Amazon CloudWatch Logs

API Reference

API Version 2014-03-28

Amazon CloudWatch Logs: API Reference

Table of Contents

Welcome

Amazon CloudWatch Logs enables you to monitor, store, and access your system, application, and custom log files. This guide provides detailed information about CloudWatch Logs actions, data types, parameters, and errors. For more information about CloudWatch Logs features, see the Amazon CloudWatch Logs User Guide.

Use the following links to get started using the CloudWatch Logs Query API:

- Actions (p. 2): An alphabetical list of all CloudWatch Logs actions.
- Data Types (p. 79): An alphabetical list of all CloudWatch Logs data types.
- Common Parameters (p. 96): Parameters that all Query actions can use.
- Common Errors (p. 98): Client and server errors that all actions can return.
- Regions and Endpoints: Supported regions and endpoints for all AWS products.

Alternatively, you can use one of the AWS SDKs to access CloudWatch Logs using an API tailored to your programming language or platform.

Developers in the AWS developer community also provide their own libraries, which you can find at the following AWS developer centers:

- Java Developer Center
- JavaScript Developer Center
- AWS Mobile Services
- PHP Developer Center
- Python Developer Center
- Ruby Developer Center
- Windows and .NET Developer Center

Actions

The following actions are supported:

CancelExportTask

Cancels the specified export task.

The task must be in the `PENDING` or `RUNNING` state.

Request Syntax

```
{
   "taskId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

taskId (p. 3)

The ID of the export task.

Type: String

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidOperationException

The operation is not valid on the specified resource.

HTTP Status Code: 400

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To cancel an export task

The following example cancels the specified task.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.CancelExportTask
{
  "taskId": "exampleTaskId"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

CreateExportTask

Creates an export task, which allows you to efficiently export data from a log group to an Amazon S3 bucket.

This is an asynchronous call. If all the required information is provided, this operation initiates an export task and responds with the ID of the task. After the task has started, you can use DescribeExportTasks (p. 27) to get the status of the export task. Each account can only have one active (`RUNNING` or `PENDING`) export task at a time. To cancel an export task, use CancelExportTask (p. 3).

You can export logs from multiple log groups or multiple time ranges to the same S3 bucket. To separate out log data for each export task, you can specify a prefix that will be used as the Amazon S3 key prefix for all exported objects.

Request Syntax

```
{
   "destination": "string",
   "destinationPrefix": "string",
   "from": number,
   "logGroupName": "string",
   "logStreamNamePrefix": "string",
   "taskName": "string",
   "to": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

destination (p. 5)

The name of S3 bucket for the exported log data. The bucket must be in the same AWS region.

Type: String

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

Required: Yes

destinationPrefix (p. 5)

The prefix used as the start of the key for every object exported. If you don't specify a value, the default is `exportedlogs`.

Type: String

Required: No

from (p. 5)

The start time of the range for the request, expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC. Events with a timestamp earlier than this time are not exported.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

logGroupName (p. 5)

The name of the log group.

Type: String

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

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

Required: Yes

logStreamNamePrefix (p. 5)

Export only log streams that match the provided prefix. If you don't specify a value, no prefix filter is applied.

Type: String

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

Pattern: `[^:*]*`

Required: No

taskName (p. 5)

The name of the export task.

Type: String

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

Required: No

to (p. 5)

The end time of the range for the request, expressed as the number of milliseconds since Jan 1, 1970 00:00:00 UTC. Events with a timestamp later than this time are not exported.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

Response Syntax

```
{
   "taskId": "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.

taskId (p. 6)

The ID of the export task.

Type: String

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

LimitExceededException

You have reached the maximum number of resources that can be created.

HTTP Status Code: 400

OperationAbortedException

Multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceAlreadyExistsException
The specified resource already exists.
HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To create an export task

The following example creates an export task that exports data from a log group to an S3 bucket.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.CreateExportTask
{
  "taskName": "my-task",
  "logGroupName": "my-log-group",
  "from": 1437584472382,
  "to": 1437584472833,
  "destination": "my-destination",
  "destinationPrefix": "my-prefix"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "taskId": "exampleTaskId"
}
```

CreateLogGroup

Creates a log group with the specified name.

You can create up to 5000 log groups per account.

You must use the following guidelines when naming a log group:

- Log group names must be unique within a region for an AWS account.
- Log group names can be between 1 and 512 characters long.
- Log group names consist of the following characters: a-z, A-Z, 0-9, '_' (underscore), '-' (hyphen), '/' (forward slash), and '.' (period).

Request Syntax

```
{
   "logGroupName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

logGroupName (p. 8)

The name of the log group.

Type: String

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

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

LimitExceededException

You have reached the maximum number of resources that can be created.

HTTP Status Code: 400

OperationAbortedException

Multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To create a log group

The following example creates a log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.CreateLogGroup
{
  "logGroupName": "my-log-group"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

CreateLogStream

Creates a log stream for the specified log group.

There is no limit on the number of log streams that you can create for a log group.

You must use the following guidelines when naming a log stream:

- Log stream names must be unique within the log group.
- Log stream names can be between 1 and 512 characters long.
- The ':' (colon) and '*' (asterisk) characters are not allowed.

Request Syntax

```
{
   "logGroupName": "string",
   "logStreamName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

logGroupName (p. 10)

The name of the log group.

Type: String

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

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

Required: Yes

logStreamName (p. 10)

The name of the log stream.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

ResourceAlreadyExistsException

The specified resource already exists.

HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To create a log stream

The following example creates a log stream for the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.CreateLogStream
{
  "logGroupName": "my-log-group",
  "logStreamName": "my-log-stream"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

DeleteDestination

Deletes the specified destination, and eventually disables all the subscription filters that publish to it. This operation does not delete the physical resource encapsulated by the destination.

Request Syntax

```
{
   "destinationName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

destinationName (p. 12)
The name of the destination.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

OperationAbortedException
Multiple requests to update the same resource were in conflict.
HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To delete a destination

The following example deletes the specified destination.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteDestination
{
  "destinationName": my-destination"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

DeleteLogGroup

Deletes the specified log group and permanently deletes all the archived log events associated with the log group.

Request Syntax

```
{
   "logGroupName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

logGroupName (p. 14)
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

OperationAbortedException
Multiple requests to update the same resource were in conflict.
HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To delete a log group

The following example deletes the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteLogGroup
{
  "logGroupName": "my-log-group"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

DeleteLogStream

Deletes the specified log stream and permanently deletes all the archived log events associated with the log stream.

Request Syntax

```
{
   "logGroupName": "string",
   "logStreamName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

logGroupName (p. 16)
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: Yes

logStreamName (p. 16)
The name of the log stream.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

OperationAbortedException
Multiple requests to update the same resource were in conflict.
HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To delete a log stream

The following example deletes the specified log stream.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteLogStream
{
  "logGroupName": "my-log-group",
  "logStreamName": "my-log-stream"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

DeleteMetricFilter

Deletes the specified metric filter.

Request Syntax

```
{
   "filterName": "string",
   "logGroupName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

filterName (p. 18)
The name of the metric filter.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: Yes

logGroupName (p. 18)
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

OperationAbortedException
Multiple requests to update the same resource were in conflict.
HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To delete a metric filter

The following example deletes the specified filter for the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteMetricFilter
{
  "logGroupName": "my-log-group",
  "filterName": "my-metric-filter"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

DeleteRetentionPolicy

Deletes the specified retention policy.

