

---

**AWS IoT**  
**API Reference**  
**Version**



## **AWS IoT: API Reference**

Copyright © 2016 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
AcceptCertificateTransfer .....	4
Request Syntax .....	4
Request Parameters .....	4
Response Elements .....	4
Errors .....	4
AttachPrincipalPolicy .....	6
Request Syntax .....	6
Request Parameters .....	6
Response Elements .....	6
Errors .....	6
AttachThingPrincipal .....	8
Request Syntax .....	8
Request Parameters .....	8
Response Elements .....	8
Errors .....	8
CancelCertificateTransfer .....	10
Request Syntax .....	10
Request Parameters .....	10
Response Elements .....	10
Errors .....	10
CreateCertificateFromCsr .....	12
Request Syntax .....	12
Request Parameters .....	12
Response Syntax .....	13
Response Elements .....	13
Errors .....	13
CreateKeysAndCertificate .....	15
Request Syntax .....	15
Request Parameters .....	15
Response Syntax .....	15
Response Elements .....	15
Errors .....	16
CreatePolicy .....	17
Request Syntax .....	17
Request Parameters .....	17
Response Syntax .....	17
Response Elements .....	17
Errors .....	18
CreatePolicyVersion .....	19
Request Syntax .....	19
Request Parameters .....	19
Response Syntax .....	19
Response Elements .....	20
Errors .....	20
CreateThing .....	22
Request Syntax .....	22
Request Parameters .....	22
Response Syntax .....	23
Response Elements .....	23
Errors .....	23
CreateThingType .....	25
Request Syntax .....	25
Request Parameters .....	25

Response Syntax .....	25
Response Elements .....	25
Errors .....	26
CreateTopicRule .....	27
Request Syntax .....	27
Request Parameters .....	28
Response Elements .....	29
Errors .....	29
DeleteCACertificate .....	30
Request Syntax .....	30
Request Parameters .....	30
Response Elements .....	30
Errors .....	30
DeleteCertificate .....	32
Request Syntax .....	32
Request Parameters .....	32
Response Elements .....	32
Errors .....	32
DeletePolicy .....	34
Request Syntax .....	34
Request Parameters .....	34
Response Elements .....	34
Errors .....	34
DeletePolicyVersion .....	36
Request Syntax .....	36
Request Parameters .....	36
Response Elements .....	36
Errors .....	36
DeleteRegistrationCode .....	38
Response Elements .....	38
Errors .....	38
DeleteThing .....	39
Request Syntax .....	39
Request Parameters .....	39
Response Elements .....	39
Errors .....	39
DeleteThingType .....	41
Request Syntax .....	41
Request Parameters .....	41
Response Elements .....	41
Errors .....	41
DeleteTopicRule .....	43
Request Syntax .....	43
Request Parameters .....	43
Response Elements .....	43
Errors .....	43
DeprecateThingType .....	44
Request Syntax .....	44
Request Parameters .....	44
Response Elements .....	44
Errors .....	44
DescribeCACertificate .....	46
Request Syntax .....	46
Request Parameters .....	46
Response Syntax .....	46
Response Elements .....	46
Errors .....	47
DescribeCertificate .....	48

Request Syntax .....	48
Request Parameters .....	48
Response Syntax .....	48
Response Elements .....	49
Errors .....	49
DescribeEndpoint .....	50
Response Syntax .....	50
Response Elements .....	50
Errors .....	50
DescribeThing .....	51
Request Syntax .....	51
Request Parameters .....	51
Response Syntax .....	51
Response Elements .....	51
Errors .....	52
DescribeThingType .....	54
Request Syntax .....	54
Request Parameters .....	54
Response Syntax .....	54
Response Elements .....	54
Errors .....	55
DetachPrincipalPolicy .....	56
Request Syntax .....	56
Request Parameters .....	56
Response Elements .....	56
Errors .....	56
DetachThingPrincipal .....	58
Request Syntax .....	58
Request Parameters .....	58
Response Elements .....	58
Errors .....	58
DisableTopicRule .....	60
Request Syntax .....	60
Request Parameters .....	60
Response Elements .....	60
Errors .....	60
EnableTopicRule .....	61
Request Syntax .....	61
Request Parameters .....	61
Response Elements .....	61
Errors .....	61
GetLoggingOptions .....	62
Response Syntax .....	62
Response Elements .....	62
Errors .....	62
GetPolicy .....	63
Request Syntax .....	63
Request Parameters .....	63
Response Syntax .....	63
Response Elements .....	63
Errors .....	64
GetPolicyVersion .....	65
Request Syntax .....	65
Request Parameters .....	65
Response Syntax .....	65
Response Elements .....	65
Errors .....	66
GetRegistrationCode .....	67

Response Syntax .....	67
Response Elements .....	67
Errors .....	67
GetTopicRule .....	68
Request Syntax .....	68
Request Parameters .....	68
Response Syntax .....	68
Response Elements .....	70
Errors .....	70
ListCACertificates .....	71
Request Syntax .....	71
Request Parameters .....	71
Response Syntax .....	71
Response Elements .....	72
Errors .....	72
ListCertificates .....	73
Request Syntax .....	73
Request Parameters .....	73
Response Syntax .....	73
Response Elements .....	74
Errors .....	74
ListCertificatesByCA .....	75
Request Syntax .....	75
Request Parameters .....	75
Response Syntax .....	76
Response Elements .....	76
Errors .....	76
ListOutgoingCertificates .....	77
Request Syntax .....	77
Request Parameters .....	77
Response Syntax .....	77
Response Elements .....	78
Errors .....	78
ListPolicies .....	79
Request Syntax .....	79
Request Parameters .....	79
Response Syntax .....	79
Response Elements .....	80
Errors .....	80
ListPolicyPrincipals .....	81
Request Syntax .....	81
Request Parameters .....	81
Response Syntax .....	81
Response Elements .....	82
Errors .....	82
ListPolicyVersions .....	83
Request Syntax .....	83
Request Parameters .....	83
Response Syntax .....	83
Response Elements .....	83
Errors .....	84
ListPrincipalPolicies .....	85
Request Syntax .....	85
Request Parameters .....	85
Response Syntax .....	85
Response Elements .....	86
Errors .....	86
ListPrincipalThings .....	87

Request Syntax .....	87
Request Parameters .....	87
Response Syntax .....	87
Response Elements .....	87
Errors .....	88
ListThingPrincipals .....	89
Request Syntax .....	89
Request Parameters .....	89
Response Syntax .....	89
Response Elements .....	89
Errors .....	89
ListThings .....	91
Request Syntax .....	91
Request Parameters .....	91
Response Syntax .....	92
Response Elements .....	92
Errors .....	92
ListThingTypes .....	94
Request Syntax .....	94
Request Parameters .....	94
Response Syntax .....	94
Response Elements .....	95
Errors .....	95
ListTopicRules .....	96
Request Syntax .....	96
Request Parameters .....	96
Response Syntax .....	96
Response Elements .....	97
Errors .....	97
RegisterCACertificate .....	98
Request Syntax .....	98
Request Parameters .....	98
Response Syntax .....	99
Response Elements .....	99
Errors .....	99
RegisterCertificate .....	101
Request Syntax .....	101
Request Parameters .....	101
Response Syntax .....	102
Response Elements .....	102
Errors .....	102
RejectCertificateTransfer .....	104
Request Syntax .....	104
Request Parameters .....	104
Response Elements .....	104
Errors .....	104
ReplaceTopicRule .....	106
Request Syntax .....	106
Request Parameters .....	107
Response Elements .....	108
Errors .....	108
SetDefaultPolicyVersion .....	109
Request Syntax .....	109
Request Parameters .....	109
Response Elements .....	109
Errors .....	109
SetLoggingOptions .....	111
Request Syntax .....	111

Request Parameters .....	111
Response Elements .....	111
Errors .....	111
TransferCertificate .....	112
Request Syntax .....	112
Request Parameters .....	112
Response Syntax .....	113
Response Elements .....	113
Errors .....	113
UpdateCACertificate .....	115
Request Syntax .....	115
Request Parameters .....	115
Response Elements .....	115
Errors .....	116
UpdateCertificate .....	117
Request Syntax .....	117
Request Parameters .....	117
Response Elements .....	117
Errors .....	118
UpdateThing .....	119
Request Syntax .....	119
Request Parameters .....	119
Response Elements .....	120
Errors .....	120
Data Types .....	121
Action .....	122
Description .....	122
Contents .....	122
AttributePayload .....	123
Description .....	123
Contents .....	123
CACertificate .....	124
Description .....	124
Contents .....	124
CACertificateDescription .....	124
Description .....	124
Contents .....	125
Certificate .....	126
Description .....	126
Contents .....	126
CertificateDescription .....	126
Description .....	126
Contents .....	126
CloudwatchAlarmAction .....	128
Description .....	128
Contents .....	128
CloudwatchMetricAction .....	128
Description .....	128
Contents .....	129
DynamoDBAction .....	129
Description .....	129
Contents .....	130
DynamoDBv2Action .....	131
Description .....	131
Contents .....	131
ElasticsearchAction .....	131
Description .....	131
Contents .....	131



FirehoseAction .....	132
Description .....	132
Contents .....	132
KeyPair .....	133
Description .....	133
Contents .....	133
KinesisAction .....	133
Description .....	133
Contents .....	133
LambdaAction .....	134
Description .....	134
Contents .....	134
LoggingOptionsPayload .....	134
Description .....	134
Contents .....	134
OutgoingCertificate .....	134
Description .....	134
Contents .....	134
Policy .....	135
Description .....	135
Contents .....	135
PolicyVersion .....	136
Description .....	136
Contents .....	136
PutItemInput .....	136
Description .....	136
Contents .....	136
RepublishAction .....	137
Description .....	137
Contents .....	137
S3Action .....	137
Description .....	137
Contents .....	137
SnsAction .....	138
Description .....	138
Contents .....	138
SqsAction .....	138
Description .....	138
Contents .....	138
ThingAttribute .....	139
Description .....	139
Contents .....	139
ThingTypeDefintion .....	140
Description .....	140
Contents .....	140
ThingTypeMetadata .....	140
Description .....	140
Contents .....	140
ThingTypeProperties .....	141
Description .....	141
Contents .....	141
TopicRule .....	141
Description .....	141
Contents .....	141
TopicRuleListItem .....	142
Description .....	142
Contents .....	142
TopicRulePayload .....	143

Description .....	143
Contents .....	143
TransferData .....	144
Description .....	144
Contents .....	144
Common Parameters .....	145
.....	145

# Welcome

---

AWS IoT provides secure, bi-directional communication between Internet-connected things (such as sensors, actuators, embedded devices, or smart appliances) and the AWS cloud. You can discover your custom IoT-Data endpoint to communicate with, configure rules for data processing and integration with other services, organize resources associated with each thing (Thing Registry), configure logging, and create and manage policies and credentials to authenticate things.

