Amazon Kinesis Service API Reference

API Reference

API Version 2013-12-02

Amazon Kinesis Service API Reference: API Reference

Welcome

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

This document was last updated on March 14, 2014.

Actions

The following actions are supported:

- CreateStream (p. 3)
- DeleteStream (p. 6)
- DescribeStream (p. 8)
- GetRecords (p. 12)
- GetShardIterator (p. 15)
- ListStreams (p. 19)
- MergeShards (p. 22)
- PutRecord (p. 25)
- SplitShard (p. 29)

CreateStream

This operation adds a new Amazon Kinesis stream to your AWS account. A stream captures and transports data records that are continuously emitted from different data sources or *producers*. Scale-out within an Amazon Kinesis stream is explicitly supported by means of shards, which are uniquely identified groups of data records in an Amazon Kinesis stream.

You specify and control the number of shards that a stream is composed of. Each open shard can support up to 5 read transactions per second, up to a maximum total of 2 MB of data read per second. Each shard can support up to 1000 write transactions per second, up to a maximum total of 1 MB data written per second. You can add shards to a stream if the amount of data input increases and you can remove shards if the amount of data input decreases.

The stream name identifies the stream. The name is scoped to the AWS account used by the application. It is also scoped by 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, Amazon Kinesis immediately returns and sets the stream status to CREATING. After the stream is created, Amazon Kinesis 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 if 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.

Note: The default limit for an AWS account is 10 shards per stream. If you need to create a stream with more than 10 shards, contact AWS Support to increase the limit on your account.

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

`CreateStream` has a limit of 5 transactions per second per account.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

ShardCount
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.

Note: The default limit for an AWS account is 10 shards per stream. If you need to create a stream with more than 10 shards, contact AWS Support to increase the limit on your account.

Type: Number

Required: Yes

StreamName

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 region. That is, two streams in two different AWS accounts can have the same name, and two streams in the same AWS account, but in two different regions, can have the same name.

Type: String

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. 43).

InvalidArgumentException

HTTP Status Code: 400

LimitExceededException

HTTP Status Code: 400

ResourceInUseException

HTTP Status Code: 400

Examples

Create a Stream

The following is an example of an Amazon Kinesis CreateStream request and response.

Sample Request

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

DeleteStream

This operation deletes a stream and all of 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 will receive 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 Amazon Kinesis completes the deletion.

Note: Amazon Kinesis might continue to accept data read and write operations, such as PutRecord (p. 25) and GetRecords (p. 12), 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.

You can use the DescribeStream (p. 8) operation to check the state of the stream, which is returned in `StreamStatus`.

`DeleteStream` has a limit of 5 transactions per second per account.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

StreamName

The name of the stream to delete.

Type: String

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. 43).

LimitExceededException

HTTP Status Code: 400

ResourceNotFoundException

HTTP Status Code: 400

Examples

Delete a Stream

The following is an example of an Amazon Kinesis DeleteStream request and response.

Sample Request

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

DescribeStream

This operation returns the following information about the stream: the current status of the stream, the stream Amazon Resource Name (ARN), and an array of shard objects that comprise the stream. For each shard object there is information about the hash key and sequence number ranges that the shard spans, and the IDs of any earlier shards that played in a role in a MergeShards (p. 22) or SplitShard (p. 29) operation that created the shard. A sequence number is the identifier associated with every record ingested in the Amazon Kinesis stream. The sequence number is assigned by the Amazon Kinesis service when a record is put into the stream.

You can limit the number of returned shards using the `Limit` parameter. The number of shards in a stream may be too large to return from a single call to `DescribeStream`. You can detect this by using the `HasMoreShards` flag in the returned output. `HasMoreShards` is set to `true` when there is more data available.

If there are more shards available, you can request more shards by using the shard ID of the last shard returned by the `DescribeStream` request, in the `ExclusiveStartShardId` parameter in a subsequent request to `DescribeStream`. `DescribeStream` is a paginated operation.