Log events do not expire if they belong to log groups without a retention policy.

Request Syntax

```
{
   "logGroupName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

logGroupName (p. 20)

The name of the log group.

Type: String

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

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

OperationAbortedException

Multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To delete a retention policy

The following example deletes the retention policy for the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteRetentionPolicy
{
  "logGroupName": "my-log-group"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

DeleteSubscriptionFilter

Deletes the specified subscription filter.

Request Syntax

```
{
   "filterName": "string",
   "logGroupName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

filterName (p. 22)
The name of the subscription filter.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: Yes

logGroupName (p. 22)
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

OperationAbortedException
Multiple requests to update the same resource were in conflict.
HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To delete a subscription filter

The following example deletes the specified subscription filter for the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DeleteSubscriptionFilter
{
  "logGroupName": "my-log-group",
  "filterName": "my-subscription-filter"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

DescribeDestinations

Lists all your destinations. The results are ASCII-sorted by destination name.

Request Syntax

```
{
   "DestinationNamePrefix": "string",
   "limit": number,
   "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

DestinationNamePrefix (p. 24)

The prefix to match. If you don't specify a value, no prefix filter is applied.

Type: String

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

Pattern: `[^:*]*`

Required: No

limit (p. 24)

The maximum number of items returned. If you don't specify a value, the default is up to 50 items.

Type: Integer

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

Required: No

nextToken (p. 24)

The token for the next set of items to return. (You received this token from a previous call.)

Type: String

Length Constraints: Minimum length of 1.

Required: No

Response Syntax

```
{
   "destinations": [
      {
         "accessPolicy": "string",
         "arn": "string",
         "creationTime": number,
         "destinationName": "string",
         "roleArn": "string",
         "targetArn": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

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

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

destinations (p. 24)
The destinations.
Type: array of Destination (p. 80) objects

nextToken (p. 24)
The token for the next set of items to return. The token expires after 24 hours.
Type: String
Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To list all destinations

The following example lists all the destinations for the account.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeDestinations
{
  "destinationNamePrefix": "my-prefix"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
```

```
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "destination": [
    {
      "destinationName": "my-destination",
      "targetArn": "arn:aws:kinesis:us-east-1:123456789012:stream/my-kinesis-
stream",
      "roleArn": "arn:aws:iam::123456789012:role/my-subscription-role",
      "arn": "arn:aws:logs:us-east-1:123456789012:destination:my-
destination",
      "creationTime": 1437584472382
    }
  ]
}
```

DescribeExportTasks

Lists the specified export tasks. You can list all your export tasks or filter the results based on task ID or task status.

Request Syntax

```
{
   "limit": number,
   "nextToken": "string",
   "statusCode": "string",
   "taskId": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

limit (p. 27)

The maximum number of items returned. If you don't specify a value, the default is up to 50 items.

Type: Integer

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

Required: No

nextToken (p. 27)

The token for the next set of items to return. (You received this token from a previous call.)

Type: String

Length Constraints: Minimum length of 1.

Required: No

statusCode (p. 27)

The status code of the export task. Specifying a status code filters the results to zero or more export tasks.

Type: String

Valid Values: `CANCELLED | COMPLETED | FAILED | PENDING | PENDING_CANCEL | RUNNING`

Required: No

taskId (p. 27)

The ID of the export task. Specifying a task ID filters the results to zero or one export tasks.

Type: String

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

Required: No

Response Syntax

```
{
   "exportTasks": [
      {
         "destination": "string",
         "destinationPrefix": "string",
```

```
         "executionInfo": {
            "completionTime": number,
            "creationTime": number
         },
         "from": number,
         "logGroupName": "string",
         "status": {
            "code": "string",
            "message": "string"
         },
         "taskId": "string",
         "taskName": "string",
         "to": number
      }
   ],
   "nextToken": "string"
}
```

Response Elements

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

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

exportTasks (p. 27)

The export tasks.

Type: array of ExportTask (p. 81) objects

nextToken (p. 27)

The token for the next set of items to return. The token expires after 24 hours.

Type: String

Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To list the export tasks that are complete

The following example lists the export tasks with the `COMPLETE` status.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
```

```
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeExportTasks
{
  "statusCode": "COMPLETE"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "exportTasks": [
    {
      "taskId": "exampleTaskId",
      "taskName": "my-task-1",
      "logGroupName": "my-log-group",
      "from": 1437584472382,
      "to": 1437584472833,
      "destination": "my-destination",
      "destinationPrefix": "my-prefix",
      "status":
        {
          "code": "COMPLETE",
          "message": "Example message"
        },
      "executionInfo":
        {
          "creationTime": 1437584472856,
          "completionTime" : 1437584472986
        }
    },
    {
      "taskId": "exampleTaskId",
      "taskName": "my-task-2",
      "logGroupName": "my-log-group",
      "from": 1437584472382,
      "to": 1437584472833,
      "destination": "my-destination",
      "destinationPrefix": "my-prefix",
      "status":
        {
          "code": "COMPLETE",
          "message": "Example message"
        },
      "executionInfo":
        {
          "creationTime": 1437584472856,
          "completionTime" : 1437584472986
```

```
        }
    }
  ]
}
```

DescribeLogGroups

Lists the specified log groups. You can list all your log groups or filter the results by prefix. The results are ASCII-sorted by log group name.

Request Syntax

```
{
   "limit": number,
   "logGroupNamePrefix": "string",
   "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

limit (p. 31)
The maximum number of items returned. If you don't specify a value, the default is up to 50 items.
Type: Integer
Valid Range: Minimum value of 1. Maximum value of 50.
Required: No

logGroupNamePrefix (p. 31)
The prefix to match.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: No

nextToken (p. 31)
The token for the next set of items to return. (You received this token from a previous call.)
Type: String
Length Constraints: Minimum length of 1.
Required: No

Response Syntax

```
{
   "logGroups": [
      {
         "arn": "string",
         "creationTime": number,
         "logGroupName": "string",
         "metricFilterCount": number,
         "retentionInDays": number,
         "storedBytes": number
      }
   ],
   "nextToken": "string"
}
```

Response Elements

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

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

logGroups (p. 31)
The log groups.
Type: array of LogGroup (p. 87) objects

nextToken (p. 31)
The token for the next set of items to return. The token expires after 24 hours.
Type: String
Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To list all log groups

The following example lists all your log groups.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeLogGroups
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
```

```
  "logGroups": [
    {
      "storageBytes": 1048576,
      "arn": "arn:aws:logs:us-east-1:123456789012:log-group:my-log-
group-1:*",
      "creationTime": 1393545600000,
      "logGroupName": "my-log-group-1",
      "metricFilterCount": 0,
      "retentionInDays": 14
    },
    {
      "storageBytes": 5242880,
      "arn": "arn:aws:logs:us-east-1:123456789012:log-group:my-log-
group-2:*",
      "creationTime": 1396224000000,
      "logGroupName": "my-log-group-2",
      "metricFilterCount": 0,
      "retentionInDays": 30
    }
  ]
}
```

DescribeLogStreams

Lists the log streams for the specified log group. You can list all the log streams or filter the results by prefix. You can also control how the results are ordered.

This operation has a limit of five transactions per second, after which transactions are throttled.

Request Syntax

```
{
   "descending": boolean,
   "limit": number,
   "logGroupName": "string",
   "logStreamNamePrefix": "string",
   "nextToken": "string",
   "orderBy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

descending (p. 34)

If the value is true, results are returned in descending order. If the value is to false, results are returned in ascending order. The default value is false.

