Amazon Kinesis Data Streams Service API Reference

API Reference

API Version 2013-12-02

aws

Amazon Kinesis Data Streams Service API Reference: API Reference

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Table of Contents

Welcome

Amazon Kinesis Data Streams is a managed service that scales elastically for real-time processing of streaming big data.

This document was last published on August 20, 2019.

Actions

The following actions are supported:

AddTagsToStream

Adds or updates tags for the specified Kinesis data stream. Each time you invoke this operation, you can specify up to 10 tags. If you want to add more than 10 tags to your stream, you can invoke this operation multiple times. In total, each stream can have up to 50 tags.

If tags have already been assigned to the stream, `AddTagsToStream` overwrites any existing tags that correspond to the specified tag keys.

AddTagsToStream (p. 3) has a limit of five transactions per second per account.

Request Syntax

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

Request Parameters

The request accepts the following data in JSON format.

StreamName (p. 3)

>   The name of the stream.
>
>   Type: String
>
>   Length Constraints: Minimum length of 1. Maximum length of 128.
>
>   Pattern: `[a-zA-Z0-9_.-]+`
>
>   Required: Yes

Tags (p. 3)

>   A set of up to 10 key-value pairs to use to create the tags.
>
>   Type: String to string map
>
>   Key Length Constraints: Minimum length of 1. Maximum length of 128.
>
>   Value Length Constraints: Minimum length of 0. Maximum length of 256.
>
>   Required: Yes

Response 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. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To add tags to a stream

The following JSON example adds two tags to the specified stream.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.AddTagsToStream
{
  "StreamName": "exampleStreamName",
  "Tags": {
     "Project" : "myProject",
     "Environment" : "Production"
   }
}
```

Sample Response

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

See Also

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

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

CreateStream

Creates a Kinesis data stream. A stream captures and transports data records that are continuously emitted from different data sources or *producers*. Scale-out within a stream is explicitly supported by means of shards, which are uniquely identified groups of data records in a stream.

You specify and control the number of shards that a stream is composed of. Each shard can support reads up to five transactions per second, up to a maximum data read total of 2 MiB per second. Each shard can support writes up to 1,000 records per second, up to a maximum data write total of 1 MiB per second. If the amount of data input increases or decreases, you can add or remove shards.

The stream name identifies the stream. The name is scoped to the AWS account used by the application. It is also scoped by AWS Region. That is, two streams in two different accounts can have the same name, and two streams in the same account, but in two different Regions, can have the same name.

`CreateStream` is an asynchronous operation. Upon receiving a `CreateStream` request, Kinesis Data Streams immediately returns and sets the stream status to `CREATING`. After the stream is created, Kinesis Data Streams sets the stream status to `ACTIVE`. You should perform read and write operations only on an `ACTIVE` stream.

You receive a `LimitExceededException` when making a `CreateStream` request when you try to do one of the following:

- Have more than five streams in the `CREATING` state at any point in time.
- Create more shards than are authorized for your account.

For the default shard limit for an AWS account, see Amazon Kinesis Data Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*. To increase this limit, contact AWS Support.

You can use `DescribeStream` to check the stream status, which is returned in `StreamStatus`.

CreateStream (p. 6) has a limit of five transactions per second per account.

Request Syntax

```
{
    "ShardCount": number,
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ShardCount (p. 6)

The number of shards that the stream will use. The throughput of the stream is a function of the number of shards; more shards are required for greater provisioned throughput.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100000.

Required: Yes

StreamName (p. 6)

A name to identify the stream. The stream name is scoped to the AWS account used by the application that creates the stream. It is also scoped by AWS Region. That is, two streams in two different AWS accounts can have the same name. Two streams in the same AWS account but in two different Regions can also have the same name.

Type: String

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

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

Example

To create a stream

The following JSON example creates a stream with three shards.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
```

```
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.CreateStream
{
    "StreamName": "exampleStreamName",
    "ShardCount":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>
```

See Also

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

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

DecreaseStreamRetentionPeriod

Decreases the Kinesis data stream's retention period, which is the length of time data records are accessible after they are added to the stream. The minimum value of a stream's retention period is 24 hours.

This operation may result in lost data. For example, if the stream's retention period is 48 hours and is decreased to 24 hours, any data already in the stream that is older than 24 hours is inaccessible.

Request Syntax

```
{
    "RetentionPeriodHours": number,
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

RetentionPeriodHours (p. 9)

The new retention period of the stream, in hours. Must be less than the current retention period.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 168.

Required: Yes

StreamName (p. 9)

The name of the stream to modify.

Type: String

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

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To decrease stream retention period

The following JSON example decreases a stream's retention period.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.DecreaseStreamRetentionPeriod
{
    "RetentionPeriodInHours": "24",
    "StreamName": "examplestream"
}
```

Sample Response

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

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

DeleteStream

Deletes a Kinesis data stream and all its shards and data. You must shut down any applications that are operating on the stream before you delete the stream. If an application attempts to operate on a deleted stream, it receives the exception `ResourceNotFoundException`.

If the stream is in the `ACTIVE` state, you can delete it. After a `DeleteStream` request, the specified stream is in the `DELETING` state until Kinesis Data Streams completes the deletion.

Note: Kinesis Data Streams might continue to accept data read and write operations, such as PutRecord (p. 66), PutRecords (p. 71), and GetRecords (p. 36), on a stream in the `DELETING` state until the stream deletion is complete.

When you delete a stream, any shards in that stream are also deleted, and any tags are dissociated from the stream.

You can use the DescribeStream (p. 19) operation to check the state of the stream, which is returned in `StreamStatus`.

DeleteStream (p. 12) has a limit of five transactions per second per account.

Request Syntax

```
{
    "EnforceConsumerDeletion": boolean,
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

EnforceConsumerDeletion (p. 12)

> If this parameter is unset (`null`) or if you set it to `false`, and the stream has registered consumers, the call to `DeleteStream` fails with a `ResourceInUseException`.
>
> Type: Boolean
>
> Required: No

StreamName (p. 12)

> The name of the stream to delete.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 128.
>
> Pattern: `[a-zA-Z0-9_.-]+`
>
> Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To delete a stream

The following JSON example deletes the specified stream.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.DeleteStream
{
    "StreamName":"exampleStreamName"
}
```

Sample Response

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

See Also

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

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

DeregisterStreamConsumer

To deregister a consumer, provide its ARN. Alternatively, you can provide the ARN of the data stream and the name you gave the consumer when you registered it. You may also provide all three parameters, as long as they don't conflict with each other. If you don't know the name or ARN of the consumer that you want to deregister, you can use the ListStreamConsumers (p. 53) operation to get a list of the descriptions of all the consumers that are currently registered with a given data stream. The description of a consumer contains its name and ARN.

This operation has a limit of five transactions per second per stream.

Request Syntax

```
{
    "ConsumerARN": "string",
    "ConsumerName": "string",
    "StreamARN": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ConsumerARN (p. 15)

The ARN returned by Kinesis Data Streams when you registered the consumer. If you don't know the ARN of the consumer that you want to deregister, you can use the ListStreamConsumers operation to get a list of the descriptions of all the consumers that are currently registered with a given data stream. The description of a consumer contains its ARN.

Type: String

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

Pattern: `^(arn):aws.*:kinesis:.*:\d{12}:.*stream\/[a-zA-Z0-9_.-]+\/consumer\/[a-zA-Z0-9_.-]+:[0-9]+`

Required: No

ConsumerName (p. 15)

The name that you gave to the consumer.

Type: String

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

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

Required: No

StreamARN (p. 15)

The ARN of the Kinesis data stream that the consumer is registered with. For more information, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

Type: String

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

Pattern: `arn:aws.*:kinesis:.*:\d{12}:stream/.+`

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. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

See Also

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

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

DescribeLimits

Describes the shard limits and usage for the account.

If you update your account limits, the old limits might be returned for a few minutes.

This operation has a limit of one transaction per second per account.

Response Syntax

```
{
    "OpenShardCount": number,
    "ShardLimit": 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.

OpenShardCount (p. 17)

> The number of open shards.
>
> Type: Integer
>
> Valid Range: Minimum value of 0. Maximum value of 1000000.

ShardLimit (p. 17)

> The maximum number of shards.
>
> Type: Integer
>
> Valid Range: Minimum value of 0. Maximum value of 1000000.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

LimitExceededException

> The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.
>
> HTTP Status Code: 400

Example

To display the shard limits for the account

The following example displays the shard limits for the account.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.DescribeLimits
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "OpenShardCount": 20,
    "ShardLimit": 70
}
```

See Also

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

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

DescribeStream

Describes the specified Kinesis data stream.

The information returned includes the stream name, Amazon Resource Name (ARN), creation time, enhanced metric configuration, and shard map. The shard map is an array of shard objects. For each shard object, there is the hash key and sequence number ranges that the shard spans, and the IDs of any earlier shards that played in a role in creating the shard. Every record ingested in the stream is identified by a sequence number, which is assigned when the record is put into the stream.

You can limit the number of shards returned by each call. For more information, see Retrieving Shards from a Stream in the *Amazon Kinesis Data Streams Developer Guide*.

There are no guarantees about the chronological order shards returned. To process shards in chronological order, use the ID of the parent shard to track the lineage to the oldest shard.

This operation has a limit of 10 transactions per second per account.

Request Syntax

```
{
    "ExclusiveStartShardId": "string",
    "Limit": number,
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ExclusiveStartShardId (p. 19)

> The shard ID of the shard to start with.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 128.
>
> Pattern: `[a-zA-Z0-9_.-]+`
>
> Required: No

Limit (p. 19)

> The maximum number of shards to return in a single call. The default value is 100. If you specify a value greater than 100, at most 100 shards are returned.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 10000.
>
> Required: No

StreamName (p. 19)

> The name of the stream to describe.
>
> Type: String

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

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

Required: Yes

Response Syntax

```
{
    "StreamDescription": {
        "EncryptionType": "string",
        "EnhancedMonitoring": [
            {
                "ShardLevelMetrics": [ "string" ]
            }
        ],
        "HasMoreShards": boolean,
        "KeyId": "string",
        "RetentionPeriodHours": number,
        "Shards": [
            {
                "AdjacentParentShardId": "string",
                "HashKeyRange": {
                    "EndingHashKey": "string",
                    "StartingHashKey": "string"
                },
                "ParentShardId": "string",
                "SequenceNumberRange": {
                    "EndingSequenceNumber": "string",
                    "StartingSequenceNumber": "string"
                },
                "ShardId": "string"
            }
        ],
        "StreamARN": "string",
        "StreamCreationTimestamp": number,
        "StreamName": "string",
        "StreamStatus": "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.

StreamDescription (p. 20)

The current status of the stream, the stream Amazon Resource Name (ARN), an array of shard objects that comprise the stream, and whether there are more shards available.

