aws

Amazon IVS

API Version 2020-07-14

Amazon IVS: Chat API Reference

Table of Contents

Welcome

Introduction

The Amazon IVS Chat control-plane API enables you to create and manage Amazon IVS Chat resources. You also need to integrate with the Amazon IVS Chat Messaging API, to enable users to interact with chat rooms in real time.

The API is an AWS regional service. For a list of supported regions and Amazon IVS Chat HTTPS service endpoints, see the Amazon IVS Chat information on the Amazon IVS page in the *AWS General Reference*.

This document describes HTTP operations. There is a separate *messaging* API for managing Chat resources; see the Amazon IVS Chat Messaging API Reference.

Notes on terminology:

- You create service applications using the Amazon IVS Chat API. We refer to these as *applications*.
- You create front-end client applications (browser and Android/iOS apps) using the Amazon IVS Chat Messaging API. We refer to these as *clients*.

Resources

The following resources are part of Amazon IVS Chat:

- **LoggingConfiguration** — A configuration that allows customers to store and record sent messages in a chat room. See the Logging Configuration endpoints for more information.
- **Room** — The central Amazon IVS Chat resource through which clients connect to and exchange chat messages. See the Room endpoints for more information.

Tagging

A *tag* is a metadata label that you assign to an AWS resource. A tag comprises a *key* and a *value*, both set by you. For example, you might set a tag as `topic:nature` to label a particular video category. See Best practices and strategies in *Tagging AWS Resources and Tag Editor* for details, including restrictions that apply to tags and "Tag naming limits and requirements"; Amazon IVS Chat has no service-specific constraints beyond what is documented there.

Tags can help you identify and organize your AWS resources. For example, you can use the same tag for different resources to indicate that they are related. You can also use tags to manage access (see Access Tags).

The Amazon IVS Chat API has these tag-related operations: TagResource, UntagResource, and ListTagsForResource. The following resource supports tagging: Room.

At most 50 tags can be applied to a resource.

API Access Security

Your Amazon IVS Chat applications (service applications and clients) must be authenticated and authorized to access Amazon IVS Chat resources. Note the differences between these concepts:

- *Authentication* is about verifying identity. Requests to the Amazon IVS Chat API must be signed to verify your identity.
- *Authorization* is about granting permissions. Your IAM roles need to have permissions for Amazon IVS Chat API requests.

Users (viewers) connect to a room using secure access tokens that you create using the CreateChatToken operation through the AWS SDK. You call CreateChatToken for every user's chat session, passing identity and authorization information about the user.

Signing API Requests

HTTP API requests must be signed with an AWS SigV4 signature using your AWS security credentials. The AWS Command Line Interface (CLI) and the AWS SDKs take care of signing the underlying API calls for you. However, if your application calls the Amazon IVS Chat HTTP API directly, it's your responsibility to sign the requests.

You generate a signature using valid AWS credentials for an IAM role that has permission to perform the requested action. For example, DeleteMessage requests must be made using an IAM role that has the `ivschat:DeleteMessage` permission.

For more information:

- Authentication and generating signatures — See Authenticating Requests (AWS Signature Version 4) in the *AWS General Reference*.
- Managing Amazon IVS permissions — See Identity and Access Management on the Security page of the *Amazon IVS User Guide*.

Amazon Resource Names (ARNs)

ARNs uniquely identify AWS resources. An ARN is required when you need to specify a resource unambiguously across all of AWS, such as in IAM policies and API calls. For more information, see Amazon Resource Names in the *AWS General Reference*.

This document was last published on March 24, 2026.

Actions

The following actions are supported:

- CreateChatToken
- CreateLoggingConfiguration
- CreateRoom
- DeleteLoggingConfiguration
- DeleteMessage
- DeleteRoom
- DisconnectUser
- GetLoggingConfiguration
- GetRoom
- ListLoggingConfigurations
- ListRooms
- ListTagsForResource
- SendEvent
- TagResource
- UntagResource
- UpdateLoggingConfiguration
- UpdateRoom

CreateChatToken

Creates an encrypted token that is used by a chat participant to establish an individual WebSocket chat connection to a room. When the token is used to connect to chat, the connection is valid for the session duration specified in the request. The token becomes invalid at the token-expiration timestamp included in the response.

Use the `capabilities` field to permit an end user to send messages or moderate a room.