Type: Boolean

Required: No

limit (p. 34)

The maximum number of items returned. If you don't specify a value, the default is up to 50 items.

Type: Integer

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

Required: No

logGroupName (p. 34)

The name of the log group.

Type: String

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

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

Required: Yes

logStreamNamePrefix (p. 34)

The prefix to match.

You cannot specify this parameter if `orderBy` is `LastEventTime`.

Type: String

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

Pattern: `[^:*]*`

Required: No

nextToken (p. 34)

The token for the next set of items to return. (You received this token from a previous call.)

Type: String

Length Constraints: Minimum length of 1.

Required: No

orderBy (p. 34)

If the value is `LogStreamName`, the results are ordered by log stream name. If the value is `LastEventTime`, the results are ordered by the event time. The default value is `LogStreamName`.

If you order the results by event time, you cannot specify the `logStreamNamePrefix` parameter.

Type: String

Valid Values: `LogStreamName | LastEventTime`

Required: No

Response Syntax

```
{
   "logStreams": [
      {
         "arn": "string",
         "creationTime": number,
         "firstEventTimestamp": number,
         "lastEventTimestamp": number,
         "lastIngestionTime": number,
         "logStreamName": "string",
         "storedBytes": number,
         "uploadSequenceToken": "string"
      }
   ],
   "nextToken": "string"
}
```

Response Elements

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

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

logStreams (p. 35)

The log streams.

Type: array of LogStream (p. 88) objects

nextToken (p. 35)

The token for the next set of items to return. The token expires after 24 hours.

Type: String

Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To list the log streams for a log group

The following example lists the log streams associated with the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeLogStreams
{
  "logGroupName": "my-log-group"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "logStreams": [
    {
      "storageBytes": 1048576,
      "arn": "arn:aws:logs:us-east-1:123456789012:log-group:my-log-
group-1:log-stream:my-log-stream-1",
      "creationTime": 1393545600000,
      "firstEventTimestamp": 1393545600000,
      "lastEventTimestamp": 1393567800000,
      "lastIngestionTime": 1393589200000,
      "logStreamName": "my-log-stream-1",
      "uploadSequenceToken":
 "88602967394531410094953670125156212707622379445839968487"
    },
    {
      "storageBytes": 5242880,
      "arn": "arn:aws:logs:us-east-1:123456789012:log-group:my-log-
group-2:log-stream:my-log-stream-2",
      "creationTime": 1396224000000,
      "firstEventTimestamp": 1396224000000,
      "lastEventTimestamp": 1396235500000,
      "lastIngestionTime": 1396225560000,
      "logStreamName": "my-log-stream-2",
```

```
      "uploadSequenceToken":
 "07622379445839968487886029673945314100949536701251562127"
    }
  ]
}
```

DescribeMetricFilters

Lists the specified metric filters. You can list all the metric filters or filter the results by log name, prefix, metric name, or metric namespace. The results are ASCII-sorted by filter name.

Request Syntax

```
{
   "filterNamePrefix": "string",
   "limit": number,
   "logGroupName": "string",
   "metricName": "string",
   "metricNamespace": "string",
   "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

filterNamePrefix (p. 38)
The prefix to match.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: No

limit (p. 38)
The maximum number of items returned. If you don't specify a value, the default is up to 50 items.
Type: Integer
Valid Range: Minimum value of 1. Maximum value of 50.
Required: No

logGroupName (p. 38)
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: No

metricName (p. 38)
The name of the CloudWatch metric.
Type: String
Length Constraints: Maximum length of 255.
Pattern: `[^:*$]*`
Required: No

metricNamespace (p. 38)
The namespace of the CloudWatch metric.
Type: String
Length Constraints: Maximum length of 255.
Pattern: `[^:*$]*`

Required: No

nextToken (p. 38)

The token for the next set of items to return. (You received this token from a previous call.)

Type: String

Length Constraints: Minimum length of 1.

Required: No

Response Syntax

```
{
   "metricFilters": [
      {
         "creationTime": number,
         "filterName": "string",
         "filterPattern": "string",
         "logGroupName": "string",
         "metricTransformations": [
            {
               "defaultValue": number,
               "metricName": "string",
               "metricNamespace": "string",
               "metricValue": "string"
            }
         ]
      }
   ],
   "nextToken": "string"
}
```

Response Elements

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

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

metricFilters (p. 39)

The metric filters.

Type: array of MetricFilter (p. 89) objects

nextToken (p. 39)

The token for the next set of items to return. The token expires after 24 hours.

Type: String

Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To list the metric filters for a log group

The following example lists the metric filters for the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeMetricFilters
{
  "logGroupName": "my-log-group"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "metricFilters": [
    {
      "creationTime": 1396224000000,
      "filterName": "my-metric-filter",
      "filterPattern": "[ip, identity, user_id, timestamp, request,
 status_code, size]",
      "logGroupName": "my-log-group",
      "metricTransformations": [
        {
          "defaultValue": "0",
          "metricValue": "$size",
          "metricNamespace": "my-app",
          "metricName": "Volume"
        }
      ]
    }
  ]
}
```

DescribeSubscriptionFilters

Lists the subscription filters for the specified log group. You can list all the subscription filters or filter the results by prefix. The results are ASCII-sorted by filter name.

Request Syntax

```
{
   "filterNamePrefix": "string",
   "limit": number,
   "logGroupName": "string",
   "nextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

filterNamePrefix (p. 41)
The prefix to match. If you don't specify a value, no prefix filter is applied.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: No

limit (p. 41)
The maximum number of items returned. If you don't specify a value, the default is up to 50 items.
Type: Integer
Valid Range: Minimum value of 1. Maximum value of 50.
Required: No

logGroupName (p. 41)
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: Yes

nextToken (p. 41)
The token for the next set of items to return. (You received this token from a previous call.)
Type: String
Length Constraints: Minimum length of 1.
Required: No