For more information about how AWS IoT works, see the [Developer Guide](#).

This document was last updated on December 9, 2016.

# Actions

---

The following actions are supported:

- [AcceptCertificateTransfer](#) (p. 4)
- [AttachPrincipalPolicy](#) (p. 6)
- [AttachThingPrincipal](#) (p. 8)
- [CancelCertificateTransfer](#) (p. 10)
- [CreateCertificateFromCsr](#) (p. 12)
- [CreateKeysAndCertificate](#) (p. 15)
- [CreatePolicy](#) (p. 17)
- [CreatePolicyVersion](#) (p. 19)
- [CreateThing](#) (p. 22)
- [CreateThingType](#) (p. 25)
- [CreateTopicRule](#) (p. 27)
- [DeleteCACertificate](#) (p. 30)
- [DeleteCertificate](#) (p. 32)
- [DeletePolicy](#) (p. 34)
- [DeletePolicyVersion](#) (p. 36)
- [DeleteRegistrationCode](#) (p. 38)
- [DeleteThing](#) (p. 39)
- [DeleteThingType](#) (p. 41)
- [DeleteTopicRule](#) (p. 43)
- [DeprecateThingType](#) (p. 44)
- [DescribeCACertificate](#) (p. 46)
- [DescribeCertificate](#) (p. 48)
- [DescribeEndpoint](#) (p. 50)
- [DescribeThing](#) (p. 51)
- [DescribeThingType](#) (p. 54)
- [DetachPrincipalPolicy](#) (p. 56)
- [DetachThingPrincipal](#) (p. 58)
- [DisableTopicRule](#) (p. 60)
- [EnableTopicRule](#) (p. 61)
- [GetLoggingOptions](#) (p. 62)

- [GetPolicy](#) (p. 63)
- [GetPolicyVersion](#) (p. 65)
- [GetRegistrationCode](#) (p. 67)
- [GetTopicRule](#) (p. 68)
- [ListCACertificates](#) (p. 71)
- [ListCertificates](#) (p. 73)
- [ListCertificatesByCA](#) (p. 75)
- [ListOutgoingCertificates](#) (p. 77)
- [ListPolicies](#) (p. 79)
- [ListPolicyPrincipals](#) (p. 81)
- [ListPolicyVersions](#) (p. 83)
- [ListPrincipalPolicies](#) (p. 85)
- [ListPrincipalThings](#) (p. 87)
- [ListThingPrincipals](#) (p. 89)
- [ListThings](#) (p. 91)
- [ListThingTypes](#) (p. 94)
- [ListTopicRules](#) (p. 96)
- [RegisterCACertificate](#) (p. 98)
- [RegisterCertificate](#) (p. 101)
- [RejectCertificateTransfer](#) (p. 104)
- [ReplaceTopicRule](#) (p. 106)
- [SetDefaultPolicyVersion](#) (p. 109)
- [SetLoggingOptions](#) (p. 111)
- [TransferCertificate](#) (p. 112)
- [UpdateCACertificate](#) (p. 115)
- [UpdateCertificate](#) (p. 117)
- [UpdateThing](#) (p. 119)

# AcceptCertificateTransfer

Accepts a pending certificate transfer. The default state of the certificate is INACTIVE.

To check for pending certificate transfers, call [ListCertificates \(p. 73\)](#) to enumerate your certificates.

## Request Syntax

```
{  
  "certificateId": "string",  
  "setAsActive": boolean  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **certificateId**

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

### **setAsActive**

Specifies whether the certificate is active.

Type: Boolean

Required: No

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**TransferAlreadyCompletedException**

You can't revert the certificate transfer because the transfer is already complete.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# AttachPrincipalPolicy

Attaches the specified policy to the specified principal (certificate or other credential).

## Request Syntax

```
{  
  "policyName": "string",  
  "principal": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **policyName**

The policy name.

Type: String

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

Pattern: [*\w+=, .@-]*+

Required: Yes

### **principal**

The principal, which can be a certificate ARN (as returned from the CreateCertificate operation) or an Amazon Cognito ID.

Type: String

Required: Yes

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **LimitExceededException**

The number of attached entities exceeds the limit.



HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# AttachThingPrincipal

Attaches the specified principal to the specified thing.

## Request Syntax

```
{  
  "principal": "string",  
  "thingName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **principal**

The principal, such as a certificate or other credential.

Type: String

Required: Yes

### **thingName**

The name of the thing.

Type: String

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

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

Required: Yes

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# CancelCertificateTransfer

Cancels a pending transfer for the specified certificate.

**Note** Only the transfer source account can use this operation to cancel a transfer. (Transfer destinations can use [RejectCertificateTransfer](#) (p. 104) instead.) After transfer, AWS IoT returns the certificate to the source account in the INACTIVE state. After the destination account has accepted the transfer, the transfer cannot be cancelled.

After a certificate transfer is cancelled, the status of the certificate changes from PENDING\_TRANSFER to INACTIVE.

## Request Syntax

```
{  
  "certificateId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **certificateId**

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**TransferAlreadyCompletedException**

You can't revert the certificate transfer because the transfer is already complete.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

## CreateCertificateFromCsr

Creates an X.509 certificate using the specified certificate signing request.

**Note** Reusing the same certificate signing request (CSR) results in a distinct certificate.

You can create multiple certificates in a batch by creating a directory, copying multiple .csr files into that directory, and then specifying that directory on the command line. The following commands show how to create a batch of certificates given a batch of CSRs.

Assuming a set of CSRs are located inside of the directory my-csr-directory:

On Linux and OS X, the command is:

```
$ ls my-csr-directory/ | xargs -I {} aws iot create-certificate-from-csr --certificate-signing-request file://my-csr-directory/{}
```

This command lists all of the CSRs in my-csr-directory and pipes each CSR file name to the aws iot create-certificate-from-csr AWS CLI command to create a certificate for the corresponding CSR.

The aws iot create-certificate-from-csr part of the command can also be run in parallel to speed up the certificate creation process:

```
$ ls my-csr-directory/ | xargs -P 10 -I {} aws iot create-certificate-from-csr --certificate-signing-request file://my-csr-directory/{}
```

On Windows PowerShell, the command to create certificates for all CSRs in my-csr-directory is:

```
> ls -Name my-csr-directory | %{aws iot create-certificate-from-csr --certificate-signing-request file://my-csr-directory/$_}
```

On a Windows command prompt, the command to create certificates for all CSRs in my-csr-directory is:

```
> forfiles /p my-csr-directory /c "cmd /c aws iot create-certificate-from-csr --certificate-signing-request file://@path"
```

## Request Syntax

```
{  
  "certificateSigningRequest": "string",  
  "setAsActive": boolean  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### certificateSigningRequest

The certificate signing request (CSR).

Type: String

Length constraints: Minimum length of 1.

Required: Yes

#### **setAsActive**

Specifies whether the certificate is active.

Type: Boolean

Required: No

## Response Syntax

```
{  
  "certificateArn": "string",  
  "certificateId": "string",  
  "certificatePem": "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.

#### **certificateArn**

The Amazon Resource Name (ARN) of the certificate. You can use the ARN as a principal for policy operations.

Type: String

#### **certificateId**

The ID of the certificate. Certificate management operations only take a certificateId.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

#### **certificatePem**

The certificate data, in PEM format.

Type: String

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

## Errors

#### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

#### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400



## CreateKeysAndCertificate

Creates a 2048-bit RSA key pair and issues an X.509 certificate using the issued public key.

**Note** This is the only time AWS IoT issues the private key for this certificate, so it is important to keep it in a secure location.

### Request Syntax

```
{  
  "setAsActive": boolean  
}
```

### Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

#### setAsActive

Specifies whether the certificate is active.

Type: Boolean

Required: No

### Response Syntax

```
{  
  "certificateArn": "string",  
  "certificateId": "string",  
  "certificatePem": "string",  
  "keyPair": {  
    "PrivateKey": "string",  
    "PublicKey": "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.

#### certificateArn

The ARN of the certificate.

Type: String

### **certificateId**

The ID of the certificate. AWS IoT issues a default subject name for the certificate (for example, AWS IoT Certificate).

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

### **certificatePem**

The certificate data, in PEM format.

Type: String

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

### **keyPair**

The generated key pair.

Type: [KeyPair](#) (p. 133) object

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# CreatePolicy

Creates an AWS IoT policy.

The created policy is the default version for the policy. This operation creates a policy version with a version identifier of **1** and sets **1** as the policy's default version.

## Request Syntax

```
{  
  "policyDocument": "string",  
  "policyName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### policyDocument

The JSON document that describes the policy. **policyDocument** must have a minimum length of 1, with a maximum length of 2048, excluding whitespace.

Type: String

Required: Yes

### policyName

The policy name.

Type: String

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

Pattern: [\w+=, .@- ]+

Required: Yes

## Response Syntax

```
{  
  "policyArn": "string",  
  "policyDocument": "string",  
  "policyName": "string",  
  "policyVersionId": "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.

**policyArn**

The policy ARN.

Type: String

**policyDocument**

The JSON document that describes the policy.

Type: String

**policyName**

The policy name.

Type: String

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

Pattern: [ \w+=, .@- ]+

**policyVersionId**

The policy version ID.

Type: String

Pattern: [ 0-9 ]+

## Errors

**InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**MalformedPolicyException**

The policy documentation is not valid.

HTTP Status Code: 400

**ResourceAlreadyExistsException**

The resource already exists.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# CreatePolicyVersion

Creates a new version of the specified AWS IoT policy. To update a policy, create a new policy version. A managed policy can have up to five versions. If the policy has five versions, you must use [DeletePolicyVersion](#) (p. 36) to delete an existing version before you create a new one.

Optionally, you can set the new version as the policy's default version. The default version is the operative version (that is, the version that is in effect for the certificates to which the policy is attached).

## Request Syntax

```
{
  "policyDocument": "string",
  "policyName": "string",
  "setAsDefault": boolean
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### policyDocument

The JSON document that describes the policy. Minimum length of 1. Maximum length of 2048, excluding whitespaces

Type: String

Required: Yes

### policyName

The policy name.

