valid Values: `Male | Female | Nonbinary | Unlisted`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Label

Service: Amazon Rekognition

Structure containing details about the detected label, including the name, detected instances, parent labels, and level of confidence.

Contents

Aliases

A list of potential aliases for a given label.

Type: Array of LabelAlias objects

Required: No

Categories

A list of the categories associated with a given label.

Type: Array of LabelCategory objects

Required: No

Confidence

Level of confidence.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Instances

If `Label` represents an object, `Instances` contains the bounding boxes for each instance of the detected object. Bounding boxes are returned for common object labels such as people, cars, furniture, apparel or pets.

Type: Array of Instance objects

Required: No

Name

The name (label) of the object or scene.

Type: String

Required: No

Parents

The parent labels for a label. The response includes all ancestor labels.

Type: Array of Parent objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LabelAlias

Service: Amazon Rekognition

A potential alias of for a given label.

Contents

Name

The name of an alias for a given label.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LabelCategory

Service: Amazon Rekognition

The category that applies to a given label.

Contents

Name

The name of a category that applies to a given label.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LabelDetection

Service: Amazon Rekognition

Information about a label detected in a video analysis request and the time the label was detected in the video.

Contents

DurationMillis

The time duration of a segment in milliseconds, I.e. time elapsed from StartTimestampMillis to EndTimestampMillis.

Type: Long

Valid Range: Minimum value of 0.

Required: No

EndTimestampMillis

The time in milliseconds defining the end of the timeline segment containing a continuously detected label.

Type: Long

Valid Range: Minimum value of 0.

Required: No

Label

Details about the detected label.

Type: Label object

Required: No

StartTimestampMillis

The time in milliseconds defining the start of the timeline segment containing a continuously detected label.

Type: Long

Valid Range: Minimum value of 0.

Required: No

Timestamp

Time, in milliseconds from the start of the video, that the label was detected. Note that `Timestamp` is not guaranteed to be accurate to the individual frame where the label first appears.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LabelDetectionSettings

Service: Amazon Rekognition

Contains the specified filters that should be applied to a list of returned GENERAL_LABELS.

Contents

GeneralLabels

Contains filters for the object labels returned by DetectLabels. Filters can be inclusive, exclusive, or a combination of both and can be applied to individual labels or entire label categories. To see a list of label categories, see Detecting Labels.

Type: GeneralLabelsSettings object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Landmark

Service: Amazon Rekognition

Indicates the location of the landmark on the face.

Contents

Type

Type of landmark.

Type: String

Valid Values: `eyeLeft | eyeRight | nose | mouthLeft | mouthRight | leftEyeBrowLeft | leftEyeBrowRight | leftEyeBrowUp | rightEyeBrowLeft | rightEyeBrowRight | rightEyeBrowUp | leftEyeLeft | leftEyeRight | leftEyeUp | leftEyeDown | rightEyeLeft | rightEyeRight | rightEyeUp | rightEyeDown | noseLeft | noseRight | mouthUp | mouthDown | leftPupil | rightPupil | upperJawlineLeft | midJawlineLeft | chinBottom | midJawlineRight | upperJawlineRight`

Required: No

X

The x-coordinate of the landmark expressed as a ratio of the width of the image. The x-coordinate is measured from the left-side of the image. For example, if the image is 700 pixels wide and the x-coordinate of the landmark is at 350 pixels, this value is 0.5.

Type: Float

Required: No

Y

The y-coordinate of the landmark expressed as a ratio of the height of the image. The y-coordinate is measured from the top of the image. For example, if the image height is 200 pixels and the y-coordinate of the landmark is at 50 pixels, this value is 0.25.

Type: Float

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LivenessOutputConfig

Service: Amazon Rekognition

Contains settings that specify the location of an Amazon S3 bucket used to store the output of a Face Liveness session. Note that the S3 bucket must be located in the caller's AWS account and in the same region as the Face Liveness end-point. Additionally, the Amazon S3 object keys are auto-generated by the Face Liveness system.

Contents

S3Bucket

The path to an AWS Amazon S3 bucket used to store Face Liveness session results.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[0-9A-Za-z\.\-_]*`

Required: Yes

S3KeyPrefix

The prefix prepended to the output files for the Face Liveness session results.

Type: String

Length Constraints: Maximum length of 950.

Pattern: `\S*`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

MatchedUser

Service: Amazon Rekognition

Contains metadata for a UserID matched with a given face.

Contents

UserId

A provided ID for the UserID. Unique within the collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

UserStatus

The status of the user matched to a provided FaceID.

Type: String

Valid Values: `ACTIVE | UPDATING | CREATING | CREATED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisDetectModerationLabelsConfig

Service: Amazon Rekognition

Configuration for Moderation Labels Detection.

Contents

MinConfidence

Specifies the minimum confidence level for the moderation labels to return. Amazon Rekognition doesn't return any labels with a confidence level lower than this specified value.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

ProjectVersion

Specifies the custom moderation model to be used during the label detection job. If not provided the pre-trained model is used.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisInput

Service: Amazon Rekognition

Contains input information for a media analysis job.

Contents

S3Object

Provides the S3 bucket name and object name.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Type: S3Object object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisJobDescription

Service: Amazon Rekognition

Description for a media analysis job.

Contents

CreationTimestamp

The Unix date and time when the job was started.

Type: Timestamp

Required: Yes

Input

Reference to the input manifest that was provided in the job creation request.

Type: MediaAnalysisInput object

Required: Yes

JobId

The identifier for a media analysis job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z0-9-_]+$`

Required: Yes

OperationsConfig

Operation configurations that were provided during job creation.

Type: MediaAnalysisOperationsConfig object

Required: Yes

OutputConfig

Output configuration that was provided in the creation request.

Type: MediaAnalysisOutputConfig object

Required: Yes

Status

The status of the media analysis job being retrieved.

Type: String

Valid Values: `CREATED | QUEUED | IN_PROGRESS | SUCCEEDED | FAILED`

Required: Yes

CompletionTimestamp

The Unix date and time when the job finished.

Type: Timestamp

Required: No

FailureDetails

Details about the error that resulted in failure of the job.

Type: MediaAnalysisJobFailureDetails object

Required: No

JobName

The name of a media analysis job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: No

KmsKeyId

KMS Key that was provided in the creation request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

Required: No

ManifestSummary

Provides statistics on input manifest and errors identified in the input manifest.

Type: MediaAnalysisManifestSummary object

Required: No

Results

Output manifest that contains prediction results.

Type: MediaAnalysisResults object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisJobFailureDetails

Service: Amazon Rekognition

Details about the error that resulted in failure of the job.

Contents

Code

Error code for the failed job.

Type: String

Valid Values: `INTERNAL_ERROR | INVALID_S3_OBJECT | INVALID_MANIFEST | INVALID_OUTPUT_CONFIG | INVALID_KMS_KEY | ACCESS_DENIED | RESOURCE_NOT_FOUND | RESOURCE_NOT_READY | THROTTLED`

Required: No

Message

Human readable error message.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisManifestSummary

Service: Amazon Rekognition

Summary that provides statistics on input manifest and errors identified in the input manifest.

Contents

S3Object

Provides the S3 bucket name and object name.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Type: S3Object object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisModelVersions

Service: Amazon Rekognition

Object containing information about the model versions of selected features in a given job.

Contents

Moderation

The Moderation base model version.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisOperationsConfig

Service: Amazon Rekognition

Configuration options for a media analysis job. Configuration is operation-specific.