`DescribeStream` has a limit of 10 transactions per second per account.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

ExclusiveStartShardId

The shard ID of the shard to start with for the stream description.

Type: String

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

Required: No

Limit

The maximum number of shards to return.

Type: Number

Required: No

StreamName

The name of the stream to describe.

Type: String

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

Required: Yes

Response Syntax

```
{
    "StreamDescription": {
        "HasMoreShards": "boolean",
        "Shards": [
            {
                "AdjacentParentShardId": "string",
                "HashKeyRange": {
                    "EndingHashKey": "string",
                    "StartingHashKey": "string"
                },
                "ParentShardId": "string",
                "SequenceNumberRange": {
                    "EndingSequenceNumber": "string",
                    "StartingSequenceNumber": "string"
                },
                "ShardId": "string"
            }
        ],
        "StreamARN": "string",
        "StreamName": "string",
        "StreamStatus": "string"
    }
}
```

Response Elements

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

StreamDescription

Contains the current status of the stream, the stream ARN, an array of shard objects that comprise the stream, and states whether there are more shards available.

Type: StreamDescription (p. 37) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 43).

LimitExceededException

HTTP Status Code: 400

ResourceNotFoundException

HTTP Status Code: 400

Examples

Obtain Information About a Stream

The following is an example of an Amazon Kinesis DescribeStream request and response.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
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": {
    "HasMoreShards": false,
    "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-1:052958737983:exampleStreamName",
    "StreamName": "exampleStreamName",
    "StreamStatus": "ACTIVE"
  }
}
```

GetRecords

This operation returns one or more data records from a shard. A `GetRecords` operation request can retrieve up to 10 MB of data.

You specify a shard iterator for the shard that you want to read data from in the `ShardIterator` parameter. The shard iterator specifies the position in the shard from which you want to start reading data records sequentially. A shard iterator specifies this position using the sequence number of a data record in the shard. For more information about the shard iterator, see GetShardIterator (p. 15).

`GetRecords` may return a partial result if the response size limit is exceeded. You will get an error, but not a partial result if the shard's provisioned throughput is exceeded, the shard iterator has expired, or an internal processing failure has occurred. Clients can request a smaller amount of data by specifying a maximum number of returned records using the `Limit` parameter. The `Limit` parameter can be set to an integer value of up to 10,000. If you set the value to an integer greater than 10,000, you will receive `InvalidArgumentException`.

A new shard iterator is returned by every `GetRecords` request in `NextShardIterator`, which you use in the `ShardIterator` parameter of the next `GetRecords` request. When you repeatedly read from an Amazon Kinesis stream use a GetShardIterator (p. 15) request to get the first shard iterator to use in your first `GetRecords` request and then use the shard iterator returned in `NextShardIterator` for subsequent reads.

`GetRecords` can return `null` for the `NextShardIterator` to reflect that the shard has been closed and that the requested shard iterator would never have returned more data.

If no items can be processed because of insufficient provisioned throughput on the shard involved in the request, `GetRecords` throws `ProvisionedThroughputExceededException`.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

Limit

The maximum number of records to return, which can be set to a value of up to 10,000.

Type: Number

Required: No

ShardIterator

The position in the shard from which you want to start sequentially reading data records.

Type: String

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

Required: Yes

Response Syntax

```
{
    "NextShardIterator": "string",
    "Records": [
        {
            "Data": "blob",
            "PartitionKey": "string",
            "SequenceNumber": "string"
        }
    ]
}
```

Response Elements

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

NextShardIterator

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 will not return any more data.

Type: String

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

Records

Type: array of Record (p. 35) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 43).

ExpiredIteratorException

HTTP Status Code: 400

InvalidArgumentException

HTTP Status Code: 400

ProvisionedThroughputExceededException

HTTP Status Code: 400

ResourceNotFoundException

HTTP Status Code: 400

Examples

Get Data from the Shards in a Stream

The following is an example of an Amazon Kinesis GetRecords request and response.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
X-Amz-Target: Kinesis_20131202.GetRecords

{
  "ShardIterator": "AAAAAAAAAAETYyAYzd665+8e0X7JT
sASDM/Hr2rSwc0X2qz93iuA3udrjTH+ikQvpQk/1ZcMMLzRdAesqwBGPnsthzU0/CBlM/U8/8oEqG
wX3pKw0XyeDNRAAZyXBo3MqkQtCpXhr942BRTjvWKhFz7OmCb2Ncfr8Tl2cB
ktooi6kJhr+djN5WYkB38Rr3akRgCl9qaU4dY=",
  "Limit": 2
}
```