Type: StreamDescription (p. 116) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To get information about a stream

The following JSON example describes the specified stream.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.DescribeStream
{
    "StreamName":"exampleStreamName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "StreamDescription": {
    "EnhancedMonitoring":[
        {
            "ShardLevelMetrics":[
                "IncomingBytes",
                "OutgoingRecords"
            ]
        }
    ],
    "HasMoreShards": false,
    "RetentionPeriodHours": 24,
    "StreamCreationTimestamp": 1.468346745E9,
    "Shards": [
      {
        "HashKeyRange": {
          "EndingHashKey": "113427455640312821154458202477256070484",
          "StartingHashKey": "0"
        },
```

```
      "SequenceNumberRange": {
        "EndingSequenceNumber": "21269319989741826081360214168359141376",
        "StartingSequenceNumber": "21267647932558653966460912964485513216"
      },
      "ShardId": "shardId-000000000000"
    },
    {
      "HashKeyRange": {
        "EndingHashKey": "226854911280625642308916404954512140969",
        "StartingHashKey": "113427455640312821154458202477256070485"
      },
      "SequenceNumberRange": {
        "StartingSequenceNumber": "21267647932558653966460912964485513217"
      },
      "ShardId": "shardId-000000000001"
    },
    {
      "HashKeyRange": {
        "EndingHashKey": "340282366920938463463374607431768211455",
        "StartingHashKey": "226854911280625642308916404954512140970"
      },
      "SequenceNumberRange": {
        "StartingSequenceNumber": "21267647932558653966460912964485513218"
      },
      "ShardId": "shardId-000000000002"
    }
  ],
  "StreamARN": "arn:aws:kinesis:us-east-1:111122223333:exampleStreamName",
  "StreamName": "exampleStreamName",
  "StreamStatus": "ACTIVE"
  }
}
```

See Also

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

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

DescribeStreamConsumer

To get the description of a registered consumer, provide the ARN of the consumer. Alternatively, you can provide the ARN of the data stream and the name you gave the consumer when you registered it. You may also provide all three parameters, as long as they don't conflict with each other. If you don't know the name or ARN of the consumer that you want to describe, you can use the ListStreamConsumers (p. 53) operation to get a list of the descriptions of all the consumers that are currently registered with a given data stream.

This operation has a limit of 20 transactions per second per stream.

Request Syntax

```
{
    "ConsumerARN": "string",
    "ConsumerName": "string",
    "StreamARN": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ConsumerARN (p. 23)

The ARN returned by Kinesis Data Streams when you registered the consumer.

Type: String

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

Pattern: `^(arn):aws.*:kinesis:.*:\d{12}:.*stream\/[a-zA-Z0-9_.-]+\/consumer\/[a-zA-Z0-9_.-]+:[0-9]+`

Required: No

ConsumerName (p. 23)

The name that you gave to the consumer.

Type: String

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

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

Required: No

StreamARN (p. 23)

The ARN of the Kinesis data stream that the consumer is registered with. For more information, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

Type: String

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

Pattern: `arn:aws.*:kinesis:.*:\d{12}:stream/.+`

Required: No

Response Syntax

```
{
    "ConsumerDescription": {
        "ConsumerARN": "string",
        "ConsumerCreationTimestamp": number,
        "ConsumerName": "string",
        "ConsumerStatus": "string",
        "StreamARN": "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.

ConsumerDescription (p. 24)

An object that represents the details of the consumer.

Type: ConsumerDescription (p. 102) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET

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

DescribeStreamSummary

Provides a summarized description of the specified Kinesis data stream without the shard list.

The information returned includes the stream name, Amazon Resource Name (ARN), status, record retention period, approximate creation time, monitoring, encryption details, and open shard count.

DescribeStreamSummary (p. 26) has a limit of 20 transactions per second per account.

Request Syntax

```
{
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

StreamName (p. 26)

The name of the stream to describe.

Type: String

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

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

Required: Yes

Response Syntax

```
{
    "StreamDescriptionSummary": {
        "ConsumerCount": number,
        "EncryptionType": "string",
        "EnhancedMonitoring": [
            {
                "ShardLevelMetrics": [ "string" ]
            }
        ],
        "KeyId": "string",
        "OpenShardCount": number,
        "RetentionPeriodHours": number,
        "StreamARN": "string",
        "StreamCreationTimestamp": number,
        "StreamName": "string",
        "StreamStatus": "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.

StreamDescriptionSummary (p. 26)

A StreamDescriptionSummary (p. 119) containing information about the stream.

Type: StreamDescriptionSummary (p. 119) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

See Also

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

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

DisableEnhancedMonitoring

Disables enhanced monitoring.

Request Syntax

```
{
    "ShardLevelMetrics": [ "string" ],
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ShardLevelMetrics (p. 28)

List of shard-level metrics to disable.

The following are the valid shard-level metrics. The value "`ALL`" disables every metric.

- `IncomingBytes`
- `IncomingRecords`
- `OutgoingBytes`
- `OutgoingRecords`
- `WriteProvisionedThroughputExceeded`
- `ReadProvisionedThroughputExceeded`
- `IteratorAgeMilliseconds`
- `ALL`

For more information, see Monitoring the Amazon Kinesis Data Streams Service with Amazon CloudWatch in the *Amazon Kinesis Data Streams Developer Guide*.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 7 items.

Valid Values: `IncomingBytes` | `IncomingRecords` | `OutgoingBytes` | `OutgoingRecords` | `WriteProvisionedThroughputExceeded` | `ReadProvisionedThroughputExceeded` | `IteratorAgeMilliseconds` | `ALL`

Required: Yes

StreamName (p. 28)

The name of the Kinesis data stream for which to disable enhanced monitoring.

Type: String

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

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

Required: Yes

Response Syntax

```
{
   "CurrentShardLevelMetrics": [ "string" ],
   "DesiredShardLevelMetrics": [ "string" ],
   "StreamName": "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.

CurrentShardLevelMetrics (p. 29)

Represents the current state of the metrics that are in the enhanced state before the operation.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 7 items.

Valid Values: `IncomingBytes | IncomingRecords | OutgoingBytes | OutgoingRecords | WriteProvisionedThroughputExceeded | ReadProvisionedThroughputExceeded | IteratorAgeMilliseconds | ALL`

DesiredShardLevelMetrics (p. 29)

Represents the list of all the metrics that would be in the enhanced state after the operation.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 7 items.

Valid Values: `IncomingBytes | IncomingRecords | OutgoingBytes | OutgoingRecords | WriteProvisionedThroughputExceeded | ReadProvisionedThroughputExceeded | IteratorAgeMilliseconds | ALL`

StreamName (p. 29)

The name of the Kinesis data stream.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To disable enhanced monitoring

The following JSON example disables enhanced monitoring for `IncomingBytes` and `OutgoingRecords` shard level metrics.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.DisableEnhancedMonitoring
{
    "ShardLevelMetrics": [
        "IncomingBytes", "OutgoingRecords"
    ],
    "StreamName": "exampleStreamName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "StreamName": "exampleStreamName",
    "CurrentShardLevelMetrics": [
        "IncomingBytes",
        "OutgoingRecords"
    ],
    "DesiredShardLevelMetrics": []
}
```

See Also

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

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

EnableEnhancedMonitoring

Enables enhanced Kinesis data stream monitoring for shard-level metrics.

Request Syntax

```
{
    "ShardLevelMetrics": [ "string" ],
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ShardLevelMetrics (p. 32)

> List of shard-level metrics to enable.
>
> The following are the valid shard-level metrics. The value "`ALL`" enables every metric.
>
> - `IncomingBytes`
> - `IncomingRecords`
> - `OutgoingBytes`
> - `OutgoingRecords`
> - `WriteProvisionedThroughputExceeded`
> - `ReadProvisionedThroughputExceeded`
> - `IteratorAgeMilliseconds`
> - `ALL`
>
> For more information, see Monitoring the Amazon Kinesis Data Streams Service with Amazon CloudWatch in the *Amazon Kinesis Data Streams Developer Guide*.
>
> Type: Array of strings
>
> Array Members: Minimum number of 1 item. Maximum number of 7 items.
>
> Valid Values: `IncomingBytes | IncomingRecords | OutgoingBytes | OutgoingRecords | WriteProvisionedThroughputExceeded | ReadProvisionedThroughputExceeded | IteratorAgeMilliseconds | ALL`
>
> Required: Yes

StreamName (p. 32)

> The name of the stream for which to enable enhanced monitoring.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 128.
>
> Pattern: `[a-zA-Z0-9_.-]+`
>
> Required: Yes

Response Syntax

```
{
   "CurrentShardLevelMetrics": [ "string" ],
   "DesiredShardLevelMetrics": [ "string" ],
   "StreamName": "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.

CurrentShardLevelMetrics (p. 33)

Represents the current state of the metrics that are in the enhanced state before the operation.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 7 items.

Valid Values: `IncomingBytes` | `IncomingRecords` | `OutgoingBytes` | `OutgoingRecords`
| `WriteProvisionedThroughputExceeded` | `ReadProvisionedThroughputExceeded` |
`IteratorAgeMilliseconds` | `ALL`

DesiredShardLevelMetrics (p. 33)

Represents the list of all the metrics that would be in the enhanced state after the operation.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 7 items.

Valid Values: `IncomingBytes` | `IncomingRecords` | `OutgoingBytes` | `OutgoingRecords`
| `WriteProvisionedThroughputExceeded` | `ReadProvisionedThroughputExceeded` |
`IteratorAgeMilliseconds` | `ALL`

StreamName (p. 33)

The name of the Kinesis data stream.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more
information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To enable enhanced monitoring

The following JSON example enables enhanced monitoring for `IncomingBytes` and `OutgoingRecords` shard level metrics.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.EnableEnhancedMonitoring
{
    "ShardLevelMetrics": [
        "IncomingBytes", "OutgoingRecords"
    ],
    "StreamName": "exampleStreamName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "StreamName": "exampleStreamName",
    "CurrentShardLevelMetrics": [],
    "DesiredShardLevelMetrics": [
        "IncomingBytes",
        "OutgoingRecords"
    ]
}
```

See Also

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

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

GetRecords

Gets data records from a Kinesis data stream's shard.

Specify a shard iterator using the `ShardIterator` parameter. The shard iterator specifies the position in the shard from which you want to start reading data records sequentially. If there are no records available in the portion of the shard that the iterator points to, GetRecords (p. 36) returns an empty list. It might take multiple calls to get to a portion of the shard that contains records.

You can scale by provisioning multiple shards per stream while considering service limits (for more information, see Amazon Kinesis Data Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*). Your application should have one thread per shard, each reading continuously from its stream. To read from a stream continually, call GetRecords (p. 36) in a loop. Use GetShardIterator (p. 41) to get the shard iterator to specify in the first GetRecords (p. 36) call. GetRecords (p. 36) returns a new shard iterator in `NextShardIterator`. Specify the shard iterator returned in `NextShardIterator` in subsequent calls to GetRecords (p. 36). If the shard has been closed, the shard iterator can't return more data and GetRecords (p. 36) returns `null` in `NextShardIterator`. You can terminate the loop when the shard is closed, or when the shard iterator reaches the record with the sequence number or other attribute that marks it as the last record to process.

Each data record can be up to 1 MiB in size, and each shard can read up to 2 MiB per second. You can ensure that your calls don't exceed the maximum supported size or throughput by using the `Limit` parameter to specify the maximum number of records that GetRecords (p. 36) can return. Consider your average record size when determining this limit. The maximum number of records that can be returned per call is 10,000.

The size of the data returned by GetRecords (p. 36) varies depending on the utilization of the shard. The maximum size of data that GetRecords (p. 36) can return is 10 MiB. If a call returns this amount of data, subsequent calls made within the next 5 seconds throw `ProvisionedThroughputExceededException`. If there is insufficient provisioned throughput on the stream, subsequent calls made within the next 1 second throw `ProvisionedThroughputExceededException`. GetRecords (p. 36) doesn't return any data when it throws an exception. For this reason, we recommend that you wait 1 second between calls to GetRecords (p. 36). However, it's possible that the application will get exceptions for longer than 1 second.

To detect whether the application is falling behind in processing, you can use the `MillisBehindLatest` response attribute. You can also monitor the stream using CloudWatch metrics and other mechanisms (see Monitoring in the *Amazon Kinesis Data Streams Developer Guide*).

Each Amazon Kinesis record includes a value, `ApproximateArrivalTimestamp`, that is set when a stream successfully receives and stores a record. This is commonly referred to as a server-side time stamp, whereas a client-side time stamp is set when a data producer creates or sends the record to a stream (a data producer is any data source putting data records into a stream, for example with PutRecords (p. 71)). The time stamp has millisecond precision. There are no guarantees about the time stamp accuracy, or that the time stamp is always increasing. For example, records in a shard or across a stream might have time stamps that are out of order.

This operation has a limit of five transactions per second per shard.

Request Syntax

```
{
    "Limit": number,
    "ShardIterator": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

Limit (p. 36)

The maximum number of records to return. Specify a value of up to 10,000. If you specify a value that is greater than 10,000, GetRecords (p. 36) throws `InvalidArgumentException`. The default value is 10,000.