Contents

DetectModerationLabels

Contains configuration options for a DetectModerationLabels job.

Type: MediaAnalysisDetectModerationLabelsConfig object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisOutputConfig

Service: Amazon Rekognition

Output configuration provided in the job creation request.

Contents

S3Bucket

Specifies the Amazon S3 bucket to contain the output of the media analysis job.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[0-9A-Za-z\.\-_]*`

Required: Yes

S3KeyPrefix

Specifies the Amazon S3 key prefix that comes after the name of the bucket you have designated for storage.

Type: String

Length Constraints: Maximum length of 800.

Pattern: `\S*`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaAnalysisResults

Service: Amazon Rekognition

Contains the results for a media analysis job created with StartMediaAnalysisJob.

Contents

ModelVersions

Information about the model versions for the features selected in a given job.

Type: MediaAnalysisModelVersions object

Required: No

S3Object

Provides the S3 bucket name and object name.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Type: S3Object object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ModerationLabel

Service: Amazon Rekognition

Provides information about a single type of inappropriate, unwanted, or offensive content found in an image or video. Each type of moderated content has a label within a hierarchical taxonomy. For a list of moderation labels in Amazon Rekognition, see Using the image and video moderation APIs. For more information, see Moderating content.

Contents

Confidence

Specifies the confidence that Amazon Rekognition has that the label has been correctly identified.

If you don't specify the `MinConfidence` parameter in the call to `DetectModerationLabels`, the operation returns labels with a confidence value greater than or equal to 50 percent.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Name

The label name for the type of unsafe content detected in the image.

Type: String

Required: No

ParentName

The name for the parent label. Labels at the top level of the hierarchy have the parent label `""`.

Type: String

Required: No

TaxonomyLevel

The level of the moderation label with regard to its taxonomy, from 1 to 3.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MouthOpen

Service: Amazon Rekognition

Indicates whether or not the mouth on the face is open, and the confidence level in the determination.

Contents

Confidence

Level of confidence in the determination.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

Boolean value that indicates whether the mouth on the face is open or not.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Mustache

Service: Amazon Rekognition

Indicates whether or not the face has a mustache, and the confidence level in the determination.

Contents

Confidence

Level of confidence in the determination.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

Boolean value that indicates whether the face has mustache or not.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NotificationChannel

Service: Amazon Rekognition

The Amazon Simple Notification Service topic to which Amazon Rekognition publishes the completion status of a video analysis operation. For more information, see Calling Amazon Rekognition Video operations. Note that the Amazon SNS topic must have a topic name that begins with *AmazonRekognition* if you are using the AmazonRekognitionServiceRole permissions policy to access the topic. For more information, see Giving access to multiple Amazon SNS topics.

Contents

RoleArn

The ARN of an IAM role that gives Amazon Rekognition publishing permissions to the Amazon SNS topic.

Type: String

Pattern: `arn:aws:iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`

Required: Yes

SNSTopicArn

The Amazon SNS topic to which Amazon Rekognition posts the completion status.

Type: String

Pattern: `(^arn:aws:sns:.*:\w{12}:.+$)`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

OutputConfig

Service: Amazon Rekognition

The S3 bucket and folder location where training output is placed.

Contents

S3Bucket

The S3 bucket where training output is placed.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[0-9A-Za-z\.\-_]*`

Required: No

S3KeyPrefix

The prefix applied to the training output files.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Parent

Service: Amazon Rekognition

A parent label for a label. A label can have 0, 1, or more parents.

Contents

Name

The name of the parent label.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PersonDetail

Service: Amazon Rekognition

Details about a person detected in a video analysis request.

Contents

BoundingBox

Bounding box around the detected person.

Type: BoundingBox object

Required: No

Face

Face details for the detected person.

Type: FaceDetail object

Required: No

Index

Identifier for the person detected person within a video. Use to keep track of the person throughout the video. The identifier is not stored by Amazon Rekognition.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PersonDetection

Service: Amazon Rekognition

Details and path tracking information for a single time a person's path is tracked in a video. Amazon Rekognition operations that track people's paths return an array of `PersonDetection` objects with elements for each time a person's path is tracked in a video.

For more information, see GetPersonTracking.

Contents

Person

Details about a person whose path was tracked in a video.

Type: PersonDetail object

Required: No

Timestamp

The time, in milliseconds from the start of the video, that the person's path was tracked. Note that `Timestamp` is not guaranteed to be accurate to the individual frame where the person's path first appears.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PersonMatch

Service: Amazon Rekognition

Information about a person whose face matches a face(s) in an Amazon Rekognition collection. Includes information about the faces in the Amazon Rekognition collection (FaceMatch), information about the person (PersonDetail), and the time stamp for when the person was detected in a video. An array of `PersonMatch` objects is returned by GetFaceSearch.

Contents

FaceMatches

Information about the faces in the input collection that match the face of a person in the video.

Type: Array of FaceMatch objects

Required: No

Person

Information about the matched person.

Type: PersonDetail object

Required: No

Timestamp

The time, in milliseconds from the beginning of the video, that the person was matched in the video.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

Point

Service: Amazon Rekognition

The X and Y coordinates of a point on an image or video frame. The X and Y values are ratios of the overall image size or video resolution. For example, if an input image is 700x200 and the values are X=0.5 and Y=0.25, then the point is at the (350,50) pixel coordinate on the image.

An array of `Point` objects, `Polygon`, is returned by DetectText and DetectCustomLabels or used to define regions of interest in Amazon Rekognition Video operations such as `CreateStreamProcessor`. `Polygon` represents a fine-grained polygon around a detected item. For more information, see Geometry.

Contents

X

The value of the X coordinate for a point on a `Polygon`.

Type: Float

Required: No

Y

The value of the Y coordinate for a point on a `Polygon`.

Type: Float

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Pose

Service: Amazon Rekognition

Indicates the pose of the face as determined by its pitch, roll, and yaw.

Contents

Pitch

Value representing the face rotation on the pitch axis.

Type: Float

Valid Range: Minimum value of -180. Maximum value of 180.

Required: No

Roll

Value representing the face rotation on the roll axis.

Type: Float

Valid Range: Minimum value of -180. Maximum value of 180.

Required: No

Yaw

Value representing the face rotation on the yaw axis.

Type: Float

Valid Range: Minimum value of -180. Maximum value of 180.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProjectDescription

Service: Amazon Rekognition

A description of an Amazon Rekognition Custom Labels project. For more information, see DescribeProjects.

Contents

AutoUpdate

Indicates whether automatic retraining will be attempted for the versions of the project. Applies only to adapters.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

CreationTimestamp

The Unix timestamp for the date and time that the project was created.

Type: Timestamp

Required: No

Datasets

Information about the training and test datasets in the project.

Type: Array of DatasetMetadata objects

Required: No

Feature

Specifies the project that is being customized.

Type: String

Valid Values: `CONTENT_MODERATION | CUSTOM_LABELS`

Required: No

ProjectArn

The Amazon Resource Name (ARN) of the project.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: No

Status

The current status of the project.

Type: String

Valid Values: `CREATING | CREATED | DELETING`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProjectPolicy

Service: Amazon Rekognition

Describes a project policy in the response from ListProjectPolicies.

Contents

CreationTimestamp

The Unix datetime for the creation of the project policy.

Type: Timestamp

Required: No

LastUpdatedTimestamp

The Unix datetime for when the project policy was last updated.