Type: String

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

Pattern: [w+=, .@- ]+

Required: Yes

### setAsDefault

Specifies whether the policy version is set as the default. When this parameter is true, the new policy version becomes the operative version (that is, the version that is in effect for the certificates to which the policy is attached).

Type: Boolean

Required: No

## Response Syntax

```
{  
  "isDefaultVersion": boolean,  
  "policyArn": "string",  
  "policyDocument": "string",  
  "policyVersionId": "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.

### **isDefaultVersion**

Specifies whether the policy version is the default.

Type: Boolean

### **policyArn**

The policy ARN.

Type: String

### **policyDocument**

The JSON document that describes the policy.

Type: String

### **policyVersionId**

The policy version ID.

Type: String

Pattern: [0-9]+

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **MalformedPolicyException**

The policy documentation is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

**VersionsLimitExceededException**

The number of policy versions exceeds the limit.

HTTP Status Code: 400

# CreateThing

Creates a thing record in the thing registry.

## Request Syntax

```
{
  "attributePayload": {
    "attributes": {
      "string": "string"
    },
    "merge": boolean
  },
  "thingName": "string",
  "thingTypeName": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### attributePayload

The attribute payload, which consists of up to three name/value pairs in a JSON document. For example:

```
{\"attributes\":{\"string1\":\"string2\"}}
```

Type: [AttributePayload](#) (p. 123) object

Required: No

### thingName

The name of the thing to create.

Type: String

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

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

Required: Yes

### thingTypeName

The name of the thing type associated with the new thing.

Type: String

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

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



Required: No

## Response Syntax

```
{  
  "thingArn": "string",  
  "thingName": "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.

### **thingArn**

The ARN of the new thing.

Type: String

### **thingName**

The name of the new thing.

Type: String

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

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceAlreadyExistsException**

The resource already exists.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# CreateThingType

Creates a new thing type.

## Request Syntax

```
{
  "thingTypeName": "string",
  "thingTypeProperties": {
    "searchableAttributes": [
      "string"
    ],
    "thingTypeDescription": "string"
  }
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **thingTypeName**

The name of the thing type.

Type: String

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

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

Required: Yes

### **thingTypeProperties**

The ThingTypeProperties for the thing type to create. It contains information about the new thing type including a description, and a list of searchable thing attribute names.

Type: [ThingTypeProperties](#) (p. 141) object

Required: No

## Response Syntax

```
{
  "thingTypeArn": "string",
  "thingTypeName": "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.

**thingTypeArn**

The Amazon Resource Name (ARN) of the thing type.

Type: String

**thingTypeName**

The name of the thing type.

Type: String

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

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

## Errors

**InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceAlreadyExistsException**

The resource already exists.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# CreateTopicRule

Creates a rule. Creating rules is an administrator-level action. Any user who has permission to create rules will be able to access data processed by the rule.

## Request Syntax

```
{
  "ruleName": "string",
  "topicRulePayload": {
    "actions": [
      {
        "cloudwatchAlarm": {
          "alarmName": "string",
          "roleArn": "string",
          "stateReason": "string",
          "stateValue": "string"
        },
        "cloudwatchMetric": {
          "metricName": "string",
          "metricNamespace": "string",
          "metricTimestamp": "string",
          "metricUnit": "string",
          "metricValue": "string",
          "roleArn": "string"
        },
        "dynamoDB": {
          "hashKeyField": "string",
          "hashKeyType": "string",
          "hashKeyValue": "string",
          "operation": "string",
          "payloadField": "string",
          "rangeKeyField": "string",
          "rangeKeyType": "string",
          "rangeKeyValue": "string",
          "roleArn": "string",
          "tableName": "string"
        },
        "dynamoDBv2": {
          "putItem": {
            "tableName": "string"
          },
          "roleArn": "string"
        },
        "elasticsearch": {
          "endpoint": "string",
          "id": "string",
          "index": "string",
          "roleArn": "string",
          "type": "string"
        },
        "firehose": {
          "deliveryStreamName": "string",
          "roleArn": "string",
          "separator": "string"
        }
      ]
    }
  }
}
```

```
    "kinesis": {
      "partitionKey": "string",
      "roleArn": "string",
      "streamName": "string"
    },
    "lambda": {
      "functionArn": "string"
    },
    "republish": {
      "roleArn": "string",
      "topic": "string"
    },
    "s3": {
      "bucketName": "string",
      "cannedAcl": "string",
      "key": "string",
      "roleArn": "string"
    },
    "sns": {
      "messageFormat": "string",
      "roleArn": "string",
      "targetArn": "string"
    },
    "sqs": {
      "queueUrl": "string",
      "roleArn": "string",
      "useBase64": boolean
    }
  },
  "awsIotSqlVersion": "string",
  "description": "string",
  "ruleDisabled": boolean,
  "sql": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **ruleName**

The name of the rule.

Type: String

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

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

Required: Yes

### **topicRulePayload**

The rule payload.

Type: [TopicRulePayload \(p. 143\)](#) object

Required: Yes

## Response Elements

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

## Errors

### **InternalException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceAlreadyExistsException**

The resource already exists.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **SqlParseException**

The Rule-SQL expression can't be parsed correctly.

HTTP Status Code: 400

# DeleteCACertificate

Deletes a registered CA certificate.

## Request Syntax

```
{  
  "certificateId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **certificateId**

The ID of the certificate to delete.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

## Response Elements

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

## Errors

### **CertificateStateException**

The certificate operation is not allowed.

HTTP Status Code: 400

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.



HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DeleteCertificate

Deletes the specified certificate.

A certificate cannot be deleted if it has a policy attached to it or if its status is set to ACTIVE. To delete a certificate, first use the [DetachPrincipalPolicy \(p. 56\)](#) API to detach all policies. Next, use the [UpdateCertificate \(p. 117\)](#) API to set the certificate to the INACTIVE status.

## Request Syntax

```
{  
  "certificateId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **certificateId**

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

## Response Elements

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

## Errors

### **CertificateStateException**

The certificate operation is not allowed.

HTTP Status Code: 400

### **DeleteConflictException**

You can't delete the resource because it is attached to one or more resources.

HTTP Status Code: 400

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

## DeletePolicy

Deletes the specified policy.

A policy cannot be deleted if it has non-default versions or it is attached to any certificate.

To delete a policy, use the DeletePolicyVersion API to delete all non-default versions of the policy; use the DetachPrincipalPolicy API to detach the policy from any certificate; and then use the DeletePolicy API to delete the policy.

When a policy is deleted using DeletePolicy, its default version is deleted with it.

## Request Syntax

```
{  
  "policyName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **policyName**

The name of the policy to delete.

Type: String

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

Pattern: [`\w+=, .@-` ]+

Required: Yes

## Response Elements

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

## Errors

### **DeleteConflictException**

You can't delete the resource because it is attached to one or more resources.

HTTP Status Code: 400

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

## DeletePolicyVersion

Deletes the specified version of the specified policy. You cannot delete the default version of a policy using this API. To delete the default version of a policy, use [DeletePolicy \(p. 34\)](#). To find out which version of a policy is marked as the default version, use [ListPolicyVersions](#).

### Request Syntax

```
{  
  "policyName": "string",  
  "policyVersionId": "string"  
}
```

### Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

#### **policyName**

The name of the policy.

Type: String

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

Pattern: [*\w+=, .@-*]+

Required: Yes

#### **policyVersionId**

The policy version ID.

Type: String

Pattern: [*0-9*]+

Required: Yes

### Response Elements

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

### Errors

#### **DeleteConflictException**

You can't delete the resource because it is attached to one or more resources.

HTTP Status Code: 400

#### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DeleteRegistrationCode

Deletes a CA certificate registration code.

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400



# DeleteThing

Deletes the specified thing.

## Request Syntax

```
{  
  "expectedVersion": number,  
  "thingName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **expectedVersion**

The expected version of the thing record in the registry. If the version of the record in the registry does not match the expected version specified in the request, the `DeleteThing` request is rejected with a `VersionConflictException`.

Type: Long

Required: No

### **thingName**

The name of the thing to delete.

Type: String

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

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

Required: Yes

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

**VersionConflictException**

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

HTTP Status Code: 400

## DeleteThingType

Deletes the specified thing type . You cannot delete a thing type if it has things associated with it. To delete a thing type, first mark it as deprecated by calling [DeprecateThingType \(p. 44\)](#), then remove any associated things by calling [UpdateThing \(p. 119\)](#) to change the thing type on any associated thing, and finally use [DeleteThingType \(p. 41\)](#) to delete the thing type.

### Request Syntax

```
{  
  "thingTypeName": "string"  
}
```

### Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

#### **thingTypeName**

The name of the thing type.

Type: String

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

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

Required: Yes

### Response Elements

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

### Errors

#### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

#### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

#### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

#### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DeleteTopicRule

Deletes the specified rule.

## Request Syntax

```
{  
  "ruleName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **ruleName**

The name of the rule.

Type: String

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

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

Required: No

## Response Elements

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

## Errors

### **InternalException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DeprecateThingType

Deprecates a thing type. You can not associate new things with deprecated thing type.

## Request Syntax

```
{  
  "thingTypeName": "string",  
  "undoDeprecate": boolean  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **thingTypeName**

The name of the thing type to deprecate.

Type: String

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

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

Required: Yes

### **undoDeprecate**

Whether to undeprecate a deprecated thing type. If **true**, the thing type will not be deprecated anymore and you can associate it with things.

Type: Boolean

Required: No

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DescribeCACertificate

Describes a registered CA certificate.

## Request Syntax

```
{  
  "certificateId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **certificateId**

The CA certificate identifier.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

## Response Syntax

```
{  
  "certificateDescription": {  
    "autoRegistrationStatus": "string",  
    "certificateArn": "string",  
    "certificateId": "string",  
    "certificatePem": "string",  
    "creationDate": number,  
    "ownedBy": "string",  
    "status": "string"  
  }  
}
```

## Response Elements

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

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

### **certificateDescription**

The CA certificate description.



Type: [CACertificateDescription](#) (p. 124) object

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DescribeCertificate

Gets information about the specified certificate.

## Request Syntax

```
{  
  "certificateId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **certificateId**

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

## Response Syntax

```
{  
  "certificateDescription": {  
    "caCertificateId": "string",  
    "certificateArn": "string",  
    "certificateId": "string",  
    "certificatePem": "string",  
    "creationDate": number,  
    "lastModifiedDate": number,  
    "ownedBy": "string",  
    "previousOwnedBy": "string",  
    "status": "string",  
    "transferData": {  
      "acceptDate": number,  
      "rejectDate": number,  
      "rejectReason": "string",  
      "transferDate": number,  
      "transferMessage": "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.

### **certificateDescription**

The description of the certificate.

Type: [CertificateDescription](#) (p. 126) object

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DescribeEndpoint

Returns a unique endpoint specific to the AWS account making the call.

## Response Syntax

```
{  
  "endpointAddress": "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.