Sample Response

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


{
  "NextShardIterator": "AAAAAAAAAAHsW8zCWf9164uy8Epue6WS3w6wmj4a4USt+CN
vMd6uXQ+HL5vAJMznqqC0DLKsIjuoiTi1BpT6nW0LN2M2D56zM5H8anHm30Gbri9ua+qaGgj+3XTyvbh
pERfrezgLHbPB/rIcVpykJbaSj5tmcXYRmFnqZBEyHwtZYFmh6hvWVFkIwLuMZLMrpWhG5r5hzkE=",

  "Records": [
    {
      "Data": "XzxkYXRhPl8w",
      "PartitionKey": "partitionKey",
      "SequenceNumber": "21269319989652663814458848515492872193"
    }
  ]
}
```

GetShardIterator

This operation returns a shard iterator in ShardIterator. The shard iterator specifies the position in the shard from which you want to start reading data records sequentially. A shard iterator specifies this position using the sequence number of a data record in a shard. A sequence number is the identifier associated with every record ingested in the Amazon Kinesis stream. The sequence number is assigned by the Amazon Kinesis service when a record is put into the stream.

You must specify the shard iterator type in the GetShardIterator request. 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, or right after the sequence number by using the AFTER_SEQUENCE_NUMBER shard iterator type, using sequence numbers returned by earlier PutRecord (p. 25), GetRecords (p. 12) or DescribeStream (p. 8) requests. You can specify the shard iterator type TRIM_HORIZON in the request to cause ShardIterator to point to the last untrimmed record in the shard in the system, which is the oldest data record in the shard. Or you can point to just after the most recent record in the shard, by using the shard iterator type LATEST, so that you always read the most recent data in the shard.

Note: Each shard iterator expires five minutes after it is returned to the requester.

When you repeatedly read from an Amazon Kinesis stream use a GetShardIterator (p. 15) request to get the first shard iterator to to use in your first GetRecords request and then use the shard iterator returned by the GetRecords request in NextShardIterator for subsequent reads. A new shard iterator is returned by every GetRecords request in NextShardIterator, which you use in the ShardIterator parameter of the next GetRecords request.

If a GetShardIterator request is made too often, you will receive a ProvisionedThroughputExceededException. For more information about throughput limits, see the Amazon Kinesis Developer Guide.

GetShardIterator can return null for its ShardIterator to indicate that the shard has been closed and that the requested iterator will return no more data. A shard can be closed by a SplitShard (p. 29) or MergeShards (p. 22) operation.

GetShardIterator has a limit of 5 transactions per second per account per open shard.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

ShardId
The shard ID of the shard to get the iterator for.

Type: String

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

Required: Yes

ShardIteratorType

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

The following are the valid shard iterator types:

- AT_SEQUENCE_NUMBER - Start reading exactly from the position denoted by a specific sequence number.
- AFTER_SEQUENCE_NUMBER - Start reading right after the position denoted by a specific sequence number.
- 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`

Required: Yes

StartingSequenceNumber

The sequence number of the data record in the shard from which to start reading from.

Type: String

Required: No

StreamName

The name of the stream.

Type: String

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

Required: Yes

Response Syntax