Type: Integer

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

Required: No

ShardIterator (p. 36)

The position in the shard from which you want to start sequentially reading data records. A shard iterator specifies this position using the sequence number of a data record in the shard.

Type: String

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

Required: Yes

Response Syntax

```
{
    "MillisBehindLatest": number,
    "NextShardIterator": "string",
    "Records": [
        {
            "ApproximateArrivalTimestamp": number,
            "Data": blob,
            "EncryptionType": "string",
            "PartitionKey": "string",
            "SequenceNumber": "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.

MillisBehindLatest (p. 37)

The number of milliseconds the GetRecords (p. 36) response is from the tip of the stream, indicating how far behind current time the consumer is. A value of zero indicates that record processing is caught up, and there are no new records to process at this moment.

Type: Long

Valid Range: Minimum value of 0.

NextShardIterator (p. 37)

The next position in the shard from which to start sequentially reading data records. If set to `null`, the shard has been closed and the requested iterator does not return any more data.

Type: String

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

Records (p. 37)

The data records retrieved from the shard.

Type: Array of Record (p. 109) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

ExpiredIteratorException

The provided iterator exceeds the maximum age allowed.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

KMSAccessDeniedException

The ciphertext references a key that doesn't exist or that you don't have access to.

HTTP Status Code: 400

KMSDisabledException

The request was rejected because the specified customer master key (CMK) isn't enabled.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource isn't valid for this request. For more information, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

KMSNotFoundException

The request was rejected because the specified entity or resource can't be found.

HTTP Status Code: 400

KMSOptInRequired

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

HTTP Status Code: 400

KMSThrottlingException

The request was denied due to request throttling. For more information about throttling, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request rate for the stream is too high, or the requested data is too large for the available throughput. Reduce the frequency or size of your requests. For more information, see Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*, and Error Retries and Exponential Backoff in AWS in the *AWS General Reference*.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To get data from the shards in a stream

The following JSON example gets data from the shards in a stream.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.GetRecords
{
  "ShardIterator": "AAAAAAAAAAETYyAYzd665+8e0X7JTsASDM/
Hr2rSwc0X2qz93iuA3udrjTH+ikQvpQk/1ZcMMLzRdAesqwBGPnsthzU0/CBlM/
U8/8oEqGwX3pKw0XyeDNRAAZyXBo3MqkQtCpXhr942BRTjvWKhFz7OmCb2Ncfr8Tl2cBktooi6kJhr
+djN5WYkB38Rr3akRgCl9qaU4dY=",
  "Limit": 25
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "MillisBehindLatest": 2100,
  "NextShardIterator": "AAAAAAAAAAHsW8zCWf9164uy8Epue6WS3w6wmj4a4USt+CNvMd6uXQ
+HL5vAJMznqqC0DLKsIjuoiTi1BpT6nW0LN2M2D56zM5H8anHm30Gbri9ua+qaGgj+3XTyvbhpERfrezgLHbPB/
rIcVpykJbaSj5tmcXYRmFnqZBEyHwtZYFmh6hvWVFkIwLuMZLMrpWhG5r5hzkE=",
  "Records": [
```

```
    {
      "Data": "XzxkYXRhPl8w",
      "PartitionKey": "partitionKey",
      "ApproximateArrivalTimestamp": 1.441215410867E9,
      "SequenceNumber": "21269319989652663814458848515492872193"
    }
  ]
}
```

See Also

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

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

GetShardIterator

Gets an Amazon Kinesis shard iterator. A shard iterator expires 5 minutes after it is returned to the requester.

A shard iterator specifies the shard position from which to start reading data records sequentially. The position is specified using the sequence number of a data record in a shard. A sequence number is the identifier associated with every record ingested in the stream, and is assigned when a record is put into the stream. Each stream has one or more shards.

You must specify the shard iterator type. For example, you can set the `ShardIteratorType` parameter to read exactly from the position denoted by a specific sequence number by using the `AT_SEQUENCE_NUMBER` shard iterator type. Alternatively, the parameter can read right after the sequence number by using the `AFTER_SEQUENCE_NUMBER` shard iterator type, using sequence numbers returned by earlier calls to PutRecord (p. 66), PutRecords (p. 71), GetRecords (p. 36), or DescribeStream (p. 19). In the request, you can specify the shard iterator type `AT_TIMESTAMP` to read records from an arbitrary point in time, `TRIM_HORIZON` to cause `ShardIterator` to point to the last untrimmed record in the shard in the system (the oldest data record in the shard), or `LATEST` so that you always read the most recent data in the shard.

When you read repeatedly from a stream, use a GetShardIterator (p. 41) request to get the first shard iterator for use in your first GetRecords (p. 36) request and for subsequent reads use the shard iterator returned by the GetRecords (p. 36) request in `NextShardIterator`. A new shard iterator is returned by every GetRecords (p. 36) request in `NextShardIterator`, which you use in the `ShardIterator` parameter of the next GetRecords (p. 36) request.

If a GetShardIterator (p. 41) request is made too often, you receive a `ProvisionedThroughputExceededException`. For more information about throughput limits, see GetRecords (p. 36), and Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*.

If the shard is closed, GetShardIterator (p. 41) returns a valid iterator for the last sequence number of the shard. A shard can be closed as a result of using SplitShard (p. 83) or MergeShards (p. 63).

GetShardIterator (p. 41) has a limit of five transactions per second per account per open shard.

Request Syntax

```
{
    "ShardId": "string",
    "ShardIteratorType": "string",
    "StartingSequenceNumber": "string",
    "StreamName": "string",
    "Timestamp": number
}
```

Request Parameters

The request accepts the following data in JSON format.

ShardId (p. 41)

The shard ID of the Kinesis Data Streams shard to get the iterator for.

Type: String

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

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

Required: Yes

ShardIteratorType (p. 41)

Determines how the shard iterator is used to start reading data records from the shard.

The following are the valid Amazon Kinesis shard iterator types:
- AT_SEQUENCE_NUMBER - Start reading from the position denoted by a specific sequence number, provided in the value `StartingSequenceNumber`.
- AFTER_SEQUENCE_NUMBER - Start reading right after the position denoted by a specific sequence number, provided in the value `StartingSequenceNumber`.
- AT_TIMESTAMP - Start reading from the position denoted by a specific time stamp, provided in the value `Timestamp`.
- TRIM_HORIZON - Start reading at the last untrimmed record in the shard in the system, which is the oldest data record in the shard.
- LATEST - Start reading just after the most recent record in the shard, so that you always read the most recent data in the shard.

Type: String

Valid Values: `AT_SEQUENCE_NUMBER | AFTER_SEQUENCE_NUMBER | TRIM_HORIZON | LATEST | AT_TIMESTAMP`

Required: Yes

StartingSequenceNumber (p. 41)

The sequence number of the data record in the shard from which to start reading. Used with shard iterator type AT_SEQUENCE_NUMBER and AFTER_SEQUENCE_NUMBER.

Type: String

Pattern: `0|([1-9]\d{0,128})`

Required: No

StreamName (p. 41)

The name of the Amazon Kinesis data stream.

Type: String

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

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

Required: Yes

Timestamp (p. 41)

The time stamp of the data record from which to start reading. Used with shard iterator type AT_TIMESTAMP. A time stamp is the Unix epoch date with precision in milliseconds. For example, `2016-04-04T19:58:46.480-00:00` or `1459799926.480`. If a record with this exact time stamp does not exist, the iterator returned is for the next (later) record. If the time stamp is older than the current trim horizon, the iterator returned is for the oldest untrimmed data record (TRIM_HORIZON).

Type: Timestamp

Required: No

Response Syntax

```
{
    "ShardIterator": "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.

ShardIterator (p. 43)

The position in the shard from which to start reading data records sequentially. A shard iterator specifies this position using the sequence number of a data record in a shard.

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. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request rate for the stream is too high, or the requested data is too large for the available throughput. Reduce the frequency or size of your requests. For more information, see Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*, and Error Retries and Exponential Backoff in AWS in the *AWS General Reference*.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To get a shard iterator

The following JSON example gets the specified shard iterator.

Sample Request

```
POST / HTTP/1.1
```

```
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.GetShardIterator
{
  "StreamName": "exampleStreamName",
  "ShardId": "shardId-000000000001",
  "ShardIteratorType": "LATEST"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "ShardIterator": "AAAAAAAAAAETYyAYzd665+8e0X7JTsASDM/
Hr2rSwc0X2qz93iuA3udrjTH+ikQvpQk/1ZcMMLzRdAesqwBGPnsthzU0/CBlM/
U8/8oEqGwX3pKw0XyeDNRAAZyXBo3MqkQtCpXhr942BRTjvWKhFz7OmCb2Ncfr8Tl2cBktooi6kJhr
+djN5WYkB38Rr3akRgCl9qaU4dY="
}
```

See Also

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

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

IncreaseStreamRetentionPeriod

Increases the Kinesis data stream's retention period, which is the length of time data records are accessible after they are added to the stream. The maximum value of a stream's retention period is 168 hours (7 days).

If you choose a longer stream retention period, this operation increases the time period during which records that have not yet expired are accessible. However, it does not make previous, expired data (older than the stream's previous retention period) accessible after the operation has been called. For example, if a stream's retention period is set to 24 hours and is increased to 168 hours, any data that is older than 24 hours remains inaccessible to consumer applications.

Request Syntax

```
{
    "RetentionPeriodHours": number,
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

RetentionPeriodHours (p. 45)

> The new retention period of the stream, in hours. Must be more than the current retention period.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 168.
>
> Required: Yes

StreamName (p. 45)

> The name of the stream to modify.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 128.
>
> Pattern: `[a-zA-Z0-9_.-]+`
>
> Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

> A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To increase stream retention period

The following JSON example increases a stream's retention period.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.IncreaseStreamRetentionPeriod
{
    "RetentionPeriodInHours": "96",
    "StreamName": "examplestream"
}
```

Sample Response

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

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET

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

ListShards

Lists the shards in a stream and provides information about each shard. This operation has a limit of 100 transactions per second per data stream.

> **Important**
> This API is a new operation that is used by the Amazon Kinesis Client Library (KCL). If you have a fine-grained IAM policy that only allows specific operations, you must update your policy to allow calls to this API. For more information, see Controlling Access to Amazon Kinesis Data Streams Resources Using IAM.

Request Syntax

```
{
    "ExclusiveStartShardId": "string",
    "MaxResults": number,
    "NextToken": "string",
    "StreamCreationTimestamp": number,
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ExclusiveStartShardId (p. 48)

Specify this parameter to indicate that you want to list the shards starting with the shard whose ID immediately follows `ExclusiveStartShardId`.

If you don't specify this parameter, the default behavior is for `ListShards` to list the shards starting with the first one in the stream.

You cannot specify this parameter if you specify `NextToken`.

Type: String

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

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

Required: No

MaxResults (p. 48)

The maximum number of shards to return in a single call to `ListShards`. The minimum value you can specify for this parameter is 1, and the maximum is 10,000, which is also the default.

When the number of shards to be listed is greater than the value of `MaxResults`, the response contains a `NextToken` value that you can use in a subsequent call to `ListShards` to list the next set of shards.

Type: Integer

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

Required: No

NextToken (p. 48)

When the number of shards in the data stream is greater than the default value for the `MaxResults` parameter, or if you explicitly specify a value for `MaxResults` that is less than the number of shards in the data stream, the response includes a pagination token named `NextToken`. You can specify this `NextToken` value in a subsequent call to `ListShards` to list the next set of shards.

Don't specify `StreamName` or `StreamCreationTimestamp` if you specify `NextToken` because the latter unambiguously identifies the stream.

You can optionally specify a value for the `MaxResults` parameter when you specify `NextToken`. If you specify a `MaxResults` value that is less than the number of shards that the operation returns if you don't specify `MaxResults`, the response will contain a new `NextToken` value. You can use the new `NextToken` value in a subsequent call to the `ListShards` operation.