Type: Timestamp

Required: No

PolicyDocument

The JSON document for the project policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2000.

Pattern: `[\u0009\u000A\u000D\u0020-\u00FF]+`

Required: No

PolicyName

The name of the project policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: No

PolicyRevisionId

The revision ID of the project policy.

Type: String

Length Constraints: Maximum length of 64.

Pattern: `[0-9A-Fa-f]+`

Required: No

ProjectArn

The Amazon Resource Name (ARN) of the project to which the project policy is attached.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProjectVersionDescription

Service: Amazon Rekognition

A description of a version of a Amazon Rekognition project version.

Contents

BaseModelVersion

The base detection model version used to create the project version.

Type: String

Required: No

BillableTrainingTimeInSeconds

The duration, in seconds, that you were billed for a successful training of the model version. This value is only returned if the model version has been successfully trained.

Type: Long

Valid Range: Minimum value of 0.

Required: No

CreationTimestamp

The Unix datetime for the date and time that training started.

Type: Timestamp

Required: No

EvaluationResult

The training results. `EvaluationResult` is only returned if training is successful.

Type: EvaluationResult object

Required: No

Feature

The feature that was customized.

Type: String

Valid Values: `CONTENT_MODERATION | CUSTOM_LABELS`

Required: No

FeatureConfig

Feature specific configuration that was applied during training.

Type: CustomizationFeatureConfig object

Required: No

KmsKeyId

The identifer for the AWS Key Management Service key (AWS KMS key) that was used to encrypt the model during training.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: `^[A-Za-z0-9][A-Za-z0-9:_/+=,@.-]{0,2048}$`

Required: No

ManifestSummary

The location of the summary manifest. The summary manifest provides aggregate data validation results for the training and test datasets.

Type: GroundTruthManifest object

Required: No

MaxInferenceUnits

The maximum number of inference units Amazon Rekognition uses to auto-scale the model. Applies only to Custom Labels projects. For more information, see StartProjectVersion.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

MinInferenceUnits

The minimum number of inference units used by the model. Applies only to Custom Labels projects. For more information, see StartProjectVersion.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

OutputConfig

The location where training results are saved.

Type: OutputConfig object

Required: No

ProjectVersionArn

The Amazon Resource Name (ARN) of the project version.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: No

SourceProjectVersionArn

If the model version was copied from a different project, `SourceProjectVersionArn` contains the ARN of the source model version.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `(^arn:[a-z\d-]+:rekognition:[a-z\d-]+:\d{12}:project\/[a-zA-Z0-9_.\-]{1,255}\/version\/[a-zA-Z0-9_.\-]{1,255}\/[0-9]+$)`

Required: No

Status

The current status of the model version.

Type: String

Valid Values: `TRAINING_IN_PROGRESS | TRAINING_COMPLETED | TRAINING_FAILED | STARTING | RUNNING | FAILED | STOPPING | STOPPED | DELETING | COPYING_IN_PROGRESS | COPYING_COMPLETED | COPYING_FAILED | DEPRECATED | EXPIRED`

Required: No

StatusMessage

A descriptive message for an error or warning that occurred.

Type: String

Required: No

TestingDataResult

Contains information about the testing results.

Type: TestingDataResult object

Required: No

TrainingDataResult

Contains information about the training results.

Type: TrainingDataResult object

Required: No

TrainingEndTimestamp

The Unix date and time that training of the model ended.

Type: Timestamp

Required: No

VersionDescription

A user-provided description of the project version.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `[a-zA-Z0-9-_. ()':,;?]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProtectiveEquipmentBodyPart

Service: Amazon Rekognition

Information about a body part detected by DetectProtectiveEquipment that contains PPE. An array of `ProtectiveEquipmentBodyPart` objects is returned for each person detected by `DetectProtectiveEquipment`.

Contents

Confidence

The confidence that Amazon Rekognition has in the detection accuracy of the detected body part.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

EquipmentDetections

An array of Personal Protective Equipment items detected around a body part.

Type: Array of EquipmentDetection objects

Required: No

Name

The detected body part.

Type: String

Valid Values: `FACE | HEAD | LEFT_HAND | RIGHT_HAND`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProtectiveEquipmentPerson

Service: Amazon Rekognition

A person detected by a call to DetectProtectiveEquipment. The API returns all persons detected in the input image in an array of `ProtectiveEquipmentPerson` objects.

Contents

BodyParts

An array of body parts detected on a person's body (including body parts without PPE).

Type: Array of ProtectiveEquipmentBodyPart objects

Required: No

BoundingBox

A bounding box around the detected person.

Type: BoundingBox object

Required: No

Confidence

The confidence that Amazon Rekognition has that the bounding box contains a person.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Id

The identifier for the detected person. The identifier is only unique for a single call to `DetectProtectiveEquipment`.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProtectiveEquipmentSummarizationAttributes

Service: Amazon Rekognition

Specifies summary attributes to return from a call to DetectProtectiveEquipment. You can specify which types of PPE to summarize. You can also specify a minimum confidence value for detections. Summary information is returned in the `Summary` (ProtectiveEquipmentSummary) field of the response from `DetectProtectiveEquipment`. The summary includes which persons in an image were detected wearing the requested types of person protective equipment (PPE), which persons were detected as not wearing PPE, and the persons in which a determination could not be made. For more information, see ProtectiveEquipmentSummary.

Contents

MinConfidence

The minimum confidence level for which you want summary information. The confidence level applies to person detection, body part detection, equipment detection, and body part coverage. Amazon Rekognition doesn't return summary information with a confidence than this specified value. There isn't a default value.

Specify a `MinConfidence` value that is between 50-100% as `DetectProtectiveEquipment` returns predictions only where the detection confidence is between 50% - 100%. If you specify a value that is less than 50%, the results are the same specifying a value of 50%.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: Yes

RequiredEquipmentTypes

An array of personal protective equipment types for which you want summary information. If a person is detected wearing a required requipment type, the person's ID is added to the `PersonsWithRequiredEquipment` array field returned in ProtectiveEquipmentSummary by `DetectProtectiveEquipment`.

Type: Array of strings

Valid Values: `FACE_COVER | HAND_COVER | HEAD_COVER`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProtectiveEquipmentSummary

Service: Amazon Rekognition

Summary information for required items of personal protective equipment (PPE) detected on persons by a call to DetectProtectiveEquipment. You specify the required type of PPE in the `SummarizationAttributes` (ProtectiveEquipmentSummarizationAttributes) input parameter. The summary includes which persons were detected wearing the required personal protective equipment (`PersonsWithRequiredEquipment`), which persons were detected as not wearing the required PPE (`PersonsWithoutRequiredEquipment`), and the persons in which a determination could not be made (`PersonsIndeterminate`).

To get a total for each category, use the size of the field array. For example, to find out how many people were detected as wearing the specified PPE, use the size of the `PersonsWithRequiredEquipment` array. If you want to find out more about a person, such as the location (BoundingBox) of the person on the image, use the person ID in each array element. Each person ID matches the ID field of a ProtectiveEquipmentPerson object returned in the `Persons` array by `DetectProtectiveEquipment`.

Contents

PersonsIndeterminate

An array of IDs for persons where it was not possible to determine if they are wearing personal protective equipment.

Type: Array of integers

Valid Range: Minimum value of 0.

Required: No

PersonsWithoutRequiredEquipment

An array of IDs for persons who are not wearing all of the types of PPE specified in the `RequiredEquipmentTypes` field of the detected personal protective equipment.