```
{
    "ShardIterator": "string"
}
```

Response Elements

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

ShardIterator

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. 43).

InvalidArgumentException
HTTP Status Code: 400

ProvisionedThroughputExceededException
HTTP Status Code: 400

ResourceNotFoundException
HTTP Status Code: 400

Examples

Get a Shard Iterator

The following is an example of an Amazon Kinesis GetShardIterator request and response.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
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+8e0X7JT
sASDM/Hr2rSwc0X2qz93iuA3udrjTH+ikQvpQk/1ZcMMLzRdAesqwBGPnsthzU0/CBlM/U8/8oEqG
wX3pKw0XyeDNRAAZyXBo3MqkQtCpXhr942BRTjvWKhFz7OmCb2Ncfr8Tl2cB
ktooi6kJhr+djN5WYkB38Rr3akRgCl9qaU4dY="
}
```

ListStreams

This operation returns an array of the names of all the streams that are associated with the AWS account making the `ListStreams` request. A given AWS account can have many streams active at one time.

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, Amazon Kinesis 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.

`ListStreams` has a limit of 5 transactions per second per account.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

ExclusiveStartStreamName

The name of the stream to start the list with.

Type: String

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

Required: No

Limit

The maximum number of streams to list.

Type: Number

Required: No

Response Syntax

```
{
    "HasMoreStreams": "boolean",
    "StreamNames": [
        "string"
    ]
}
```

Response Elements

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

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

Type: Boolean

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

Type: array of Strings

Errors

For information about the errors that are common to all actions, see Common Errors (p. 43).

LimitExceededException
HTTP Status Code: 400

Examples

List the Streams for an AWS Account

The following is an example of an Amazon Kinesis ListStreams request and response.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
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"
  ]
}
```

MergeShards

This operation merges two adjacent shards in a 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. The operation requires that you specify the shard to be merged and the adjacent shard for a given stream. For more information about merging shards, see the Amazon Kinesis Developer Guide.

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

You can use the DescribeStream (p. 8) operation to check the state of the stream, which is returned in `StreamStatus`.

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

You use the DescribeStream (p. 8) operation 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. 3), DeleteStream (p. 6), `MergeShards` or SplitShard (p. 29), you will receive a `LimitExceededException`.

`MergeShards` has limit of 5 transactions per second per account.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

AdjacentShardToMerge

The shard ID of the adjacent shard for the merge.

Type: String

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

Required: Yes

ShardToMerge

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.

Required: Yes

StreamName

The name of the stream for the merge.

Type: String

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. 43).

InvalidArgumentException

HTTP Status Code: 400

LimitExceededException

HTTP Status Code: 400

ResourceInUseException

HTTP Status Code: 400

ResourceNotFoundException

HTTP Status Code: 400

Examples

Merge Two Adjacent Shards

The following is an example of an Amazon Kinesis MergeShards request and response.

Sample Request

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

```
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
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>
```

PutRecord

This operation puts a data record into an Amazon Kinesis stream from a producer. This operation must be called to send data from the producer into the Amazon Kinesis stream for real-time ingestion and subsequent processing. The `PutRecord` operation requires the name of the stream that captures, stores, and transports the data; a partition key; and the data blob itself. The data blob could be a segment from a log file, geographic/location data, website clickstream data, or any other data type.

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

Partition keys are Unicode strings, with a maximum length limit of 256 bytes. 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 the Amazon Kinesis 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 generally increase over time. To guarantee strictly increasing ordering, use the `SequenceNumberForOrdering` parameter. For more information, see the Amazon Kinesis Developer Guide.

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

Data records are accessible for only 24 hours from the time that they are added to an Amazon Kinesis stream.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

Data

The data blob to put into the record, which is Base64-encoded when the blob is serialized. The maximum size of the data blob (the payload after Base64-decoding) is 50 kilobytes (KB)

Type: Blob

Length constraints: Minimum length of 0. Maximum length of 51200.

Required: Yes

ExplicitHashKey

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

Type: String

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 bytes. Amazon Kinesis 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 will map to the same shard within the stream.

Type: String

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

Required: Yes

SequenceNumberForOrdering

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 PutRecordResult (p. 34) when putting record *n-1*). If this parameter is not set, records will be coarsely ordered based on arrival time.

Type: String

Required: No

StreamName

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

Type: String

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

Required: Yes

Response Syntax