The `attributes` field securely attaches structured data to the chat session; the data is included within each message sent by the end user and received by other participants in the room. Common use cases for attributes include passing end-user profile data like an icon, display name, colors, badges, and other display features.

Encryption keys are owned by Amazon IVS Chat and never used directly by your application.

Request Syntax

```
POST /CreateChatToken HTTP/1.1
Content-type: application/json

{
   "attributes": {
      "string" : "string"
   },
   "capabilities": [ "string" ],
   "roomIdentifier": "string",
   "sessionDurationInMinutes": number,
   "userId": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

attributes

Application-provided attributes to encode into the token and attach to a chat session. Map keys and values can contain UTF-8 encoded text. The maximum length of this field is 1 KB total.

Type: String to string map

Required: No

capabilities

Set of capabilities that the user is allowed to perform in the room. Default: None (the capability to view messages is implicitly included in all requests).

Type: Array of strings

Valid Values: `SEND_MESSAGE | DISCONNECT_USER | DELETE_MESSAGE`

Required: No

roomIdentifier

Identifier of the room that the client is trying to access. Currently this must be an ARN.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

Required: Yes

sessionDurationInMinutes

Session duration (in minutes), after which the session expires. Default: 60 (1 hour).

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 180.

Required: No

userId

Application-provided ID that uniquely identifies the user associated with this token. This can be any UTF-8 encoded text.

Type: String

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

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "sessionExpirationTime": "string",
   "token": "string",
   "tokenExpirationTime": "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.

sessionExpirationTime

Time after which an end user's session is no longer valid. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

token

The issued client token, encrypted.

Type: String

tokenExpirationTime

Time after which the token is no longer valid and cannot be used to connect to a room. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

Errors

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

AccessDeniedException

HTTP Status Code: 403

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2
- AWS SDK for .NET V4
- 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

CreateLoggingConfiguration

Creates a logging configuration that allows clients to store and record sent messages.

Request Syntax

```
POST /CreateLoggingConfiguration HTTP/1.1
Content-type: application/json

{
   "destinationConfiguration": { ... },
   "name": "string",
   "tags": {
      "string" : "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

destinationConfiguration

> A complex type that contains a destination configuration for where chat content will be logged. There can be only one type of destination (`cloudWatchLogs`, `firehose`, or `s3`) in a `destinationConfiguration`.
>
> Type: DestinationConfiguration object
>
> **Note:** This object is a Union. Only one member of this object can be specified or returned.
>
> Required: Yes

name

> Logging-configuration name. The value does not need to be unique.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

Required: No

tags

Tags to attach to the resource. Array of maps, each of the form `string:string`
(`key:value`). See [Best practices and strategies](#) in *Tagging AWS Resources and Tag Editor* for
details, including restrictions that apply to tags and "Tag naming limits and requirements";
Amazon IVS Chat has no constraints on tags beyond what is documented there.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createTime": "string",
   "destinationConfiguration": { ... },
   "id": "string",
   "name": "string",
   "state": "string",
   "tags": {
      "string" : "string"
   },
   "updateTime": "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.

arn

Logging-configuration ARN, assigned by the system.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

createTime

Time when the logging configuration was created. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

destinationConfiguration

A complex type that contains a destination configuration for where chat content will be logged, from the request. There is only one type of destination (`cloudWatchLogs`, `firehose`, or `s3`) in a `destinationConfiguration`.

Type: DestinationConfiguration object

Note: This object is a Union. Only one member of this object can be specified or returned.

id

Logging-configuration ID, generated by the system. This is a relative identifier, the part of the ARN that uniquely identifies the logging configuration.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

name

Logging-configuration name, from the request (if specified).

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

state

The state of the logging configuration. When the state is `ACTIVE`, the configuration is ready to log chat content.

Type: String

Valid Values: `ACTIVE`

tags

Tags attached to the resource, from the request (if specified). Array of maps, each of the form `string:string (key:value)`.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

updateTime

Time of the logging configuration's last update. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

Errors

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

AccessDeniedException

HTTP Status Code: 403

ConflictException

HTTP Status Code: 409

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ServiceQuotaExceededException

HTTP Status Code: 402

ValidationException

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 V2](#)
- [AWS SDK for .NET V4](#)
- [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](#)

CreateRoom

Creates a room that allows clients to connect and pass messages.