> **Important**
> Tokens expire after 300 seconds. When you obtain a value for `NextToken` in the response to a call to `ListShards`, you have 300 seconds to use that value. If you specify an expired token in a call to `ListShards`, you get `ExpiredNextTokenException`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1048576.

Required: No

StreamCreationTimestamp (p. 48)

Specify this input parameter to distinguish data streams that have the same name. For example, if you create a data stream and then delete it, and you later create another data stream with the same name, you can use this input parameter to specify which of the two streams you want to list the shards for.

You cannot specify this parameter if you specify the `NextToken` parameter.

Type: Timestamp

Required: No

StreamName (p. 48)

The name of the data stream whose shards you want to list.

You cannot specify this parameter if you specify the `NextToken` parameter.

Type: String

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

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

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "Shards": [
      {
         "AdjacentParentShardId": "string",
```

```
        "HashKeyRange": {
            "EndingHashKey": "string",
            "StartingHashKey": "string"
        },
        "ParentShardId": "string",
        "SequenceNumberRange": {
            "EndingSequenceNumber": "string",
            "StartingSequenceNumber": "string"
        },
        "ShardId": "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. 49)

When the number of shards in the data stream is greater than the default value for the `MaxResults` parameter, or if you explicitly specify a value for `MaxResults` that is less than the number of shards in the data stream, the response includes a pagination token named `NextToken`. You can specify this `NextToken` value in a subsequent call to `ListShards` to list the next set of shards. For more information about the use of this pagination token when calling the `ListShards` operation, see ListShards:NextToken (p. 49).

> **Important**
> Tokens expire after 300 seconds. When you obtain a value for `NextToken` in the response to a call to `ListShards`, you have 300 seconds to use that value. If you specify an expired token in a call to `ListShards`, you get `ExpiredNextTokenException`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1048576.

Shards (p. 49)

An array of JSON objects. Each object represents one shard and specifies the IDs of the shard, the shard's parent, and the shard that's adjacent to the shard's parent. Each object also contains the starting and ending hash keys and the starting and ending sequence numbers for the shard.

Type: Array of Shard (p. 112) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

ExpiredNextTokenException

The pagination token passed to the operation is expired.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To list the shards of a stream

The following JSON example lists three of the shards of the stream that is specified using `StreamName`. If the stream has more than three shards, the response includes a pagination token named `NextToken`. You can use `NextToken` in your input to a subsequent call to `ListShards` to get a listing of the next set of shards. When there are no more shards to be listed, the response doesn't contain `NextToken`.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.ListShards
{
    "StreamName": "exampleStreamName",
    "MaxResults": 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>
{
    "NextToken": "AAAAAAAAAAGK9EEG0sJqVhCUS2JsgigQ5dcpB4q9PYswrH2oK44Skbjtm+WR0xA7/
hrAFFsohevH1/OyPnbzKBS1byPyCZuVcokYtQe/b1m4c0SCI7jctPT0oUTLRdwSRirKm9dp9YC/EL
+kZHOvYAUnztVGsOAPEFC3ECf/
bVC927bDZBbRRzy/44OHfWmrCLcbcWqehRh5D14WnL3yLsumhiHDkyuxSlkBepauvMnNLtTOlRtmQ5Q5reoujfq2gzeCSOtLcfXgBMz
    "Shards": [
```

```
        {
            "ShardId": "shardId-000000000001",
            "HashKeyRange": {
                "EndingHashKey": "68056473384187692692674921486353642280",
                "StartingHashKey": "34028236692093846346337460743176821145"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "49579844037727333356165064238440708846556371693205002258"
            }
        },
        {
            "ShardId": "shardId-000000000002",
            "HashKeyRange": {
                "EndingHashKey": "102084710076281539039012382229530463436",
                "StartingHashKey": "68056473384187692692674921486353642281"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "49579844037749634101363594861582244564829020124710982690"
            }
        },
        {
            "ShardId": "shardId-000000000003",
            "HashKeyRange": {
                "EndingHashKey": "136112946768375385385349842972707284581",
                "StartingHashKey": "102084710076281539039012382229530463437"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "49579844037771934846562125484723780283101668556216963122"
            }
        }
    ]
}
```

See Also

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

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

ListStreamConsumers

Lists the consumers registered to receive data from a stream using enhanced fan-out, and provides information about each consumer.

This operation has a limit of 5 transactions per second per stream.

Request Syntax

```
{
    "MaxResults": number,
    "NextToken": "string",
    "StreamARN": "string",
    "StreamCreationTimestamp": number
}
```

Request Parameters

The request accepts the following data in JSON format.

MaxResults (p. 53)

The maximum number of consumers that you want a single call of `ListStreamConsumers` to return.

Type: Integer

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

Required: No

NextToken (p. 53)

When the number of consumers that are registered with the data stream is greater than the default value for the `MaxResults` parameter, or if you explicitly specify a value for `MaxResults` that is less than the number of consumers that are registered with the data stream, the response includes a pagination token named `NextToken`. You can specify this `NextToken` value in a subsequent call to `ListStreamConsumers` to list the next set of registered consumers.

Don't specify `StreamName` or `StreamCreationTimestamp` if you specify `NextToken` because the latter unambiguously identifies the stream.

You can optionally specify a value for the `MaxResults` parameter when you specify `NextToken`. If you specify a `MaxResults` value that is less than the number of consumers that the operation returns if you don't specify `MaxResults`, the response will contain a new `NextToken` value. You can use the new `NextToken` value in a subsequent call to the `ListStreamConsumers` operation to list the next set of consumers.

> **Important**
> Tokens expire after 300 seconds. When you obtain a value for `NextToken` in the response to a call to `ListStreamConsumers`, you have 300 seconds to use that value. If you specify an expired token in a call to `ListStreamConsumers`, you get `ExpiredNextTokenException`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1048576.

Required: No

StreamARN (p. 53)

The ARN of the Kinesis data stream for which you want to list the registered consumers. For more information, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

Type: String

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

Pattern: `arn:aws.*:kinesis:.*:\d{12}:stream/.+`

Required: Yes

StreamCreationTimestamp (p. 53)

Specify this input parameter to distinguish data streams that have the same name. For example, if you create a data stream and then delete it, and you later create another data stream with the same name, you can use this input parameter to specify which of the two streams you want to list the consumers for.

You can't specify this parameter if you specify the NextToken parameter.

Type: Timestamp

Required: No

Response Syntax

```
{
    "Consumers": [
        {
            "ConsumerARN": "string",
            "ConsumerCreationTimestamp": number,
            "ConsumerName": "string",
            "ConsumerStatus": "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.

Consumers (p. 54)

An array of JSON objects. Each object represents one registered consumer.

Type: Array of Consumer (p. 100) objects

NextToken (p. 54)

When the number of consumers that are registered with the data stream is greater than the default value for the `MaxResults` parameter, or if you explicitly specify a value for `MaxResults` that is less than the number of registered consumers, the response includes a pagination token named `NextToken`. You can specify this `NextToken` value in a subsequent call to

`ListStreamConsumers` to list the next set of registered consumers. For more information about the use of this pagination token when calling the `ListStreamConsumers` operation, see .

> **Important**
> Tokens expire after 300 seconds. When you obtain a value for `NextToken` in the response to a call to `ListStreamConsumers`, you have 300 seconds to use that value. If you specify an expired token in a call to `ListStreamConsumers`, you get `ExpiredNextTokenException`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1048576.

Errors

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

ExpiredNextTokenException

The pagination token passed to the operation is expired.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

See Also

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

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

- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

ListStreams

Lists your Kinesis data streams.

The number of streams may be too large to return from a single call to `ListStreams`. You can limit the number of returned streams using the `Limit` parameter. If you do not specify a value for the `Limit` parameter, Kinesis Data Streams uses the default limit, which is currently 10.

You can detect if there are more streams available to list by using the `HasMoreStreams` flag from the returned output. If there are more streams available, you can request more streams by using the name of the last stream returned by the `ListStreams` request in the `ExclusiveStartStreamName` parameter in a subsequent request to `ListStreams`. The group of stream names returned by the subsequent request is then added to the list. You can continue this process until all the stream names have been collected in the list.

has a limit of five transactions per second per account.

Request Syntax

```
{
    "ExclusiveStartStreamName": "string",
    "Limit": number
}
```

Request Parameters

The request accepts the following data in JSON format.

ExclusiveStartStreamName (p. 57)

The name of the stream to start the list with.

Type: String

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

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

Required: No

Limit (p. 57)

The maximum number of streams to list.

Type: Integer

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

Required: No

Response Syntax

```
{
    "HasMoreStreams": boolean,
    "StreamNames": [ "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.

HasMoreStreams (p. 57)

If set to `true`, there are more streams available to list.

Type: Boolean

StreamNames (p. 57)

The names of the streams that are associated with the AWS account making the `ListStreams` request.

Type: Array of strings

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

Example

To list your streams

The following JSON example lists your streams, starting with the specified stream.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.ListStreams
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
```

```
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "HasMoreStreams": false,
  "StreamNames": [
    "exampleStreamName"
  ]
}
```

See Also

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

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

ListTagsForStream

Lists the tags for the specified Kinesis data stream. This operation has a limit of five transactions per second per account.

Request Syntax

```
{
   "ExclusiveStartTagKey": "string",
   "Limit": number,
   "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ExclusiveStartTagKey (p. 60)

The key to use as the starting point for the list of tags. If this parameter is set, `ListTagsForStream` gets all tags that occur after `ExclusiveStartTagKey`.

Type: String

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

Required: No

Limit (p. 60)

The number of tags to return. If this number is less than the total number of tags associated with the stream, `HasMoreTags` is set to `true`. To list additional tags, set `ExclusiveStartTagKey` to the last key in the response.

Type: Integer

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

Required: No

StreamName (p. 60)

The name of the stream.

Type: String

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

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

Required: Yes

Response Syntax

```
{
   "HasMoreTags": boolean,
   "Tags": [
```

```
        {
            "Key": "string",
            "Value": "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.

HasMoreTags (p. 60)

If set to `true`, more tags are available. To request additional tags, set `ExclusiveStartTagKey` to the key of the last tag returned.

Type: Boolean

Tags (p. 60)

A list of tags associated with `StreamName`, starting with the first tag after `ExclusiveStartTagKey` and up to the specified `Limit`.

Type: Array of Tag (p. 125) objects

Array Members: Minimum number of 0 items.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To list the tags for a stream

The following JSON example lists the tags for the specified stream.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.ListTagsForStream
{
   "StreamName": "exampleStreamName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "HasMoreTags": "false",
  "Tags" : [
     {
        "Key": "Project",
        "Value": "myProject"
     },
     {
        "Key": "Environment",
        "Value": "Production"
     }
   ]
}
```

See Also

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

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

MergeShards

Merges two adjacent shards in a Kinesis data stream and combines them into a single shard to reduce the stream's capacity to ingest and transport data. Two shards are considered adjacent if the union of the hash key ranges for the two shards form a contiguous set with no gaps. For example, if you have two shards, one with a hash key range of 276...381 and the other with a hash key range of 382...454, then you could merge these two shards into a single shard that would have a hash key range of 276...454. After the merge, the single child shard receives data for all hash key values covered by the two parent shards.

`MergeShards` is called when there is a need to reduce the overall capacity of a stream because of excess capacity that is not being used. You must specify the shard to be merged and the adjacent shard for a stream. For more information about merging shards, see Merge Two Shards in the *Amazon Kinesis Data Streams Developer Guide*.

If the stream is in the `ACTIVE` state, you can call `MergeShards`. If a stream is in the `CREATING`, `UPDATING`, or `DELETING` state, `MergeShards` returns a `ResourceInUseException`. If the specified stream does not exist, `MergeShards` returns a `ResourceNotFoundException`.

You can use DescribeStream (p. 19) to check the state of the stream, which is returned in `StreamStatus`.

`MergeShards` is an asynchronous operation. Upon receiving a `MergeShards` request, Amazon Kinesis Data Streams immediately returns a response and sets the `StreamStatus` to `UPDATING`. After the operation is completed, Kinesis Data Streams sets the `StreamStatus` to `ACTIVE`. Read and write operations continue to work while the stream is in the `UPDATING` state.

You use DescribeStream (p. 19) to determine the shard IDs that are specified in the `MergeShards` request.

If you try to operate on too many streams in parallel using CreateStream (p. 6), DeleteStream (p. 12), `MergeShards`, or SplitShard (p. 83), you receive a `LimitExceededException`.

`MergeShards` has a limit of five transactions per second per account.

Request Syntax

```
{
   "AdjacentShardToMerge": "string",
   "ShardToMerge": "string",
   "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

AdjacentShardToMerge (p. 63)

The shard ID of the adjacent shard for the merge.