```
{
    "SequenceNumber": "string",
    "ShardId": "string"
}
```

Response Elements

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

SequenceNumber

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

ShardId

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

Type: String

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 43).

InvalidArgumentException

HTTP Status Code: 400

ProvisionedThroughputExceededException

HTTP Status Code: 400

ResourceNotFoundException

HTTP Status Code: 400

Examples

Add Data to a Stream

The following is an example of an Amazon Kinesis PutRecord request and response.

Sample Request

```
POST / HTTP/1.1
Host: kinesis.<region>.<domain>
x-amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=content-
type;date;host;user-agent;x-amz-date;x-amz-target;x-amzn-requestid, Signa
ture=<Signature>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.1
Content-Length: <PayloadSizeBytes>
Connection: Keep-Alive
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"
}
```

SplitShard

This operation splits a shard into two new shards in the 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 stream because of an expected increase in the volume of data records being ingested.

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

The `SplitShard` operation requires that you 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 simply 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 about splitting shards, see the Amazon Kinesis Developer Guide.

You can use the DescribeStream (p. 8) operation 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, Amazon Kinesis immediately returns a response and sets the stream status to UPDATING. After the operation is completed, Amazon Kinesis 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, then Amazon Kinesis returns a `ResourceInUseException`.

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

Note: The default limit for an AWS account is 10 shards per stream. If you need to create a stream with more than 10 shards, contact AWS Support to increase the limit on your account.

If you try to operate on too many streams in parallel using CreateStream (p. 3), DeleteStream (p. 6), MergeShards (p. 22) or SplitShard (p. 29), you will receive a `LimitExceededException`.

`SplitShard` has limit of 5 transactions per second per account.

Request Syntax

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

Request Parameters

For information about the common parameters that all actions use, see Common Parameters (p. 39).

The request accepts the following data in JSON format.

NewStartingHashKey

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

Required: Yes

ShardToSplit

The shard ID of the shard to split.

Type: String

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

Required: Yes

StreamName

The name of the stream for the shard split.

Type: String

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. 43).

InvalidArgumentException

HTTP Status Code: 400

LimitExceededException

HTTP Status Code: 400

ResourceInUseException

HTTP Status Code: 400

ResourceNotFoundException

HTTP Status Code: 400

Examples

Split a Shard

The following is an example of an Amazon Kinesis SplitShard request and response.

Sample Request

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

Data Types

The Amazon Kinesis Service API Reference API contains several data types that various actions use. This section describes each data type in detail.

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

The following data types are supported:

DescribeStreamResult

Description

Represents the output of a `DescribeStream` operation.

Contents

StreamDescription

Contains the current status of the stream, the stream ARN, an array of shard objects that comprise the stream, and states whether there are more shards available.

Type: StreamDescription (p. 37) object

Required: Yes

GetRecordsResult

Description

Represents the output of a `GetRecords` operation.

Contents

NextShardIterator

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 will not return any more data.

Type: String

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

Required: No

Records

Type: array of Record (p. 35) objects

Required: Yes

GetShardIteratorResult

Description

Represents the output of a `GetShardIterator` operation.

Contents

ShardIterator

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.

Required: No

HashKeyRange

Description

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

Required: Yes

StartingHashKey

The starting hash key of the hash key range.

Type: String

Required: Yes

ListStreamsResult

Description

Represents the output of a `ListStreams` operation.

Contents

HasMoreStreams

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

Type: Boolean

Required: Yes

StreamNames

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

Type: array of Strings

Required: Yes

PutRecordResult

Description

Represents the output of a `PutRecord` operation.

Contents

SequenceNumber

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

Required: Yes

ShardId

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

Type: String

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

Required: Yes

Record

Description

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

Contents

Data

The data blob. The data in the blob is both opaque and immutable to the Amazon Kinesis service, which does not inspect, interpret, or change the data in the blob in any way. The maximum size of the data blob (the payload after Base64-decoding) is 50 kilobytes (KB)

Type: Blob

Length constraints: Minimum length of 0. Maximum length of 51200.

Required: Yes

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 for the record in the Amazon Kinesis stream.

Type: String

Required: Yes

