
AWS IoT Greengrass

API Reference, Version 2



AWS IoT Greengrass: API Reference, Version 2

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Welcome	1
Actions	2
CancelDeployment	3
Request Syntax	3
URI Request Parameters	3
Request Body	3
Response Syntax	3
Response Elements	3
Errors	3
See Also	4
CreateComponentVersion	5
Request Syntax	5
URI Request Parameters	6
Request Body	6
Response Syntax	7
Response Elements	7
Errors	8
See Also	9
CreateDeployment	10
Request Syntax	10
URI Request Parameters	11
Request Body	11
Response Syntax	12
Response Elements	12
Errors	13
See Also	13
DeleteComponent	15
Request Syntax	15
URI Request Parameters	15
Request Body	15
Response Syntax	15
Response Elements	15
Errors	15
See Also	16
DeleteCoreDevice	17
Request Syntax	17
URI Request Parameters	17
Request Body	17
Response Syntax	17
Response Elements	17
Errors	17
See Also	18
DescribeComponent	19
Request Syntax	19
URI Request Parameters	19
Request Body	19
Response Syntax	19
Response Elements	20
Errors	21
See Also	21
GetComponent	23
Request Syntax	23
URI Request Parameters	23
Request Body	23

Response Syntax	23
Response Elements	23
Errors	24
See Also	24
GetComponentVersionArtifact	26
Request Syntax	26
URI Request Parameters	26
Request Body	26
Response Syntax	26
Response Elements	26
Errors	27
See Also	27
GetCoreDevice	29
Request Syntax	29
URI Request Parameters	29
Request Body	29
Response Syntax	29
Response Elements	29
Errors	30
See Also	31
GetDeployment	32
Request Syntax	32
URI Request Parameters	32
Request Body	32
Response Syntax	32
Response Elements	33
Errors	35
See Also	35
ListComponents	37
Request Syntax	37
URI Request Parameters	37
Request Body	37
Response Syntax	37
Response Elements	38
Errors	38
See Also	38
ListComponentVersions	40
Request Syntax	40
URI Request Parameters	40
Request Body	40
Response Syntax	40
Response Elements	40
Errors	41
See Also	41
ListCoreDevices	43
Request Syntax	43
URI Request Parameters	43
Request Body	43
Response Syntax	43
Response Elements	44
Errors	44
See Also	44
ListDeployments	46
Request Syntax	46
URI Request Parameters	46
Request Body	46
Response Syntax	46

Response Elements	47
Errors	47
See Also	47
ListEffectiveDeployments	49
Request Syntax	49
URI Request Parameters	49
Request Body	49
Response Syntax	49
Response Elements	50
Errors	50
See Also	50
ListInstalledComponents	52
Request Syntax	52
URI Request Parameters	52
Request Body	52
Response Syntax	52
Response Elements	52
Errors	53
See Also	53
ListTagsForResource	55
Request Syntax	55
URI Request Parameters	55
Request Body	55
Response Syntax	55
Response Elements	55
Errors	56
See Also	56
ResolveComponentCandidates	57
Request Syntax	57
URI Request Parameters	57
Request Body	57
Response Syntax	58
Response Elements	58
Errors	58
See Also	59
TagResource	60
Request Syntax	60
URI Request Parameters	60
Request Body	60
Response Syntax	60
Response Elements	61
Errors	61
See Also	61
UntagResource	62
Request Syntax	62
URI Request Parameters	62
Request Body	62
Response Syntax	62
Response Elements	62
Errors	62
See Also	63
Data Types	64
CloudComponentStatus	65
Contents	65
See Also	65
Component	66
Contents	66

See Also	66
ComponentCandidate	67
Contents	67
See Also	67
ComponentConfigurationUpdate	68
Contents	68
See Also	68
ComponentDependencyRequirement	69
Contents	69
See Also	69
ComponentDeploymentSpecification	70
Contents	70
See Also	70
ComponentLatestVersion	71
Contents	71
See Also	72
ComponentPlatform	73
Contents	73
See Also	73
ComponentRunWith	74
Contents	74
See Also	74
ComponentVersionListItem	75
Contents	75
See Also	75
CoreDevice	76
Contents	76
See Also	76
Deployment	77
Contents	77
See Also	78
DeploymentComponentUpdatePolicy	79
Contents	79
See Also	79
DeploymentConfigurationValidationPolicy	80
Contents	80
See Also	80
DeploymentIoTJobConfiguration	81
Contents	81
See Also	81
DeploymentPolicies	82
Contents	82
See Also	82
EffectiveDeployment	83
Contents	83
See Also	84
InstalledComponent	85
Contents	85
See Also	85
IoTJobAbortConfig	87
Contents	87
See Also	87
IoTJobAbortCriteria	88
Contents	88
See Also	88
IoTJobExecutionsRolloutConfig	90
Contents	90

See Also	90
IoTJobExponentialRolloutRate	91
Contents	91
See Also	91
IoTJobRateIncreaseCriteria	92
Contents	92
See Also	92
IoTJobTimeoutConfig	93
Contents	93
See Also	93
LambdaContainerParams	94
Contents	94
See Also	94
LambdaDeviceMount	95
Contents	95
See Also	95
LambdaEventSource	96
Contents	96
See Also	96
LambdaExecutionParameters	97
Contents	97
See Also	98
LambdaFunctionRecipeSource	99
Contents	99
See Also	100
LambdaLinuxProcessParams	101
Contents	101
See Also	101
LambdaVolumeMount	102
Contents	102
See Also	102
ResolvedComponentVersion	103
Contents	103
See Also	103
ValidationExceptionField	105
Contents	105
See Also	105
Common Parameters	106
Common Errors	108

Welcome

AWS IoT Greengrass brings local compute, messaging, data management, sync, and ML inference capabilities to edge devices. This enables devices to collect and analyze data closer to the source of information, react autonomously to local events, and communicate securely with each other on local networks. Local devices can also communicate securely with AWS IoT Core and export IoT data to the AWS Cloud. AWS IoT Greengrass developers can use AWS Lambda functions and components to create and deploy applications to fleets of edge devices for local operation.

AWS IoT Greengrass Version 2 provides a new major version of the AWS IoT Greengrass Core software, new APIs, and a new console. Use this API reference to learn how to use the AWS IoT Greengrass V2 API operations to manage components, manage deployments, and core devices.

For more information, see [What is AWS IoT Greengrass?](#) in the *AWS IoT Greengrass V2 Developer Guide*.

This document was last published on June 14, 2021.

Actions

The following actions are supported:

- [CancelDeployment](#) (p. 3)
- [CreateComponentVersion](#) (p. 5)
- [CreateDeployment](#) (p. 10)
- [DeleteComponent](#) (p. 15)
- [DeleteCoreDevice](#) (p. 17)
- [DescribeComponent](#) (p. 19)
- [GetComponent](#) (p. 23)
- [GetComponentVersionArtifact](#) (p. 26)
- [GetCoreDevice](#) (p. 29)
- [GetDeployment](#) (p. 32)
- [ListComponents](#) (p. 37)
- [ListComponentVersions](#) (p. 40)
- [ListCoreDevices](#) (p. 43)
- [ListDeployments](#) (p. 46)
- [ListEffectiveDeployments](#) (p. 49)
- [ListInstalledComponents](#) (p. 52)
- [ListTagsForResource](#) (p. 55)
- [ResolveComponentCandidates](#) (p. 57)
- [TagResource](#) (p. 60)
- [UntagResource](#) (p. 62)

CancelDeployment

Cancels a deployment. This operation cancels the deployment for devices that haven't yet received it. If a device already received the deployment, this operation doesn't change anything for that device.