Response Syntax

```
{
   "nextToken": "string",
   "subscriptionFilters": [
      {
         "creationTime": number,
```

```
         "destinationArn": "string",
         "filterName": "string",
         "filterPattern": "string",
         "logGroupName": "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.

nextToken (p. 41)

The token for the next set of items to return. The token expires after 24 hours.

Type: String

Length Constraints: Minimum length of 1.

subscriptionFilters (p. 41)

The subscription filters.

Type: array of SubscriptionFilter (p. 95) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To list the subscription filters for a log group

The following example lists the subscription filters for the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
```

```
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.DescribeSubscriptionFilters
{
  "logGroupName": "my-log-group"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "subscriptionFilters": [
    {
      "creationTime": 1396224000000,
      "logGroupName": "my-log-group",
      "filterName": "my-subscription-ilter",
      "filterPattern": "[ip, identity, user_id, timestamp, request,
 status_code = 500, size]",
      "destinationArn": "arn:aws:kinesis:us-east-1:123456789012:stream/my-
kinesis-stream",
      "roleArn": "arn:aws:iam::123456789012:role/my-subscription-role"
    }
  ]
}
```

FilterLogEvents

Lists log events from the specified log group. You can list all the log events or filter the results using a filter pattern, a time range, and the name of the log stream.

By default, this operation returns as many log events as can fit in 1MB (up to 10,000 log events), or all the events found within the time range that you specify. If the results include a token, then there are more log events available, and you can get additional results by specifying the token in a subsequent call.

Request Syntax

```
{
   "endTime": number,
   "filterPattern": "string",
   "interleaved": boolean,
   "limit": number,
   "logGroupName": "string",
   "logStreamNames": [ "string" ],
   "nextToken": "string",
   "startTime": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

endTime (p. 44)
The end of the time range. Events with a timestamp later than this time are not returned.

Type: Long

Valid Range: Minimum value of 0.

Required: No

filterPattern (p. 44)
The filter pattern to use. If not provided, all the events are matched.

Type: String

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

Required: No

interleaved (p. 44)
If the value is true, the operation makes a best effort to provide responses that contain events from multiple log streams within the log group interleaved in a single response. If the value is false all the matched log events in the first log stream are searched first, then those in the next log stream, and so on. The default is false.

Type: Boolean

Required: No

limit (p. 44)
The maximum number of events to return. The default is 10,000 events.

Type: Integer

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

Required: No

logGroupName (p. 44)
The name of the log group.

Type: String

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

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

Required: Yes

logStreamNames (p. 44)

Optional list of log stream names.

Type: array of Strings

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

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

Pattern: `[^:*]*`

Required: No

nextToken (p. 44)

The token for the next set of events to return. (You received this token from a previous call.)

Type: String

Length Constraints: Minimum length of 1.

Required: No

startTime (p. 44)

The start of the time range. Events with a timestamp prior to this time are not returned.

Type: Long

Valid Range: Minimum value of 0.

Required: No

Response Syntax

```
{
   "events": [
      {
         "eventId": "string",
         "ingestionTime": number,
         "logStreamName": "string",
         "message": "string",
         "timestamp": number
      }
   ],
   "nextToken": "string",
   "searchedLogStreams": [
      {
         "logStreamName": "string",
         "searchedCompletely": boolean
      }
   ]
}
```

Response Elements

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

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

events (p. 45)

The matched events.

Type: array of FilteredLogEvent (p. 85) objects

nextToken (p. 45)
The token to use when requesting the next set of items. The token expires after 24 hours.
Type: String
Length Constraints: Minimum length of 1.

searchedLogStreams (p. 45)
Indicates which log streams have been searched and whether each has been searched completely.
Type: array of SearchedLogStream (p. 94) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To list the events in a log group that contain a pattern

The following example lists the events for the specified log group that contain 'ERROR'.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.FilterLogEvents
{
  "logGroupName": "my-log-group",
  "filterPattern": "ERROR"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
```

```
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "events": [
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378988,
      "message": "ERROR Event 1",
      "logStreamName": "my-log-stream-1",
      "eventId": "31132629274945519779805322857203735586714454643391594505"
    },
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378988,
      "message": "ERROR Event 2",
      "logStreamName": "my-log-stream-2",
      "eventId": "31132629274945519779805322857203735586814454643391594505"
    },
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378989,
      "message": "ERROR Event 3",
      "logStreamName": "my-log-stream-3"
      "eventId": "31132629274945519779805322857203735586824454643391594505"
    }
  ],
  "searchedLogStreams": [
    {
      "searchedCompletely": true,
      "logStreamName": "my-log-stream-1"
    },
    {
      "searchedCompletely": true,
      "logStreamName": "my-log-stream-2"
    },
    {
      "searchedCompletely": false,
      "logStreamName": "my-log-stream-3"
    },
  ],
  "nextToken": "ZNUEPl7FcQuXbIH4Swk9D9eFu2XBg-ijZIZlvzz4ea9zZRjw-
MMtQtvcoMdmq4T29K7Q6Y1e_KvyfpcT_f_tUw"
}
```

GetLogEvents

Lists log events from the specified log stream. You can list all the log events or filter using a time range.

By default, this operation returns as many log events as can fit in a response size of 1MB (up to 10,000 log events). If the results include tokens, there are more log events available. You can get additional log events by specifying one of the tokens in a subsequent call.

Request Syntax

```
{
   "endTime": number,
   "limit": number,
   "logGroupName": "string",
   "logStreamName": "string",
   "nextToken": "string",
   "startFromHead": boolean,
   "startTime": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

endTime (p. 48)

The end of the time range. Events with a timestamp later than this time are not included.

Type: Long

Valid Range: Minimum value of 0.

Required: No

limit (p. 48)

The maximum number of log events returned. If you don't specify a value, the maximum is as many log events as can fit in a response size of 1MB, up to 10,000 log events.

Type: Integer

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

Required: No

logGroupName (p. 48)

The name of the log group.

Type: String

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

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

Required: Yes

logStreamName (p. 48)

The name of the log stream.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

nextToken (p. 48)

The token for the next set of items to return. (You received this token from a previous call.)

Type: String

Length Constraints: Minimum length of 1.

Required: No

startFromHead (p. 48)

If the value is true, the earliest log events are returned first. If the value is false, the latest log events are returned first. The default value is false.

Type: Boolean

Required: No

startTime (p. 48)

The start of the time range. Events with a timestamp earlier than this time are not included.

Type: Long

Valid Range: Minimum value of 0.

Required: No

Response Syntax

```
{
   "events": [
      {
         "ingestionTime": number,
         "message": "string",
         "timestamp": number
      }
   ],
   "nextBackwardToken": "string",
   "nextForwardToken": "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.

events (p. 49)

The events.

Type: array of OutputLogEvent (p. 92) objects

nextBackwardToken (p. 49)

The token for the next set of items in the backward direction. The token expires after 24 hours.

Type: String

Length Constraints: Minimum length of 1.

nextForwardToken (p. 49)

The token for the next set of items in the forward direction. The token expires after 24 hours.