Request Syntax

```
POST /CreateRoom HTTP/1.1
Content-type: application/json

{
   "loggingConfigurationIdentifiers": [ "string" ],
   "maximumMessageLength": number,
   "maximumMessageRatePerSecond": number,
   "messageReviewHandler": {
      "fallbackResult": "string",
      "uri": "string"
   },
   "name": "string",
   "tags": {
      "string" : "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

loggingConfigurationIdentifiers

Array of logging-configuration identifiers attached to the room.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 3 items.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

Required: No

maximumMessageLength

Maximum number of characters in a single message. Messages are expected to be UTF-8 encoded and this limit applies specifically to rune/code-point count, not number of bytes. Default: 500.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

Required: No

maximumMessageRatePerSecond

Maximum number of messages per second that can be sent to the room (by all clients). Default: 10.

Type: Integer

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

Required: No

messageReviewHandler

Configuration information for optional review of messages.

Type: MessageReviewHandler object

Required: No

name

Room name. The value does not need to be unique.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

Required: No

tags

Tags to attach to the resource. Array of maps, each of the form `string:string` (`key:value`). See [Best practices and strategies](#) in *Tagging AWS Resources and Tag Editor* for details, including restrictions that apply to tags and "Tag naming limits and requirements"; Amazon IVS Chat has no constraints beyond what is documented there.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createTime": "string",
   "id": "string",
   "loggingConfigurationIdentifiers": [ "string" ],
   "maximumMessageLength": number,
   "maximumMessageRatePerSecond": number,
   "messageReviewHandler": {
      "fallbackResult": "string",
      "uri": "string"
   },
   "name": "string",
   "tags": {
      "string" : "string"
   },
   "updateTime": "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.

arn

Room ARN, assigned by the system.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

createTime

Time when the room was created. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

id

Room ID, generated by the system. This is a relative identifier, the part of the ARN that uniquely identifies the room.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

loggingConfigurationIdentifiers

Array of logging configurations attached to the room, from the request (if specified).

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 3 items.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

maximumMessageLength

Maximum number of characters in a single message, from the request (if specified).

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

maximumMessageRatePerSecond

Maximum number of messages per second that can be sent to the room (by all clients), from the request (if specified).

Type: Integer

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

messageReviewHandler

Configuration information for optional review of messages.

Type: MessageReviewHandler object

name

Room name, from the request (if specified).

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

tags

Tags attached to the resource, from the request (if specified).

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

updateTime

Time of the room's last update. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

Errors

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

AccessDeniedException

HTTP Status Code: 403

ConflictException

HTTP Status Code: 409

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ServiceQuotaExceededException

HTTP Status Code: 402

ValidationException

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 V2](#)

- [AWS SDK for .NET V4](#)

- [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](#)

DeleteLoggingConfiguration

Deletes the specified logging configuration.

Request Syntax

```
POST /DeleteLoggingConfiguration HTTP/1.1
Content-type: application/json

{
    "identifier": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

identifier

Identifier of the logging configuration to be deleted.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

Required: Yes

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Errors

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

AccessDeniedException

HTTP Status Code: 403

ConflictException

HTTP Status Code: 409

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2
- AWS SDK for .NET V4
- 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](#)

DeleteMessage

Sends an event to a specific room which directs clients to delete a specific message; that is, unrender it from view and delete it from the client's chat history. This event's `EventName` is `aws:DELETE_MESSAGE`. This replicates the DeleteMessage WebSocket operation in the Amazon IVS Chat Messaging API.

Request Syntax

```
POST /DeleteMessage HTTP/1.1
Content-type: application/json

{
   "id": "string",
   "reason": "string",
   "roomIdentifier": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

id

ID of the message to be deleted. This is the `Id` field in the received message (see Message (Subscribe) in the Chat Messaging API).

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

Required: Yes

reason

Reason for deleting the message.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

roomIdentifier

Identifier of the room where the message should be deleted. Currently this must be an ARN.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "id": "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.

id

Operation identifier, generated by Amazon IVS Chat.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

Errors

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

AccessDeniedException

HTTP Status Code: 403

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ThrottlingException

HTTP Status Code: 429

ValidationException

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 V2
- AWS SDK for .NET V4
- 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](#)

DeleteRoom

Deletes the specified room.

Request Syntax

```
POST /DeleteRoom HTTP/1.1
Content-type: application/json

{
   "identifier": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

identifier

Identifier of the room to be deleted. Currently this must be an ARN.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

Required: Yes

Response Syntax

```
HTTP/1.1 204
```

Response Elements

If the action is successful, the service sends back an HTTP 204 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

HTTP Status Code: 403

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2](#)
- [AWS SDK for .NET V4](#)
- [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](#)

DisconnectUser

Disconnects all connections using a specified user ID from a room. This replicates the
DisconnectUser WebSocket operation in the Amazon IVS Chat Messaging API.