Request Syntax

```
POST /greengrass/v2/deployments/deploymentId/cancel HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

deploymentId (p. 3)

The ID of the deployment.

Length Constraints: Minimum length of 1.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
  "message": "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.

message (p. 3)

A message that communicates if the cancel was successful.

Type: String

Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateComponentVersion

Creates a component. Components are software that run on AWS IoT Greengrass core devices. After you develop and test a component on your core device, you can use this operation to upload your component to AWS IoT Greengrass. Then, you can deploy the component to other core devices.

You can use this operation to do the following:

- **Create components from recipes**

Create a component from a recipe, which is a file that defines the component's metadata, parameters, dependencies, lifecycle, artifacts, and platform capability. For more information, see [AWS IoT Greengrass component recipe reference](#) in the *AWS IoT Greengrass V2 Developer Guide*.

To create a component from a recipe, specify `inlineRecipe` when you call this operation.

- **Create components from Lambda functions**

Create a component from an AWS Lambda function that runs on AWS IoT Greengrass. This creates a recipe and artifacts from the Lambda function's deployment package. You can use this operation to migrate Lambda functions from AWS IoT Greengrass V1 to AWS IoT Greengrass V2.

This function only accepts Lambda functions that use the following runtimes:

- Python 2.7 – `python2.7`
- Python 3.7 – `python3.7`
- Python 3.8 – `python3.8`
- Java 8 – `java8`
- Node.js 10 – `nodejs10.x`
- Node.js 12 – `nodejs12.x`

To create a component from a Lambda function, specify `lambdaFunction` when you call this operation.

Request Syntax

```
POST /greengrass/v2/createComponentVersion HTTP/1.1  
Content-type: application/json
```

```
{  
  "inlineRecipe": blob,  
  "lambdaFunction": {  
    "componentDependencies": {  
      "string": {  
        "dependencyType": "string",  
        "versionRequirement": "string"  
      }  
    },  
    "componentLambdaParameters": {  
      "environmentVariables": {  
        "string": "string"  
      }  
    },  
    "eventSources": [  
      {  
        "topic": "string",  
        "type": "string"  
      }  
    ],  
    "execArgs": [ "string" ],  
  }  
}
```

```
"inputPayloadEncodingType": "string",
"linuxProcessParams": {
  "containerParams": {
    "devices": [
      {
        "addGroupOwner": boolean,
        "path": "string",
        "permission": "string"
      }
    ],
    "memorySizeInKB": number,
    "mountROSysfs": boolean,
    "volumes": [
      {
        "addGroupOwner": boolean,
        "destinationPath": "string",
        "permission": "string",
        "sourcePath": "string"
      }
    ]
  }
},
"isolationMode": "string"
},
"maxIdleTimeInSeconds": number,
"maxInstancesCount": number,
"maxQueueSize": number,
"pinned": boolean,
"statusTimeoutInSeconds": number,
"timeoutInSeconds": number
},
"componentName": "string",
"componentPlatforms": [
  {
    "attributes": {
      "string" : "string"
    },
    "name": "string"
  }
],
"componentVersion": "string",
"lambdaArn": "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.

[inlineRecipe \(p. 5\)](#)

The recipe to use to create the component. The recipe defines the component's metadata, parameters, dependencies, lifecycle, artifacts, and platform compatibility.

You must specify either `inlineRecipe` or `lambdaFunction`.

Type: Base64-encoded binary data object

Required: No

[lambdaFunction \(p. 5\)](#)

The parameters to create a component from a Lambda function.

You must specify either `inlineRecipe` or `lambdaFunction`.

Type: [LambdaFunctionRecipeSource \(p. 99\)](#) object

Required: No

[tags \(p. 5\)](#)

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^(?!aws:)[a-zA-Z+-. _:/]+`

Value Length Constraints: Maximum length of 256.

Required: No

Response Syntax

```
HTTP/1.1 201
Content-type: application/json

{
  "arn": "string",
  "componentName": "string",
  "componentVersion": "string",
  "creationTimestamp": number,
  "status": {
    "componentState": "string",
    "errors": {
      "string" : "string"
    },
    "message": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 201 response.

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

[arn \(p. 7\)](#)

The [ARN](#) of the component version.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

componentName (p. 7)

The name of the component.

Type: String

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

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

componentVersion (p. 7)

The version of the component.

Type: String

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

Pattern: `[0-9a-zA-Z-.]+`

creationTimestamp (p. 7)

The time at which the component was created, expressed in ISO 8601 format.

Type: Timestamp

status (p. 7)

The status of the component version in AWS IoT Greengrass V2. This status is different from the status of the component on a core device.

Type: [CloudComponentStatus \(p. 65\)](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ServiceQuotaExceededException

Your request exceeds a service quota. For example, you might have the maximum number of components that you can create.

HTTP Status Code: 402

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateDeployment

Creates a continuous deployment for a target, which is a AWS IoT Greengrass core device or group of core devices. When you add a new core device to a group of core devices that has a deployment, AWS IoT Greengrass deploys that group's deployment to the new device.

You can define one deployment for each target. When you create a new deployment for a target that has an existing deployment, you replace the previous deployment. AWS IoT Greengrass applies the new deployment to the target devices.

Every deployment has a revision number that indicates how many deployment revisions you define for a target. Use this operation to create a new revision of an existing deployment. This operation returns the revision number of the new deployment when you create it.

For more information, see the [Create deployments](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Request Syntax

```
POST /greengrass/v2/deployments HTTP/1.1
Content-type: application/json

{
  "components": {
    "string": {
      "componentVersion": "string",
      "configurationUpdate": {
        "merge": "string",
        "reset": [ "string" ]
      },
      "runWith": {
        "posixUser": "string"
      }
    }
  },
  "deploymentName": "string",
  "deploymentPolicies": {
    "componentUpdatePolicy": {
      "action": "string",
      "timeoutInSeconds": number
    },
    "configurationValidationPolicy": {
      "timeoutInSeconds": number
    },
    "failureHandlingPolicy": "string"
  },
  "iotJobConfiguration": {
    "abortConfig": {
      "criteriaList": [
        {
          "action": "string",
          "failureType": "string",
          "minNumberOfExecutedThings": number,
          "thresholdPercentage": number
        }
      ]
    }
  },
  "jobExecutionsRolloutConfig": {
    "exponentialRate": {
      "baseRatePerMinute": number,
      "incrementFactor": number,
      "rateIncreaseCriteria": {
```

```
        "numberOfNotifiedThings": number,  
        "numberOfSucceededThings": number  
    },  
    },  
    "maximumPerMinute": number  
},  
"timeoutConfig": {  
    "inProgressTimeoutInMinutes": number  
}  
},  
"tags": {  
    "string" : "string"  
},  
"targetArn": "string"  
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

components (p. 10)

The components to deploy. This is a dictionary, where each key is the name of a component, and each key's value is the version and configuration to deploy for that component.

Type: String to [ComponentDeploymentSpecification](#) (p. 70) object map

Key Length Constraints: Minimum length of 1.

Required: No

deploymentName (p. 10)

The name of the deployment.

You can create deployments without names. If you create a deployment without a name, the AWS IoT Greengrass V2 console shows the deployment name as `<targetType>:<targetName>`, where `targetType` and `targetName` are the type and name of the deployment target.

Type: String

Length Constraints: Minimum length of 1.

Required: No

deploymentPolicies (p. 10)

The deployment policies for the deployment. These policies define how the deployment updates components and handles failure.

Type: [DeploymentPolicies](#) (p. 82) object

Required: No

iotJobConfiguration (p. 10)

The job configuration for the deployment configuration. The job configuration specifies the rollout, timeout, and stop configurations for the deployment configuration.

Type: [DeploymentIoTJobConfiguration \(p. 81\)](#) object

Required: No

[tags \(p. 10\)](#)

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^(?!aws:)[a-zA-Z+-._: /]+`

Value Length Constraints: Maximum length of 256.