### **endpointAddress**

The endpoint. The format of the endpoint is as follows: *identifier.iot.region.amazonaws.com*.

Type: String

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DescribeThing

Gets information about the specified thing.

## Request Syntax

```
{  
  "thingName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **thingName**

The name of the thing.

Type: String

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

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

Required: Yes

## Response Syntax

```
{  
  "attributes":  
    {  
      "string":  
        "string"  
    },  
  "defaultClientId": "string",  
  "thingName": "string",  
  "thingTypeName": "string",  
  "version": number  
}
```

## Response Elements

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

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

### **attributes**

The thing attributes.

Type: String to String map

**defaultClientId**

The default client ID.

Type: String

**thingName**

The name of the thing.

Type: String

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

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

**thingTypeName**

The thing type name.

Type: String

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

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

**version**

The current version of the thing record in the registry.

**Note**

To avoid unintentional changes to the information in the registry, you can pass the version information in the `expectedVersion` parameter of the `UpdateThing` and `DeleteThing` calls.

Type: Long

## Errors

**InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400



## DescribeThingType

Gets information about the specified thing type.

### Request Syntax

```
{  
  "thingTypeName": "string"  
}
```

### Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

#### **thingTypeName**

The name of the thing type.

Type: String

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

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

Required: Yes

### Response Syntax

```
{  
  "thingTypeMetadata": {  
    "creationDate": number,  
    "deprecated": boolean,  
    "deprecationDate": number  
  },  
  "thingTypeName": "string",  
  "thingTypeProperties": {  
    "searchableAttributes": [  
      "string"  
    ],  
    "thingTypeDescription": "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.



### **thingTypeMetadata**

The ThingTypeMetadata contains additional information about the thing type including: creation date and time, a value indicating whether the thing type is deprecated, and a date and time when time was deprecated.

Type: [ThingTypeMetadata \(p. 140\)](#) object

### **thingTypeName**

The name of the thing type.

Type: String

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

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

### **thingTypeProperties**

The ThingTypeProperties contains information about the thing type including description, and a list of searchable thing attribute names.

Type: [ThingTypeProperties \(p. 141\)](#) object

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DetachPrincipalPolicy

Removes the specified policy from the specified certificate.

## Request Syntax

```
{  
  "policyName": "string",  
  "principal": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **policyName**

The name of the policy to detach.

Type: String

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

Pattern: [w+, .@- ]+

Required: Yes

### **principal**

The principal.

If the principal is a certificate, specify the certificate ARN. If the principal is an Amazon Cognito identity, specify the identity ID.

Type: String

Required: Yes

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DetachThingPrincipal

Detaches the specified principal from the specified thing.

## Request Syntax

```
{  
  "principal": "string",  
  "thingName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **principal**

If the principal is a certificate, this value must be ARN of the certificate. If the principal is an Amazon Cognito identity, this value must be the ID of the Amazon Cognito identity.

Type: String

Required: Yes

### **thingName**

The name of the thing.

Type: String

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

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

Required: Yes

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# DisableTopicRule

Disables the specified rule.

## Request Syntax

```
{  
  "ruleName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **ruleName**

The name of the rule to disable.

Type: String

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

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

Required: Yes

## Response Elements

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

## Errors

### **InternalException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# EnableTopicRule

Enables the specified rule.

## Request Syntax

```
{  
  "ruleName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **ruleName**

The name of the topic rule to enable.

Type: String

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

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

Required: Yes

## Response Elements

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

## Errors

### **InternalException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# GetLoggingOptions

Gets the logging options.

## Response Syntax

```
{  
  "logLevel": "string",  
  "roleArn": "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.

### logLevel

The logging level.

Type: String

Valid Values: DEBUG | INFO | ERROR | WARN | DISABLED

### roleArn

The ARN of the IAM role that grants access.

Type: String

## Errors

### InternalException

An unexpected error has occurred.

HTTP Status Code: 500

### InvalidRequestException

The request is not valid.

HTTP Status Code: 400

### ServiceUnavailableException

The service is temporarily unavailable.

HTTP Status Code: 500



# GetPolicy

Gets information about the specified policy with the policy document of the default version.

## Request Syntax

```
{  
  "policyName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **policyName**

The name of the policy.

Type: String

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

Pattern: [*\w+=, .@-*]+

Required: Yes

## Response Syntax

```
{  
  "defaultVersionId": "string",  
  "policyArn": "string",  
  "policyDocument": "string",  
  "policyName": "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.

### **defaultVersionId**

The default policy version ID.

Type: String

Pattern: [*0-9*]+

### **policyArn**

The policy ARN.

Type: String

**policyDocument**

The JSON document that describes the policy.

Type: String

**policyName**

The policy name.

Type: String

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

Pattern: [\w+=, .@- ]+

## Errors

**InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# GetPolicyVersion

Gets information about the specified policy version.

## Request Syntax

```
{  
  "policyName": "string",  
  "policyVersionId": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **policyName**

The name of the policy.

Type: String

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

Pattern: [\w+=, .@- ]+

Required: Yes

### **policyVersionId**

The policy version ID.

Type: String

Pattern: [0-9 ]+

Required: Yes

## Response Syntax

```
{  
  "isDefaultVersion": boolean,  
  "policyArn": "string",  
  "policyDocument": "string",  
  "policyName": "string",  
  "policyVersionId": "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.

**isDefaultVersion**

Specifies whether the policy version is the default.

Type: Boolean

**policyArn**

The policy ARN.

Type: String

**policyDocument**

The JSON document that describes the policy.

Type: String

**policyName**

The policy name.

Type: String

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

Pattern: [ \w+=, .@- ]+

**policyVersionId**

The policy version ID.

Type: String

Pattern: [0-9 ]+

## Errors

**InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# GetRegistrationCode

Gets a registration code used to register a CA certificate with AWS IoT.

## Response Syntax

```
{  
  "registrationCode": "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.

### registrationCode

The CA certificate registration code.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

## Errors

### InternalFailureException

An unexpected error has occurred.

HTTP Status Code: 500

### InvalidRequestException

The request is not valid.

HTTP Status Code: 400

### ServiceUnavailableException

The service is temporarily unavailable.

HTTP Status Code: 500

### ThrottlingException

The rate exceeds the limit.

HTTP Status Code: 400

### UnauthorizedException

You are not authorized to perform this operation.

HTTP Status Code: 400

# GetTopicRule

Gets information about the specified rule.

## Request Syntax

```
{  
  "ruleName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### ruleName

The name of the rule.

Type: String

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

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

Required: Yes

## Response Syntax

```
{  
  "rule": {  
    "actions": [  
      {  
        "cloudwatchAlarm": {  
          "alarmName": "string",  
          "roleArn": "string",  
          "stateReason": "string",  
          "stateValue": "string"  
        },  
        "cloudwatchMetric": {  
          "metricName": "string",  
          "metricNamespace": "string",  
          "metricTimestamp": "string",  
          "metricUnit": "string",  
          "metricValue": "string",  
          "roleArn": "string"  
        },  
        "dynamoDB": {  
          "hashKeyField": "string",  
          "hashKeyType": "string",  
          "hashKeyValue": "string",  
          "operation": "string",
```

```
        "payloadField": "string",
        "rangeKeyField": "string",
        "rangeKeyType": "string",
        "rangeKeyValue": "string",
        "roleArn": "string",
        "tableName": "string"
    },
    "dynamoDBv2": {
        "putItem": {
            "tableName": "string"
        },
        "roleArn": "string"
    },
    "elasticsearch": {
        "endpoint": "string",
        "id": "string",
        "index": "string",
        "roleArn": "string",
        "type": "string"
    },
    "firehose": {
        "deliveryStreamName": "string",
        "roleArn": "string",
        "separator": "string"
    },
    "kinesis": {
        "partitionKey": "string",
        "roleArn": "string",
        "streamName": "string"
    },
    "lambda": {
        "functionArn": "string"
    },
    "republish": {
        "roleArn": "string",
        "topic": "string"
    },
    "s3": {
        "bucketName": "string",
        "cannedAcl": "string",
        "key": "string",
        "roleArn": "string"
    },
    "sns": {
        "messageFormat": "string",
        "roleArn": "string",
        "targetArn": "string"
    },
    "sqs": {
        "queueUrl": "string",
        "roleArn": "string",
        "useBase64": boolean
    }
}
],
"awsIotSqlVersion": "string",
"createdAt": number,
"description": "string",
"ruleDisabled": boolean,
```

```
    "ruleName": "string",  
    "sql": "string"  
  },  
  "ruleArn": "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.

### **rule**

The rule.

Type: [TopicRule](#) (p. 141) object

### **ruleArn**

The rule ARN.

Type: String

## Errors

### **InternalException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400



## ListCACertificates

Lists the CA certificates registered for your AWS account.

The results are paginated with a default page size of 25. You can use the returned marker to retrieve additional results.

### Request Syntax

```
{
  "ascendingOrder": boolean,
  "marker": "string",
  "pageSize": number
}
```

### Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

#### **ascendingOrder**

Determines the order of the results.

Type: Boolean

Required: No

#### **marker**

The marker for the next set of results.

Type: String

Required: No

#### **pageSize**

The result page size.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

### Response Syntax

```
{
  "certificates": [
    {
      "certificateArn": "string",
      "certificateId": "string",
      "creationDate": number,
    }
  ]
}
```

```
        "status": "string"  
      }  
    ],  
    "nextMarker": "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.

### **certificates**

The CA certificates registered in your AWS account.

Type: array of [CACertificate](#) (p. 124) objects

### **nextMarker**

The current position within the list of CA certificates.

Type: String

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

## ListCertificates

Lists the certificates registered in your AWS account.

The results are paginated with a default page size of 25. You can use the returned marker to retrieve additional results.

### Request Syntax

```
{
  "ascendingOrder": boolean,
  "marker": "string",
  "pageSize": number
}
```

### Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

#### **ascendingOrder**

Specifies the order for results. If True, the results are returned in ascending order, based on the creation date.

Type: Boolean

Required: No

#### **marker**

The marker for the next set of results.

Type: String

Required: No

#### **pageSize**

The result page size.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

### Response Syntax

```
{
  "certificates": [
    {
      "certificateArn": "string",
      "certificateId": "string",

```

```
        "creationDate": number,  
        "status": "string"  
    },  
    "nextMarker": "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.