Request Syntax

```
POST /DisconnectUser HTTP/1.1
Content-type: application/json

{
   "reason": "string",
   "roomIdentifier": "string",
   "userId": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

reason

Reason for disconnecting the user.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

roomIdentifier

Identifier of the room from which the user's clients should be disconnected. Currently this must
be an ARN.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

Required: Yes

userId

ID of the user (connection) to disconnect from the room.

Type: String

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

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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

HTTP Status Code: 403

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ThrottlingException

HTTP Status Code: 429

ValidationException

    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 V2](#)
- [AWS SDK for .NET V4](#)
- [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](#)

GetLoggingConfiguration

Gets the specified logging configuration.

Request Syntax

```
POST /GetLoggingConfiguration HTTP/1.1
Content-type: application/json

{
    "identifier": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

identifier

Identifier of the logging configuration to be retrieved.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
```

```
    "arn": "string",
    "createTime": "string",
    "destinationConfiguration": { ... },
    "id": "string",
    "name": "string",
    "state": "string",
    "tags": {
       "string" : "string"
    },
    "updateTime": "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.

arn

Logging-configuration ARN, from the request (if `identifier` was an ARN).

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

createTime

Time when the logging configuration was created. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

destinationConfiguration

A complex type that contains a destination configuration for where chat content will be logged. There is only one type of destination (`cloudWatchLogs`, `firehose`, or `s3`) in a `destinationConfiguration`.

Type: DestinationConfiguration object

Note: This object is a Union. Only one member of this object can be specified or returned.

id

Logging-configuration ID, generated by the system. This is a relative identifier, the part of the ARN that uniquely identifies the logging configuration.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

name

Logging-configuration name. This value does not need to be unique.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

state

The state of the logging configuration. When the state is `ACTIVE`, the configuration is ready to log chat content.

Type: String

Valid Values: `CREATING | CREATE_FAILED | DELETING | DELETE_FAILED | UPDATING | UPDATE_FAILED | ACTIVE`

tags

Tags attached to the resource. Array of maps, each of the form `string:string` (`key:value`).

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

updateTime

> Time of the logging configuration's last update. This is an ISO 8601 timestamp; *note that this is returned as a string.*
>
> Type: Timestamp

Errors

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

AccessDeniedException

> HTTP Status Code: 403

ResourceNotFoundException

> HTTP Status Code: 404

ValidationException

> 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 V2
- AWS SDK for .NET V4
- 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](#)

GetRoom

Gets the specified room.

Request Syntax

```
POST /GetRoom HTTP/1.1
Content-type: application/json

{
    "identifier": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

identifier

Identifier of the room for which the configuration is to be retrieved. Currently this must be an ARN.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
```

```
      "arn": "string",
      "createTime": "string",
      "id": "string",
      "loggingConfigurationIdentifiers": [ "string" ],
      "maximumMessageLength": number,
      "maximumMessageRatePerSecond": number,
      "messageReviewHandler": {
         "fallbackResult": "string",
         "uri": "string"
      },
      "name": "string",
      "tags": {
         "string" : "string"
      },
      "updateTime": "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.

arn

Room ARN, from the request (if `identifier` was an ARN).

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

createTime

Time when the room was created. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

id

Room ID, generated by the system. This is a relative identifier, the part of the ARN that uniquely identifies the room.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

loggingConfigurationIdentifiers

Array of logging configurations attached to the room.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 3 items.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

maximumMessageLength

Maximum number of characters in a single message. Messages are expected to be UTF-8 encoded and this limit applies specifically to rune/code-point count, not number of bytes. Default: 500.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

maximumMessageRatePerSecond

Maximum number of messages per second that can be sent to the room (by all clients). Default: 10.

Type: Integer

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

messageReviewHandler

Configuration information for optional review of messages.

Type: MessageReviewHandler object

name

Room name. The value does not need to be unique.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

tags

Tags attached to the resource. Array of maps, each of the form `string:string`
`(key:value)`.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

updateTime

Time of the room's last update. This is an ISO 8601 timestamp; *note that this is returned as a
string*.

Type: Timestamp

Errors

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

AccessDeniedException

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2](#)
- [AWS SDK for .NET V4](#)
- [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](#)

ListLoggingConfigurations

Gets summary information about all your logging configurations in the AWS region where the API request is processed.

Request Syntax

```
POST /ListLoggingConfigurations HTTP/1.1
Content-type: application/json

{
    "maxResults": number,
    "nextToken": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

maxResults

Maximum number of logging configurations to return. Default: 50.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 50.