Required: No

[targetArn \(p. 10\)](#)

The [ARN](#) of the target AWS IoT thing or thing group.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:(thing|thinggroup)/.+`

Required: Yes

Response Syntax

```
HTTP/1.1 201
Content-type: application/json

{
  "deploymentId": "string",
  "iotJobArn": "string",
  "iotJobId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 201 response.

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

[deploymentId \(p. 12\)](#)

The ID of the deployment.

Type: String

Length Constraints: Minimum length of 1.

[iotJobArn \(p. 12\)](#)

The [ARN](#) of the AWS IoT job that applies the deployment to target devices.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:job/.+
iotJobId \(p. 12\)`

The ID of the AWS IoT job that applies the deployment to target devices.

Type: String

Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteComponent

Deletes a version of a component from AWS IoT Greengrass.

Note

This operation deletes the component's recipe and artifacts. As a result, deployments that refer to this component version will fail. If you have deployments that use this component version, you can remove the component from the deployment or update the deployment to use a valid version.

Request Syntax

```
DELETE /greengrass/v2/components/arn HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

[arn \(p. 15\)](#)

The [ARN](#) of the component version.

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: Yes

Request Body

The request does not have a request body.

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 \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteCoreDevice

Deletes a AWS IoT Greengrass core device, which is an AWS IoT thing. This operation removes the core device from the list of core devices. This operation doesn't delete the AWS IoT thing. For more information about how to delete the AWS IoT thing, see [DeleteThing](#) in the *AWS IoT API Reference*.

Request Syntax

```
DELETE /greengrass/v2/coreDevices/coreDeviceThingName HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

coreDeviceThingName (p. 17)

The name of the core device. This is also the name of the AWS IoT thing.

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/1.1 204
```

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](#) (p. 108).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DescribeComponent

Retrieves metadata for a version of a component.

Request Syntax

```
GET /greengrass/v2/components/arn/metadata HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

arn (p. 19)

The [ARN](#) of the component version.

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
  "arn": "string",
  "componentName": "string",
  "componentVersion": "string",
  "creationTimestamp": number,
  "description": "string",
  "platforms": [
    {
      "attributes": {
        "string": "string"
      },
      "name": "string"
    }
  ],
  "publisher": "string",
  "status": {
    "componentState": "string",
    "errors": {
      "string": "string"
    },
    "message": "string"
  },
  "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.

arn (p. 19)

The [ARN](#) of the component version.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

componentName (p. 19)

The name of the component.

Type: String

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

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

componentVersion (p. 19)

The version of the component.

Type: String

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

Pattern: `[0-9a-zA-Z-.]+`

creationTimestamp (p. 19)

The time at which the component was created, expressed in ISO 8601 format.

Type: Timestamp

description (p. 19)

The description of the component version.

Type: String

platforms (p. 19)

The platforms that the component version supports.

Type: Array of [ComponentPlatform](#) (p. 73) objects

publisher (p. 19)

The publisher of the component version.

Type: String

status (p. 19)

The status of the component version in AWS IoT Greengrass V2. This status is different from the status of the component on a core device.

Type: [CloudComponentStatus](#) (p. 65) object

tags (p. 19)

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^(?!aws:)[a-zA-Z+*=._:/]+`

Value Length Constraints: Maximum length of 256.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)

- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetComponent

Gets the recipe for a version of a component. Core devices can call this operation to identify the artifacts and requirements to install a component.

Request Syntax

```
GET /greengrass/v2/components/arn?recipeOutputFormat=recipeOutputFormat HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

[arn \(p. 23\)](#)

The [ARN](#) of the component version.

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: Yes

[recipeOutputFormat \(p. 23\)](#)

The format of the recipe.

Valid Values: `JSON` | `YAML`

Request Body

The request does not have a request body.

Response Syntax

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

{
  "recipe": blob,
  "recipeOutputFormat": "string",
  "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.

[recipe \(p. 23\)](#)

The recipe of the component version.

Type: Base64-encoded binary data object

recipeOutputFormat (p. 23)

The format of the recipe.

Type: String

Valid Values: JSON | YAML

tags (p. 23)

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^(?!aws:)[a-zA-Z+-. _:/]+`

Value Length Constraints: Maximum length of 256.

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 108).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetComponentVersionArtifact

Gets the pre-signed URL to download a public component artifact. Core devices call this operation to identify the URL that they can use to download an artifact to install.

Request Syntax

```
GET /greengrass/v2/components/arn/artifacts/artifactName+ HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

arn (p. 26)

The [ARN](#) of the component version. Specify the ARN of a public component version.

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: Yes

artifactName (p. 26)

The name of the artifact.

You can use the [GetComponent](#) operation to download the component recipe, which includes the URI of the artifact. The artifact name is the section of the URI after the scheme. For example, in the artifact URI `greengrass:SomeArtifact.zip`, the artifact name is `SomeArtifact.zip`.

Length Constraints: Minimum length of 1.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
  "preSignedUrl": "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.

[preSignedUrl \(p. 26\)](#)

The URL of the artifact.

Type: String

Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetCoreDevice

Retrieves metadata for a AWS IoT Greengrass core device.

Request Syntax

```
GET /greengrass/v2/coreDevices/coreDeviceThingName HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

[coreDeviceThingName \(p. 29\)](#)

The name of the core device. This is also the name of the AWS IoT thing.

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
Content-type: application/json

{
  "architecture": "string",
  "coreDeviceThingName": "string",
  "coreVersion": "string",
  "lastStatusUpdateTimestamp": number,
  "platform": "string",
  "status": "string",
  "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.

[architecture \(p. 29\)](#)

The computer architecture of the core device.

Type: String

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

coreDeviceThingName (p. 29)

The name of the core device. This is also the name of the AWS IoT thing.

Type: String

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

coreVersion (p. 29)

The version of the AWS IoT Greengrass Core software that the core device runs. This version is equivalent to the version of the AWS IoT Greengrass nucleus component that runs on the core device. For more information, see the [AWS IoT Greengrass nucleus component](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String

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

lastStatusUpdateTimestamp (p. 29)

The time at which the core device's status last updated, expressed in ISO 8601 format.

Type: Timestamp

platform (p. 29)

The operating system platform that the core device runs.

Type: String

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

status (p. 29)

The status of the core device. The core device status can be:

- **HEALTHY** – The AWS IoT Greengrass Core software and all components run on the core device without issue.
- **UNHEALTHY** – The AWS IoT Greengrass Core software or a component is in a failed state on the core device.

Type: String

Valid Values: **HEALTHY** | **UNHEALTHY**

tags (p. 29)

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^(?!aws:)[a-zA-Z+-=._:/]+`

Value Length Constraints: Maximum length of 256.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetDeployment

Gets a deployment. Deployments define the components that run on AWS IoT Greengrass core devices.