Type: String

Length Constraints: Minimum length of 1.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To list all the events for a log stream

The following example lists all events for the specified log stream.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.GetLogEvents
{
  "logGroupName": "my-log-group",
  "logStreamName": "my-log-stream"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "events": [
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378988,
      "message": "Example event 1"
    },
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378988,
      "message": "Example event 2"
    },
    {
      "ingestionTime": 1396035394997,
      "timestamp": 1396035378989,
      "message": "Example event 3"
```

```
    }
  ],
  "nextBackwardToken":
 "b/311326292749455197798053228572037355867144546433915945 05",
  "nextForwardToken":
 "f/311326293237841517645873875382051322016993977594038845 44"
}
```

PutDestination

Creates or updates a destination. A destination encapsulates a physical resource (such as a Kinesis stream) and enables you to subscribe to a real-time stream of log events of a different account, ingested using PutLogEvents (p. 57). Currently, the only supported physical resource is a Amazon Kinesis stream belonging to the same account as the destination.

A destination controls what is written to its Amazon Kinesis stream through an access policy. By default, `PutDestination` does not set any access policy with the destination, which means a cross-account user cannot call PutSubscriptionFilter (p. 65) against this destination. To enable this, the destination owner must call PutDestinationPolicy (p. 55) after `PutDestination`.

Request Syntax

```
{
   "destinationName": "string",
   "roleArn": "string",
   "targetArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

destinationName (p. 52)

A name for the destination.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

roleArn (p. 52)

The ARN of an IAM role that grants CloudWatch Logs permissions to call Amazon Kinesis PutRecord on the destination stream.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

targetArn (p. 52)

The ARN of an Amazon Kinesis stream to deliver matching log events to.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

Response Syntax

```
{
   "destination": {
      "accessPolicy": "string",
      "arn": "string",
      "creationTime": number,
```

```
      "destinationName": "string",
      "roleArn": "string",
      "targetArn": "string"
   }
}
```

Response Elements

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

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

destination (p. 52)

The destination.

Type: Destination (p. 80) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

OperationAbortedException

Multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To create or update a destination

The following example creates the specified destination.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutDestination
{
  "destinationName": "my-destination",
  "targetArn": "arn:aws:kinesis:us-east-1:123456789012:stream/my-kinesis-
stream",
  "roleArn": "arn:aws:iam::123456789012:role/my-subscription-role"
```

```
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "destination": [
    {
      "destinationName": "my-destination",
      "targetArn": "arn:aws:kinesis:us-east-1:123456789012:stream/my-kinesis-
stream",
      "roleArn": "arn:aws:iam::123456789012:role/my-subscription-role",
      "arn": "arn:aws:logs:us-east-1:123456789012:destination:my-
destination",
      "creationTime": 1437584472382
    }
  ]
}
```

PutDestinationPolicy

Creates or updates an access policy associated with an existing destination. An access policy is an IAM policy document that is used to authorize claims to register a subscription filter against a given destination.

Request Syntax

```
{
   "accessPolicy": "string",
   "destinationName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

accessPolicy (p. 55)

An IAM policy document that authorizes cross-account users to deliver their log events to the associated destination.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

destinationName (p. 55)

A name for an existing destination.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

OperationAbortedException

Multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To create or update an access policy of a destination

The following example updates the access policy of the specified destination.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutDestinationPolicy
{
  "destinationName": "my-destination",
  "accessPolicy": "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Effect
\": \"Allow\", \"Principal\": { \"AWS\": \"logs.us-east-1.amazonaws.com\"},
 \"Action\": \"logs:PutSubscriptionFilter\",\"Resource\": \"arn:aws:logs:us-
east-1:123456789012:destination:my-destination\"}]}"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

PutLogEvents

Uploads a batch of log events to the specified log stream.

You must include the sequence token obtained from the response of the previous call. An upload in a newly created log stream does not require a sequence token. You can also get the sequence token using DescribeLogStreams (p. 34).

The batch of events must satisfy the following constraints:

- The maximum batch size is 1,048,576 bytes, and this size is calculated as the sum of all event messages in UTF-8, plus 26 bytes for each log event.
- None of the log events in the batch can be more than 2 hours in the future.
- None of the log events in the batch can be older than 14 days or the retention period of the log group.
- The log events in the batch must be in chronological ordered by their timestamp.
- The maximum number of log events in a batch is 10,000.
- A batch of log events in a single PutLogEvents request cannot span more than 24 hours. Otherwise, the PutLogEvents operation will fail.

Request Syntax

```
{
   "logEvents": [
      {
         "message": "string",
         "timestamp": number
      }
   ],
   "logGroupName": "string",
   "logStreamName": "string",
   "sequenceToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

logEvents (p. 57)
The log events.
Type: array of InputLogEvent (p. 86) objects
Array Members: Minimum number of 1 item. Maximum number of 10000 items.
Required: Yes

logGroupName (p. 57)
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: Yes

logStreamName (p. 57)
The name of the log stream.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

sequenceToken (p. 57)

The sequence token.

Type: String

Length Constraints: Minimum length of 1.

Required: No

Response Syntax

```
{
   "nextSequenceToken": "string",
   "rejectedLogEventsInfo": {
      "expiredLogEventEndIndex": number,
      "tooNewLogEventStartIndex": number,
      "tooOldLogEventEndIndex": 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.

nextSequenceToken (p. 58)

The next sequence token.

Type: String

Length Constraints: Minimum length of 1.

rejectedLogEventsInfo (p. 58)

The rejected events.

Type: RejectedLogEventsInfo (p. 93) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

DataAlreadyAcceptedException

The event was already logged.

HTTP Status Code: 400

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

InvalidSequenceTokenException

The sequence token is not valid.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To upload log events into a log stream

The following example uploads the specified log events to the specified log stream.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutLogEvents
{
  "logGroupName": "my-log-group",
  "logStreamName": "my-log-stream",
  "logEvents": [
    {
      "timestamp": 1396035378988,
      "message": "Example event 1"
    },
    {
      "timestamp": 1396035378988,
      "message": "Example event 2"
    },
    {
      "timestamp": 1396035378989,
      "message": "Example event 3"
    }
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "nextSequenceToken":
 "49536701251539826331025683274032969384950891766572122113"
}
```

PutMetricFilter

Creates or updates a metric filter and associates it with the specified log group. Metric filters allow you to configure rules to extract metric data from log events ingested through PutLogEvents (p. 57).

The maximum number of metric filters that can be associated with a log group is 100.

Request Syntax

```
{
   "filterName": "string",
   "filterPattern": "string",
   "logGroupName": "string",
   "metricTransformations": [
      {
         "defaultValue": number,
         "metricName": "string",
         "metricNamespace": "string",
         "metricValue": "string"
      }
   ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

filterName (p. 60)

A name for the metric filter.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

filterPattern (p. 60)

A filter pattern for extracting metric data out of ingested log events.

Type: String

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

Required: Yes

logGroupName (p. 60)

The name of the log group.

Type: String

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

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

Required: Yes

metricTransformations (p. 60)

A collection of information needed to define how metric data gets emitted.