Required: No

nextToken

The first logging configurations to retrieve. This is used for pagination; see the `nextToken` response field.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "loggingConfigurations": [
      {
         "arn": "string",
         "createTime": "string",
         "destinationConfiguration": { ... },
         "id": "string",
         "name": "string",
         "state": "string",
         "tags": {
            "string" : "string"
         },
         "updateTime": "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.

loggingConfigurations

List of the matching logging configurations (summary information only). There is only one type of destination (`cloudWatchLogs`, `firehose`, or `s3`) in a `destinationConfiguration`.

Type: Array of LoggingConfigurationSummary objects

nextToken

If there are more logging configurations than `maxResults`, use `nextToken` in the request to get the next set.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Errors

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

AccessDeniedException

HTTP Status Code: 403

ValidationException

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 V2
- [AWS SDK for .NET V4](AWS SDK for .NET V4)
- [AWS SDK for C++](AWS SDK for C++)
- [AWS SDK for Go v2](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 PHP V3)
- AWS SDK for Python
- AWS SDK for Ruby V3

ListRooms

Gets summary information about all your rooms in the AWS region where the API request is processed. Results are sorted in descending order of `updateTime`.

Request Syntax

```
POST /ListRooms HTTP/1.1
Content-type: application/json

{
    "loggingConfigurationIdentifier": "string",
    "maxResults": number,
    "messageReviewHandlerUri": "string",
    "name": "string",
    "nextToken": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

loggingConfigurationIdentifier

Logging-configuration identifier.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

Required: No

maxResults

Maximum number of rooms to return. Default: 50.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 50.

Required: No

messageReviewHandlerUri

Filters the list to match the specified message review handler URI.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 170.

Pattern: `$|^arn:aws:lambda:[a-z0-9-]+:[0-9]{12}:function:.+.*`

Required: No

name

Filters the list to match the specified room name.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

Required: No

nextToken

The first room to retrieve. This is used for pagination; see the `nextToken` response field.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
```

```
      "nextToken": "string",
      "rooms": [
         {
            "arn": "string",
            "createTime": "string",
            "id": "string",
            "loggingConfigurationIdentifiers": [ "string" ],
            "messageReviewHandler": {
               "fallbackResult": "string",
               "uri": "string"
            },
            "name": "string",
            "tags": {
               "string" : "string"
            },
            "updateTime": "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 there are more rooms than `maxResults`, use `nextToken` in the request to get the next set.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1024.

rooms

List of the matching rooms (summary information only).

Type: Array of RoomSummary objects

Errors

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

AccessDeniedException

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2](#)
- [AWS SDK for .NET V4](#)
- [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

Gets information about AWS tags for the specified ARN.

Request Syntax

```
GET /tags/resourceArn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

resourceArn

The ARN of the resource to be retrieved. The ARN must be URL-encoded.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:[a-z-]/[a-zA-Z0-9-]+`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "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

Tags attached to the resource. Array of maps, each of the form `string:string` (`key:value`).

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

Errors

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

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2
- AWS SDK for .NET V4
- 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](#)

SendEvent

Sends an event to a room. Use this within your application's business logic to send events to clients of a room; e.g., to notify clients to change the way the chat UI is rendered.

Request Syntax

```
POST /SendEvent HTTP/1.1
Content-type: application/json

{
   "attributes": {
      "string" : "string"
   },
   "eventName": "string",
   "roomIdentifier": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

attributes

   Application-defined metadata to attach to the event sent to clients. The maximum length of the metadata is 4 KB total.

   Type: String to string map

   Required: No

eventName

   Application-defined name of the event to send to clients.

   Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Required: Yes

roomIdentifier

Identifier of the room to which the event will be sent. Currently this must be an ARN.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
    "id": "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.

id

An identifier generated by Amazon IVS Chat. This identifier must be used in subsequent operations for this message, such as DeleteMessage.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

Errors

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

AccessDeniedException

HTTP Status Code: 403

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ThrottlingException

HTTP Status Code: 429

ValidationException

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 V2
- AWS SDK for .NET V4
- 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

Adds or updates tags for the AWS resource with the specified ARN.

Request Syntax