Type: Array of integers

Valid Range: Minimum value of 0.

Required: No

PersonsWithRequiredEquipment

An array of IDs for persons who are wearing detected personal protective equipment.

Type: Array of integers

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RegionOfInterest

Service: Amazon Rekognition

Specifies a location within the frame that Rekognition checks for objects of interest such as text, labels, or faces. It uses a `BoundingBox` or `Polygon` to set a region of the screen.

A word, face, or label is included in the region if it is more than half in that region. If there is more than one region, the word, face, or label is compared with all regions of the screen. Any object of interest that is more than half in a region is kept in the results.

Contents

BoundingBox

The box representing a region of interest on screen.

Type: BoundingBox object

Required: No

Polygon

Specifies a shape made up of up to 10 `Point` objects to define a region of interest.

Type: Array of Point objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

S3Destination

Service: Amazon Rekognition

The Amazon S3 bucket location to which Amazon Rekognition publishes the detailed inference results of a video analysis operation. These results include the name of the stream processor resource, the session ID of the stream processing session, and labeled timestamps and bounding boxes for detected labels.

Contents

Bucket

The name of the Amazon S3 bucket you want to associate with the streaming video project. You must be the owner of the Amazon S3 bucket.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[0-9A-Za-z\.\-_]*`

Required: No

KeyPrefix

The prefix value of the location within the bucket that you want the information to be published to. For more information, see Using prefixes.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

S3Object

Service: Amazon Rekognition

Provides the S3 bucket name and object name.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Contents

Bucket

Name of the S3 bucket.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[0-9A-Za-z\.\-_]*`

Required: No

Name

S3 object key name.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

Version

If the bucket is versioning enabled, you can specify the object version.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SearchedFace

Service: Amazon Rekognition

Provides face metadata such as FaceId, BoundingBox, Confidence of the input face used for search.

Contents

FaceId

Unique identifier assigned to the face.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SearchedFaceDetails

Service: Amazon Rekognition

Contains data regarding the input face used for a search.

Contents

FaceDetail

Structure containing attributes of the face that the algorithm detected.

A `FaceDetail` object contains either the default facial attributes or all facial attributes. The default attributes are `BoundingBox`, `Confidence`, `Landmarks`, `Pose`, and `Quality`.

GetFaceDetection is the only Amazon Rekognition Video stored video operation that can return a `FaceDetail` object with all attributes. To specify which attributes to return, use the `FaceAttributes` input parameter for StartFaceDetection. The following Amazon Rekognition Video operations return only the default attributes. The corresponding Start operations don't have a `FaceAttributes` input parameter:

- GetCelebrityRecognition
- GetPersonTracking
- GetFaceSearch

The Amazon Rekognition Image DetectFaces and IndexFaces operations can return all facial attributes. To specify which attributes to return, use the `Attributes` input parameter for `DetectFaces`. For `IndexFaces`, use the `DetectAttributes` input parameter.

Type: FaceDetail object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SearchedUser

Service: Amazon Rekognition

Contains metadata about a User searched for within a collection.

Contents

UserId

A provided ID for the UserID. Unique within the collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SegmentDetection

Service: Amazon Rekognition

A technical cue or shot detection segment detected in a video. An array of `SegmentDetection` objects containing all segments detected in a stored video is returned by GetSegmentDetection.

Contents

DurationFrames

The duration of a video segment, expressed in frames.

Type: Long

Valid Range: Minimum value of 0.

Required: No

DurationMillis

The duration of the detected segment in milliseconds.

Type: Long

Valid Range: Minimum value of 0.

Required: No

DurationSMPTE

The duration of the timecode for the detected segment in SMPTE format.

Type: String

Required: No

EndFrameNumber

The frame number at the end of a video segment, using a frame index that starts with 0.

Type: Long

Valid Range: Minimum value of 0.

Required: No

EndTimecodeSMPTE

The frame-accurate SMPTE timecode, from the start of a video, for the end of a detected segment. `EndTimecode` is in *HH:MM:SS:fr* format (and *;fr* for drop frame-rates).

Type: String

Required: No

EndTimestampMillis

The end time of the detected segment, in milliseconds, from the start of the video. This value is rounded down.

Type: Long

Required: No

ShotSegment

If the segment is a shot detection, contains information about the shot detection.

Type: ShotSegment object

Required: No

StartFrameNumber

The frame number of the start of a video segment, using a frame index that starts with 0.

Type: Long

Valid Range: Minimum value of 0.

Required: No

StartTimecodeSMPTE

The frame-accurate SMPTE timecode, from the start of a video, for the start of a detected segment. `StartTimecode` is in *HH:MM:SS:fr* format (and *;fr* for drop frame-rates).

Type: String

Required: No

StartTimestampMillis

The start time of the detected segment in milliseconds from the start of the video. This value is rounded down. For example, if the actual timestamp is 100.6667 milliseconds, Amazon Rekognition Video returns a value of 100 millis.

Type: Long

Required: No

TechnicalCueSegment

If the segment is a technical cue, contains information about the technical cue.

Type: TechnicalCueSegment object

Required: No

Type

The type of the segment. Valid values are `TECHNICAL_CUE` and `SHOT`.

Type: String

Valid Values: `TECHNICAL_CUE | SHOT`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SegmentTypeInfo

Service: Amazon Rekognition

Information about the type of a segment requested in a call to StartSegmentDetection. An array of `SegmentTypeInfo` objects is returned by the response from GetSegmentDetection.

Contents

ModelVersion

The version of the model used to detect segments.

Type: String

Required: No

Type

The type of a segment (technical cue or shot detection).

Type: String

Valid Values: `TECHNICAL_CUE | SHOT`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ShotSegment

Service: Amazon Rekognition

Information about a shot detection segment detected in a video. For more information, see SegmentDetection.

Contents

Confidence

The confidence that Amazon Rekognition Video has in the accuracy of the detected segment.

Type: Float

Valid Range: Minimum value of 50. Maximum value of 100.

Required: No

Index

An Identifier for a shot detection segment detected in a video.

Type: Long

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Smile

Service: Amazon Rekognition

Indicates whether or not the face is smiling, and the confidence level in the determination.

Contents

Confidence

Level of confidence in the determination.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

Boolean value that indicates whether the face is smiling or not.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StartSegmentDetectionFilters

Service: Amazon Rekognition

Filters applied to the technical cue or shot detection segments. For more information, see StartSegmentDetection.

Contents

ShotFilter

Filters that are specific to shot detections.

Type: StartShotDetectionFilter object

Required: No

TechnicalCueFilter

Filters that are specific to technical cues.

Type: StartTechnicalCueDetectionFilter object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StartShotDetectionFilter

Service: Amazon Rekognition

Filters for the shot detection segments returned by `GetSegmentDetection`. For more information, see StartSegmentDetectionFilters.

Contents

MinSegmentConfidence

Specifies the minimum confidence that Amazon Rekognition Video must have in order to return a detected segment. Confidence represents how certain Amazon Rekognition is that a segment is correctly identified. 0 is the lowest confidence. 100 is the highest confidence. Amazon Rekognition Video doesn't return any segments with a confidence level lower than this specified value.

If you don't specify `MinSegmentConfidence`, the `GetSegmentDetection` returns segments with confidence values greater than or equal to 50 percent.

Type: Float

Valid Range: Minimum value of 50. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StartTechnicalCueDetectionFilter

Service: Amazon Rekognition

Filters for the technical segments returned by GetSegmentDetection. For more information, see StartSegmentDetectionFilters.