SequenceNumberRange

Description

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

Required: No

StartingSequenceNumber

The starting sequence number for the range.

Type: String

Required: Yes

Shard

Description

A uniquely identified group of data records in an Amazon Kinesis 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.

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. 33) object

Required: Yes

ParentShardId

The shard Id of the shard's parent.

Type: String

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

Required: No

SequenceNumberRange

The range of possible sequence numbers for the shard.

Type: SequenceNumberRange (p. 35) object

Required: Yes

ShardId

The unique identifier of the shard within the Amazon Kinesis stream.

Type: String

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

Required: Yes

StreamDescription

Description

Represents the output of a `DescribeStream` operation.

Contents

HasMoreShards

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

Type: Boolean

Required: Yes

Shards

The shards that comprise the stream.

Type: array of Shard (p. 36) objects

Required: Yes

StreamARN

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

Type: String

Required: Yes

StreamName

The name of the stream being described.

Type: String

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

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. Upon receiving a CreateStream (p. 3) request, Amazon Kinesis immediately returns and sets `StreamStatus` to CREATING.
- DELETING - The stream is being deleted. After a DeleteStream (p. 6) request, the specified stream is in the DELETING state until Amazon Kinesis 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

Common Parameters

This section lists the request parameters that all actions use. Any action-specific parameters are listed in the topic for the action.

Action

The action to be performed.

Default: None

Type: string

Required: Yes

AuthParams

The parameters that are required to authenticate a Conditional request. Contains:

- AWSAccessKeyID
- SignatureVersion
- Timestamp
- Signature

Default: None

Required: Conditional

AWSAccessKeyId

The access key ID that corresponds to the secret access key that you used to sign the request.

Default: None

Type: string

Required: Yes

Expires

The date and time when the request signature expires, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

SecurityToken

The temporary security token that was obtained through a call to AWS Security Token Service. For a list of services that support AWS Security Token Service, go to Using Temporary Security Credentials to Access AWS in **Using Temporary Security Credentials**.

Default: None

Type: string

Required: No

Signature

The digital signature that you created for the request. For information about generating a signature, go to the service's developer documentation.

Default: None

Type: string

Required: Yes

SignatureMethod

The hash algorithm that you used to create the request signature.

Default: None

Type: string

Valid Values: `HmacSHA256 | HmacSHA1`

Required: Yes

SignatureVersion

The signature version you use to sign the request. Set this to the value that is recommended for your service.

Default: None

Type: string

Required: Yes

Timestamp

The date and time when the request was signed, expressed in the format YYYY-MM-DDThh:mm:ssZ, as specified in the ISO 8601 standard.

Condition: Requests must include either *Timestamp* or *Expires*, but not both.

Default: None

Type: string

Required: Conditional

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Default: None

Type: string

Required: Yes

Common Parameters for Signature V4 Signing

The following table lists the parameters that all actions use for signing Signature Version 4 requests. Any action-specific parameters are listed in the topic for that action. To view sample requests, see Examples of Signed Signature Version 4 Requests or Signature Version 4 Test Suite in the *Amazon Web Services General Reference* .

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: *access_key*/*YYYYMMDD*/*region*/*service*/aws4_request.

For more information, see Task 2: Create a String to Sign for Signature Version 4 in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service. For a list of services that support AWS Security Token Service, go to Using Temporary Security Credentials to Access AWS in *Using Temporary Security Credentials*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Task 1: Create a Canonical Request For Signature Version 4 in the *Amazon Web Services General Reference* .

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

Common Errors

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

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

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

HTTP Status Code: 500

InvalidAction

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

HTTP Status Code: 400

InvalidClientTokenId

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

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

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

HTTP Status Code: 400

InvalidQueryParameter

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

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

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

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

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

HTTP Status Code: 403

RequestExpired

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

HTTP Status Code: 400

ServiceUnavailable

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

HTTP Status Code: 503

Throttling

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

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

HTTP Status Code: 400