Type: String

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

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

Required: Yes

ShardToMerge (p. 63)

The shard ID of the shard to combine with the adjacent shard for the merge.

Type: String

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

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

Required: Yes

StreamName (p. 63)

The name of the stream for the merge.

Type: String

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

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To merge two adjacent shards

The following JSON example merges two adjacent shards.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.MergeShards
{
  "StreamName": "exampleStreamName",
  "ShardToMerge": "shardId-000000000000",
  "AdjacentShardToMerge": "shardId-000000000001"
}
```

Sample Response

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

See Also

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

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

PutRecord

Writes a single data record into an Amazon Kinesis data stream. Call `PutRecord` to send data into the stream for real-time ingestion and subsequent processing, one record at a time. Each shard can support writes up to 1,000 records per second, up to a maximum data write total of 1 MiB per second.

You must specify the name of the stream that captures, stores, and transports the data; a partition key; and the data blob itself.

The data blob can be any type of data; for example, a segment from a log file, geographic/location data, website clickstream data, and so on.

The partition key is used by Kinesis Data Streams to distribute data across shards. Kinesis Data Streams segregates the data records that belong to a stream into multiple shards, using the partition key associated with each data record to determine the shard to which a given data record belongs.

Partition keys are Unicode strings, with a maximum length limit of 256 characters for each key. An MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards using the hash key ranges of the shards. You can override hashing the partition key to determine the shard by explicitly specifying a hash value using the `ExplicitHashKey` parameter. For more information, see Adding Data to a Stream in the *Amazon Kinesis Data Streams Developer Guide*.

`PutRecord` returns the shard ID of where the data record was placed and the sequence number that was assigned to the data record.

Sequence numbers increase over time and are specific to a shard within a stream, not across all shards within a stream. To guarantee strictly increasing ordering, write serially to a shard and use the `SequenceNumberForOrdering` parameter. For more information, see Adding Data to a Stream in the *Amazon Kinesis Data Streams Developer Guide*.

> **Important**
> After you write a record to a stream, you cannot modify that record or its order within the stream.

If a `PutRecord` request cannot be processed because of insufficient provisioned throughput on the shard involved in the request, `PutRecord` throws `ProvisionedThroughputExceededException`.

By default, data records are accessible for 24 hours from the time that they are added to a stream. You can use IncreaseStreamRetentionPeriod (p. 45) or DecreaseStreamRetentionPeriod (p. 9) to modify this retention period.

Request Syntax

```
{
   "Data": blob,
   "ExplicitHashKey": "string",
   "PartitionKey": "string",
   "SequenceNumberForOrdering": "string",
   "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

Data (p. 66)

The data blob to put into the record, which is base64-encoded when the blob is serialized. When the data blob (the payload before base64-encoding) is added to the partition key size, the total size must not exceed the maximum record size (1 MiB).

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 0. Maximum length of 1048576.

Required: Yes

ExplicitHashKey (p. 66)

The hash value used to explicitly determine the shard the data record is assigned to by overriding the partition key hash.

Type: String

Pattern: `0|([1-9]\d{0,38})`

Required: No

PartitionKey (p. 66)

Determines which shard in the stream the data record is assigned to. Partition keys are Unicode strings with a maximum length limit of 256 characters for each key. Amazon Kinesis Data Streams uses the partition key as input to a hash function that maps the partition key and associated data to a specific shard. Specifically, an MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards. As a result of this hashing mechanism, all data records with the same partition key map to the same shard within the stream.

Type: String

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

Required: Yes

SequenceNumberForOrdering (p. 66)

Guarantees strictly increasing sequence numbers, for puts from the same client and to the same partition key. Usage: set the `SequenceNumberForOrdering` of record *n* to the sequence number of record *n-1* (as returned in the result when putting record *n-1*). If this parameter is not set, records are coarsely ordered based on arrival time.

Type: String

Pattern: `0|([1-9]\d{0,128})`

Required: No

StreamName (p. 66)

The name of the stream to put the data record into.

Type: String

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

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

Required: Yes

Response Syntax

```
{
    "EncryptionType": "string",
    "SequenceNumber": "string",
    "ShardId": "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.

EncryptionType (p. 68)

The encryption type to use on the record. This parameter can be one of the following values:
- `NONE`: Do not encrypt the records in the stream.
- `KMS`: Use server-side encryption on the records in the stream using a customer-managed AWS KMS key.

Type: String

Valid Values: `NONE | KMS`

SequenceNumber (p. 68)

The sequence number identifier that was assigned to the put data record. The sequence number for the record is unique across all records in the stream. A sequence number is the identifier associated with every record put into the stream.

Type: String

Pattern: `0|([1-9]\d{0,128})`

ShardId (p. 68)

The shard ID of the shard where the data record was placed.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

KMSAccessDeniedException

The ciphertext references a key that doesn't exist or that you don't have access to.

HTTP Status Code: 400

KMSDisabledException

The request was rejected because the specified customer master key (CMK) isn't enabled.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource isn't valid for this request. For more information, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

KMSNotFoundException

The request was rejected because the specified entity or resource can't be found.

HTTP Status Code: 400

KMSOptInRequired

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

HTTP Status Code: 400

KMSThrottlingException

The request was denied due to request throttling. For more information about throttling, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request rate for the stream is too high, or the requested data is too large for the available throughput. Reduce the frequency or size of your requests. For more information, see Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*, and Error Retries and Exponential Backoff in AWS in the *AWS General Reference*.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To add data to a stream

The following JSON example adds data to the specified stream.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
```

```
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.PutRecord
{
  "StreamName": "exampleStreamName",
  "Data": "XzxkYXRhPl8x",
  "PartitionKey": "partitionKey"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
  "SequenceNumber": "21269319989653637946712965403778482177",
  "ShardId": "shardId-000000000001"
}
```

See Also

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

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

PutRecords

Writes multiple data records into a Kinesis data stream in a single call (also referred to as a `PutRecords` request). Use this operation to send data into the stream for data ingestion and processing.

Each `PutRecords` request can support up to 500 records. Each record in the request can be as large as 1 MiB, up to a limit of 5 MiB for the entire request, including partition keys. Each shard can support writes up to 1,000 records per second, up to a maximum data write total of 1 MiB per second.

You must specify the name of the stream that captures, stores, and transports the data; and an array of request `Records`, with each record in the array requiring a partition key and data blob. The record size limit applies to the total size of the partition key and data blob.

The data blob can be any type of data; for example, a segment from a log file, geographic/location data, website clickstream data, and so on.

The partition key is used by Kinesis Data Streams as input to a hash function that maps the partition key and associated data to a specific shard. An MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards. As a result of this hashing mechanism, all data records with the same partition key map to the same shard within the stream. For more information, see Adding Data to a Stream in the *Amazon Kinesis Data Streams Developer Guide*.

Each record in the `Records` array may include an optional parameter, `ExplicitHashKey`, which overrides the partition key to shard mapping. This parameter allows a data producer to determine explicitly the shard where the record is stored. For more information, see Adding Multiple Records with PutRecords in the *Amazon Kinesis Data Streams Developer Guide*.

The `PutRecords` response includes an array of response `Records`. Each record in the response array directly correlates with a record in the request array using natural ordering, from the top to the bottom of the request and response. The response `Records` array always includes the same number of records as the request array.

The response `Records` array includes both successfully and unsuccessfully processed records. Kinesis Data Streams attempts to process all records in each `PutRecords` request. A single record failure does not stop the processing of subsequent records. As a result, PutRecords doesn't guarantee the ordering of records. If you need to read records in the same order they are written to the stream, use PutRecord (p. 66) instead of `PutRecords`, and write to the same shard.

A successfully processed record includes `ShardId` and `SequenceNumber` values. The `ShardId` parameter identifies the shard in the stream where the record is stored. The `SequenceNumber` parameter is an identifier assigned to the put record, unique to all records in the stream.

An unsuccessfully processed record includes `ErrorCode` and `ErrorMessage` values. `ErrorCode` reflects the type of error and can be one of the following values: `ProvisionedThroughputExceededException` or `InternalFailure`. `ErrorMessage` provides more detailed information about the `ProvisionedThroughputExceededException` exception including the account ID, stream name, and shard ID of the record that was throttled. For more information about partially successful responses, see Adding Multiple Records with PutRecords in the *Amazon Kinesis Data Streams Developer Guide*.

> **Important**
> After you write a record to a stream, you cannot modify that record or its order within the stream.

By default, data records are accessible for 24 hours from the time that they are added to a stream. You can use IncreaseStreamRetentionPeriod (p. 45) or DecreaseStreamRetentionPeriod (p. 9) to modify this retention period.

Request Syntax

```
{
    "Records": [
        {
            "Data": blob,
            "ExplicitHashKey": "string",
            "PartitionKey": "string"
        }
    ],
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

Records (p. 72)

> The records associated with the request.
>
> Type: Array of PutRecordsRequestEntry (p. 106) objects
>
> Array Members: Minimum number of 1 item. Maximum number of 500 items.
>
> Required: Yes

StreamName (p. 72)

> The stream name associated with the request.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 128.
>
> Pattern: `[a-zA-Z0-9_.-]+`
>
> Required: Yes

Response Syntax

```
{
    "EncryptionType": "string",
    "FailedRecordCount": number,
    "Records": [
        {
            "ErrorCode": "string",
            "ErrorMessage": "string",
            "SequenceNumber": "string",
            "ShardId": "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.

EncryptionType (p. 72)

> The encryption type used on the records. This parameter can be one of the following values:
> - `NONE`: Do not encrypt the records.
> - `KMS`: Use server-side encryption on the records using a customer-managed AWS KMS key.
>
> Type: String
>
> Valid Values: `NONE | KMS`

FailedRecordCount (p. 72)

> The number of unsuccessfully processed records in a `PutRecords` request.
>
> Type: Integer
>
> Valid Range: Minimum value of 1. Maximum value of 100000.

Records (p. 72)

> An array of successfully and unsuccessfully processed record results, correlated with the request by natural ordering. A record that is successfully added to a stream includes `SequenceNumber` and `ShardId` in the result. A record that fails to be added to a stream includes `ErrorCode` and `ErrorMessage` in the result.
>
> Type: Array of PutRecordsResultEntry (p. 107) objects
>
> Array Members: Minimum number of 1 item. Maximum number of 500 items.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

> A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.
>
> HTTP Status Code: 400

KMSAccessDeniedException

> The ciphertext references a key that doesn't exist or that you don't have access to.
>
> HTTP Status Code: 400

KMSDisabledException

> The request was rejected because the specified customer master key (CMK) isn't enabled.
>
> HTTP Status Code: 400

KMSInvalidStateException

> The request was rejected because the state of the specified resource isn't valid for this request. For more information, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.
>
> HTTP Status Code: 400

KMSNotFoundException

The request was rejected because the specified entity or resource can't be found.