### **certificates**

The descriptions of the certificates.

Type: array of [Certificate](#) (p. 126) objects

### **nextMarker**

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

Type: String

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ListCertificatesByCA

List the device certificates signed by the specified CA certificate.

## Request Syntax

```
{  
  "ascendingOrder": boolean,  
  "caCertificateId": "string",  
  "marker": "string",  
  "pageSize": number  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **ascendingOrder**

Specifies the order for results. If True, the results are returned in ascending order, based on the creation date.

Type: Boolean

Required: No

### **caCertificateId**

The ID of the CA certificate. This operation will list all registered device certificate that were signed by this CA certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

### **marker**

The marker for the next set of results.

Type: String

Required: No

### **pageSize**

The result page size.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

## Response Syntax

```
{
  "certificates": [
    {
      "certificateArn": "string",
      "certificateId": "string",
      "creationDate": number,
      "status": "string"
    }
  ],
  "nextMarker": "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.

### **certificates**

The device certificates signed by the specified CA certificate.

Type: array of [Certificate \(p. 126\)](#) objects

### **nextMarker**

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

Type: String

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ListOutgoingCertificates

Lists certificates that are being transferred but not yet accepted.

## Request Syntax

```
{  
  "ascendingOrder": boolean,  
  "marker": "string",  
  "pageSize": number  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### ascendingOrder

Specifies the order for results. If True, the results are returned in ascending order, based on the creation date.

Type: Boolean

Required: No

### marker

The marker for the next set of results.

Type: String

Required: No

### pageSize

The result page size.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

## Response Syntax

```
{  
  "nextMarker": "string",  
  "outgoingCertificates": [  
    {  
      "certificateArn": "string",  
      "certificateId": "string",  
      "creationDate": number,  
      "transferDate": number,  
    }  
  ]  
}
```

```
        "transferMessage": "string",  
        "transferredTo": "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.

### **nextMarker**

The marker for the next set of results.

Type: String

### **outgoingCertificates**

The certificates that are being transferred but not yet accepted.

Type: array of [OutgoingCertificate](#) (p. 134) objects

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400



# ListPolicies

Lists your policies.

## Request Syntax

```
{  
  "ascendingOrder": boolean,  
  "marker": "string",  
  "pageSize": number  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **ascendingOrder**

Specifies the order for results. If true, the results are returned in ascending creation order.

Type: Boolean

Required: No

### **marker**

The marker for the next set of results.

Type: String

Required: No

### **pageSize**

The result page size.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

## Response Syntax

```
{  
  "nextMarker": "string",  
  "policies": [  
    {  
      "policyArn": "string",  
      "policyName": "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.

### **nextMarker**

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

Type: String

### **policies**

The descriptions of the policies.

Type: array of [Policy \(p. 135\)](#) objects

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ListPolicyPrincipals

Lists the principals associated with the specified policy.

## Request Syntax

```
{  
  "ascendingOrder": boolean,  
  "marker": "string",  
  "pageSize": number,  
  "policyName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **ascendingOrder**

Specifies the order for results. If true, the results are returned in ascending creation order.

Type: Boolean

Required: No

### **marker**

The marker for the next set of results.

Type: String

Required: No

### **pageSize**

The result page size.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

### **policyName**

The policy name.

Type: String

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

Pattern: [*\w+=, .@- ]+*

Required: Yes

## Response Syntax

---

```
{  
  "nextMarker": "string",  
  "principals": [  
    "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.

### **nextMarker**

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

Type: String

### **principals**

The descriptions of the principals.

Type: array of Strings

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

## ListPolicyVersions

Lists the versions of the specified policy and identifies the default version.

### Request Syntax

```
{  
  "policyName": "string"  
}
```

### Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

#### **policyName**

The policy name.

Type: String

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

Pattern: [\w+=, .@-]+

Required: Yes

### Response Syntax

```
{  
  "policyVersions": [  
    {  
      "createDate": number,  
      "isDefaultVersion": boolean,  
      "versionId": "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.

#### **policyVersions**

The policy versions.

Type: array of [PolicyVersion](#) (p. 136) objects

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ListPrincipalPolicies

Lists the policies attached to the specified principal. If you use an Cognito identity, the ID must be in [AmazonCognito Identity format](#).

## Request Syntax

```
{  
  "ascendingOrder": boolean,  
  "marker": "string",  
  "pageSize": number,  
  "principal": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **ascendingOrder**

Specifies the order for results. If true, results are returned in ascending creation order.

Type: Boolean

Required: No

### **marker**

The marker for the next set of results.

Type: String

Required: No

### **pageSize**

The result page size.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

### **principal**

The principal.

Type: String

Required: Yes

## Response Syntax

```
{
```

```
"nextMarker": "string",  
"policies": [  
  {  
    "policyArn": "string",  
    "policyName": "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.

### **nextMarker**

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

Type: String

### **policies**

The policies.

Type: array of [Policy \(p. 135\)](#) objects

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400



# ListPrincipalThings

Lists the things associated with the specified principal.

## Request Syntax

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

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### maxResults

The maximum number of results to return in this operation.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

### nextToken

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

Type: String

Required: No

### principal

The principal.

Type: String

Required: Yes

## Response Syntax

```
{  
  "nextToken": "string",  
  "things": [  
    "string"  
  ]  
}
```

## Response Elements

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

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

**nextToken**

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

Type: String

**things**

The things.

Type: array of Strings

## Errors

**InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ListThingPrincipals

Lists the principals associated with the specified thing.

## Request Syntax

```
{  
  "thingName": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **thingName**

The name of the thing.

Type: String

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

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

Required: Yes

## Response Syntax

```
{  
  "principals": [  
    "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.

### **principals**

The principals associated with the thing.

Type: array of Strings

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ListThings

Lists your things. Use the **attributeName** and **attributeValue** parameters to filter your things. For example, calling `ListThings` with `attributeName=Color` and `attributeValue=Red` retrieves all things in the registry that contain an attribute **Color** with the value **Red**.

## Request Syntax

```
{
  "attributeName": "string",
  "attributeValue": "string",
  "maxResults": number,
  "nextToken": "string",
  "thingTypeName": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **attributeName**

The attribute name used to search for things.

Type: String

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

Pattern: [a-zA-Z0-9\_.,@/:#-]+

Required: No

### **attributeValue**

The attribute value used to search for things.

Type: String

Length constraints: Minimum length of 0. Maximum length of 800.

Pattern: [a-zA-Z0-9\_.,@/:#-]\*

Required: No

### **maxResults**

The maximum number of results to return in this operation.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

### **nextToken**

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

Type: String

Required: No

#### **thingTypeName**

The name of the thing type used to search for things.

Type: String

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

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

Required: No

## Response Syntax

```
{
  "nextToken": "string",
  "things": [
    {
      "attributes": {
        "string": "string"
      },
      "thingName": "string",
      "thingTypeName": "string",
      "version": number
    }
  ]
}
```

## Response Elements

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

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

#### **nextToken**

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

Type: String

#### **things**

The things.

Type: array of [ThingAttribute](#) (p. 139) objects

## Errors

#### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

#### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ListThingTypes

Lists the existing thing types.

## Request Syntax

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

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### maxResults

The maximum number of results to return in this operation.

Type: Number

Valid range: Minimum value of 1. Maximum value of 250.

Required: No

### nextToken

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

Type: String

Required: No

### thingTypeName

The name of the thing type.

Type: String

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

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

Required: No

## Response Syntax

```
{
  "nextToken": "string",
  "thingTypes": [
    {
      "thingTypeMetadata": {
```



```
        "creationDate": number,
        "deprecated": boolean,
        "deprecationDate": number
    },
    "thingTypeName": "string",
    "thingTypeProperties": {
        "searchableAttributes": [
            "string"
        ],
        "thingTypeDescription": "string"
    }
}
]
```

## Response Elements

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

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

### **nextToken**

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

Type: String

### **thingTypes**

The thing types.

Type: array of [ThingTypeDefinition](#) (p. 140) objects

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ListTopicRules

Lists the rules for the specific topic.

## Request Syntax

```
{  
  "maxResults": number,  
  "nextToken": "string",  
  "ruleDisabled": boolean,  
  "topic": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### maxResults

The maximum number of results to return.

Type: Number

Valid range: Minimum value of 1. Maximum value of 10000.

Required: No

### nextToken

A token used to retrieve the next value.

Type: String

Required: No

### ruleDisabled

Specifies whether the rule is disabled.

Type: Boolean

Required: No

### topic

The topic.

Type: String

Required: No

## Response Syntax

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

```
"rules": [  
  {  
    "createdAt": number,  
    "ruleArn": "string",  
    "ruleDisabled": boolean,  
    "ruleName": "string",  
    "topicPattern": "string"  
  }  
]
```

## Response Elements

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

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

### **nextToken**

A token used to retrieve the next value.

Type: String

### **rules**

The rules.

Type: array of [TopicRuleListItem](#) (p. 142) objects

## Errors

### **InternalException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

# RegisterCACertificate

Registers a CA certificate with AWS IoT. This CA certificate can then be used to sign device certificates, which can be then registered with AWS IoT. You can register up to 10 CA certificates per AWS account that have the same subject field and public key. This enables you to have up to 10 certificate authorities sign your device certificates. If you have more than one CA certificate registered, make sure you pass the CA certificate when you register your device certificates with the RegisterCertificate API.

## Request Syntax

```
{
  "allowAutoRegistration": boolean,
  "caCertificate": "string",
  "setAsActive": boolean,
  "verificationCertificate": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **allowAutoRegistration**

Allows this CA certificate to be used for auto registration of device certificates.

Type: Boolean

Required: No

### **caCertificate**

The CA certificate.

Type: String

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

Required: Yes

### **setAsActive**

A boolean value that specifies if the CA certificate is set to active.

Type: Boolean

Required: No

### **verificationCertificate**

The private key verification certificate.