Contents

BlackFrame

A filter that allows you to control the black frame detection by specifying the black levels and pixel coverage of black pixels in a frame. Videos can come from multiple sources, formats, and time periods, with different standards and varying noise levels for black frames that need to be accounted for.

Type: BlackFrame object

Required: No

MinSegmentConfidence

Specifies the minimum confidence that Amazon Rekognition Video must have in order to return a detected segment. Confidence represents how certain Amazon Rekognition is that a segment is correctly identified. 0 is the lowest confidence. 100 is the highest confidence. Amazon Rekognition Video doesn't return any segments with a confidence level lower than this specified value.

If you don't specify `MinSegmentConfidence`, `GetSegmentDetection` returns segments with confidence values greater than or equal to 50 percent.

Type: Float

Valid Range: Minimum value of 50. Maximum value of 100.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StartTextDetectionFilters

Service: Amazon Rekognition

Set of optional parameters that let you set the criteria text must meet to be included in your response. `WordFilter` looks at a word's height, width and minimum confidence. `RegionOfInterest` lets you set a specific region of the screen to look for text in.

Contents

RegionsOfInterest

Filter focusing on a certain area of the frame. Uses a `BoundingBox` object to set the region of the screen.

Type: Array of RegionOfInterest objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: No

WordFilter

Filters focusing on qualities of the text, such as confidence or size.

Type: DetectionFilter object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessingStartSelector

Service: Amazon Rekognition

This is a required parameter for label detection stream processors and should not be used to start a face search stream processor.

Contents

KVSStreamStartSelector

Specifies the starting point in the stream to start processing. This can be done with a producer timestamp or a fragment number in a Kinesis stream.

Type: KinesisVideoStreamStartSelector object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessingStopSelector

Service: Amazon Rekognition

Specifies when to stop processing the stream. You can specify a maximum amount of time to process the video.

Contents

MaxDurationInSeconds

Specifies the maximum amount of time in seconds that you want the stream to be processed. The largest amount of time is 2 minutes. The default is 10 seconds.

Type: Long

Valid Range: Minimum value of 1. Maximum value of 120.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessor

Service: Amazon Rekognition

An object that recognizes faces or labels in a streaming video. An Amazon Rekognition stream processor is created by a call to CreateStreamProcessor. The request parameters for `CreateStreamProcessor` describe the Kinesis video stream source for the streaming video, face recognition parameters, and where to stream the analysis resullts.

Contents

Name

Name of the Amazon Rekognition stream processor.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-]+`

Required: No

Status

Current status of the Amazon Rekognition stream processor.

Type: String

Valid Values: `STOPPED | STARTING | RUNNING | FAILED | STOPPING | UPDATING`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessorDataSharingPreference

Service: Amazon Rekognition

Allows you to opt in or opt out to share data with Rekognition to improve model performance. You can choose this option at the account level or on a per-stream basis. Note that if you opt out at the account level this setting is ignored on individual streams.

Contents

OptIn

If this option is set to true, you choose to share data with Rekognition to improve model performance.

Type: Boolean

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessorInput

Service: Amazon Rekognition

Information about the source streaming video.

Contents

KinesisVideoStream

The Kinesis video stream input stream for the source streaming video.

Type: KinesisVideoStream object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessorNotificationChannel

Service: Amazon Rekognition

The Amazon Simple Notification Service topic to which Amazon Rekognition publishes the object detection results and completion status of a video analysis operation.

Amazon Rekognition publishes a notification the first time an object of interest or a person is detected in the video stream. For example, if Amazon Rekognition detects a person at second 2, a pet at second 4, and a person again at second 5, Amazon Rekognition sends 2 object class detected notifications, one for a person at second 2 and one for a pet at second 4.

Amazon Rekognition also publishes an an end-of-session notification with a summary when the stream processing session is complete.

Contents

SNSTopicArn

The Amazon Resource Number (ARN) of the Amazon Amazon Simple Notification Service topic to which Amazon Rekognition posts the completion status.

Type: String

Pattern: `(^arn:aws:sns:.*:\w{12}:.+$)`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessorOutput

Service: Amazon Rekognition

Information about the Amazon Kinesis Data Streams stream to which a Amazon Rekognition Video stream processor streams the results of a video analysis. For more information, see CreateStreamProcessor.

Contents

KinesisDataStream

The Amazon Kinesis Data Streams stream to which the Amazon Rekognition stream processor streams the analysis results.

Type: KinesisDataStream object

Required: No

S3Destination

The Amazon S3 bucket location to which Amazon Rekognition publishes the detailed inference results of a video analysis operation.

Type: S3Destination object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessorSettings

Service: Amazon Rekognition

Input parameters used in a streaming video analyzed by a Amazon Rekognition stream processor. You can use `FaceSearch` to recognize faces in a streaming video, or you can use `ConnectedHome` to detect labels.

Contents

ConnectedHome

Label detection settings to use on a streaming video. Defining the settings is required in the request parameter for CreateStreamProcessor. Including this setting in the `CreateStreamProcessor` request enables you to use the stream processor for label detection. You can then select what you want the stream processor to detect, such as people or pets. When the stream processor has started, one notification is sent for each object class specified. For example, if packages and pets are selected, one SNS notification is published the first time a package is detected and one SNS notification is published the first time a pet is detected, as well as an end-of-session summary.

Type: ConnectedHomeSettings object

Required: No

FaceSearch

Face search settings to use on a streaming video.

Type: FaceSearchSettings object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamProcessorSettingsForUpdate

Service: Amazon Rekognition

The stream processor settings that you want to update. `ConnectedHome` settings can be updated to detect different labels with a different minimum confidence.

Contents

ConnectedHomeForUpdate

The label detection settings you want to use for your stream processor.

Type: ConnectedHomeSettingsForUpdate object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Summary

Service: Amazon Rekognition

The S3 bucket that contains the training summary. The training summary includes aggregated evaluation metrics for the entire testing dataset and metrics for each individual label.

You get the training summary S3 bucket location by calling DescribeProjectVersions.

Contents

S3Object

Provides the S3 bucket name and object name.

The region for the S3 bucket containing the S3 object must match the region you use for Amazon Rekognition operations.

For Amazon Rekognition to process an S3 object, the user must have permission to access the S3 object. For more information, see How Amazon Rekognition works with IAM.

Type: S3Object object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Sunglasses

Service: Amazon Rekognition

Indicates whether or not the face is wearing sunglasses, and the confidence level in the determination.

Contents

Confidence

Level of confidence in the determination.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

Value

Boolean value that indicates whether the face is wearing sunglasses or not.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TechnicalCueSegment

Service: Amazon Rekognition

Information about a technical cue segment. For more information, see SegmentDetection.

Contents

Confidence

The confidence that Amazon Rekognition Video has in the accuracy of the detected segment.

Type: Float

Valid Range: Minimum value of 50. Maximum value of 100.

Required: No

Type

The type of the technical cue.

Type: String

Valid Values: `ColorBars | EndCredits | BlackFrames | OpeningCredits | StudioLogo | Slate | Content`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TestingData

Service: Amazon Rekognition

The dataset used for testing. Optionally, if `AutoCreate` is set, Amazon Rekognition uses the training dataset to create a test dataset with a temporary split of the training dataset.

Contents

Assets

The assets used for testing.

Type: Array of Asset objects

Required: No

AutoCreate

If specified, Rekognition splits training dataset to create a test dataset for the training job.