HTTP Status Code: 400

KMSOptInRequired

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

HTTP Status Code: 400

KMSThrottlingException

The request was denied due to request throttling. For more information about throttling, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request rate for the stream is too high, or the requested data is too large for the available throughput. Reduce the frequency or size of your requests. For more information, see Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*, and Error Retries and Exponential Backoff in AWS in the *AWS General Reference*.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Examples

To add data to a stream, with complete success

The following JSON example adds data to the specified stream with a successful response.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.PutRecords
{
    "Records": [
        {
            "Data": "XzxkYXRhPl8x",
            "PartitionKey": "partitionKey1"
        },
        {
            "Data": "f1PxFQo92Afh",
            "PartitionKey": "partitionKey2"
        },
        {
```

```
            "Data": "Gi4sEdd08HypA",
            "PartitionKey": "partitionKey3"
        }
    ],
    "StreamName": "exampleStreamName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "FailedRecordCount": 0,
    "Records": [
        {
            "SequenceNumber": "49543463076548007577105092703039560359975228518395019266",
            "ShardId": "shardId-000000000000"
        },
        {
            "SequenceNumber": "49543463076570308322303623326179887152428262250726293522",
            "ShardId": "shardId-000000000001"
        },
        {
            "SequenceNumber": "49543463076570308322303623326179887152428262250726293588",
            "ShardId": "shardId-000000000003"
        }
    ]
}
```

To add data to a stream, with a partially successful response

The following JSON example adds data to the specified stream with a partially successful response and contains failed records.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.PutRecords
{
    "Records": [
        {
            "Data": "XzxkYXRhPl8x",
            "PartitionKey": "partitionKey1"
        },
        {
            "Data": "f1PxFQo92Afh",
            "PartitionKey": "partitionKey2"
        },
        {
            "Data": "Gi4sEdd08HypA",
            "PartitionKey": "partitionKey3"
        }
```

```
    ],
    "StreamName": "exampleStreamName"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "FailedRecordCount": 2,
    "Records": [
        {
            "SequenceNumber": "49543463076548007577105092703039560359975228518395012686",
            "ShardId": "shardId-000000000000"
        },
        {
            "ErrorCode": "ProvisionedThroughputExceededException",
            "ErrorMessage": "Rate exceeded for shard shardId-000000000001 in stream
 exampleStreamName under account 111111111111."
        },
        {
            "ErrorCode": "InternalFailure",
            "ErrorMessage": "Internal service failure."
        }
    ]
}
```

See Also

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

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

RegisterStreamConsumer

Registers a consumer with a Kinesis data stream. When you use this operation, the consumer you register can then call SubscribeToShard (p. 92) to receive data from the stream using enhanced fan-out, at a rate of up to 2 MiB per second for every shard you subscribe to. This rate is unaffected by the total number of consumers that read from the same stream.

You can register up to 5 consumers per stream. However, you can request a limit increase using the Kinesis Data Streams limits form. A given consumer can only be registered with one stream at a time.

For an example of how to use this operations, see Enhanced Fan-Out Using the Kinesis Data Streams API.

The use of this operation has a limit of five transactions per second per account.

Request Syntax

```
{
   "ConsumerName": "string",
   "StreamARN": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

ConsumerName (p. 77)

For a given Kinesis data stream, each consumer must have a unique name. However, consumer names don't have to be unique across data streams.

Type: String

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

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

Required: Yes

StreamARN (p. 77)

The ARN of the Kinesis data stream that you want to register the consumer with. For more info, see Amazon Resource Names (ARNs) and AWS Service Namespaces.

Type: String

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

Pattern: `arn:aws.*:kinesis:.*:\d{12}:stream/.+`

Required: Yes

Response Syntax

```
{
   "Consumer": {
      "ConsumerARN": "string",
```

```
        "ConsumerCreationTimestamp": number,
        "ConsumerName": "string",
        "ConsumerStatus": "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.

Consumer (p. 77)

An object that represents the details of the consumer you registered. When you register a consumer, it gets an ARN that is generated by Kinesis Data Streams.

Type: Consumer (p. 100) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

See Also

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

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

- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

RemoveTagsFromStream

Removes tags from the specified Kinesis data stream. Removed tags are deleted and cannot be recovered after this operation successfully completes.

If you specify a tag that does not exist, it is ignored.

RemoveTagsFromStream (p. 80) has a limit of five transactions per second per account.

Request Syntax

```
{
    "StreamName": "string",
    "TagKeys": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

StreamName (p. 80)

> The name of the stream.
>
> Type: String
>
> Length Constraints: Minimum length of 1. Maximum length of 128.
>
> Pattern: `[a-zA-Z0-9_.-]+`
>
> Required: Yes

TagKeys (p. 80)

> A list of tag keys. Each corresponding tag is removed from the stream.
>
> Type: Array of strings
>
> Array Members: Minimum number of 1 item. Maximum number of 50 items.
>
> Length Constraints: Minimum length of 1. Maximum length of 128.
>
> 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. 126).

InvalidArgumentException

> A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To remove tags from a stream

The following JSON example removes the specified tag from the specified stream.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.RemoveTagsFromStream
{
   "StreamName": "exampleStreamName",
   "TagKeys": ["Project", "Environment"]
}
```

Sample Response

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

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET

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

SplitShard

Splits a shard into two new shards in the Kinesis data stream, to increase the stream's capacity to ingest and transport data. `SplitShard` is called when there is a need to increase the overall capacity of a stream because of an expected increase in the volume of data records being ingested.

You can also use `SplitShard` when a shard appears to be approaching its maximum utilization; for example, the producers sending data into the specific shard are suddenly sending more than previously anticipated. You can also call `SplitShard` to increase stream capacity, so that more Kinesis Data Streams applications can simultaneously read data from the stream for real-time processing.

You must specify the shard to be split and the new hash key, which is the position in the shard where the shard gets split in two. In many cases, the new hash key might be the average of the beginning and ending hash key, but it can be any hash key value in the range being mapped into the shard. For more information, see Split a Shard in the *Amazon Kinesis Data Streams Developer Guide*.

You can use DescribeStream (p. 19) to determine the shard ID and hash key values for the `ShardToSplit` and `NewStartingHashKey` parameters that are specified in the `SplitShard` request.

`SplitShard` is an asynchronous operation. Upon receiving a `SplitShard` request, Kinesis Data Streams immediately returns a response and sets the stream status to `UPDATING`. After the operation is completed, Kinesis Data Streams sets the stream status to `ACTIVE`. Read and write operations continue to work while the stream is in the `UPDATING` state.

You can use `DescribeStream` to check the status of the stream, which is returned in `StreamStatus`. If the stream is in the `ACTIVE` state, you can call `SplitShard`. If a stream is in `CREATING` or `UPDATING` or `DELETING` states, `DescribeStream` returns a `ResourceInUseException`.

If the specified stream does not exist, `DescribeStream` returns a `ResourceNotFoundException`. If you try to create more shards than are authorized for your account, you receive a `LimitExceededException`.

For the default shard limit for an AWS account, see Kinesis Data Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*. To increase this limit, contact AWS Support.

If you try to operate on too many streams simultaneously using CreateStream (p. 6), DeleteStream (p. 12), MergeShards (p. 63), and/or SplitShard (p. 83), you receive a `LimitExceededException`.

`SplitShard` has a limit of five transactions per second per account.

Request Syntax

```
{
    "NewStartingHashKey": "string",
    "ShardToSplit": "string",
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

NewStartingHashKey (p. 83)

> A hash key value for the starting hash key of one of the child shards created by the split. The hash key range for a given shard constitutes a set of ordered contiguous positive integers. The

value for `NewStartingHashKey` must be in the range of hash keys being mapped into the shard. The `NewStartingHashKey` hash key value and all higher hash key values in hash key range are distributed to one of the child shards. All the lower hash key values in the range are distributed to the other child shard.

Type: String

Pattern: `0|([1-9]\d{0,38})`

Required: Yes

ShardToSplit (p. 83)

The shard ID of the shard to split.

Type: String

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

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

Required: Yes

StreamName (p. 83)

The name of the stream for the shard split.

Type: String

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

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To split a shard

The following JSON example splits the specified shard.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.SplitShard
{
  "StreamName": "exampleStreamName",
  "ShardToSplit": "shardId-000000000000",
  "NewStartingHashKey": "10"
}
```

Sample Response

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

See Also

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

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

StartStreamEncryption

Enables or updates server-side encryption using an AWS KMS key for a specified stream.

Starting encryption is an asynchronous operation. Upon receiving the request, Kinesis Data Streams returns immediately and sets the status of the stream to `UPDATING`. After the update is complete, Kinesis Data Streams sets the status of the stream back to `ACTIVE`. Updating or applying encryption normally takes a few seconds to complete, but it can take minutes. You can continue to read and write data to your stream while its status is `UPDATING`. Once the status of the stream is `ACTIVE`, encryption begins for records written to the stream.

API Limits: You can successfully apply a new AWS KMS key for server-side encryption 25 times in a rolling 24-hour period.

Note: It can take up to 5 seconds after the stream is in an `ACTIVE` status before all records written to the stream are encrypted. After you enable encryption, you can verify that encryption is applied by inspecting the API response from `PutRecord` or `PutRecords`.

Request Syntax

```
{
    "EncryptionType": "string",
    "KeyId": "string",
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

EncryptionType (p. 86)

The encryption type to use. The only valid value is `KMS`.

Type: String

Valid Values: `KMS`

Required: Yes

KeyId (p. 86)

The GUID for the customer-managed AWS KMS key to use for encryption. This value can be a globally unique identifier, a fully specified Amazon Resource Name (ARN) to either an alias or a key, or an alias name prefixed by "alias/".You can also use a master key owned by Kinesis Data Streams by specifying the alias `aws/kinesis`.

- Key ARN example: `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012`
- Alias ARN example: `arn:aws:kms:us-east-1:123456789012:alias/MyAliasName`
- Globally unique key ID example: `12345678-1234-1234-1234-123456789012`
- Alias name example: `alias/MyAliasName`
- Master key owned by Kinesis Data Streams: `alias/aws/kinesis`

Type: String

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

Required: Yes

StreamName (p. 86)

The name of the stream for which to start encrypting records.

Type: String

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

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

KMSAccessDeniedException

The ciphertext references a key that doesn't exist or that you don't have access to.

HTTP Status Code: 400

KMSDisabledException

The request was rejected because the specified customer master key (CMK) isn't enabled.

HTTP Status Code: 400

KMSInvalidStateException

The request was rejected because the state of the specified resource isn't valid for this request. For more information, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

KMSNotFoundException

The request was rejected because the specified entity or resource can't be found.

HTTP Status Code: 400

KMSOptInRequired

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

HTTP Status Code: 400

KMSThrottlingException

The request was denied due to request throttling. For more information about throttling, see Limits in the *AWS Key Management Service Developer Guide*.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

See Also

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

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

StopStreamEncryption

Disables server-side encryption for a specified stream.

Stopping encryption is an asynchronous operation. Upon receiving the request, Kinesis Data Streams returns immediately and sets the status of the stream to `UPDATING`. After the update is complete, Kinesis Data Streams sets the status of the stream back to `ACTIVE`. Stopping encryption normally takes a few seconds to complete, but it can take minutes. You can continue to read and write data to your stream while its status is `UPDATING`. Once the status of the stream is `ACTIVE`, records written to the stream are no longer encrypted by Kinesis Data Streams.

API Limits: You can successfully disable server-side encryption 25 times in a rolling 24-hour period.

Note: It can take up to 5 seconds after the stream is in an `ACTIVE` status before all records written to the stream are no longer subject to encryption. After you disabled encryption, you can verify that encryption is not applied by inspecting the API response from `PutRecord` or `PutRecords`.

Request Syntax

```
{
    "EncryptionType": "string",
    "KeyId": "string",
    "StreamName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

EncryptionType (p. 89)

The encryption type. The only valid value is `KMS`.