Type: String

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

Required: Yes

## Response Syntax

```
{  
  "certificateArn": "string",  
  "certificateId": "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.

### **certificateArn**

The CA certificate ARN.

Type: String

### **certificateId**

The CA certificate identifier.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

## Errors

### **CertificateValidationException**

The certificate is invalid.

HTTP Status Code: 400

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **LimitExceededException**

The number of attached entities exceeds the limit.

HTTP Status Code: 400

### **RegistrationCodeValidationException**

The registration code is invalid.

HTTP Status Code: 400

### **ResourceAlreadyExistsException**

The resource already exists.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# RegisterCertificate

Registers a device certificate with AWS IoT. If you have more than one CA certificate that has the same subject field, you must specify the CA certificate that was used to sign the device certificate being registered.

## Request Syntax

```
{  
  "caCertificatePem": "string",  
  "certificatePem": "string",  
  "setAsActive": boolean,  
  "status": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### caCertificatePem

The CA certificate used to sign the device certificate being registered.

Type: String

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

Required: No

### certificatePem

The certificate data, in PEM format.

Type: String

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

Required: Yes

### setAsActive

A boolean value that specifies if the CA certificate is set to active.

Type: Boolean

Required: No

### status

The status of the register certificate request.

Type: String

Valid Values: ACTIVE | INACTIVE | REVOKED | PENDING\_TRANSFER | REGISTER\_INACTIVE | PENDING\_ACTIVATION

Required: No

## Response Syntax

```
{  
  "certificateArn": "string",  
  "certificateId": "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.

### **certificateArn**

The certificate ARN.

Type: String

### **certificateId**

The certificate identifier.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

## Errors

### **CertificateConflictException**

Unable to verify the CA certificate used to sign the device certificate you are attempting to register. This happens when you have registered more than one CA certificate that has the same subject field and public key.

HTTP Status Code: 400

### **CertificateStateException**

The certificate operation is not allowed.

HTTP Status Code: 400

### **CertificateValidationException**

The certificate is invalid.

HTTP Status Code: 400

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceAlreadyExistsException**

The resource already exists.

HTTP Status Code: 400



**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# RejectCertificateTransfer

Rejects a pending certificate transfer. After AWS IoT rejects a certificate transfer, the certificate status changes from **PENDING\_TRANSFER** to **INACTIVE**.

To check for pending certificate transfers, call [ListCertificates](#) (p. 73) to enumerate your certificates.

This operation can only be called by the transfer destination. After it is called, the certificate will be returned to the source's account in the **INACTIVE** state.

## Request Syntax

```
{
  "certificateId": "string",
  "rejectReason": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **certificateId**

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

### **rejectReason**

The reason the certificate transfer was rejected.

Type: String

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

Required: No

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

**InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**TransferAlreadyCompletedException**

You can't revert the certificate transfer because the transfer is already complete.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# ReplaceTopicRule

Replaces the specified rule. You must specify all parameters for the new rule. Creating rules is an administrator-level action. Any user who has permission to create rules will be able to access data processed by the rule.

## Request Syntax

```
{
  "ruleName": "string",
  "topicRulePayload": {
    "actions": [
      {
        "cloudwatchAlarm": {
          "alarmName": "string",
          "roleArn": "string",
          "stateReason": "string",
          "stateValue": "string"
        },
        "cloudwatchMetric": {
          "metricName": "string",
          "metricNamespace": "string",
          "metricTimestamp": "string",
          "metricUnit": "string",
          "metricValue": "string",
          "roleArn": "string"
        },
        "dynamoDB": {
          "hashKeyField": "string",
          "hashKeyType": "string",
          "hashKeyValue": "string",
          "operation": "string",
          "payloadField": "string",
          "rangeKeyField": "string",
          "rangeKeyType": "string",
          "rangeKeyValue": "string",
          "roleArn": "string",
          "tableName": "string"
        },
        "dynamoDBv2": {
          "putItem": {
            "tableName": "string"
          },
          "roleArn": "string"
        },
        "elasticsearch": {
          "endpoint": "string",
          "id": "string",
          "index": "string",
          "roleArn": "string",
          "type": "string"
        },
        "firehose": {
          "deliveryStreamName": "string",
          "roleArn": "string",
          "separator": "string"
        }
      ]
    }
  }
}
```

```

    },
    "kinesis": {
      "partitionKey": "string",
      "roleArn": "string",
      "streamName": "string"
    },
    "lambda": {
      "functionArn": "string"
    },
    "republish": {
      "roleArn": "string",
      "topic": "string"
    },
    "s3": {
      "bucketName": "string",
      "cannedAcl": "string",
      "key": "string",
      "roleArn": "string"
    },
    "sns": {
      "messageFormat": "string",
      "roleArn": "string",
      "targetArn": "string"
    },
    "sqs": {
      "queueUrl": "string",
      "roleArn": "string",
      "useBase64": boolean
    }
  },
  "awsIotSqlVersion": "string",
  "description": "string",
  "ruleDisabled": boolean,
  "sql": "string"
}

```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### ruleName

The name of the rule.

Type: String

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

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

Required: Yes

### topicRulePayload

The rule payload.

Type: [TopicRulePayload \(p. 143\)](#) object

Required: Yes

## Response Elements

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

## Errors

### **InternalException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **SqlParseException**

The Rule-SQL expression can't be parsed correctly.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# SetDefaultPolicyVersion

Sets the specified version of the specified policy as the policy's default (operative) version. This action affects all certificates to which the policy is attached. To list the principals the policy is attached to, use the ListPrincipalPolicy API.

## Request Syntax

```
{
  "policyName": "string",
  "policyVersionId": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **policyName**

The policy name.

Type: String

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

Pattern: [*\w+=, .@-*]+

Required: Yes

### **policyVersionId**

The policy version ID.

Type: String

Pattern: [*0-9*]+

Required: Yes

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

**ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

**ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

**ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

**UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400



# SetLoggingOptions

Sets the logging options.

## Request Syntax

```
{
  "loggingOptionsPayload": {
    "logLevel": "string",
    "roleArn": "string"
  }
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **loggingOptionsPayload**

The logging options payload.

Type: [LoggingOptionsPayload](#) (p. 134) object

Required: Yes

## Response Elements

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

## Errors

### **InternalException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

# TransferCertificate

Transfers the specified certificate to the specified AWS account.

You can cancel the transfer until it is acknowledged by the recipient.

No notification is sent to the transfer destination's account. It is up to the caller to notify the transfer target.

The certificate being transferred must not be in the ACTIVE state. You can use the UpdateCertificate API to deactivate it.

The certificate must not have any policies attached to it. You can use the DetachPrincipalPolicy API to detach them.

## Request Syntax

```
{
  "certificateId": "string",
  "targetAwsAccount": "string",
  "transferMessage": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters \(p. 145\)](#).

The request requires the following data in JSON format.

### **certificateId**

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

### **targetAwsAccount**

The AWS account.

Type: String

Pattern: [0-9]{12}

Required: Yes

### **transferMessage**

The transfer message.

Type: String

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

Required: No

## Response Syntax

```
{  
  "transferredCertificateArn": "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.

### **transferredCertificateArn**

The ARN of the certificate.

Type: String

## Errors

### **CertificateStateException**

The certificate operation is not allowed.

HTTP Status Code: 400

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **TransferConflictException**

You can't transfer the certificate because authorization policies are still attached.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# UpdateCACertificate

Updates a registered CA certificate.

## Request Syntax

```
{  
  "certificateId": "string",  
  "newAutoRegistrationStatus": "string",  
  "newStatus": "string"  
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### **certificateId**

The CA certificate identifier.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

### **newAutoRegistrationStatus**

The new value for the auto registration status. Valid values are: "ENABLE" or "DISABLE".

Type: String

Valid Values: ENABLE | DISABLE

Required: No

### **newStatus**

The updated status of the CA certificate.

**Note:** The status value REGISTER\_INACTIVE is deprecated and should not be used.

Type: String

Valid Values: ACTIVE | INACTIVE

Required: No

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

# UpdateCertificate

Updates the status of the specified certificate. This operation is idempotent.

Moving a certificate from the ACTIVE state (including REVOKED) will not disconnect currently connected devices, but these devices will be unable to reconnect.

The ACTIVE state is required to authenticate devices connecting to AWS IoT using a certificate.

## Request Syntax

```
{
  "certificateId": "string",
  "newStatus": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### certificateId

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: Yes

### newStatus

The new status.

**Note:** Setting the status to PENDING\_TRANSFER will result in an exception being thrown. PENDING\_TRANSFER is a status used internally by AWS IoT. It is not intended for developer use.

**Note:** The status value REGISTER\_INACTIVE is deprecated and should not be used.

Type: String

Valid Values: ACTIVE | INACTIVE | REVOKED | PENDING\_TRANSFER | REGISTER\_INACTIVE | PENDING\_ACTIVATION

Required: Yes

## Response Elements

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

## Errors

### **CertificateStateException**

The certificate operation is not allowed.

HTTP Status Code: 400

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400



# UpdateThing

Updates the data for a thing.

## Request Syntax

```
{
  "attributePayload": {
    "attributes": {
      "string" :
        "string"
    },
    "merge": boolean
  },
  "expectedVersion": number,
  "removeThingType": boolean,
  "thingName": "string",
  "thingTypeName": "string"
}
```

## Request Parameters

For information about the common parameters that all actions use, see [Common Parameters](#) (p. 145).

The request requires the following data in JSON format.

### attributePayload

A list of thing attributes, a JSON string containing name-value pairs. For example:

```
{\"attributes\":{\"name1\": \"value2\"}}
```

This data is used to add new attributes or update existing attributes.

Type: [AttributePayload](#) (p. 123) object

Required: No

### expectedVersion

The expected version of the thing record in the registry. If the version of the record in the registry does not match the expected version specified in the request, the `UpdateThing` request is rejected with a `VersionConflictException`.

Type: Long

Required: No

### removeThingType

Remove a thing type association. If **true**, the association is removed.

Type: Boolean

Required: No

### thingName

The name of the thing to update.

Type: String

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

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

Required: Yes

#### **thingTypeName**

The name of the thing type.

Type: String

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

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

Required: No

## Response Elements

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

## Errors

### **InternalFailureException**

An unexpected error has occurred.

HTTP Status Code: 500

### **InvalidRequestException**

The request is not valid.

HTTP Status Code: 400

### **ResourceNotFoundException**

The specified resource does not exist.

HTTP Status Code: 400

### **ServiceUnavailableException**

The service is temporarily unavailable.

HTTP Status Code: 500

### **ThrottlingException**

The rate exceeds the limit.

HTTP Status Code: 400

### **UnauthorizedException**

You are not authorized to perform this operation.