Type: array of MetricTransformation (p. 91) objects

Array Members: Fixed number of 1 item.

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

LimitExceededException
You have reached the maximum number of resources that can be created.
HTTP Status Code: 400

OperationAbortedException
Multiple requests to update the same resource were in conflict.
HTTP Status Code: 400

ResourceNotFoundException
The specified resource does not exist.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Example

To create or update a metric filter

The following example creates a metric filter for the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutMetricFilter
{
  "logGroupName": "my-log-group",
  "filterName": "my-metric-filter",
  "filterPattern": "[ip, identity, user_id, timestamp, request, status_code,
 size]",
  "metricTransformations": [
    {
      "defaultValue": "0",
      "metricValue": "$size",
      "metricNamespace": "MyApp",
```

```
      "metricName": "Volume"
    }
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

PutRetentionPolicy

Sets the retention of the specified log group. A retention policy allows you to configure the number of days you want to retain log events in the specified log group.

Request Syntax

```
{
   "logGroupName": "string",
   "retentionInDays": number
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

logGroupName (p. 63)

The name of the log group.

Type: String

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

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

Required: Yes

retentionInDays (p. 63)

The number of days to retain the log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, and 3653.

Type: Integer

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

OperationAbortedException

Multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To create or update a retention policy for a log group

The following example creates a 30-day retention policy for the specified log group.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutRetentionPolicy
{
  "logGroupName": "my-log-group",
  "retentionInDays": 30
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

PutSubscriptionFilter

Creates or updates a subscription filter and associates it with the specified log group. Subscription filters allow you to subscribe to a real-time stream of log events ingested through PutLogEvents (p. 57) and have them delivered to a specific destination. Currently, the supported destinations are:

- An Amazon Kinesis stream belonging to the same account as the subscription filter, for same-account delivery.
- A logical destination that belongs to a different account, for cross-account delivery.
- An Amazon Kinesis Firehose stream that belongs to the same account as the subscription filter, for same-account delivery.
- An AWS Lambda function that belongs to the same account as the subscription filter, for same-account delivery.

There can only be one subscription filter associated with a log group.

Request Syntax

```
{
   "destinationArn": "string",
   "filterName": "string",
   "filterPattern": "string",
   "logGroupName": "string",
   "roleArn": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

destinationArn (p. 65)

The ARN of the destination to deliver matching log events to. Currently, the supported destinations are:

- An Amazon Kinesis stream belonging to the same account as the subscription filter, for same-account delivery.
- A logical destination (specified using an ARN) belonging to a different account, for cross-account delivery.
- An Amazon Kinesis Firehose stream belonging to the same account as the subscription filter, for same-account delivery.
- An AWS Lambda function belonging to the same account as the subscription filter, for same-account delivery.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

filterName (p. 65)

A name for the subscription filter.

Type: String

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

Pattern: `[^:*]*`

Required: Yes

filterPattern (p. 65)

A filter pattern for subscribing to a filtered stream of log events.

Type: String

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

Required: Yes

logGroupName (p. 65)

The name of the log group.

Type: String

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

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

Required: Yes

roleArn (p. 65)

The ARN of an IAM role that grants CloudWatch Logs permissions to deliver ingested log events to the destination stream. You don't need to provide the ARN when you are working with a logical destination for cross-account delivery.

Type: String

Length Constraints: Minimum length of 1.

Required: No

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException

A parameter is specified incorrectly.

HTTP Status Code: 400

LimitExceededException

You have reached the maximum number of resources that can be created.

HTTP Status Code: 400

OperationAbortedException

Multiple requests to update the same resource were in conflict.

HTTP Status Code: 400

ResourceNotFoundException

The specified resource does not exist.

HTTP Status Code: 400

ServiceUnavailableException

The service cannot complete the request.

HTTP Status Code: 500

Example

To create or update a subscription filter

The following example creates a subscription filter.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.PutSubscriptionFilter
{
  "logGroupName": "my-log-group",
  "filterName": "my-subscription-filter",
  "filterPattern": "[ip, identity, user_id, timestamp, request, status_code =
 500, size]",
  "destinationArn": "arn:aws:kinesis:us-east-1:123456789012:stream/my-
kinesis-stream",
  "roleArn": "arn:aws:iam::123456789012:role/my-subscription-role"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
```

TestMetricFilter

Tests the filter pattern of a metric filter against a sample of log event messages. You can use this operation to validate the correctness of a metric filter pattern.

Request Syntax

```
{
   "filterPattern": "string",
   "logEventMessages": [ "string" ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 96).

The request accepts the following data in JSON format.

filterPattern (p. 68)

A symbolic description of how CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

Type: String

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

Required: Yes

logEventMessages (p. 68)

The log event messages to test.

Type: array of Strings

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

Length Constraints: Minimum length of 1.

Required: Yes

Response Syntax

```
{
   "matches": [
      {
         "eventMessage": "string",
         "eventNumber": number,
         "extractedValues": {
            "string" : "string"
         }
      }
   ]
}
```

Response Elements

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

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

matches (p. 68)
The matched events.
Type: array of MetricFilterMatchRecord (p. 90) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 98).

InvalidParameterException
A parameter is specified incorrectly.
HTTP Status Code: 400

ServiceUnavailableException
The service cannot complete the request.
HTTP Status Code: 500

Examples

To test a metric filter pattern on Apache access.log events

The following example tests the specified metric filter pattern.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[ip, identity, user_id, timestamp, request, status_code,
 size]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 200 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 500 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 200 4355"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
```

```
Date: <Date>
{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 200 1534",
      "extractedValues": {
        "$status_code": "200",
        "$identity": "-",
        "$request": "GET /apache_pb.gif HTTP/1.0",
        "$size": "1534,",
        "$user_id": "frank",
        "$ip": "127.0.0.1",
        "$timestamp": "10/Oct/2000:13:25:15 -0700"
      }
    },
    {
      "eventNumber": 1,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 500 5324",
      "extractedValues": {
        "$status_code": "500",
        "$identity": "-",
        "$request": "GET /apache_pb.gif HTTP/1.0",
        "$size": "5324,",
        "$user_id": "frank",
        "$ip": "127.0.0.1",
        "$timestamp": "10/Oct/2000:13:35:22 -0700"
      }
    },
    {
      "eventNumber": 2,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 200 4355",
      "extractedValues": {
        "$status_code": "200",
        "$identity": "-",
        "$request": "GET /apache_pb.gif HTTP/1.0",
        "$size": "4355",
        "$user_id": "frank",
        "$ip": "127.0.0.1",
        "$timestamp": "10/Oct/2000:13:50:35 -0700"
      }
    }
  ]
}
```

To test a metric filter pattern on Apache access.log events without specifying all the fields

The following example tests the specified metric filter pattern.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
```

```
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[..., size]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 200 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 500 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 200 4355"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 200 1534",
      "extractedValues": {
        "$size": "1534",
        "$6": "200",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 1,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 500 5324",
      "extractedValues": {
        "$size": "5324",
        "$6": "500",
        "$4": "10/Oct/2000:13:35:22 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
```

```
    {
      "eventNumber": 2,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 200 4355",
      "extractedValues": {
        "$size": "4355",
        "$6": "200",
        "$4": "10/Oct/2000:13:50:35 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    }
  ]
}
```