Type: String

Valid Values: `KMS`

Required: Yes

KeyId (p. 89)

The GUID for the customer-managed AWS KMS key to use for encryption. This value can be a globally unique identifier, a fully specified Amazon Resource Name (ARN) to either an alias or a key, or an alias name prefixed by "alias/".You can also use a master key owned by Kinesis Data Streams by specifying the alias `aws/kinesis`.

- Key ARN example: `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012`
- Alias ARN example: `arn:aws:kms:us-east-1:123456789012:alias/MyAliasName`
- Globally unique key ID example: `12345678-1234-1234-1234-123456789012`
- Alias name example: `alias/MyAliasName`
- Master key owned by Kinesis Data Streams: `alias/aws/kinesis`

Type: String

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

Required: Yes

StreamName (p. 89)

The name of the stream on which to stop encrypting records.

Type: String

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

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

Required: Yes

Response Elements

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot

- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

SubscribeToShard

This operation establishes an HTTP/2 connection between the consumer you specify in the `ConsumerARN` parameter and the shard you specify in the `ShardId` parameter. After the connection is successfully established, Kinesis Data Streams pushes records from the shard to the consumer over this connection. Before you call this operation, call RegisterStreamConsumer (p. 77) to register the consumer with Kinesis Data Streams.

When the `SubscribeToShard` call succeeds, your consumer starts receiving events of type SubscribeToShardEvent (p. 122) over the HTTP/2 connection for up to 5 minutes, after which time you need to call `SubscribeToShard` again to renew the subscription if you want to continue to receive records.

You can make one call to `SubscribeToShard` per second per registered consumer per shard. For example, if you have a 4000 shard stream and two registered stream consumers, you can make one `SubscribeToShard` request per second for each combination of shard and registered consumer, allowing you to subscribe both consumers to all 4000 shards in one second.

If you call `SubscribeToShard` again with the same `ConsumerARN` and `ShardId` within 5 seconds of a successful call, you'll get a `ResourceInUseException`. If you call `SubscribeToShard` 5 seconds or more after a successful call, the first connection will expire and the second call will take over the subscription.

For an example of how to use this operations, see Enhanced Fan-Out Using the Kinesis Data Streams API.

Request Syntax

```
{
    "ConsumerARN": "string",
    "ShardId": "string",
    "StartingPosition": {
        "SequenceNumber": "string",
        "Timestamp": number,
        "Type": "string"
    }
}
```

Request Parameters

The request accepts the following data in JSON format.

ConsumerARN (p. 92)

For this parameter, use the value you obtained when you called RegisterStreamConsumer (p. 77).

Type: String

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

Pattern: `^(arn):aws.*:kinesis:.*:\d{12}:.*stream\/[a-zA-Z0-9_.-]+\/consumer\/[a-zA-Z0-9_.-]+:[0-9]+`

Required: Yes

ShardId (p. 92)

The ID of the shard you want to subscribe to. To see a list of all the shards for a given stream, use ListShards (p. 48).

Type: String

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

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

Required: Yes

StartingPosition (p. 92)

Type: StartingPosition (p. 114) object

Required: Yes

Response Syntax

```
{
    "EventStream": {
        "InternalFailureException": {
        },
        "KMSAccessDeniedException": {
        },
        "KMSDisabledException": {
        },
        "KMSInvalidStateException": {
        },
        "KMSNotFoundException": {
        },
        "KMSOptInRequired": {
        },
        "KMSThrottlingException": {
        },
        "ResourceInUseException": {
        },
        "ResourceNotFoundException": {
        },
        "SubscribeToShardEvent": {
            "ContinuationSequenceNumber": "string",
            "MillisBehindLatest": number,
            "Records": [
                {
                    "ApproximateArrivalTimestamp": number,
                    "Data": blob,
                    "EncryptionType": "string",
                    "PartitionKey": "string",
                    "SequenceNumber": "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.

EventStream (p. 93)

The event stream that your consumer can use to read records from the shard.

Type: SubscribeToShardEventStream (p. 123) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

See Also

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

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

UpdateShardCount

Updates the shard count of the specified stream to the specified number of shards.

Updating the shard count is an asynchronous operation. Upon receiving the request, Kinesis Data Streams returns immediately and sets the status of the stream to `UPDATING`. After the update is complete, Kinesis Data Streams sets the status of the stream back to `ACTIVE`. Depending on the size of the stream, the scaling action could take a few minutes to complete. You can continue to read and write data to your stream while its status is `UPDATING`.

To update the shard count, Kinesis Data Streams performs splits or merges on individual shards. This can cause short-lived shards to be created, in addition to the final shards. These short-lived shards count towards your total shard limit for your account in the Region.

When using this operation, we recommend that you specify a target shard count that is a multiple of 25% (25%, 50%, 75%, 100%). You can specify any target value within your shard limit. However, if you specify a target that isn't a multiple of 25%, the scaling action might take longer to complete.

This operation has the following default limits. By default, you cannot do the following:

- Scale more than twice per rolling 24-hour period per stream
- Scale up to more than double your current shard count for a stream
- Scale down below half your current shard count for a stream
- Scale up to more than 500 shards in a stream
- Scale a stream with more than 500 shards down unless the result is less than 500 shards
- Scale up to more than the shard limit for your account

For the default limits for an AWS account, see Streams Limits in the *Amazon Kinesis Data Streams Developer Guide*. To request an increase in the call rate limit, the shard limit for this API, or your overall shard limit, use the limits form.

Request Syntax

```
{
    "ScalingType": "string",
    "StreamName": "string",
    "TargetShardCount": number
}
```

Request Parameters

The request accepts the following data in JSON format.

ScalingType (p. 95)

The scaling type. Uniform scaling creates shards of equal size.

Type: String

Valid Values: `UNIFORM_SCALING`

Required: Yes

StreamName (p. 95)

The name of the stream.

Type: String

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

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

Required: Yes

TargetShardCount (p. 95)

The new number of shards.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100000.

Required: Yes

Response Syntax

```
{
    "CurrentShardCount": number,
    "StreamName": "string",
    "TargetShardCount": 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.

CurrentShardCount (p. 96)

The current number of shards.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100000.

StreamName (p. 96)

The name of the stream.

Type: String

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

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

TargetShardCount (p. 96)

The updated number of shards.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100000.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 126).

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used. For more information, see the returned message.

HTTP Status Code: 400

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed.