Request Syntax

```
GET /greengrass/v2/deployments/deploymentId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

deploymentId (p. 32)

The ID of the deployment.

Length Constraints: Minimum length of 1.

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

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

{
  "components": {
    "string": {
      "componentVersion": "string",
      "configurationUpdate": {
        "merge": "string",
        "reset": [ "string" ]
      },
      "runWith": {
        "posixUser": "string"
      }
    }
  },
  "creationTimestamp": number,
  "deploymentId": "string",
  "deploymentName": "string",
  "deploymentPolicies": {
    "componentUpdatePolicy": {
      "action": "string",
      "timeoutInSeconds": number
    },
    "configurationValidationPolicy": {
      "timeoutInSeconds": number
    },
    "failureHandlingPolicy": "string"
  },
  "deploymentStatus": "string",
```

```

"iotJobArn": "string",
"iotJobConfiguration": {
  "abortConfig": {
    "criteriaList": [
      {
        "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": number,
        "thresholdPercentage": number
      }
    ]
  },
  "jobExecutionsRolloutConfig": {
    "exponentialRate": {
      "baseRatePerMinute": number,
      "incrementFactor": number,
      "rateIncreaseCriteria": {
        "numberOfNotifiedThings": number,
        "numberOfSucceededThings": number
      }
    },
    "maximumPerMinute": number
  },
  "timeoutConfig": {
    "inProgressTimeoutInMinutes": number
  }
},
"iotJobId": "string",
"isLatestForTarget": boolean,
"revisionId": "string",
"tags": {
  "string" : "string"
},
"targetArn": "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.

components (p. 32)

The components to deploy. This is a dictionary, where each key is the name of a component, and each key's value is the version and configuration to deploy for that component.

Type: String to [ComponentDeploymentSpecification](#) (p. 70) object map

Key Length Constraints: Minimum length of 1.

creationTimestamp (p. 32)

The time at which the deployment was created, expressed in ISO 8601 format.

Type: Timestamp

deploymentId (p. 32)

The ID of the deployment.

Type: String

Length Constraints: Minimum length of 1.

deploymentName (p. 32)

The name of the deployment.

You can create deployments without names. If you create a deployment without a name, the AWS IoT Greengrass V2 console shows the deployment name as `<targetType>:<targetName>`, where `targetType` and `targetName` are the type and name of the deployment target.

Type: String

deploymentPolicies (p. 32)

The deployment policies for the deployment. These policies define how the deployment updates components and handles failure.

Type: [DeploymentPolicies \(p. 82\)](#) object

deploymentStatus (p. 32)

The status of the deployment.

Type: String

Valid Values: ACTIVE | COMPLETED | CANCELED | FAILED | INACTIVE

iotJobArn (p. 32)

The [ARN](#) of the AWS IoT job that applies the deployment to target devices.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:job/.+`

iotJobConfiguration (p. 32)

The job configuration for the deployment configuration. The job configuration specifies the rollout, timeout, and stop configurations for the deployment configuration.

Type: [DeploymentIoTJobConfiguration \(p. 81\)](#) object

iotJobId (p. 32)

The ID of the AWS IoT job that applies the deployment to target devices.

Type: String

isLatestForTarget (p. 32)

Whether or not the deployment is the latest revision for its target.

Type: Boolean

revisionId (p. 32)

The revision number of the deployment.

Type: String

Length Constraints: Minimum length of 1.

tags (p. 32)

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^(?!aws:)[a-zA-Z+-._: /]+`

Value Length Constraints: Maximum length of 256.

targetArn (p. 32)

The [ARN](#) of the target AWS IoT thing or thing group.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:(thing|thinggroup)/.+`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListComponents

Retrieves a paginated list of component summaries. This list includes components that you have permission to view.

Request Syntax

```
GET /greengrass/v2/components?maxResults=maxResults&nextToken=nextToken&scope=scope
HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults (p. 37)

The maximum number of results to be returned per paginated request.

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

nextToken (p. 37)

The token to be used for the next set of paginated results.

scope (p. 37)

The scope of the components to list.

Default: PRIVATE

Valid Values: PRIVATE | PUBLIC

Request Body

The request does not have a request body.

Response Syntax

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

{
  "components": [
    {
      "arn": "string",
      "componentName": "string",
      "latestVersion": {
        "arn": "string",
        "componentVersion": "string",
        "creationTimestamp": number,
        "description": "string",
        "platforms": [
          {
            "attributes": {
              "string": "string"
            },
            "name": "string"
          }
        ]
      }
    }
  ]
}
```

```
    },  
    "publisher": "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.

components (p. 37)

A list that summarizes each component.

Type: Array of [Component](#) (p. 66) objects

nextToken (p. 37)

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 108).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListComponentVersions

Retrieves a paginated list of all versions for a component.

Request Syntax

```
GET /greengrass/v2/components/arn/versions?maxResults=maxResults&nextToken=nextToken
HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

[arn \(p. 40\)](#)

The [ARN](#) of the component version.

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+`

Required: Yes

[maxResults \(p. 40\)](#)

The maximum number of results to be returned per paginated request.

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

[nextToken \(p. 40\)](#)

The token to be used for the next set of paginated results.

Request Body

The request does not have a request body.

Response Syntax

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

{
  "componentVersions": [
    {
      "arn": "string",
      "componentName": "string",
      "componentVersion": "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.

componentVersions (p. 40)

A list of versions that exist for the component.

Type: Array of [ComponentVersionListItem \(p. 75\)](#) objects

nextToken (p. 40)

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListCoreDevices

Retrieves a paginated list of AWS IoT Greengrass core devices.

Request Syntax

```
GET /greengrass/v2/coreDevices?  
maxResults=maxResults&nextToken=nextToken&status=status&thingGroupArn=thingGroupArn  
HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

maxResults (p. 43)

The maximum number of results to be returned per paginated request.

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

nextToken (p. 43)

The token to be used for the next set of paginated results.

status (p. 43)

The core device status by which to filter. If you specify this parameter, the list includes only core devices that have this status. Choose one of the following options:

- **HEALTHY** – The AWS IoT Greengrass Core software and all components run on the core device without issue.
- **UNHEALTHY** – The AWS IoT Greengrass Core software or a component is in a failed state on the core device.

Valid Values: **HEALTHY** | **UNHEALTHY**

thingGroupArn (p. 43)

The [ARN](#) of the AWS IoT thing group by which to filter. If you specify this parameter, the list includes only core devices that are members of this thing group.

Pattern: `arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:thinggroup/.+`

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200  
Content-type: application/json  
  
{  
  "coreDevices": [  
    {  
      "coreDeviceThingName": "string",  
      "lastStatusUpdateTimestamp": number,  
    }  
  ]  
}
```

```
    "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.

coreDevices (p. 43)

A list that summarizes each core device.

Type: Array of [CoreDevice](#) (p. 76) objects

nextToken (p. 43)

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 108).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListDeployments

Retrieves a paginated list of deployments.

Request Syntax

```
GET /greengrass/v2/deployments?  
historyFilter=historyFilter&maxResults=maxResults&nextToken=nextToken&targetArn=targetArn  
HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

historyFilter (p. 46)

The filter for the list of deployments. Choose one of the following options:

- ALL – The list includes all deployments.
- LATEST_ONLY – The list includes only the latest revision of each deployment.

Default: LATEST_ONLY

Valid Values: ALL | LATEST_ONLY

maxResults (p. 46)

The maximum number of results to be returned per paginated request.

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

nextToken (p. 46)

The token to be used for the next set of paginated results.

targetArn (p. 46)

The [ARN](#) of the target AWS IoT thing or thing group.

Pattern: arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:(thing|thinggroup)/.+

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200  
Content-type: application/json  
  
{  
  "deployments": [  
    {  
      "creationTimestamp": number,  
      "deploymentId": "string",  
      "deploymentName": "string",  
      "deploymentStatus": "string",
```

```
    "isLatestForTarget": boolean,  
    "revisionId": "string",  
    "targetArn": "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.

deployments (p. 46)

A list that summarizes each deployment.

Type: Array of [Deployment](#) (p. 77) objects

nextToken (p. 46)

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 108).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListEffectiveDeployments

Retrieves a paginated list of deployment jobs that AWS IoT Greengrass sends to AWS IoT Greengrass core devices.

Request Syntax

```
GET /greengrass/v2/coreDevices/coreDeviceThingName/effectiveDeployments?  
maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

coreDeviceThingName (p. 49)

The name of the core device. This is also the name of the AWS IoT thing.

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

Required: Yes

maxResults (p. 49)

The maximum number of results to be returned per paginated request.