HTTP Status Code: 400

### **VersionConflictException**

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

HTTP Status Code: 400

# Data Types

---

The AWS IoT API contains several data types that various actions use. This section describes each data type in detail.

**Note**

The order of each element in the response is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- [Action](#) (p. 122)
- [AttributePayload](#) (p. 123)
- [CACertificate](#) (p. 124)
- [CACertificateDescription](#) (p. 124)
- [Certificate](#) (p. 126)
- [CertificateDescription](#) (p. 126)
- [CloudwatchAlarmAction](#) (p. 128)
- [CloudwatchMetricAction](#) (p. 128)
- [DynamoDBAction](#) (p. 129)
- [DynamoDBv2Action](#) (p. 131)
- [ElasticsearchAction](#) (p. 131)
- [FirehoseAction](#) (p. 132)
- [KeyPair](#) (p. 133)
- [KinesisAction](#) (p. 133)
- [LambdaAction](#) (p. 134)
- [LoggingOptionsPayload](#) (p. 134)
- [OutgoingCertificate](#) (p. 134)
- [Policy](#) (p. 135)
- [PolicyVersion](#) (p. 136)
- [PutItemInput](#) (p. 136)
- [RepublishAction](#) (p. 137)
- [S3Action](#) (p. 137)
- [SnsAction](#) (p. 138)
- [SqsAction](#) (p. 138)

- [ThingAttribute](#) (p. 139)
- [ThingTypeDefinition](#) (p. 140)
- [ThingTypeMetadata](#) (p. 140)
- [ThingTypeProperties](#) (p. 141)
- [TopicRule](#) (p. 141)
- [TopicRuleListItem](#) (p. 142)
- [TopicRulePayload](#) (p. 143)
- [TransferData](#) (p. 144)

## Action

### Description

Describes the actions associated with a rule.

### Contents

#### **cloudwatchAlarm**

Change the state of a CloudWatch alarm.

Type: [CloudwatchAlarmAction](#) (p. 128) object

Required: No

#### **cloudwatchMetric**

Capture a CloudWatch metric.

Type: [CloudwatchMetricAction](#) (p. 128) object

Required: No

#### **dynamoDB**

Write to a DynamoDB table.

Type: [DynamoDBAction](#) (p. 129) object

Required: No

#### **dynamoDBv2**

Write to a DynamoDB table. This is a new version of the DynamoDB action. It allows you to write each attribute in an MQTT message payload into a separate DynamoDB column.

Type: [DynamoDBv2Action](#) (p. 131) object

Required: No

#### **elasticsearch**

Write data to an Amazon Elasticsearch Service domain.

Type: [ElasticsearchAction](#) (p. 131) object

Required: No

#### **firehose**

Write to an Amazon Kinesis Firehose stream.

Type: [FirehoseAction](#) (p. 132) object

Required: No

**kinesis**

Write data to an Amazon Kinesis stream.

Type: [KinesisAction \(p. 133\)](#) object

Required: No

**lambda**

Invoke a Lambda function.

Type: [LambdaAction \(p. 134\)](#) object

Required: No

**republish**

Publish to another MQTT topic.

Type: [RepublishAction \(p. 137\)](#) object

Required: No

**s3**

Write to an Amazon S3 bucket.

Type: [S3Action \(p. 137\)](#) object

Required: No

**sns**

Publish to an Amazon SNS topic.

Type: [SnsAction \(p. 138\)](#) object

Required: No

**sqs**

Publish to an Amazon SQS queue.

Type: [SqsAction \(p. 138\)](#) object

Required: No

## AttributePayload

### Description

The attribute payload.

### Contents

**attributes**

A JSON string containing up to three key-value pair in JSON format. For example:

```
{"attributes\" : {\"string1\" : \"string2\"}}
```

Type: String to String map

Required: No

**merge**

Specifies whether the list of attributes provided in the `AttributePayload` is merged with the attributes stored in the registry, instead of overwriting them.

To remove an attribute, call `UpdateThing` with an empty attribute value.

**Note**

The `merge` attribute is only valid when calling `UpdateThing`.

Type: Boolean

Required: No

## CACertificate

### Description

A CA certificate.

### Contents

**certificateArn**

The ARN of the CA certificate.

Type: String

Required: No

**certificateId**

The ID of the CA certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: `(0x)?[a-fA-F0-9]+`

Required: No

**creationDate**

The date the CA certificate was created.

Type: DateTime

Required: No

**status**

The status of the CA certificate.

The status value `REGISTER_INACTIVE` is deprecated and should not be used.

Type: String

Valid Values: `ACTIVE` | `INACTIVE`

Required: No

## CACertificateDescription

### Description

Describes a CA certificate.

## Contents

### **autoRegistrationStatus**

Whether the CA certificate configured for auto registration of device certificates. Valid values are "ENABLE" and "DISABLE"

Type: String

Valid Values: `ENABLE` | `DISABLE`

Required: No

### **certificateArn**

The CA certificate ARN.

Type: String

Required: No

### **certificateId**

The CA certificate ID.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: `(0x)?[a-fA-F0-9]+`

Required: No

### **certificatePem**

The CA certificate data, in PEM format.

Type: String

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

Required: No

### **creationDate**

The date the CA certificate was created.

Type: DateTime

Required: No

### **ownedBy**

The owner of the CA certificate.

Type: String

Pattern: `[0-9]{12}`

Required: No

### **status**

The status of a CA certificate.

Type: String

Valid Values: `ACTIVE` | `INACTIVE`

Required: No

# Certificate

## Description

Information about a certificate.

## Contents

### **certificateArn**

The ARN of the certificate.

Type: String

Required: No

### **certificateId**

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: No

### **creationDate**

The date and time the certificate was created.

Type: DateTime

Required: No

### **status**

The status of the certificate.

The status value REGISTER\_INACTIVE is deprecated and should not be used.

Type: String

Valid Values: ACTIVE | INACTIVE | REVOKED | PENDING\_TRANSFER | REGISTER\_INACTIVE | PENDING\_ACTIVATION

Required: No

# CertificateDescription

## Description

Describes a certificate.

## Contents

### **caCertificateId**

The certificate ID of the CA certificate used to sign this certificate.



Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: No

**certificateArn**

The ARN of the certificate.

Type: String

Required: No

**certificateId**

The ID of the certificate.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: No

**certificatePem**

The certificate data, in PEM format.

Type: String

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

Required: No

**creationDate**

The date and time the certificate was created.

Type: DateTime

Required: No

**lastModifiedDate**

The date and time the certificate was last modified.

Type: DateTime

Required: No

**ownedBy**

The ID of the AWS account that owns the certificate.

Type: String

Pattern: [0-9]{12}

Required: No

**previousOwnedBy**

The ID of the AWS account of the previous owner of the certificate.

Type: String

Pattern: [0-9]{12}

Required: No

**status**

The status of the certificate.

Type: String

Valid Values: ACTIVE | INACTIVE | REVOKED | PENDING\_TRANSFER | REGISTER\_INACTIVE | PENDING\_ACTIVATION

Required: No

**transferData**

The transfer data.

Type: [TransferData](#) (p. 144) object

Required: No

## CloudwatchAlarmAction

### Description

Describes an action that updates a CloudWatch alarm.

### Contents

**alarmName**

The CloudWatch alarm name.

Type: String

Required: Yes

**roleArn**

The IAM role that allows access to the CloudWatch alarm.

Type: String

Required: Yes

**stateReason**

The reason for the alarm change.

Type: String

Required: Yes

**stateValue**

The value of the alarm state. Acceptable values are: OK, ALARM, INSUFFICIENT\_DATA.

Type: String

Required: Yes

## CloudwatchMetricAction

### Description

Describes an action that captures a CloudWatch metric.

## Contents

### **metricName**

The CloudWatch metric name.

Type: String

Required: Yes

### **metricNamespace**

The CloudWatch metric namespace name.

Type: String

Required: Yes

### **metricTimestamp**

An optional [Unix timestamp](#).

Type: String

Required: No

### **metricUnit**

The [metric unit](#) supported by CloudWatch.

Type: String

Required: Yes

### **metricValue**

The CloudWatch metric value.

Type: String

Required: Yes

### **roleArn**

The IAM role that allows access to the CloudWatch metric.

Type: String

Required: Yes

## DynamoDBAction

### Description

Describes an action to write to a DynamoDB table.

The `tableName`, `hashKeyField`, and `rangeKeyField` values must match the values used when you created the table.

The `hashKeyValue` and `rangeKeyValue` fields use a substitution template syntax. These templates provide data at runtime. The syntax is as follows: `${sql-expression}`.

You can specify any valid expression in a WHERE or SELECT clause, including JSON properties, comparisons, calculations, and functions. For example, the following field uses the third level of the topic:

```
"hashKeyValue": "${topic(3)}"
```

The following field uses the timestamp:

```
"rangeKeyValue": "${timestamp()}"
```

## Contents

### **hashKeyField**

The hash key name.

Type: String

Required: Yes

### **hashKeyType**

The hash key type. Valid values are "STRING" or "NUMBER"

Type: String

Valid Values: STRING | NUMBER

Required: No

### **hashKeyValue**

The hash key value.

Type: String

Required: Yes

### **operation**

The type of operation to be performed. This follows the substitution template, so it can be `${operation}`, but the substitution must result in one of the following: INSERT, UPDATE, or DELETE.

Type: String

Required: No

### **payloadField**

The action payload. This name can be customized.

Type: String

Required: No

### **rangeKeyField**

The range key name.

Type: String

Required: No

### **rangeKeyType**

The range key type. Valid values are "STRING" or "NUMBER"

Type: String

Valid Values: STRING | NUMBER

Required: No

### **rangeKeyValue**

The range key value.

Type: String

Required: No

**roleArn**

The ARN of the IAM role that grants access to the DynamoDB table.

Type: String

Required: Yes

**tableName**

The name of the DynamoDB table.

Type: String

Required: Yes

## DynamoDBv2Action

### Description

Describes an action to write to a DynamoDB table.

This DynamoDB action writes each attribute in the message payload into it's own column in the DynamoDB table.

### Contents

**putItem**

Specifies the DynamoDB table to which the message data will be written. For example:

```
{ "dynamoDBv2": { "roleArn": "aws:iam:12341251:my-role" "putItem":  
  { "tableName": "my-table" } } }
```

Each attribute in the message payload will be written to a separate column in the DynamoDB database.

Type: [PutItemInput](#) (p. 136) object

Required: No

**roleArn**

The ARN of the IAM role that grants access to the DynamoDB table.