```
POST /tags/resourceArn HTTP/1.1
Content-type: application/json

{
   "tags": {
      "string" : "string"
   }
}
```

URI Request Parameters

The request uses the following URI parameters.

resourceArn

The ARN of the resource to be tagged. The ARN must be URL-encoded.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:[a-z-]/[a-zA-Z0-9-]+`

Required: Yes

Request Body

The request accepts the following data in JSON format.

tags

Array of tags to be added or updated. Array of maps, each of the form `string:string` (`key:value`). See Best practices and strategies in *Tagging AWS Resources and Tag Editor* for details, including restrictions that apply to tags and "Tag naming limits and requirements"; Amazon IVS Chat has no constraints beyond what is documented there.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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.

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2

- [AWS SDK for .NET V4](#)

- [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

Removes tags from the resource with the specified ARN.

Request Syntax

```
DELETE /tags/resourceArn?tagKeys=tagKeys HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

resourceArn

The ARN of the resource to be untagged. The ARN must be URL-encoded.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:[a-z-]/[a-zA-Z0-9-]+`

Required: Yes

tagKeys

Array of tag keys (strings) for the tags to be removed. See Best practices and strategies in *Tagging AWS Resources and Tag Editor* for details, including restrictions that apply to tags and "Tag naming limits and requirements"; Amazon IVS Chat has no constraints on tags beyond what is documented there.

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

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

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
```

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](#).

InternalServerException

HTTP Status Code: 500

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2](#)
- [AWS SDK for .NET V4](#)
- [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](#)

UpdateLoggingConfiguration

Updates a specified logging configuration.

Request Syntax

```
POST /UpdateLoggingConfiguration HTTP/1.1
Content-type: application/json

{
   "destinationConfiguration": { ... },
   "identifier": "string",
   "name": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

destinationConfiguration

A complex type that contains a destination configuration for where chat content will be logged. There can be only one type of destination (cloudWatchLogs, firehose, or s3) in a destinationConfiguration.

Type: DestinationConfiguration object

Note: This object is a Union. Only one member of this object can be specified or returned.

Required: No

identifier

Identifier of the logging configuration to be updated.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

Required: Yes

name

Logging-configuration name. The value does not need to be unique.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createTime": "string",
   "destinationConfiguration": { ... },
   "id": "string",
   "name": "string",
   "state": "string",
   "tags": {
      "string" : "string"
   },
   "updateTime": "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.

arn

Logging-configuration ARN, from the request (if `identifier` was an ARN).

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

createTime

Time when the logging configuration was created. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

destinationConfiguration

A complex type that contains a destination configuration for where chat content will be logged, from the request. There is only one type of destination (`cloudWatchLogs`, `firehose`, or `s3`) in a `destinationConfiguration`.

Type: DestinationConfiguration object

Note: This object is a Union. Only one member of this object can be specified or returned.

id

Logging-configuration ID, generated by the system. This is a relative identifier, the part of the ARN that uniquely identifies the room.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

name

Logging-configuration name, from the request (if specified).

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

state

The state of the logging configuration. When the state is `ACTIVE`, the configuration is ready to log chat content.

Type: String

Valid Values: `ACTIVE`

tags

Tags attached to the resource. Array of maps, each of the form `string:string` (`key:value`).

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

updateTime

Time of the logging configuration's last update. This is an ISO 8601 timestamp; *note that this is returned as a string.*

Type: Timestamp

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

HTTP Status Code: 403

ConflictException

HTTP Status Code: 409

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2](#)
- [AWS SDK for .NET V4](#)
- [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](#)

UpdateRoom

Updates a room's configuration.