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

nextToken (p. 49)

The token to be used for the next set of paginated results.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200  
Content-type: application/json  
  
{  
  "effectiveDeployments": [  
    {  
      "coreDeviceExecutionStatus": "string",  
      "creationTimestamp": number,  
      "deploymentId": "string",  
      "deploymentName": "string",  
      "description": "string",  
      "iotJobArn": "string",  
      "iotJobId": "string",  
      "modifiedTimestamp": number,  
      "reason": "string",  
      "targetArn": "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.

effectiveDeployments (p. 49)

A list that summarizes each deployment on the core device.

Type: Array of [EffectiveDeployment \(p. 83\)](#) objects

nextToken (p. 49)

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListInstalledComponents

Retrieves a paginated list of the components that a AWS IoT Greengrass core device runs.

Request Syntax

```
GET /greengrass/v2/coreDevices/coreDeviceThingName/installedComponents?  
maxResults=maxResults&nextToken=nextToken HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

coreDeviceThingName (p. 52)

The name of the core device. This is also the name of the AWS IoT thing.

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

Required: Yes

maxResults (p. 52)

The maximum number of results to be returned per paginated request.

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

nextToken (p. 52)

The token to be used for the next set of paginated results.

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200  
Content-type: application/json  
  
{  
  "installedComponents": [  
    {  
      "componentName": "string",  
      "componentVersion": "string",  
      "isRoot": boolean,  
      "lifecycleState": "string",  
      "lifecycleStateDetails": "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.

installedComponents (p. 52)

A list that summarizes each component on the core device.

Type: Array of [InstalledComponent \(p. 85\)](#) objects

nextToken (p. 52)

The token for the next set of results, or null if there are no additional results.

Type: String

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)

- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListTagsForResource

Retrieves the list of tags for an AWS IoT Greengrass resource.

Request Syntax

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

URI Request Parameters

The request uses the following URI parameters.

resourceArn (p. 55)

The [ARN](#) of the resource.

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):(components|deployments|coreDevices):.+`

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 (p. 55)

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^(?!aws:)[a-zA-Z+-._:/$]+`

Value Length Constraints: Maximum length of 256.

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ResolveComponentCandidates

Retrieves a list of components that meet the component, version, and platform requirements of a deployment. AWS IoT Greengrass core devices call this operation when they receive a deployment to identify the components to install.

This operation identifies components that meet all dependency requirements for a deployment. If the requirements conflict, then this operation returns an error and the deployment fails. For example, this occurs if component A requires version $>2.0.0$ and component B requires version $<2.0.0$ of a component dependency.

When you specify the component candidates to resolve, AWS IoT Greengrass compares each component's digest from the core device with the component's digest in the AWS Cloud. If the digests don't match, then AWS IoT Greengrass specifies to use the version from the AWS Cloud.

Important

To use this operation, you must use the data plane API endpoint and authenticate with an AWS IoT device certificate. For more information, see [AWS IoT Greengrass endpoints and quotas](#).

Request Syntax

```
POST /greengrass/v2/resolveComponentCandidates HTTP/1.1  
Content-type: application/json
```

```
{  
  "componentCandidates": [  
    {  
      "componentName": "string",  
      "componentVersion": "string",  
      "versionRequirements": {  
        "string": "string"  
      }  
    }  
  ],  
  "platform": {  
    "attributes": {  
      "string": "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.

componentCandidates (p. 57)

The list of components to resolve.

Type: Array of [ComponentCandidate](#) (p. 67) objects

Required: Yes

platform (p. 57)

The platform to use to resolve compatible components.