Type: Boolean

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TestingDataResult

Service: Amazon Rekognition

Sagemaker Groundtruth format manifest files for the input, output and validation datasets that are used and created during testing.

Contents

Input

The testing dataset that was supplied for training.

Type: TestingData object

Required: No

Output

The subset of the dataset that was actually tested. Some images (assets) might not be tested due to file formatting and other issues.

Type: TestingData object

Required: No

Validation

The location of the data validation manifest. The data validation manifest is created for the test dataset during model training.

Type: ValidationData object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TextDetection

Service: Amazon Rekognition

Information about a word or line of text detected by DetectText.

The `DetectedText` field contains the text that Amazon Rekognition detected in the image.

Every word and line has an identifier (`Id`). Each word belongs to a line and has a parent identifier (`ParentId`) that identifies the line of text in which the word appears. The word `Id` is also an index for the word within a line of words.

For more information, see Detecting text.

Contents

Confidence

The confidence that Amazon Rekognition has in the accuracy of the detected text and the accuracy of the geometry points around the detected text.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

DetectedText

The word or line of text recognized by Amazon Rekognition.

Type: String

Required: No

Geometry

The location of the detected text on the image. Includes an axis aligned coarse bounding box surrounding the text and a finer grain polygon for more accurate spatial information.

Type: Geometry object

Required: No

Id

The identifier for the detected text. The identifier is only unique for a single call to `DetectText`.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

ParentId

The Parent identifier for the detected text identified by the value of `ID`. If the type of detected text is `LINE`, the value of `ParentId` is `Null`.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

Type

The type of text that was detected.

Type: String

Valid Values: `LINE | WORD`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TextDetectionResult

Service: Amazon Rekognition

Information about text detected in a video. Incudes the detected text, the time in milliseconds from the start of the video that the text was detected, and where it was detected on the screen.

Contents

TextDetection

Details about text detected in a video.

Type: TextDetection object

Required: No

Timestamp

The time, in milliseconds from the start of the video, that the text was detected. Note that `Timestamp` is not guaranteed to be accurate to the individual frame where the text first appears.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TrainingData

Service: Amazon Rekognition

The dataset used for training.

Contents

Assets

A manifest file that contains references to the training images and ground-truth annotations.

Type: Array of Asset objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TrainingDataResult

Service: Amazon Rekognition

The data validation manifest created for the training dataset during model training.

Contents

Input

The training data that you supplied.

Type: TrainingData object

Required: No

Output

Reference to images (assets) that were actually used during training with trained model predictions.

Type: TrainingData object

Required: No

Validation

A manifest that you supplied for training, with validation results for each line.

Type: ValidationData object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UnindexedFace

Service: Amazon Rekognition

A face that IndexFaces detected, but didn't index. Use the `Reasons` response attribute to determine why a face wasn't indexed.

Contents

FaceDetail

The structure that contains attributes of a face that `IndexFacesdetected`, but didn't index.

Type: FaceDetail object

Required: No

Reasons

An array of reasons that specify why a face wasn't indexed.

- EXTREME_POSE - The face is at a pose that can't be detected. For example, the head is turned too far away from the camera.
- EXCEEDS_MAX_FACES - The number of faces detected is already higher than that specified by the `MaxFaces` input parameter for `IndexFaces`.
- LOW_BRIGHTNESS - The image is too dark.
- LOW_SHARPNESS - The image is too blurry.
- LOW_CONFIDENCE - The face was detected with a low confidence.
- SMALL_BOUNDING_BOX - The bounding box around the face is too small.

Type: Array of strings

Valid Values: `EXCEEDS_MAX_FACES | EXTREME_POSE | LOW_BRIGHTNESS | LOW_SHARPNESS | LOW_CONFIDENCE | SMALL_BOUNDING_BOX | LOW_FACE_QUALITY`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UnsearchedFace

Service: Amazon Rekognition

Face details inferred from the image but not used for search. The response attribute contains reasons for why a face wasn't used for Search.

Contents

FaceDetails

Structure containing attributes of the face that the algorithm detected.

A `FaceDetail` object contains either the default facial attributes or all facial attributes. The default attributes are `BoundingBox`, `Confidence`, `Landmarks`, `Pose`, and `Quality`.

GetFaceDetection is the only Amazon Rekognition Video stored video operation that can return a `FaceDetail` object with all attributes. To specify which attributes to return, use the `FaceAttributes` input parameter for StartFaceDetection. The following Amazon Rekognition Video operations return only the default attributes. The corresponding Start operations don't have a `FaceAttributes` input parameter:

- GetCelebrityRecognition
- GetPersonTracking
- GetFaceSearch

The Amazon Rekognition Image DetectFaces and IndexFaces operations can return all facial attributes. To specify which attributes to return, use the `Attributes` input parameter for `DetectFaces`. For `IndexFaces`, use the `DetectAttributes` input parameter.

Type: FaceDetail object

Required: No

Reasons

Reasons why a face wasn't used for Search.

Type: Array of strings

Valid Values: `FACE_NOT_LARGEST | EXCEEDS_MAX_FACES | EXTREME_POSE | LOW_BRIGHTNESS | LOW_SHARPNESS | LOW_CONFIDENCE | SMALL_BOUNDING_BOX | LOW_FACE_QUALITY`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UnsuccessfulFaceAssociation

Service: Amazon Rekognition

Contains metadata like FaceId, UserID, and Reasons, for a face that was unsuccessfully associated.

Contents

Confidence

Match confidence with the UserID, provides information regarding if a face association was unsuccessful because it didn't meet UserMatchThreshold.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

FaceId

A unique identifier assigned to the face.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

Reasons

The reason why the association was unsuccessful.

Type: Array of strings

Valid Values: `FACE_NOT_FOUND | ASSOCIATED_TO_A_DIFFERENT_USER | LOW_MATCH_CONFIDENCE`

Required: No

UserId

A provided ID for the UserID. Unique within the collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UnsuccessfulFaceDeletion

Service: Amazon Rekognition

Contains metadata like FaceId, UserID, and Reasons, for a face that was unsuccessfully deleted.

Contents

FaceId

A unique identifier assigned to the face.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

Reasons

The reason why the deletion was unsuccessful.

Type: Array of strings

Valid Values: `ASSOCIATED_TO_AN_EXISTING_USER | FACE_NOT_FOUND`

Required: No

UserId

A provided ID for the UserID. Unique within the collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UnsuccessfulFaceDisassociation

Service: Amazon Rekognition

Contains metadata like FaceId, UserID, and Reasons, for a face that was unsuccessfully disassociated.

Contents

FaceId

A unique identifier assigned to the face.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

Reasons

The reason why the deletion was unsuccessful.

Type: Array of strings

Valid Values: `FACE_NOT_FOUND | ASSOCIATED_TO_A_DIFFERENT_USER`

Required: No

UserId

A provided ID for the UserID. Unique within the collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

User

Service: Amazon Rekognition

Metadata of the user stored in a collection.

Contents

UserId

A provided ID for the User. Unique within the collection.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Pattern: `[a-zA-Z0-9_.\-:]+`

Required: No

UserStatus

Communicates if the UserID has been updated with latest set of faces to be associated with the UserID.

Type: String

Valid Values: `ACTIVE | UPDATING | CREATING | CREATED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UserMatch

Service: Amazon Rekognition

Provides UserID metadata along with the confidence in the match of this UserID with the input face.