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

HTTP Status Code: 400

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

HTTP Status Code: 400

Example

To update the shard count of the specified stream from 2 to 4

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Authorization: <AuthParams>
Connection: Keep-Alive
X-Amz-Date: <Date>
X-Amz-Target: Kinesis_20131202.UpdateShardCount
{
   "StreamName": "exampleStreamName",
   "TargetShardCount": 4,
   "ScalingType": "UNIFORM_SCALING"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
   "TargetShardCount": 4,
   "StreamName": "exampleStreamName",
   "CurrentShardCount": 2
}
```

See Also

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

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

Data Types

The Amazon Kinesis 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:

Consumer

An object that represents the details of the consumer you registered. This type of object is returned by RegisterStreamConsumer (p. 77).

Contents

ConsumerARN

When you register a consumer, Kinesis Data Streams generates an ARN for it. You need this ARN to be able to call SubscribeToShard (p. 92).

If you delete a consumer and then create a new one with the same name, it won't have the same ARN. That's because consumer ARNs contain the creation timestamp. This is important to keep in mind if you have IAM policies that reference consumer ARNs.

Type: String

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

Pattern: `^(arn):aws.*:kinesis:.*:\d{12}:.*stream\/[a-zA-Z0-9_.-]+\/consumer\/` `[a-zA-Z0-9_.-]+:[0-9]+`

Required: Yes

ConsumerCreationTimestamp

Type: Timestamp

Required: Yes

ConsumerName

The name of the consumer is something you choose when you register the consumer.

Type: String

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

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

Required: Yes

ConsumerStatus

A consumer can't read data while in the `CREATING` or `DELETING` states.

Type: String

Valid Values: `CREATING | DELETING | ACTIVE`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go

- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

ConsumerDescription

An object that represents the details of a registered consumer. This type of object is returned by
DescribeStreamConsumer (p. 23).

Contents

ConsumerARN

When you register a consumer, Kinesis Data Streams generates an ARN for it. You need this ARN to
be able to call SubscribeToShard (p. 92).

If you delete a consumer and then create a new one with the same name, it won't have the same
ARN. That's because consumer ARNs contain the creation timestamp. This is important to keep in
mind if you have IAM policies that reference consumer ARNs.

Type: String

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

Pattern: `^(arn):aws.*:kinesis:.*:\d{12}:.*stream\/[a-zA-Z0-9_.-]+\/consumer\/`
`[a-zA-Z0-9_.-]+:[0-9]+`

Required: Yes

ConsumerCreationTimestamp

Type: Timestamp

Required: Yes

ConsumerName

The name of the consumer is something you choose when you register the consumer.

Type: String

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

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

Required: Yes

ConsumerStatus

A consumer can't read data while in the `CREATING` or `DELETING` states.

Type: String

Valid Values: `CREATING | DELETING | ACTIVE`

Required: Yes

StreamARN

The ARN of the stream with which you registered the consumer.

Type: String

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

Pattern: `arn:aws.*:kinesis:.*:\d{12}:stream/.+`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

EnhancedMetrics

Represents enhanced metrics types.

Contents

ShardLevelMetrics

List of shard-level metrics.

The following are the valid shard-level metrics. The value "`ALL`" enhances every metric.

- `IncomingBytes`
- `IncomingRecords`
- `OutgoingBytes`
- `OutgoingRecords`
- `WriteProvisionedThroughputExceeded`
- `ReadProvisionedThroughputExceeded`
- `IteratorAgeMilliseconds`
- `ALL`

For more information, see Monitoring the Amazon Kinesis Data Streams Service with Amazon CloudWatch in the *Amazon Kinesis Data Streams Developer Guide*.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 7 items.

Valid Values: `IncomingBytes | IncomingRecords | OutgoingBytes | OutgoingRecords | WriteProvisionedThroughputExceeded | ReadProvisionedThroughputExceeded | IteratorAgeMilliseconds | ALL`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

HashKeyRange

The range of possible hash key values for the shard, which is a set of ordered contiguous positive integers.

Contents

EndingHashKey

The ending hash key of the hash key range.

Type: String

Pattern: `0|([1-9]\d{0,38})`

Required: Yes

StartingHashKey

The starting hash key of the hash key range.

Type: String

Pattern: `0|([1-9]\d{0,38})`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

PutRecordsRequestEntry

Represents the output for `PutRecords`.

Contents

Data

The data blob to put into the record, which is base64-encoded when the blob is serialized. When the data blob (the payload before base64-encoding) is added to the partition key size, the total size must not exceed the maximum record size (1 MiB).

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 0. Maximum length of 1048576.

Required: Yes

ExplicitHashKey

The hash value used to determine explicitly the shard that the data record is assigned to by overriding the partition key hash.

Type: String

Pattern: `0|([1-9]\d{0,38})`

Required: No

PartitionKey

Determines which shard in the stream the data record is assigned to. Partition keys are Unicode strings with a maximum length limit of 256 characters for each key. Amazon Kinesis Data Streams uses the partition key as input to a hash function that maps the partition key and associated data to a specific shard. Specifically, an MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards. As a result of this hashing mechanism, all data records with the same partition key map to the same shard within the stream.

Type: String

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

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

PutRecordsResultEntry

Represents the result of an individual record from a `PutRecords` request. A record that is successfully added to a stream includes `SequenceNumber` and `ShardId` in the result. A record that fails to be added to the stream includes `ErrorCode` and `ErrorMessage` in the result.

Contents

ErrorCode

The error code for an individual record result. `ErrorCodes` can be either `ProvisionedThroughputExceededException` or `InternalFailure`.

Type: String

Required: No

ErrorMessage

The error message for an individual record result. An `ErrorCode` value of `ProvisionedThroughputExceededException` has an error message that includes the account ID, stream name, and shard ID. An `ErrorCode` value of `InternalFailure` has the error message "Internal Service Failure".

Type: String

Required: No

SequenceNumber

The sequence number for an individual record result.

Type: String

Pattern: `0|([1-9]\d{0,128})`

Required: No

ShardId

The shard ID for an individual record result.

Type: String

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

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

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java

- AWS SDK for Ruby V2

Record

The unit of data of the Kinesis data stream, which is composed of a sequence number, a partition key, and a data blob.

Contents

ApproximateArrivalTimestamp

The approximate time that the record was inserted into the stream.

Type: Timestamp

Required: No

Data

The data blob. The data in the blob is both opaque and immutable to Kinesis Data Streams, which does not inspect, interpret, or change the data in the blob in any way. When the data blob (the payload before base64-encoding) is added to the partition key size, the total size must not exceed the maximum record size (1 MiB).

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 0. Maximum length of 1048576.

Required: Yes

EncryptionType

The encryption type used on the record. This parameter can be one of the following values:

- `NONE`: Do not encrypt the records in the stream.
- `KMS`: Use server-side encryption on the records in the stream using a customer-managed AWS KMS key.

Type: String

Valid Values: `NONE | KMS`

Required: No

PartitionKey

Identifies which shard in the stream the data record is assigned to.

Type: String

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

Required: Yes

SequenceNumber

The unique identifier of the record within its shard.

Type: String

Pattern: `0|([1-9]\d{0,128})`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SequenceNumberRange

The range of possible sequence numbers for the shard.

Contents

EndingSequenceNumber

The ending sequence number for the range. Shards that are in the OPEN state have an ending sequence number of `null`.

Type: String

Pattern: `0|([1-9]\d{0,128})`

Required: No

StartingSequenceNumber

The starting sequence number for the range.

Type: String

Pattern: `0|([1-9]\d{0,128})`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Shard

A uniquely identified group of data records in a Kinesis data stream.

Contents

AdjacentParentShardId

The shard ID of the shard adjacent to the shard's parent.

Type: String

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

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

Required: No

HashKeyRange

The range of possible hash key values for the shard, which is a set of ordered contiguous positive integers.

Type: HashKeyRange (p. 105) object

Required: Yes

ParentShardId

The shard ID of the shard's parent.

Type: String

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

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

Required: No

SequenceNumberRange

The range of possible sequence numbers for the shard.

Type: SequenceNumberRange (p. 111) object

Required: Yes

ShardId

The unique identifier of the shard within the stream.

Type: String

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

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

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

StartingPosition

Contents

SequenceNumber

The sequence number of the data record in the shard from which to start streaming. To specify a sequence number, set `StartingPosition` to `AT_SEQUENCE_NUMBER` or `AFTER_SEQUENCE_NUMBER`.

Type: String

Pattern: `0|([1-9]\d{0,128})`

Required: No

Timestamp

The time stamp of the data record from which to start reading. To specify a time stamp, set `StartingPosition` to `Type AT_TIMESTAMP`. A time stamp is the Unix epoch date with precision in milliseconds. For example, `2016-04-04T19:58:46.480-00:00` or `1459799926.480`. If a record with this exact time stamp does not exist, records will be streamed from the next (later) record. If the time stamp is older than the current trim horizon, records will be streamed from the oldest untrimmed data record (`TRIM_HORIZON`).

Type: Timestamp

Required: No

Type

You can set the starting position to one of the following values:

`AT_SEQUENCE_NUMBER`: Start streaming from the position denoted by the sequence number specified in the `SequenceNumber` field.

`AFTER_SEQUENCE_NUMBER`: Start streaming right after the position denoted by the sequence number specified in the `SequenceNumber` field.

`AT_TIMESTAMP`: Start streaming from the position denoted by the time stamp specified in the `Timestamp` field.

`TRIM_HORIZON`: Start streaming at the last untrimmed record in the shard, which is the oldest data record in the shard.

`LATEST`: Start streaming just after the most recent record in the shard, so that you always read the most recent data in the shard.

Type: String

Valid Values: `AT_SEQUENCE_NUMBER | AFTER_SEQUENCE_NUMBER | TRIM_HORIZON | LATEST | AT_TIMESTAMP`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

StreamDescription

Represents the output for .

Contents

EncryptionType

The server-side encryption type used on the stream. This parameter can be one of the following values:

- `NONE`: Do not encrypt the records in the stream.
- `KMS`: Use server-side encryption on the records in the stream using a customer-managed AWS KMS key.

Type: String

Valid Values: `NONE | KMS`

Required: No

EnhancedMonitoring

Represents the current enhanced monitoring settings of the stream.

Type: Array of objects

Required: Yes

HasMoreShards

If set to `true`, more shards in the stream are available to describe.

Type: Boolean

Required: Yes

KeyId

The GUID for the customer-managed AWS KMS key to use for encryption. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".You can also use a master key owned by Kinesis Data Streams by specifying the alias `aws/kinesis`.

- Key ARN example: `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012`
- Alias ARN example: `arn:aws:kms:us-east-1:123456789012:alias/MyAliasName`
- Globally unique key ID example: `12345678-1234-1234-1234-123456789012`
- Alias name example: `alias/MyAliasName`
- Master key owned by Kinesis Data Streams: `alias/aws/kinesis`

Type: String

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

Required: No

RetentionPeriodHours

The current retention period, in hours.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 168.

Required: Yes

Shards

The shards that comprise the stream.

Type: Array of Shard (p. 112) objects

Required: Yes

StreamARN

The Amazon Resource Name (ARN) for the stream being described.

Type: String

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

Pattern: `arn:aws.*:kinesis:.*:\d{12}:stream/.+`

Required: Yes

StreamCreationTimestamp

The approximate time that the stream was created.

Type: Timestamp

Required: Yes

StreamName

The name of the stream being described.

Type: String

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

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

Required: Yes

StreamStatus

The current status of the stream being described. The stream status is one of the following states:

- `CREATING` - The stream is being created. Kinesis Data Streams immediately returns and sets `StreamStatus` to `CREATING`.
- `DELETING` - The stream is being deleted. The specified stream is in the `DELETING` state until Kinesis Data Streams completes the deletion.
- `ACTIVE` - The stream exists and is ready for read and write operations or deletion. You should perform read and write operations only on an `ACTIVE` stream.
- `UPDATING` - Shards in the stream are being merged or split. Read and write operations continue to work while the stream is in the `UPDATING` state.

Type: String

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

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

StreamDescriptionSummary

Represents the output for DescribeStreamSummary (p. 26)

Contents

ConsumerCount

The number of enhanced fan-out consumers registered with the stream.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 1000000.

Required: No

EncryptionType

The encryption type used. This value is one of the following:

- `KMS`
- `NONE`

Type: String

Valid Values: `NONE | KMS`

Required: No

EnhancedMonitoring

Represents the current enhanced monitoring settings of the stream.

Type: Array of EnhancedMetrics (p. 104) objects

Required: Yes

KeyId

The GUID for the customer-managed AWS KMS key to use for encryption. This value can be a globally unique identifier, a fully specified ARN to either an alias or a key, or an alias name prefixed by "alias/".You can also use a master key owned by Kinesis Data Streams by specifying the alias `aws/kinesis`.

- Key ARN example: `arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012`
- Alias ARN example: `arn:aws:kms:us-east-1:123456789012:alias/MyAliasName`
- Globally unique key ID example: `12345678-1234-1234-1234-123456789012`
- Alias name example: `alias/MyAliasName`
- Master key owned by Kinesis Data Streams: `alias/aws/kinesis`

Type: String

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

Required: No

OpenShardCount

The number of open shards in the stream.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 1000000.

Required: Yes

RetentionPeriodHours

The current retention period, in hours.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100000.

Required: Yes

StreamARN

The Amazon Resource Name (ARN) for the stream being described.

Type: String

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

Pattern: `arn:aws.*:kinesis:.*:\d{12}:stream/.+`

Required: Yes

StreamCreationTimestamp

The approximate time that the stream was created.

Type: Timestamp

Required: Yes

StreamName

The name of the stream being described.

Type: String

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

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

Required: Yes

StreamStatus

The current status of the stream being described. The stream status is one of the following states:

- `CREATING` - The stream is being created. Kinesis Data Streams immediately returns and sets `StreamStatus` to `CREATING`.
- `DELETING` - The stream is being deleted. The specified stream is in the `DELETING` state until Kinesis Data Streams completes the deletion.
- `ACTIVE` - The stream exists and is ready for read and write operations or deletion. You should perform read and write operations only on an `ACTIVE` stream.
- `UPDATING` - Shards in the stream are being merged or split. Read and write operations continue to work while the stream is in the `UPDATING` state.

Type: String

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

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SubscribeToShardEvent

After you call SubscribeToShard (p. 92), Kinesis Data Streams sends events of this type over an HTTP/2 connection to your consumer.

Contents

ContinuationSequenceNumber

Use this as `SequenceNumber` in the next call to SubscribeToShard (p. 92), with `StartingPosition` set to `AT_SEQUENCE_NUMBER` or `AFTER_SEQUENCE_NUMBER`. Use `ContinuationSequenceNumber` for checkpointing because it captures your shard progress even when no data is written to the shard.

Type: String

Pattern: `0|([1-9]\d{0,128})`

Required: Yes

MillisBehindLatest

The number of milliseconds the read records are from the tip of the stream, indicating how far behind current time the consumer is. A value of zero indicates that record processing is caught up, and there are no new records to process at this moment.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

Records

Type: Array of Record (p. 109) objects

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

SubscribeToShardEventStream

This is a tagged union for all of the types of events an enhanced fan-out consumer can receive over HTTP/2 after a call to SubscribeToShard (p. 92).

Contents

InternalFailureException

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

Type: Exception
HTTP Status Code:

Required: No

KMSAccessDeniedException

The ciphertext references a key that doesn't exist or that you don't have access to.

Type: Exception
HTTP Status Code:

Required: No

KMSDisabledException

The request was rejected because the specified customer master key (CMK) isn't enabled.

Type: Exception
HTTP Status Code:

Required: No

KMSInvalidStateException

The request was rejected because the state of the specified resource isn't valid for this request. For more information, see How Key State Affects Use of a Customer Master Key in the *AWS Key Management Service Developer Guide*.

Type: Exception
HTTP Status Code:

Required: No

KMSNotFoundException

The request was rejected because the specified entity or resource can't be found.

Type: Exception
HTTP Status Code:

Required: No

KMSOptInRequired

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

Type: Exception
HTTP Status Code:

Required: No

KMSThrottlingException

The request was denied due to request throttling. For more information about throttling, see Limits in the *AWS Key Management Service Developer Guide*.

Type: Exception
HTTP Status Code:

Required: No

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource must be in the `ACTIVE` state.

Type: Exception
HTTP Status Code:

Required: No

ResourceNotFoundException

The requested resource could not be found. The stream might not be specified correctly.

Type: Exception
HTTP Status Code:

Required: No

SubscribeToShardEvent

After you call SubscribeToShard (p. 92), Kinesis Data Streams sends events of this type to your consumer. For an example of how to handle these events, see Enhanced Fan-Out Using the Kinesis Data Streams API.

Type: SubscribeToShardEvent (p. 122) object

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Tag

Metadata assigned to the stream, consisting of a key-value pair.

Contents

Key

A unique identifier for the tag. Maximum length: 128 characters. Valid characters: Unicode letters, digits, white space, _ . / = + - % @

Type: String

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

Required: Yes

Value

An optional string, typically used to describe or define the tag. Maximum length: 256 characters. Valid characters: Unicode letters, digits, white space, _ . / = + - % @

Type: String

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

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

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

HTTP Status Code: 500

InvalidAction

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

HTTP Status Code: 400

InvalidClientTokenId

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

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

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

HTTP Status Code: 400

InvalidQueryParameter

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

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

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

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

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

HTTP Status Code: 403

RequestExpired

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

HTTP Status Code: 400

ServiceUnavailable

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

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

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

HTTP Status Code: 400