Request Syntax

```
POST /UpdateRoom HTTP/1.1
Content-type: application/json

{
   "identifier": "string",
   "loggingConfigurationIdentifiers": [ "string" ],
   "maximumMessageLength": number,
   "maximumMessageRatePerSecond": number,
   "messageReviewHandler": {
      "fallbackResult": "string",
      "uri": "string"
   },
   "name": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

identifier

Identifier of the room to be updated. Currently this must be an ARN.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

Required: Yes

loggingConfigurationIdentifiers

Array of logging-configuration identifiers attached to the room.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 3 items.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

Required: No

maximumMessageLength

The maximum number of characters in a single message. Messages are expected to be UTF-8 encoded and this limit applies specifically to rune/code-point count, not number of bytes. Default: 500.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

Required: No

maximumMessageRatePerSecond

Maximum number of messages per second that can be sent to the room (by all clients). Default: 10.

Type: Integer

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

Required: No

messageReviewHandler

Configuration information for optional review of messages. Specify an empty `uri` string to disassociate a message review handler from the specified room.

Type: MessageReviewHandler object

Required: No

name

Room name. The value does not need to be unique.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: [a-zA-Z0-9-_]*

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "arn": "string",
   "createTime": "string",
   "id": "string",
   "loggingConfigurationIdentifiers": [ "string" ],
   "maximumMessageLength": number,
   "maximumMessageRatePerSecond": number,
   "messageReviewHandler": {
      "fallbackResult": "string",
      "uri": "string"
   },
   "name": "string",
   "tags": {
      "string" : "string"
   },
   "updateTime": "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.

arn

Room ARN, from the request (if `identifier` was an ARN).

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

createTime

Time when the room was created. This is an ISO 8601 timestamp; *note that this is returned as a string.*

Type: Timestamp

id

Room ID, generated by the system. This is a relative identifier, the part of the ARN that uniquely identifies the room.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

loggingConfigurationIdentifiers

Array of logging configurations attached to the room, from the request (if specified).

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 3 items.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

maximumMessageLength

Maximum number of characters in a single message, from the request (if specified).

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 500.

maximumMessageRatePerSecond

Maximum number of messages per second that can be sent to the room (by all clients), from the request (if specified).

Type: Integer

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

messageReviewHandler

Configuration information for optional review of messages.

Type: MessageReviewHandler object

name

Room name, from the request (if specified).

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

tags

Tags attached to the resource. Array of maps, each of the form `string:string` `(key:value)`.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

updateTime

Time of the room's last update. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

Errors

For information about the errors that are common to all actions, see [Common Errors](#).

AccessDeniedException

HTTP Status Code: 403

PendingVerification

HTTP Status Code: 403

ResourceNotFoundException

HTTP Status Code: 404

ValidationException

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 V2](#)
- [AWS SDK for .NET V4](#)
- [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 Amazon Interactive Video Service Chat API contains several data types that various actions use. This section describes each data type in detail.

> **Note**
>
> The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- CloudWatchLogsDestinationConfiguration
- DestinationConfiguration
- FirehoseDestinationConfiguration
- LoggingConfigurationSummary
- MessageReviewHandler
- RoomSummary
- S3DestinationConfiguration
- ValidationExceptionField

CloudWatchLogsDestinationConfiguration

Specifies a CloudWatch Logs location where chat logs will be stored.

Contents

logGroupName

Name of the Amazon Cloudwatch Logs destination where chat activity will be logged.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 512.

Pattern: [\.\-_/#A-Za-z0-9]+

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 C++)
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DestinationConfiguration

A complex type that describes a location where chat logs will be stored. Each member represents the configuration of one log destination. For logging, you define only one type of destination (for CloudWatch Logs, Kinesis Firehose, or S3).

Contents

> ⚠️ **Important**
>
> This data type is a UNION, so only one of the following members can be specified when used or returned.

cloudWatchLogs

An Amazon CloudWatch Logs destination configuration where chat activity will be logged.

Type: CloudWatchLogsDestinationConfiguration object

Required: No

firehose

An Amazon Kinesis Data Firehose destination configuration where chat activity will be logged.

Type: FirehoseDestinationConfiguration object

Required: No

s3

An Amazon S3 destination configuration where chat activity will be logged.

Type: S3DestinationConfiguration 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](#)

FirehoseDestinationConfiguration

Specifies a Kinesis Firehose location where chat logs will be stored.

Contents

deliveryStreamName

Name of the Amazon Kinesis Firehose delivery stream where chat activity will be logged.

Type: String

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

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

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](#)

LoggingConfigurationSummary

Summary information about a logging configuration.

Contents

arn

> Logging-configuration ARN.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 128.