Contents

Similarity

Describes the UserID metadata.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 100.

Required: No

User

Confidence in the match of this UserID with the input face.

Type: MatchedUser object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ValidationData

Service: Amazon Rekognition

Contains the Amazon S3 bucket location of the validation data for a model training job.

The validation data includes error information for individual JSON lines in the dataset. For more information, see Debugging a Failed Model Training.

You get the `ValidationData` object for the training dataset (TrainingDataResult) and the test dataset (TestingDataResult) by calling DescribeProjectVersions.

The assets array contains a single Asset object. The GroundTruthManifest field of the Asset object contains the S3 bucket location of the validation data.

Contents

Assets

The assets that comprise the validation data.

Type: Array of Asset objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Versions

Service: Amazon Rekognition

Object specifying the acceptable range of challenge versions.

Contents

Maximum

The desired maximum version for the challenge.

Type: String

Length Constraints: Minimum length of 5. Maximum length of 11.

Pattern: `^(0|[1-9]\d{0,2})\.(0|[1-9]\d{0,2})\.(0|[1-9]\d{0,2})$`

Required: No

Minimum

The desired minimum version for the challenge.

Type: String

Length Constraints: Minimum length of 5. Maximum length of 11.

Pattern: `^(0|[1-9]\d{0,2})\.(0|[1-9]\d{0,2})\.(0|[1-9]\d{0,2})$`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Video

Service: Amazon Rekognition

Video file stored in an Amazon S3 bucket. Amazon Rekognition video start operations such as StartLabelDetection use `Video` to specify a video for analysis. The supported file formats are .mp4, .mov and .avi.

Contents

S3Object

The Amazon S3 bucket name and file name for the video.

Type: S3Object object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

VideoMetadata

Service: Amazon Rekognition

Information about a video that Amazon Rekognition analyzed. `Videometadata` is returned in every page of paginated responses from a Amazon Rekognition video operation.

Contents

Codec

Type of compression used in the analyzed video.

Type: String

Required: No

ColorRange

A description of the range of luminance values in a video, either LIMITED (16 to 235) or FULL (0 to 255).

Type: String

Valid Values: `FULL | LIMITED`

Required: No

DurationMillis

Length of the video in milliseconds.

Type: Long

Valid Range: Minimum value of 0.

Required: No

Format

Format of the analyzed video. Possible values are MP4, MOV and AVI.

Type: String

Required: No

FrameHeight

Vertical pixel dimension of the video.

Type: Long

Valid Range: Minimum value of 0.

Required: No

FrameRate

Number of frames per second in the video.

Type: Float

Required: No

FrameWidth

Horizontal pixel dimension of the video.

Type: Long

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon Rekognition Streaming

The following data types are supported by Amazon Rekognition Streaming:

- BoundingBox

- ChallengeConfig
- ClientChallenge
- ClientSessionInformationEvent
- ColorDisplayed
- ColorSequence
- DisconnectionEvent
- FaceMovementAndLightClientChallenge
- FaceMovementAndLightServerChallenge
- FreshnessColor
- InitialFace
- LivenessRequestStream
- LivenessResponseStream
- OvalParameters
- ServerChallenge
- ServerSessionInformationEvent
- SessionInformation
- TargetFace
- VideoEvent

BoundingBox

Service: Amazon Rekognition Streaming

Identifies the bounding box around a given label, face, text, object of interest, or personal protective equipment. The left (x-coordinate) and top (y-coordinate) are coordinates representing the top and left sides of the bounding box. Note that the upper-left corner of the image is the origin (0,0).

The top and left values returned are ratios of the overall image size. For example, if the input image is 700x200 pixels, and the top-left coordinate of the bounding box is 350x50 pixels, the API returns a left value of 0.5 (350/700) and a top value of 0.25 (50/200).

The width and height values represent the dimensions of the bounding box as a ratio of the overall image dimension. For example, if the input image is 700x200 pixels, and the bounding box width is 70 pixels, the width returned is 0.1.

Contents

Height

Height of the bounding box as a ratio of the overall image height.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: Yes

Left

Left coordinate of the bounding box as a ratio of overall image width.

Type: Float

Valid Range: Minimum value of -1. Maximum value of 1.

Required: Yes

Top

Top coordinate of the bounding box as a ratio of overall image height.

Type: Float

Valid Range: Minimum value of -1. Maximum value of 1.

Required: Yes

Width

Width of the bounding box as a ratio of the overall image width.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ChallengeConfig

Service: Amazon Rekognition Streaming

Configuration options for Face Liveness challenges performed at the client side.

Contents

BlazeFaceDetectionThreshold

Threshold for face detection done using blaze face.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

FaceDistanceThreshold

Threshold for face distance threshold face liveness challenge.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

FaceDistanceThresholdMax

Threshold for the maximum distance that face can be from the camera before face oval match challenge starts.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

FaceDistanceThresholdMin

Threshold for the minimum distance that face can be from the camera before face oval match challenge starts.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

FaceIouHeightThreshold

Threshold for height during face IOU (Intersection over Union) match.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

FaceIouWidthThreshold

Threshold for width during face IOU (Intersection over Union) match.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

OvalFitTimeout

Timeout limit in which the end-users need to fit in the oval, in miliseconds.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 10000.

Required: No

OvalHeightWidthRatio

Height to width ratio of oval used for face oval match.

Type: Float

Required: No

OvalIouHeightThreshold

Threshold for height during oval IOU (Intersection over Union) match.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

OvalIouThreshold

Threshold for face oval match using IoU (Intersection over Union).

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

OvalIouWidthThreshold

Threshold for width during oval IOU (Intersection over Union) match.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClientChallenge

Service: Amazon Rekognition Streaming

Object containing information for Face Liveness challenges performed at the client side.

Contents

> **Important**
>
> This data type is a UNION, so only one of the following members can be specified when used or returned.

FaceMovementAndLightChallenge

Information about a face movement and light challenge type.

Type: FaceMovementAndLightClientChallenge object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClientSessionInformationEvent

Service: Amazon Rekognition Streaming

Any information that the client needs to send for the streaming session. For face movement challenge, it will contain information like initial face position and target face position.

Contents

Challenge

Contains information on FaceMovementAndLightChellenge, TargetFace, and ColorDisplayed, for a given Challenge.

Type: ClientChallenge object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ColorDisplayed

Service: Amazon Rekognition Streaming

Contains information regarding `SequenceNumber`, `CurrentColor`, `PreviousColor`, and `CurrentColorStartTimestamp`.

Contents

CurrentColor

The current color being displayed on the screen (flat or downscrolled). It will contain a separate RGB list having 3 elements (each element can have values between 0 to 255).

Type: FreshnessColor object

Required: Yes

CurrentColorStartTimestamp

Epoch timestamp of when the current color was displayed on the user screen.

Type: Long

Valid Range: Minimum value of 1640995200000. Maximum value of 2272147200000.

Required: Yes

SequenceNumber

The order in which the current color was displayed on the screen.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 50.

Required: Yes

PreviousColor

The previous color displayed on screen (flat or downscrolled). It will contain a separate RGB list having 3 elements (each element can have values between 0 to 255).

Type: FreshnessColor object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ColorSequence

Service: Amazon Rekognition Streaming

A color sequence to be displayed on the user's screen.

Contents

DownscrollDuration

Duration in milliseconds for which a given color in the color sequence will down-scroll before taking over full screen.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1000.