To test a metric filter pattern on Apache access.log events without specifying any fields

The following example tests the specified metric filter pattern.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 200 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 500 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 200 4355"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "matches": [
```

```
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 200 1534",
      "extractedValues": {
        "$7": "1534",
        "$6": "200",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 1,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 500 5324",
      "extractedValues": {
        "$7": "5324",
        "$6": "500",
        "$4": "10/Oct/2000:13:35:22 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 2,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 200 4355",
      "extractedValues": {
        "$7": "4355",
        "$6": "200",
        "$4": "10/Oct/2000:13:50:35 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    }
  ]
}
```

To test a metric filter pattern that matches successful requests in Apache access.log events

The following example tests the specified metric filter pattern.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
```

```
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[..., status_code=200, size]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 200 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 500 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 200 4355"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 200 1534",
      "extractedValues": {
        "$status_code": "200",
        "$size": "1534",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
      "eventNumber": 2,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /
apache_pb.gif HTTP/1.0\" 200 4355",
      "extractedValues": {
        "$status_code": "200",
        "$size": "4355",
        "$4": "10/Oct/2000:13:50:35 -0700",
        "$5": "GET /apache_pb.gif HTTP/1.0",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    }
  ]
}
```

To test a metric filter pattern that matches 4XX response codes for html pages in Apache access.log events

The following example tests the specified metric filter pattern.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "[..., request=*.html*, status_code=4*,]",
  "logEventMessages": [
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /index.html
 HTTP/1.0\" 404 1534",
    "127.0.0.1 - frank [10/Oct/2000:13:35:22 -0700] \"GET /about-us/
index.html HTTP/1.0\" 200 5324",
    "127.0.0.1 - frank [10/Oct/2000:13:50:35 -0700] \"GET /apache_pb.gif
 HTTP/1.0\" 404 4355",
    "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /products/
index.html HTTP/1.0\" 400 1534",
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "matches": [
    {
      "eventNumber": 0,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /
index.html HTTP/1.0\" 404 1534",
      "extractedValues": {
        "$status_code": "404",
        "$request": "GET /index.html HTTP/1.0",
        "$7": "1534",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    },
    {
```

```
      "eventNumber": 3,
      "eventMessage": "127.0.0.1 - frank [10/Oct/2000:13:25:15 -0700] \"GET /
products/index.html HTTP/1.0\" 400 1534",
      "extractedValues": {
        "$status_code": "400",
        "$request": "GET /products/index.html HTTP/1.0",
        "$7": "1534",
        "$4": "10/Oct/2000:13:25:15 -0700",
        "$2": "-",
        "$3": "frank",
        "$1": "127.0.0.1"
      }
    }
  ]
}
```

To test a metric filter pattern that matches occurrences of "[ERROR]" in log events

The following example tests the specified metric filter pattern.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "\"[ERROR]\"",
  "logEventMessages": [
    "02 May 2014 00:34:12,525 [INFO] Starting the application",
    "02 May 2014 00:35:14,245 [DEBUG] Database connection established",
    "02 May 2014 00:34:14,663 [INFO] Executing SQL Query",
    "02 May 2014 00:34:16,142 [ERROR] Unhanded exception:
 InvalidQueryException",
    "02 May 2014 00:34:16,224 [ERROR] Terminating the application"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "matches": [
```

```
    {
      "eventNumber": 3,
      "eventMessage": "02 May 2014 00:34:16,142 [ERROR] Unhanded exception:
 InvalidQueryException",
      "extractedValues": {}
    },
    {
      "eventNumber": 4,
      "eventMessage": "02 May 2014 00:34:16,224 [ERROR] Terminating the
 application",
      "extractedValues": {}
    }
  ]
}
```

To test a metric filter pattern that matches occurrences of "[ERROR]" and "Exception" in log events

The following example tests the specified metric filter pattern.

Sample Request

```
POST / HTTP/1.1
Host: logs.<region>.<domain>
X-Amz-Date: <DATE>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>,
 SignedHeaders=content-type;date;host;user-agent;x-amz-date;x-amz-target;x-
amzn-requestid, Signature=<Signature>
User-Agent: <UserAgentString>
Accept: application/json
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Logs_20140328.TestMetricFilter
{
  "filterPattern": "\"[ERROR]\" Exception",
  "logEventMessages": [
    "02 May 2014 00:34:12,525 [INFO] Starting the application",
    "02 May 2014 00:35:14,245 [DEBUG] Database connection established",
    "02 May 2014 00:34:14,663 [INFO] Executing SQL Query",
    "02 May 2014 00:34:16,142 [ERROR] Unhanded exception:
 InvalidQueryException",
    "02 May 2014 00:34:16,224 [ERROR] Terminating the application"
  ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "matches": [
    {
      "eventNumber": 3,
```

```
      "eventMessage": "02 May 2014 00:34:16,142 [ERROR] Unhanded exception:
 InvalidQueryException",
      "extractedValues": {}
    }
  ]
}
```

Data Types

The Amazon CloudWatch Logs API contains several data types that various actions use. This section describes each data type in detail.

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

The following data types are supported:

Destination

Represents a cross-account destination that receives subscription log events.

Contents

accessPolicy

An IAM policy document that governs which AWS accounts can create subscription filters against this destination.

Type: String

Length Constraints: Minimum length of 1.

Required: No

arn

The ARN of this destination.

Type: String

Required: No

creationTime

The creation time of the destination.

Type: Long

Valid Range: Minimum value of 0.

Required: No

destinationName

The name of the destination.

Type: String

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

Pattern: `[^:*]*`

Required: No

roleArn

A role for impersonation, used when delivering log events to the target.

Type: String

Length Constraints: Minimum length of 1.

Required: No

targetArn

The Amazon Resource Name (ARN) of the physical target where the log events will be delivered (for example, a Kinesis stream).

Type: String

Length Constraints: Minimum length of 1.

Required: No

ExportTask

Represents an export task.

Contents

destination
The name of Amazon S3 bucket to which the log data was exported.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Required: No

destinationPrefix
The prefix that was used as the start of Amazon S3 key for every object exported.
Type: String
Required: No

executionInfo
Execution info about the export task.
Type: ExportTaskExecutionInfo (p. 83) object
Required: No

from
The start time. Events with a timestamp prior to this time are not exported.
Type: Long
Valid Range: Minimum value of 0.
Required: No

logGroupName
The name of the log group from which logs data was exported.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: No

status
The status of the export task.
Type: ExportTaskStatus (p. 84) object
Required: No

taskId
The ID of the export task.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Required: No

taskName
The name of the export task.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Required: No

to
The end time. Events with a timestamp later than this time are not exported.
Type: Long
Valid Range: Minimum value of 0.
Required: No

ExportTaskExecutionInfo

Represents the status of an export task.