Type: [ComponentPlatform \(p. 73\)](#) object

Required: Yes

Response Syntax

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

{
  "resolvedComponentVersions": [
    {
      "arn": "string",
      "componentName": "string",
      "componentVersion": "string",
      "recipe": blob
    }
  ]
}
```

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.

resolvedComponentVersions (p. 58)

A list of components that meet the requirements that you specify in the request. This list includes each component's recipe that you can use to install the component.

Type: Array of [ResolvedComponentVersion \(p. 103\)](#) objects

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 108\)](#).

AccessDeniedException

You don't have permission to perform the action.

HTTP Status Code: 403

ConflictException

Your request has conflicting operations. This can occur if you're trying to perform more than one operation on the same resource at the same time.

HTTP Status Code: 409

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ThrottlingException

Your request exceeded a request rate quota. For example, you might have exceeded the amount of times that you can retrieve device or deployment status per second.

HTTP Status Code: 429

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

TagResource

Adds tags to an AWS IoT Greengrass resource. If a tag already exists for the resource, this operation updates the tag's value.

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 (p. 60)

The [ARN](#) of the resource to tag.

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):(components|deployments|coreDevices):.+`

Required: Yes

Request Body

The request accepts the following data in JSON format.

tags (p. 60)

A list of key-value pairs that contain metadata for the resource. For more information, see [Tag your resources](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^(?!aws:)[a-zA-Z+-=._:/]+$`

Value Length Constraints: 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 \(p. 108\)](#).

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UntagResource

Removes a tag from an AWS IoT Greengrass resource.

Request Syntax

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

URI Request Parameters

The request uses the following URI parameters.

resourceArn (p. 62)

The [ARN](#) of the resource to untag.

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):(components|deployments|coreDevices):.+`

Required: Yes

tagKeys (p. 62)

A list of keys for tags to remove from the resource.

Array Members: Minimum number of 1 item. Maximum number of 50 items.

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

Pattern: `^(?!aws:)[a-zA-Z+-._: /]+`

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](#) (p. 108).

InternalServerErrorException

AWS IoT Greengrass can't process your request right now. Try again later.

HTTP Status Code: 500

ResourceNotFoundException

The requested resource can't be found.

HTTP Status Code: 404

ValidationException

The request isn't valid. This can occur if your request contains malformed JSON or unsupported characters.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

Data Types

The AWS IoT Greengrass V2 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:

- [CloudComponentStatus](#) (p. 65)
- [Component](#) (p. 66)
- [ComponentCandidate](#) (p. 67)
- [ComponentConfigurationUpdate](#) (p. 68)
- [ComponentDependencyRequirement](#) (p. 69)
- [ComponentDeploymentSpecification](#) (p. 70)
- [ComponentLatestVersion](#) (p. 71)
- [ComponentPlatform](#) (p. 73)
- [ComponentRunWith](#) (p. 74)
- [ComponentVersionListItem](#) (p. 75)
- [CoreDevice](#) (p. 76)
- [Deployment](#) (p. 77)
- [DeploymentComponentUpdatePolicy](#) (p. 79)
- [DeploymentConfigurationValidationPolicy](#) (p. 80)
- [DeploymentIoTJobConfiguration](#) (p. 81)
- [DeploymentPolicies](#) (p. 82)
- [EffectiveDeployment](#) (p. 83)
- [InstalledComponent](#) (p. 85)
- [IoTJobAbortConfig](#) (p. 87)
- [IoTJobAbortCriteria](#) (p. 88)
- [IoTJobExecutionsRolloutConfig](#) (p. 90)
- [IoTJobExponentialRolloutRate](#) (p. 91)
- [IoTJobRateIncreaseCriteria](#) (p. 92)
- [IoTJobTimeoutConfig](#) (p. 93)
- [LambdaContainerParams](#) (p. 94)
- [LambdaDeviceMount](#) (p. 95)
- [LambdaEventSource](#) (p. 96)
- [LambdaExecutionParameters](#) (p. 97)
- [LambdaFunctionRecipeSource](#) (p. 99)
- [LambdaLinuxProcessParams](#) (p. 101)
- [LambdaVolumeMount](#) (p. 102)
- [ResolvedComponentVersion](#) (p. 103)
- [ValidationExceptionField](#) (p. 105)

CloudComponentStatus

Contains the status of a component in the AWS IoT Greengrass service.

Contents

componentState

The state of the component.

Type: String

Valid Values: `REQUESTED` | `INITIATED` | `DEPLOYABLE` | `FAILED` | `DEPRECATED`

Required: No

errors

A dictionary of errors that communicate why the component is in an error state. For example, if AWS IoT Greengrass can't access an artifact for the component, then `errors` contains the artifact's URI as a key, and the error message as the value for that key.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Value Length Constraints: Minimum length of 1.

Required: No

message

A message that communicates details, such as errors, about the status of the component.

Type: String

Length Constraints: Minimum length 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Component

Contains information about a component.

Contents

arn

The [ARN](#) of the component version.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+`

Required: No

componentName

The name of the component.

Type: String

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

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

Required: No

latestVersion

The latest version of the component and its details.

Type: [ComponentLatestVersion](#) (p. 71) 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ComponentCandidate

Contains information about a component that is a candidate to deploy to a AWS IoT Greengrass core device.

Contents

componentName

The name of the component.

Type: String

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

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

Required: No

componentVersion

The version of the component.

Type: String

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

Pattern: [0-9a-zA-Z- .+]+

Required: No

versionRequirements

The version requirements for the component's dependencies. AWS IoT Greengrass core devices get the version requirements from component recipes.

AWS IoT Greengrass V2 uses semantic version constraints. For more information, see [Semantic Versioning](#).

Type: String to string map

Key Length Constraints: Minimum length of 1.

Value Length Constraints: Minimum length 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ComponentConfigurationUpdate

Contains information about a deployment's update to a component's configuration on Greengrass core devices. For more information, see [Update component configurations](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Contents

merge

A serialized JSON string that contains the configuration object to merge to target devices. The core device merges this configuration with the component's existing configuration. If this is the first time a component deploys on a device, the core device merges this configuration with the component's default configuration. This means that the core device keeps its existing configuration for keys and values that you don't specify in this object. For more information, see [Merge configuration updates](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 65536.

Required: No

reset

The list of configuration nodes to reset to default values on target devices. Use JSON pointers to specify each node to reset. JSON pointers start with a forward slash (/) and use forward slashes to separate the key for each level in the object. For more information, see the [JSON pointer specification](#) and [Reset configuration updates](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: Array of strings

Length Constraints: Minimum length of 0. 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ComponentDependencyRequirement

Contains information about a component dependency for a Lambda function component.

Contents

dependencyType

The type of this dependency. Choose from the following options:

- `SOFT` – The component doesn't restart if the dependency changes state.
- `HARD` – The component restarts if the dependency changes state.

Default: `HARD`

Type: String

Valid Values: `HARD` | `SOFT`

Required: No

versionRequirement

The component version requirement for the component dependency.

AWS IoT Greengrass V2 uses semantic version constraints. For more information, see [Semantic Versioning](#).

Type: String

Length Constraints: Minimum length 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ComponentDeploymentSpecification

Contains information about a component to deploy.

Contents

componentVersion

The version of the component.

Type: String

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

Pattern: [0-9a-zA-Z-.+]+

Required: No

configurationUpdate

The configuration updates to deploy for the component. You can define *reset* updates and *merge* updates. A reset updates the keys that you specify to the default configuration for the component. A merge updates the core device's component configuration with the keys and values that you specify. The AWS IoT Greengrass Core software applies reset updates before it applies merge updates. For more information, see [Update component configurations](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: [ComponentConfigurationUpdate \(p. 68\)](#) object

Required: No

runWith

The system user and group that the AWS IoT Greengrass Core software uses to run component processes on the core device. If you omit this parameter, the AWS IoT Greengrass Core software uses the system user and group that you configure for the core device. For more information, see [Configure the user and group that run components](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: [ComponentRunWith \(p. 74\)](#) 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ComponentLatestVersion

Contains information about the latest version of a component.

Contents

arn

The [ARN](#) of the component version.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: No

componentVersion

The version of the component.

Type: String

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

Pattern: `[0-9a-zA-Z-.+]+`

Required: No

creationTimestamp

The time at which the component was created, expressed in ISO 8601 format.

Type: Timestamp

Required: No

description

The description of the component version.

Type: String

Length Constraints: Minimum length of 1.

Required: No

platforms

The platforms that the component version supports.

Type: Array of [ComponentPlatform \(p. 73\)](#) objects

Required: No

publisher

The publisher of the component version.

Type: String

Length Constraints: Minimum length 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ComponentPlatform

Contains information about a platform that a component supports.

Contents

attributes

A dictionary of attributes for the platform. The AWS IoT Greengrass Core software defines the `os` and `platform` by default. You can specify additional platform attributes for a core device when you deploy the AWS IoT Greengrass nucleus component. For more information, see the [AWS IoT Greengrass nucleus component](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Value Length Constraints: Minimum length of 1.

Required: No

name

The friendly name of the platform. This name helps you identify the platform.

If you omit this parameter, AWS IoT Greengrass creates a friendly name from the `os` and `architecture` of the platform.