Type: String

Required: No

## ElasticsearchAction

### Description

Describes an action that writes data to an Amazon Elasticsearch Service domain.

### Contents

**endpoint**

The endpoint of your Elasticsearch domain.

Type: String

Pattern: `https?://.*`

Required: Yes

**id**

The unique identifier for the document you are storing.

Type: String

Required: Yes

**index**

The Elasticsearch index where you want to store your data.

Type: String

Required: Yes

**roleArn**

The IAM role ARN that has access to Elasticsearch.

Type: String

Required: Yes

**type**

The type of document you are storing.

Type: String

Required: Yes

## FirehoseAction

### Description

Describes an action that writes data to an Amazon Kinesis Firehose stream.

### Contents

**deliveryStreamName**

The delivery stream name.

Type: String

Required: Yes

**roleArn**

The IAM role that grants access to the Amazon Kinesis Firehose stream.

Type: String

Required: Yes

**separator**

A character separator that will be used to separate records written to the Firehose stream. Valid values are: `\n` (newline), `\t` (tab), `\r\n` (Windows newline), `,` (comma).

Type: String

Pattern: ([\n\t])|(\x\n)|(.)

Required: No

## KeyPair

### Description

Describes a key pair.

### Contents

#### **PrivateKey**

The private key.

Type: String

Length constraints: Minimum length of 1.

Required: No

#### **PublicKey**

The public key.

Type: String

Length constraints: Minimum length of 1.

Required: No

## KinesisAction

### Description

Describes an action to write data to an Amazon Kinesis stream.

### Contents

#### **partitionKey**

The partition key.

Type: String

Required: No

#### **roleArn**

The ARN of the IAM role that grants access to the Amazon Kinesis stream.

Type: String

Required: Yes

#### **streamName**

The name of the Amazon Kinesis stream.

Type: String

Required: Yes

## LambdaAction

### Description

Describes an action to invoke a Lambda function.

### Contents

**functionArn**

The ARN of the Lambda function.

Type: String

Required: Yes

## LoggingOptionsPayload

### Description

Describes the logging options payload.

### Contents

**logLevel**

The logging level.

Type: String

Valid Values: `DEBUG` | `INFO` | `ERROR` | `WARN` | `DISABLED`

Required: No

**roleArn**

The ARN of the IAM role that grants access.

Type: String

Required: Yes

## OutgoingCertificate

### Description

A certificate that has been transferred but not yet accepted.

### Contents

**certificateArn**

The certificate ARN.



Type: String

Required: No

**certificateId**

The certificate ID.

Type: String

Length constraints: Minimum length of 64. Maximum length of 64.

Pattern: (0x)?[a-fA-F0-9]+

Required: No

**creationDate**

The certificate creation date.

Type: DateTime

Required: No

**transferDate**

The date the transfer was initiated.

Type: DateTime

Required: No

**transferMessage**

The transfer message.

Type: String

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

Required: No

**transferredTo**

The AWS account to which the transfer was made.

Type: String

Pattern: [0-9]{12}

Required: No

## Policy

### Description

Describes an AWS IoT policy.

### Contents

**policyArn**

The policy ARN.

Type: String

Required: No

**policyName**

The policy name.

Type: String

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

Pattern: [ \w+=, .@- ]+

Required: No

## PolicyVersion

### Description

Describes a policy version.

### Contents

**createDate**

The date and time the policy was created.

Type: DateTime

Required: No

**isDefaultVersion**

Specifies whether the policy version is the default.

Type: Boolean

Required: No

**versionId**

The policy version ID.

Type: String

Pattern: [ 0-9 ]+

Required: No

## PutItemInput

### Description

The input for the DynamoActionVS action that specifies the DynamoDB table to which the message data will be written.

### Contents

**tableName**

The table where the message data will be written

Type: String

Required: Yes

## RepublishAction

### Description

Describes an action to republish to another topic.

### Contents

**roleArn**

The ARN of the IAM role that grants access.

Type: String

Required: Yes

**topic**

The name of the MQTT topic.

Type: String

Required: Yes

## S3Action

### Description

Describes an action to write data to an Amazon S3 bucket.

### Contents

**bucketName**

The Amazon S3 bucket.

Type: String

Required: Yes

**cannedAcl**

The Amazon S3 canned ACL that controls access to the object identified by the object key. For more information, see [S3 canned ACLs](#).

Type: String

Valid Values: `private` | `public-read` | `public-read-write` | `aws-exec-read` | `authenticated-read` | `bucket-owner-read` | `bucket-owner-full-control` | `log-delivery-write`

Required: No

**key**

The object key.

Type: String

Required: Yes

**roleArn**

The ARN of the IAM role that grants access.

Type: String

Required: Yes

## SnsAction

### Description

Describes an action to publish to an Amazon SNS topic.

### Contents

**messageFormat**

The message format of the message to publish. Optional. Accepted values are "JSON" and "RAW". The default value of the attribute is "RAW". SNS uses this setting to determine if the payload should be parsed and relevant platform-specific bits of the payload should be extracted. To read more about SNS message formats, see <http://docs.aws.amazon.com/sns/latest/dg/json-formats.html> refer to their official documentation.

Type: String

Valid Values: RAW | JSON

Required: No

**roleArn**

The ARN of the IAM role that grants access.

Type: String

Required: Yes

**targetArn**

The ARN of the SNS topic.

Type: String

Required: Yes

## SqsAction

### Description

Describes an action to publish data to an Amazon SQS queue.

### Contents

**queueUrl**

The URL of the Amazon SQS queue.

Type: String

Required: Yes

**roleArn**

The ARN of the IAM role that grants access.

Type: String

Required: Yes

**useBase64**

Specifies whether to use Base64 encoding.

Type: Boolean

Required: No

## ThingAttribute

### Description

The properties of the thing, including thing name, thing type name, and a list of thing attributes.

### Contents

**attributes**

A list of thing attributes which are name-value pairs.

Type: String to String map

Required: No

**thingName**

The name of the thing.

Type: String

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

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

Required: No

**thingTypeName**

The name of the thing type, if the thing has been associated with a type.

Type: String

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

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

Required: No

**version**

The version of the thing record in the registry.

Type: Long

Required: No

# ThingTypeDefinition

## Description

The definition of the thing type, including thing type name and description.

## Contents

### **thingTypeMetadata**

The ThingTypeMetadata contains additional information about the thing type including: creation date and time, a value indicating whether the thing type is deprecated, and a date and time when time was deprecated.

Type: [ThingTypeMetadata \(p. 140\)](#) object

Required: No

### **thingTypeName**

The name of the thing type.

Type: String

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

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

Required: No

### **thingTypeProperties**

The ThingTypeProperties for the thing type.

Type: [ThingTypeProperties \(p. 141\)](#) object

Required: No

# ThingTypeMetadata

## Description

The ThingTypeMetadata contains additional information about the thing type including: creation date and time, a value indicating whether the thing type is deprecated, and a date and time when time was deprecated.

## Contents

### **creationDate**

The date and time when the thing type was created.

Type: DateTime

Required: No

### **deprecated**

Whether the thing type is deprecated. If **true**, no new things could be associated with this type.

Type: Boolean

Required: No

**deprecationDate**

The date and time when the thing type was deprecated.

Type: DateTime

Required: No

## ThingTypeProperties

### Description

The ThingTypeProperties contains information about the thing type including: a thing type description, and a list of searchable thing attribute names.

### Contents

**searchableAttributes**

A list of searchable thing attribute names.

Type: array of Strings

Required: No

**thingTypeDescription**

The description of the thing type.

Type: String

Length constraints: Minimum length of 0. Maximum length of 2028.

Pattern: [ \p{Graph}\x20 ] \*

Required: No

## TopicRule

### Description

Describes a rule.

### Contents

**actions**

The actions associated with the rule.

Type: array of [Action \(p. 122\)](#) objects

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: No

**awsIotSqlVersion**

The version of the SQL rules engine to use when evaluating the rule.

Type: String

Required: No

**createdAt**

The date and time the rule was created.

Type: DateTime

Required: No

**description**

The description of the rule.

Type: String

Required: No

**ruleDisabled**

Specifies whether the rule is disabled.

Type: Boolean

Required: No

**ruleName**

The name of the rule.

Type: String

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

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

Required: No

**sql**

The SQL statement used to query the topic. When using a SQL query with multiple lines, be sure to escape the newline characters.

Type: String

Required: No

## TopicRuleListItem

### Description

Describes a rule.

### Contents

**createdAt**

The date and time the rule was created.

Type: DateTime

Required: No

**ruleArn**

The rule ARN.



Type: String

Required: No

**ruleDisabled**

Specifies whether the rule is disabled.

Type: Boolean

Required: No

**ruleName**

The name of the rule.

Type: String

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

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

Required: No

**topicPattern**

The pattern for the topic names that apply.

Type: String

Required: No

## TopicRulePayload

### Description

Describes a rule.

### Contents

**actions**

The actions associated with the rule.

Type: array of [Action \(p. 122\)](#) objects

Length constraints: Minimum of 0 item(s) in the list. Maximum of 10 item(s) in the list.

Required: Yes

**awsIotSqlVersion**

The version of the SQL rules engine to use when evaluating the rule.

Type: String

Required: No

**description**

The description of the rule.

Type: String

Required: No

**ruleDisabled**

Specifies whether the rule is disabled.

Type: Boolean

Required: No

**sql**

The SQL statement used to query the topic. For more information, see [AWS IoT SQL Reference](#) in the *AWS IoT Developer Guide*.

Type: String

Required: Yes

## TransferData

### Description

Data used to transfer a certificate to an AWS account.

### Contents

**acceptDate**

The date the transfer was accepted.

Type: DateTime

Required: No

**rejectDate**

The date the transfer was rejected.

Type: DateTime

Required: No

**rejectReason**

The reason why the transfer was rejected.

Type: String

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

Required: No

**transferDate**

The date the transfer took place.

Type: DateTime

Required: No

**transferMessage**

The transfer message.

Type: String

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

Required: No

# Common Parameters

---

The following table lists the parameters that all actions use for signing Signature Version 4 requests. Any action-specific parameters are listed in the topic for that action. To view sample requests, see [Examples of Signed Signature Version 4 Requests](#) or [Signature Version 4 Test Suite](#) 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'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see [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. For a list of services that support AWS Security Token Service, go to [Using Temporary Security Credentials to Access AWS](#) in *Using Temporary Security Credentials*.

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