Required: Yes

FlatDisplayDuration

Duration in milliseconds for which a given flat color in the color sequence will be displayed on the full screen.

Type: Float

Valid Range: Minimum value of 0. Maximum value of 1000.

Required: Yes

FreshnessColor

Represents the colors in a given ColorSequence to be flashed to the end user, with each color represented in RGB values.

Type: FreshnessColor object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DisconnectionEvent

Service: Amazon Rekognition Streaming

Contains information regarding if a disconnection event took place during the Face Liveness session. Disconnection events may indicate processing is complete at the server side.

Contents

TimestampMillis

The timestamp that a disconnection event took place at.

Type: Long

Valid Range: Minimum value of 1640995200000. Maximum value of 2272147200000.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FaceMovementAndLightClientChallenge

Service: Amazon Rekognition Streaming

Holds information about the type of face movement and light challenges that need to be performed.

Contents

ChallengeId

An ID that uniquely identifies a challenge from a client.

Type: String

Length Constraints: Fixed length of 36.

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: Yes

ColorDisplayed

Contains information regarding SequenceNumber, CurrentColor, PreviousColor, and CurrentColorStartTimestamp for a challenge.

Type: ColorDisplayed object

Required: No

InitialFace

Contains the bounding box of the initial face position for the user on the device screen. Also contains an epoch timestamp of when the user was detected in this position. Used for Face Liveness detection.

Type: InitialFace object

Required: No

TargetFace

Contains the face position bounding box for the user on the device's screen, set at the target location constructed for the challenge. Generated using a precise oval location depending on the provided video

dimensions in the input stream. Contains start and end epoch timestamp of when the user was detected in this position.

Type: TargetFace object

Required: No

VideoEndTimestamp

Epoch timestamp indicating end of video recording.

Type: Long

Valid Range: Minimum value of 1640995200000. Maximum value of 2272147200000.

Required: No

VideoStartTimestamp

Epoch timestamp indicating start of video recording.

Type: Long

Valid Range: Minimum value of 1640995200000. Maximum value of 2272147200000.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FaceMovementAndLightServerChallenge

Service: Amazon Rekognition Streaming

Contains information regarding the `OvalParameters` and `LightChallengeType` for a challenge.

Contents

ChallengeConfig

Configurations for attributes of the Face Liveness movement and light challenges.

Type: ChallengeConfig object

Required: Yes

ColorSequences

Used to generate a list of color sequences to be displayed on a user's screen.

Type: Array of ColorSequence objects

Required: Yes

LightChallengeType

Information on the type of colored light challenge.

Type: String

Valid Values: `SEQUENTIAL`

Required: Yes

OvalParameters

The parameters needed for an oval to display and to complete oval match challenge.

Type: OvalParameters object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FreshnessColor

Service: Amazon Rekognition Streaming

Represents the colors to be flashed to the end user, each color represented in RGB values.

Contents

RGB

RGB values for a color to be flashed to the end user.

Type: Array of integers

Array Members: Fixed number of 3 items.

Valid Range: Minimum value of 0. Maximum value of 255.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

InitialFace

Service: Amazon Rekognition Streaming

Contains bounding box of initial face position of the user on the device screen. Contains an epoch timestamp of when the user was detected in this position. Used for Face Liveness detection.

Contents

BoundingBox

The bounding box for the initial face position of the user on the device screen.

Type: BoundingBox object

Required: Yes

InitialFaceDetectedTimestamp

Timestamp that a face was initially detected at.

Type: Long

Valid Range: Minimum value of 1640995200000. Maximum value of 2272147200000.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LivenessRequestStream

Service: Amazon Rekognition Streaming

Contains information regarding video events and client session details for a Face Liveness verification request stream.

Contents

ClientSessionInformationEvent

Information event for the client session for a given Face Liveness request stream.

Type: ClientSessionInformationEvent object

Required: No

VideoEvent

Contains video events for a Face Liveness request stream.

Type: VideoEvent object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LivenessResponseStream

Service: Amazon Rekognition Streaming

Contains information event details and exception information for a Face Liveness verification response stream.

Contents

DisconnectionEvent

Notification that disconnection event has occurred to a Face Liveness session.

Type: DisconnectionEvent object

Required: No

InternalServerException

Notification that an InternalServerException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 500

Required: No

ServerSessionInformationEvent

Details for a server session information event.

Type: ServerSessionInformationEvent object

Required: No

ServiceQuotaExceededException

Notification that a ServiceQuotaExceededException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 429

Required: No

ServiceUnavailableException

Notification that a ServiceUnavailableException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 503

Required: No

ThrottlingException

Notification that a ThrottlingException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 429

Required: No

ValidationException

Notification that a ValidationException occurred during a Face Liveness session.

Type: Exception
HTTP Status Code: 400

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

OvalParameters

Service: Amazon Rekognition Streaming

Oval parameters need for oval display to complete oval match challenge.

Contents

CenterX

X-axis co-ordinate of for center of oval

Type: Float

Valid Range: Minimum value of 0. Maximum value of 9999.

Required: Yes

CenterY

Y-axis co-ordinate of for center of oval

Type: Float

Valid Range: Minimum value of 0. Maximum value of 9999.

Required: Yes

Height

Height of the oval

Type: Float

Valid Range: Minimum value of 0. Maximum value of 9999.

Required: Yes

Width

Width of the oval

Type: Float

Valid Range: Minimum value of 0. Maximum value of 9999.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServerChallenge

Service: Amazon Rekognition Streaming

Information on the challenge sent by the server.

Contents

> **Important**
>
> This data type is a UNION, so only one of the following members can be specified when used or returned.

FaceMovementAndLightChallenge

Paramteters and configuration information for the face movement and light sequence challenges.

Type: FaceMovementAndLightServerChallenge object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ServerSessionInformationEvent

Service: Amazon Rekognition Streaming

Event containing session information for the server.

Contents

SessionInformation

Contains the session information sent by the server..

Type: SessionInformation object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SessionInformation

Service: Amazon Rekognition Streaming

Information required for the streaming session.

Contents

Challenge

Object containing information of Face Liveness challenge to be performed by the client. The information is used to construct the challenges at the client's SDK.

Type: ServerChallenge object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TargetFace

Service: Amazon Rekognition Streaming

Contains bounding box of face position of the user on the device screen at target location constructed for the challenge. This is generated using the random offsets provided by the server to the client at session start. Also contains start and end epoch timestamp of when the user was detected in this position.

Contents

BoundingBox

A bounding box for the target face.

Type: BoundingBox object

Required: Yes

FaceDetectedInTargetPositionEndTimestamp

Ending timestamp at which a face was detected in the target position.

Type: Long

Valid Range: Minimum value of 1640995200000. Maximum value of 2272147200000.

Required: Yes

FaceDetectedInTargetPositionStartTimestamp

Starting timestamp at which a face was detected in the target position.

Type: Long

Valid Range: Minimum value of 1640995200000. Maximum value of 2272147200000.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

VideoEvent

Service: Amazon Rekognition Streaming

Contains the video chunk blob.

Contents

TimestampMillis

Milisecond timestamp for a video chunk in a video event.

Type: Long

Valid Range: Minimum value of 1640995200000. Maximum value of 2272147200000.

Required: No

VideoChunk

The video chunk for a video event.

Type: Base64-encoded binary data object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signing AWS API requests in the *IAM User Guide*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Elements of an AWS API request signature in the *IAM User Guide*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS STS, see AWS services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Create a signed AWS API request in the *IAM User Guide*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400