Type: String

Length Constraints: Minimum length 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ComponentRunWith

Contains information system user and group that the AWS IoT Greengrass Core software uses to run component processes on the core device. For more information, see [Configure the user and group that run components](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Contents

posixUser

The POSIX system user and (optional) group to use to run this component. Specify the user and group separated by a colon (:) in the following format: `user:group`. The group is optional. If you don't specify a group, the AWS IoT Greengrass Core software uses the primary user for the group.

Type: String

Length Constraints: Minimum length 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ComponentVersionListItem

Contains information about a component version in a list.

Contents

arn

The [ARN](#) of the component version.

Type: String

Length Constraints: Minimum length of 1.

Required: No

componentName

The name of the component.

Type: String

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

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

Required: No

componentVersion

The version of the component.

Type: String

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

Pattern: [0-9a-zA-Z- .+]+

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CoreDevice

Contains information about a AWS IoT Greengrass core device, which is an AWS IoT thing that runs the AWS IoT Greengrass Core software.

Contents

coreDeviceThingName

The name of the core device. This is also the name of the AWS IoT thing.

Type: String

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

Required: No

lastStatusUpdateTimestamp

The time at which the core device's status last updated, expressed in ISO 8601 format.

Type: Timestamp

Required: No

status

The status of the core device. Core devices can have the following statuses:

- **HEALTHY** – The AWS IoT Greengrass Core software and all components run on the core device without issue.
- **UNHEALTHY** – The AWS IoT Greengrass Core software or a component is in a failed state on the core device.

Type: String

Valid Values: **HEALTHY** | **UNHEALTHY**

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Deployment

Contains information about a deployment.

Contents

creationTimestamp

The time at which the deployment was created, expressed in ISO 8601 format.

Type: Timestamp

Required: No

deploymentId

The ID of the deployment.

Type: String

Length Constraints: Minimum length of 1.

Required: No

deploymentName

The name of the deployment.

You can create deployments without names. If you create a deployment without a name, the AWS IoT Greengrass V2 console shows the deployment name as `<targetType>:<targetName>`, where `targetType` and `targetName` are the type and name of the deployment target.

Type: String

Length Constraints: Minimum length of 1.

Required: No

deploymentStatus

The status of the deployment.

Type: String

Valid Values: `ACTIVE` | `COMPLETED` | `CANCELED` | `FAILED` | `INACTIVE`

Required: No

isLatestForTarget

Whether or not the deployment is the latest revision for its target.

Type: Boolean

Required: No

revisionId

The revision number of the deployment.

Type: String

Length Constraints: Minimum length of 1.

Required: No

targetArn

The [ARN](#) of the target AWS IoT thing or thing group.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:(thing|thinggroup)/.+`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DeploymentComponentUpdatePolicy

Contains information about a deployment's policy that defines when components are safe to update.

Each component on a device can report whether or not it's ready to update. After a component and its dependencies are ready, they can apply the update in the deployment. You can configure whether or not the deployment notifies components of an update and waits for a response. You specify the amount of time each component has to respond to the update notification.

Contents

action

Whether or not to notify components and wait for components to become safe to update. Choose from the following options:

- `NOTIFY_COMPONENTS` – The deployment notifies each component before it stops and updates that component. Components can use the [SubscribeToComponentUpdates](#) IPC operation to receive these notifications. Then, components can respond with the [DeferComponentUpdate](#) IPC operation. For more information, see [Create deployments](#) in the *AWS IoT Greengrass V2 Developer Guide*.
- `SKIP_NOTIFY_COMPONENTS` – The deployment doesn't notify components or wait for them to be safe to update.

Default: `NOTIFY_COMPONENTS`

Type: String

Valid Values: `NOTIFY_COMPONENTS` | `SKIP_NOTIFY_COMPONENTS`

Required: No

timeoutInSeconds

The amount of time in seconds that each component on a device has to report that it's safe to update. If the component waits for longer than this timeout, then the deployment proceeds on the device.

Default: 60

Type: Integer

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DeploymentConfigurationValidationPolicy

Contains information about how long a component on a core device can validate its configuration updates before it times out. Components can use the [SubscribeToValidateConfigurationUpdates](#) IPC operation to receive notifications when a deployment specifies a configuration update. Then, components can respond with the [SendConfigurationValidityReport](#) IPC operation. For more information, see [Create deployments](#) in the *AWS IoT Greengrass V2 Developer Guide*.

Contents

timeoutInSeconds

The amount of time in seconds that a component can validate its configuration updates. If the validation time exceeds this timeout, then the deployment proceeds for the device.

Default: 30

Type: Integer

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DeploymentIoTJobConfiguration

Contains information about an AWS IoT job configuration.

Contents

abortConfig

The stop configuration for the job. This configuration defines when and how to stop a job rollout.

Type: [IoTJobAbortConfig \(p. 87\)](#) object

Required: No

jobExecutionsRolloutConfig

The rollout configuration for the job. This configuration defines the rate at which the job rolls out to the fleet of target devices.

Type: [IoTJobExecutionsRolloutConfig \(p. 90\)](#) object

Required: No

timeoutConfig

The timeout configuration for the job. This configuration defines the amount of time each device has to complete the job.

Type: [IoTJobTimeoutConfig \(p. 93\)](#) 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

DeploymentPolicies

Contains information about policies that define how a deployment updates components and handles failure.

Contents

componentUpdatePolicy

The component update policy for the configuration deployment. This policy defines when it's safe to deploy the configuration to devices.

Type: [DeploymentComponentUpdatePolicy \(p. 79\)](#) object

Required: No

configurationValidationPolicy

The configuration validation policy for the configuration deployment. This policy defines how long each component has to validate its configure updates.

Type: [DeploymentConfigurationValidationPolicy \(p. 80\)](#) object

Required: No

failureHandlingPolicy

The failure handling policy for the configuration deployment. This policy defines what to do if the deployment fails.

Default: `ROLLBACK`

Type: String

Valid Values: `ROLLBACK` | `DO_NOTHING`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

EffectiveDeployment

Contains information about a deployment job that AWS IoT Greengrass sends to a AWS IoT Greengrass core device.

Contents

coreDeviceExecutionStatus

The status of the deployment job on the AWS IoT Greengrass core device.

Type: String

Valid Values: `IN_PROGRESS` | `QUEUED` | `FAILED` | `COMPLETED` | `TIMED_OUT` | `CANCELED` | `REJECTED`

Required: Yes

creationTimestamp

The time at which the deployment was created, expressed in ISO 8601 format.

Type: Timestamp

Required: Yes

deploymentId

The ID of the deployment.

Type: String

Required: Yes

deploymentName

The name of the deployment.

You can create deployments without names. If you create a deployment without a name, the AWS IoT Greengrass V2 console shows the deployment name as `<targetType>:<targetName>`, where `targetType` and `targetName` are the type and name of the deployment target.

Type: String

Required: Yes

description

The description of the deployment job.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 500.

Required: No

iotJobArn

The [ARN](#) of the AWS IoT job that applies the deployment to target devices.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:job/.+`

Required: No

iotJobId

The ID of the AWS IoT job that applies the deployment to target devices.

Type: String

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

Required: No

modifiedTimestamp

The time at which the deployment job was last modified, expressed in ISO 8601 format.

Type: Timestamp

Required: Yes

reason

The reason code for the update, if the job was updated.

Type: String

Required: No

targetArn

The [ARN](#) of the target AWS IoT thing or thing group.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:iot:[^:]+:[0-9]+:(thing|thinggroup)/.+`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

InstalledComponent

Contains information about a component on a AWS IoT Greengrass core device.

Contents

componentName

The name of the component.

Type: String

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

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

Required: No

componentVersion

The version of the component.

Type: String

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

Pattern: [0-9a-zA-Z-.]+

Required: No

isRoot

Whether or not the component is a root component.

Type: Boolean

Required: No

lifecycleState

The lifecycle state of the component.

Type: String

Valid Values: NEW | INSTALLED | STARTING | RUNNING | STOPPING | ERRORED |
BROKEN | FINISHED

Required: No

lifecycleStateDetails

The details about the lifecycle state of the component.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1000.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IoTJobAbortConfig

Contains a list of criteria that define when and how to cancel a configuration deployment.

Contents

criteriaList

The list of criteria that define when and how to cancel the configuration deployment.

Type: Array of [IoTJobAbortCriteria \(p. 88\)](#) objects

Array Members: Minimum number of 1 item.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IoTJobAbortCriteria

Contains criteria that define when and how to cancel a job.

The deployment stops if the following conditions are true:

1. The number of things that receive the deployment exceeds the `minNumberOfExecutedThings`.
2. The percentage of failures with type `failureType` exceeds the `thresholdPercentage`.

Contents

action

The action to perform when the criteria are met.

Type: String

Valid Values: CANCEL

Required: Yes

failureType

The type of job deployment failure that can cancel a job.

Type: String

Valid Values: FAILED | REJECTED | TIMED_OUT | ALL

Required: Yes

minNumberOfExecutedThings

The minimum number of things that receive the configuration before the job can cancel.

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

thresholdPercentage

The minimum percentage of `failureType` failures that occur before the job can cancel.

This parameter supports up to two digits after the decimal (for example, you can specify 10.9 or 10.99, but not 10.999).

Type: Double

Valid Range: Maximum value of 100.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IoTJobExecutionsRolloutConfig