Contents

completionTime

The completion time of the export task.

Type: Long

Valid Range: Minimum value of 0.

Required: No

creationTime

The creation time of the export task.

Type: Long

Valid Range: Minimum value of 0.

Required: No

ExportTaskStatus

Represents the status of an export task.

Contents

code

The status code of the export task.

Type: String

Valid Values: `CANCELLED | COMPLETED | FAILED | PENDING | PENDING_CANCEL | RUNNING`

Required: No

message

The status message related to the status code.

Type: String

Required: No

FilteredLogEvent

Represents a matched event.

Contents

eventId
The ID of the event.
Type: String
Required: No

ingestionTime
The time the event was ingested.
Type: Long
Valid Range: Minimum value of 0.
Required: No

logStreamName
The name of the log stream this event belongs to.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: No

message
The data contained in the log event.
Type: String
Length Constraints: Minimum length of 1.
Required: No

timestamp
The time the event occurred.
Type: Long
Valid Range: Minimum value of 0.
Required: No

InputLogEvent

Represents a log event, which is a record of activity that was recorded by the application or resource being monitored.

Contents

message

The raw event message.

Type: String

Length Constraints: Minimum length of 1.

Required: Yes

timestamp

The time the event occurred.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

LogGroup

Represents a log group.

Contents

arn
The Amazon Resource Name (ARN) of the log group.
Type: String
Required: No

creationTime
The creation time of the log group.
Type: Long
Valid Range: Minimum value of 0.
Required: No

logGroupName
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: No

metricFilterCount
The number of metric filters.
Type: Integer
Required: No

retentionInDays
The number of days to retain the log events in the specified log group. Possible values are: 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, and 3653.
Type: Integer
Required: No

storedBytes
The number of bytes stored.
Type: Long
Valid Range: Minimum value of 0.
Required: No

LogStream

Represents a log stream, which is a sequence of log events from a single emitter of logs.

Contents

arn
The Amazon Resource Name (ARN) of the log stream.
Type: String
Required: No

creationTime
The creation time of the stream.
Type: Long
Valid Range: Minimum value of 0.
Required: No

firstEventTimestamp
The time of the first event.
Type: Long
Valid Range: Minimum value of 0.
Required: No

lastEventTimestamp
The time of the last event.
Type: Long
Valid Range: Minimum value of 0.
Required: No

lastIngestionTime
The ingestion time.
Type: Long
Valid Range: Minimum value of 0.
Required: No

logStreamName
The name of the log stream.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: No

storedBytes
The number of bytes stored.
Type: Long
Valid Range: Minimum value of 0.
Required: No

uploadSequenceToken
The sequence token.
Type: String
Length Constraints: Minimum length of 1.
Required: No

MetricFilter

Metric filters express how CloudWatch Logs would extract metric observations from ingested log events and transform them into metric data in a CloudWatch metric.

Contents

creationTime
The creation time of the metric filter.
Type: Long
Valid Range: Minimum value of 0.
Required: No

filterName
The name of the metric filter.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: No

filterPattern
A symbolic description of how CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.
Type: String
Length Constraints: Minimum length of 0. Maximum length of 1024.
Required: No

logGroupName
The name of the log group.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[\.\-_/#A-Za-z0-9]+`
Required: No

metricTransformations
The metric transformations.
Type: array of MetricTransformation (p. 91) objects
Array Members: Fixed number of 1 item.
Required: No

MetricFilterMatchRecord

Represents a matched event.

Contents

eventMessage
The raw event data.
Type: String
Length Constraints: Minimum length of 1.
Required: No

eventNumber
The event number.
Type: Long
Required: No

extractedValues
The values extracted from the event data by the filter.
Type: String to String map
Required: No

MetricTransformation

Indicates how to transform ingested log events into metric data in a CloudWatch metric.

Contents

defaultValue

(Optional) The value to emit when a filter pattern does not match a log event. This value can be null.

Type: Double

Required: No

metricName

The name of the CloudWatch metric.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `[^:*$]*`

Required: Yes

metricNamespace

The namespace of the CloudWatch metric.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `[^:*$]*`

Required: Yes

metricValue

The value to publish to the CloudWatch metric when a filter pattern matches a log event.

Type: String

Length Constraints: Maximum length of 100.

Required: Yes

OutputLogEvent

Represents a log event.

Contents

ingestionTime
The time the event was ingested.
Type: Long
Valid Range: Minimum value of 0.
Required: No

message
The data contained in the log event.
Type: String
Length Constraints: Minimum length of 1.
Required: No

timestamp
The time the event occurred.
Type: Long
Valid Range: Minimum value of 0.
Required: No

RejectedLogEventsInfo

Represents the rejected events.

Contents

expiredLogEventEndIndex
The expired log events.
Type: Integer
Required: No

tooNewLogEventStartIndex
The log events that are too new.
Type: Integer
Required: No

tooOldLogEventEndIndex
The log events that are too old.
Type: Integer
Required: No

SearchedLogStream

Represents the search status of a log stream.

Contents

logStreamName
The name of the log stream.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 512.
Pattern: `[^:*]*`
Required: No

searchedCompletely
Indicates whether all the events in this log stream were searched.
Type: Boolean
Required: No

SubscriptionFilter

Represents a subscription filter.

Contents

creationTime

The creation time of the subscription filter.

Type: Long

Valid Range: Minimum value of 0.

Required: No

destinationArn

The Amazon Resource Name (ARN) of the destination.

Type: String

Length Constraints: Minimum length of 1.

Required: No

filterName

The name of the subscription filter.

Type: String

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

Pattern: `[^:*]*`

Required: No

filterPattern

A symbolic description of how CloudWatch Logs should interpret the data in each log event. For example, a log event may contain timestamps, IP addresses, strings, and so on. You use the filter pattern to specify what to look for in the log event message.

Type: String

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

Required: No

logGroupName

The name of the log group.

Type: String

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

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

Required: No

roleArn

Type: String

Length Constraints: Minimum length of 1.

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

Common Errors

This section lists the common errors that all actions return. Any action-specific errors are listed in the topic for the action.

IncompleteSignature
The request signature does not conform to AWS standards.

HTTP Status Code: 400

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

HTTP Status Code: 500

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

HTTP Status Code: 400

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

HTTP Status Code: 403

InvalidParameterCombination
Parameters that must not be used together were used together.

HTTP Status Code: 400

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

HTTP Status Code: 400

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

HTTP Status Code: 400

MalformedQueryString
The query string contains a syntax error.

HTTP Status Code: 404

MissingAction
The request is missing an action or a required parameter.

HTTP Status Code: 400

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

HTTP Status Code: 403

MissingParameter
A required parameter for the specified action is not supplied.

HTTP Status Code: 400

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

HTTP Status Code: 403

RequestExpired
The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable
The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

Throttling
The request was denied due to request throttling.

HTTP Status Code: 400

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

HTTP Status Code: 400