>
> Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`
>
> Required: No

createTime

> Time when the logging configuration was created. This is an ISO 8601 timestamp; *note that this is returned as a string*.
>
> Type: Timestamp
>
> Required: No

destinationConfiguration

> A complex type that contains a destination configuration for where chat content will be logged.
>
> Type: DestinationConfiguration object
>
> **Note:** This object is a Union. Only one member of this object can be specified or returned.
>
> Required: No

id

> Logging-configuration ID, generated by the system. This is a relative identifier, the part of the ARN that uniquely identifies the room.
>
> Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

Required: No

name

Logging-configuration name. The value does not need to be unique.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

Required: No

state

The state of the logging configuration. When this is `ACTIVE`, the configuration is ready for logging chat content.

Type: String

Valid Values: `CREATING | CREATE_FAILED | DELETING | DELETE_FAILED | UPDATING | UPDATE_FAILED | ACTIVE`

Required: No

tags

Tags to attach to the resource. Array of maps, each of the form `string:string` (`key:value`). See [Best practices and strategies](#) in *Tagging AWS Resources and Tag Editor* for details, including restrictions that apply to tags and "Tag naming limits and requirements"; Amazon IVS Chat has no constraints on tags beyond what is documented there.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

Required: No

updateTime

Time of the logging configuration's last update. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: 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](#)

MessageReviewHandler

Configuration information for optional message review.

Contents

fallbackResult

Specifies the fallback behavior (whether the message is allowed or denied) if the handler does not return a valid response, encounters an error, or times out. (For the timeout period, see Service Quotas.) If allowed, the message is delivered with returned content to all users connected to the room. If denied, the message is not delivered to any user. Default: ALLOW.

Type: String

Valid Values: ALLOW | DENY

Required: No

uri

Identifier of the message review handler. Currently this must be an ARN of a lambda function.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 170.

Pattern: $|^arn:aws:lambda:[a-z0-9-]+:[0-9]{12}:function:.+.*

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

RoomSummary

Summary information about a room.

Contents

arn

Room ARN.

Type: String

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:room/[a-zA-Z0-9-]+`

Required: No

createTime

Time when the room was created. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: Timestamp

Required: No

id

Room ID, generated by the system. This is a relative identifier, the part of the ARN that uniquely identifies the room.

Type: String

Length Constraints: Fixed length of 12.

Pattern: `[a-zA-Z0-9]+`

Required: No

loggingConfigurationIdentifiers

List of logging-configuration identifiers attached to the room.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 3 items.

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

Pattern: `arn:aws:ivschat:[a-z0-9-]+:[0-9]+:logging-configuration/[a-zA-Z0-9-]+`

Required: No

messageReviewHandler

Configuration information for optional review of messages.

Type: MessageReviewHandler object

Required: No

name

Room name. The value does not need to be unique.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 128.

Pattern: `[a-zA-Z0-9-_]*`

Required: No

tags

Tags attached to the resource. Array of maps, each of the form `string:string` (`key:value`). See [Best practices and strategies](#) in *Tagging AWS Resources and Tag Editor* for details, including restrictions that apply to tags and "Tag naming limits and requirements"; Amazon IVS Chat has no constraints beyond what is documented there.

Type: String to string map

Map Entries: Minimum number of 0 items. Maximum number of 50 items.

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

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

Required: No

updateTime

Time of the room's last update. This is an ISO 8601 timestamp; *note that this is returned as a string*.

Type: 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](#)

S3DestinationConfiguration

Specifies an S3 location where chat logs will be stored.

Contents

bucketName

Name of the Amazon S3 bucket where chat activity will be logged.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 63.

Pattern: `[a-z0-9-.]+`

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](#)

ValidationExceptionField

This object is used in the ValidationException error.

Contents

message

Explanation of the reason for the validation error.

Type: String

Required: Yes

name

Name of the field which failed validation.

Type: String

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](#)

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: 403

ExpiredTokenException

The security token included in the request is expired

HTTP Status Code: 403

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 403

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

MalformedHttpRequestException

Problems with the request at the HTTP level, e.g. we can't decompress the body according to the decompression algorithm specified by the content-encoding.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 401

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestAbortedException

Convenient exception that can be used when a request is aborted before a reply is sent back (e.g. client closed connection).

HTTP Status Code: 400

RequestEntityTooLargeException

Problems with the request at the HTTP level. The request entity is too large.

HTTP Status Code: 413

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

RequestTimeoutException

Problems with the request at the HTTP level. Reading the Request timed out.

HTTP Status Code: 408

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

UnrecognizedClientException

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

UnknownOperationException

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 404

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400