Contains information about the rollout configuration for a job. This configuration defines the rate at which the job deploys a configuration to a fleet of target devices.

Contents

exponentialRate

The exponential rate to increase the job rollout rate.

Type: [IoTJobExponentialRolloutRate \(p. 91\)](#) object

Required: No

maximumPerMinute

The maximum number of devices that receive a pending job notification, per minute.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IoTJobExponentialRolloutRate

Contains information about an exponential rollout rate for a configuration deployment job.

Contents

baseRatePerMinute

The minimum number of devices that receive a pending job notification, per minute, when the job starts. This parameter defines the initial rollout rate of the job.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: Yes

incrementFactor

The exponential factor to increase the rollout rate for the job.

This parameter supports up to one digit after the decimal (for example, you can specify 1.5, but not 1.55).

Type: Double

Valid Range: Minimum value of 1. Maximum value of 5.

Required: Yes

rateIncreaseCriteria

The criteria to increase the rollout rate for the job.

Type: [IoTJobRateIncreaseCriteria \(p. 92\)](#) 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IoTJobRateIncreaseCriteria

Contains information about criteria to meet before a job increases its rollout rate. Specify either `numberOfNotifiedThings` or `numberOfSucceededThings`.

Contents

numberOfNotifiedThings

The number of devices to receive the job notification before the rollout rate increases.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

numberOfSucceededThings

The number of devices to successfully run the configuration job before the rollout rate increases.

Type: Integer

Valid Range: Minimum 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

IoTJobTimeoutConfig

Contains information about the timeout configuration for a job.

Contents

inProgressTimeoutInMinutes

The amount of time, in minutes, that devices have to complete the job. The timer starts when the job status is set to `IN_PROGRESS`. If the job status doesn't change to a terminal state before the time expires, then the job status is set to `TIMED_OUT`.

The timeout interval must be between 1 minute and 7 days (10080 minutes).

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LambdaContainerParams

Contains information about a container in which AWS Lambda functions run on AWS IoT Greengrass core devices.

Contents

devices

The list of system devices that the container can access.

Type: Array of [LambdaDeviceMount \(p. 95\)](#) objects

Required: No

memorySizeInKB

The memory size of the container, expressed in kilobytes.

Default: 16384 (16 MB)

Type: Integer

Required: No

mountROSysfs

Whether or not the container can read information from the device's `/sys` folder.

Default: `false`

Type: Boolean

Required: No

volumes

The list of volumes that the container can access.

Type: Array of [LambdaVolumeMount \(p. 102\)](#) 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LambdaDeviceMount

Contains information about a device that Linux processes in a container can access.

Contents

addGroupOwner

Whether or not to add the component's system user as an owner of the device.

Default: `false`

Type: Boolean

Required: No

path

The mount path for the device in the file system.

Type: String

Required: Yes

permission

The permission to access the device: read/only (`ro`) or read/write (`rw`).

Default: `ro`

Type: String

Valid Values: `ro` | `rw`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LambdaEventSource

Contains information about an event source for an AWS Lambda function. The event source defines the topics on which this Lambda function subscribes to receive messages that run the function.

Contents

topic

The topic to which to subscribe to receive event messages.

Type: String

Required: Yes

type

The type of event source. Choose from the following options:

- `PUB_SUB` – Subscribe to local publish/subscribe messages. This event source type doesn't support MQTT wildcards (+ and #) in the event source topic.
- `IOT_CORE` – Subscribe to AWS IoT Core MQTT messages. This event source type supports MQTT wildcards (+ and #) in the event source topic.

Type: String

Valid Values: `PUB_SUB` | `IOT_CORE`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LambdaExecutionParameters

Contains parameters for a Lambda function that runs on AWS IoT Greengrass.

Contents

environmentVariables

The map of environment variables that are available to the Lambda function when it runs.

Type: String to string map

Key Length Constraints: Minimum length of 1.

Required: No

eventSources

The list of event sources to which to subscribe to receive work messages. The Lambda function runs when it receives a message from an event source. You can subscribe this function to local publish/subscribe messages and AWS IoT Core MQTT messages.

Type: Array of [LambdaEventSource \(p. 96\)](#) objects

Required: No

execArgs

The list of arguments to pass to the Lambda function when it runs.

Type: Array of strings

Required: No

inputPayloadEncodingType

The encoding type that the Lambda function supports.

Default: `json`

Type: String

Valid Values: `json` | `binary`

Required: No

linuxProcessParams

The parameters for the Linux process that contains the Lambda function.

Type: [LambdaLinuxProcessParams \(p. 101\)](#) object

Required: No

maxIdleTimeInSeconds

The maximum amount of time in seconds that a non-pinned Lambda function can idle before the AWS IoT Greengrass Core software stops its process.

Type: Integer

Required: No

maxInstancesCount

The maximum number of instances that a non-pinned Lambda function can run at the same time.

Type: Integer

Required: No

maxQueueSize

The maximum size of the message queue for the Lambda function component. The AWS IoT Greengrass core stores messages in a FIFO (first-in-first-out) queue until it can run the Lambda function to consume each message.

Type: Integer

Required: No

pinned

Whether or not the Lambda function is pinned, or long-lived.

- A pinned Lambda function starts when AWS IoT Greengrass starts and keeps running in its own container.
- A non-pinned Lambda function starts only when it receives a work item and exists after it idles for `maxIdleTimeInSeconds`. If the function has multiple work items, the AWS IoT Greengrass Core software creates multiple instances of the function.

Default: `true`

Type: Boolean

Required: No

statusTimeoutInSeconds

The interval in seconds at which a pinned (also known as long-lived) Lambda function component sends status updates to the Lambda manager component.

Type: Integer

Required: No

timeoutInSeconds

The maximum amount of time in seconds that the Lambda function can process a work item.

Type: Integer

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LambdaFunctionRecipeSource

Contains information about an AWS Lambda function to import to create a component.

Contents

componentDependencies

The component versions on which this Lambda function component depends.

Type: String to [ComponentDependencyRequirement \(p. 69\)](#) object map

Key Length Constraints: Minimum length of 1.

Required: No

componentLambdaParameters

The system and runtime parameters for the Lambda function as it runs on the AWS IoT Greengrass core device.

Type: [LambdaExecutionParameters \(p. 97\)](#) object

Required: No

componentName

The name of the component.

Defaults to the name of the Lambda function.

Type: String

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

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

Required: No

componentPlatforms

The platforms that the component version supports.

Type: Array of [ComponentPlatform \(p. 73\)](#) objects

Required: No

componentVersion

The version of the component.

Defaults to the version of the Lambda function as a semantic version. For example, if your function version is 3, the component version becomes 3.0.0.

Type: String

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

Pattern: [0-9a-zA-Z-.]+

Required: No

lambdaArn

The [ARN](#) of the Lambda function. The ARN must include the version of the function to import. You can't use version aliases like `$LATEST`.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:lambda:[^:]+:[0-9]+:function:[a-zA-Z0-9-_]+:[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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LambdaLinuxProcessParams

Contains parameters for a Linux process that contains an AWS Lambda function.

Contents

containerParams

The parameters for the container in which the Lambda function runs.

Type: [LambdaContainerParams \(p. 94\)](#) object

Required: No

isolationMode

The isolation mode for the process that contains the Lambda function. The process can run in an isolated runtime environment inside the AWS IoT Greengrass container, or as a regular process outside any container.

Default: `GreengrassContainer`

Type: String

Valid Values: `GreengrassContainer` | `NoContainer`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

LambdaVolumeMount

Contains information about a volume that Linux processes in a container can access. When you define a volume, the AWS IoT Greengrass Core software mounts the source files to the destination inside the container.

Contents

addGroupOwner

Whether or not to add the AWS IoT Greengrass user group as an owner of the volume.

Default: `false`

Type: Boolean

Required: No

destinationPath

The path to the logical volume in the file system.

Type: String

Required: Yes

permission

The permission to access the volume: read/only (`ro`) or read/write (`rw`).

Default: `ro`

Type: String

Valid Values: `ro` | `rw`

Required: No

sourcePath

The path to the physical volume in the file system.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ResolvedComponentVersion

Contains information about a component version that is compatible to run on a AWS IoT Greengrass core device.

Contents

arn

The [ARN](#) of the component version.

Type: String

Pattern: `arn:aws(-cn|-us-gov)?:greengrass:[^:]+:(aws|[0-9]+):components:[^:]+:versions:[^:]+`

Required: No

componentName

The name of the component.

Type: String

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

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

Required: No

componentVersion

The version of the component.

Type: String

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

Pattern: `[0-9a-zA-Z-.]+`

Required: No

recipe

The recipe of the component version.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ValidationExceptionField

Contains information about a validation exception field.

Contents

message

The message of the exception field.

Type: String

Required: Yes

name

The name of the exception field.

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 Go](#)
- [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 [Signature Version 4 Signing Process](#) in the *Amazon Web Services General Reference*.

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 [Task 2: Create a String to Sign for Signature Version 4](#) in the *Amazon Web Services General Reference*.

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'THHMMSS'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 [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.

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 Security Token Service, go to [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, 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 [Task 1: Create a Canonical Request For Signature Version 4](#) in the *Amazon Web Services General Reference*.

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

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

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