Amazon Kinesis Streams Service API Reference

API Reference

API Version 2013-12-02

Amazon Kinesis Streams Service API Reference: API Reference

Table of Contents

Welcome

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

This document was last published on September 8, 2016.

Actions

The following actions are supported:

- AddTagsToStream (p. 3)
- CreateStream (p. 5)
- DecreaseStreamRetentionPeriod (p. 8)
- DeleteStream (p. 10)
- DescribeStream (p. 12)
- DisableEnhancedMonitoring (p. 16)
- EnableEnhancedMonitoring (p. 19)
- GetRecords (p. 22)
- GetShardIterator (p. 26)
- IncreaseStreamRetentionPeriod (p. 30)
- ListStreams (p. 32)
- ListTagsForStream (p. 34)
- MergeShards (p. 37)
- PutRecord (p. 40)
- PutRecords (p. 43)
- RemoveTagsFromStream (p. 48)
- SplitShard (p. 50)

AddTagsToStream

Adds or updates tags for the specified Amazon Kinesis stream. Each stream can have up to 10 tags.

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

Request Syntax

```
{
   "StreamName (p. 3)": "string",
   "Tags (p. 3)": {
      "string" : "string"
   }
}
```

Request Parameters

The request requires 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)

The set of key-value pairs to use to create the tags.

Type: String to String map

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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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>
```

CreateStream

Creates an Amazon Kinesis 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 5 transactions per second, up to a maximum data read total of 2 MB per second. Each shard can support writes up to 1,000 records per second, up to a maximum data write total of 1 MB 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.

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

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

CreateStream (p. 5) has a limit of 5 transactions per second per account.

Request Syntax

```
{
   "ShardCount (p. 5)": number,
   "StreamName (p. 5)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

ShardCount (p. 5)

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.

DefaultShardLimit;

Type: Integer

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

Required: Yes

StreamName (p. 5)

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.

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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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>
```

DecreaseStreamRetentionPeriod

Decreases the Amazon Kinesis 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 (p. 8)": number,
   "StreamName (p. 8)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

RetentionPeriodHours (p. 8)

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

Type: Integer

Valid Range: Minimum value of 24. Maximum value of 168.

Required: Yes

StreamName (p. 8)

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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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>
```

DeleteStream

Deletes an Amazon Kinesis 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 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. 40), PutRecords (p. 43), and GetRecords (p. 22), 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. 12) operation to check the state of the stream, which is returned in `StreamStatus`.

DeleteStream (p. 10) has a limit of 5 transactions per second per account.

Request Syntax

```
{
   "StreamName (p. 10)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

StreamName (p. 10)

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. 63).

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed (5).

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>
```

DescribeStream

Describes the specified Amazon Kinesis stream.

The stream information includes 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 using the `Limit` parameter. In the output, if `HasMoreShards` is set to `true`, there are more shards available. If there are more shards available, you can request them using the shard ID of the last shard returned. Specify this ID in the `ExclusiveStartShardId` parameter in a subsequent call. For more information, see Retrieving Shards from a Stream in the *Amazon Kinesis Streams Developer Guide*.

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

DescribeStream (p. 12) has a limit of 10 transactions per second per account.

Request Syntax

```
{
   "ExclusiveStartShardId (p. 12)": "string",
   "Limit (p. 12)": number,
   "StreamName (p. 12)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

ExclusiveStartShardId (p. 12)

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. 12)

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. 12)

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 (p. 13)": {
      "EnhancedMonitoring (p. 61)": [
         {
            "ShardLevelMetrics (p. 54)": [ "string" ]
         }
      ],
      "HasMoreShards (p. 61)": boolean,
      "RetentionPeriodHours (p. 61)": number,
      "Shards (p. 61)": [
         {
            "AdjacentParentShardId (p. 60)": "string",
            "HashKeyRange (p. 60)": {
               "EndingHashKey (p. 55)": "string",
               "StartingHashKey (p. 55)": "string"
            },
            "ParentShardId (p. 60)": "string",
            "SequenceNumberRange (p. 60)": {
               "EndingSequenceNumber (p. 59)": "string",
               "StartingSequenceNumber (p. 59)": "string"
            },
            "ShardId (p. 60)": "string"
         }
      ],
      "StreamARN (p. 61)": "string",
      "StreamName (p. 61)": "string",
      "StreamStatus (p. 61)": "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. 13)

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

Type: StreamDescription (p. 61) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 63).

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed (5).

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,
    "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"
  }
}
```

DisableEnhancedMonitoring

Disables enhanced monitoring.

Request Syntax

```
{
   "ShardLevelMetrics (p. 16)": [ "string" ],
   "StreamName (p. 16)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

ShardLevelMetrics (p. 16)

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 Streams Service with Amazon CloudWatch in the *Amazon Kinesis 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. 16)

The name of the Amazon Kinesis 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 (p. 17)": [ "string" ],
   "DesiredShardLevelMetrics (p. 17)": [ "string" ],
   "StreamName (p. 17)": "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. 16)

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. 16)

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. 16)

The name of the Amazon Kinesis 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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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": []
}
```

EnableEnhancedMonitoring

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

Request Syntax

```
{
   "ShardLevelMetrics (p. 19)": [ "string" ],
   "StreamName (p. 19)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

ShardLevelMetrics (p. 19)

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 Streams Service with Amazon CloudWatch in the *Amazon Kinesis 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. 19)

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 (p. 20)": [ "string" ],
   "DesiredShardLevelMetrics (p. 20)": [ "string" ],
   "StreamName (p. 20)": "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. 19)

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. 19)

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. 19)

The name of the Amazon Kinesis 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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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"
    ]
}
```

GetRecords

Gets data records from an Amazon Kinesis 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. 22) returns an empty list. Note that 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 Streams Limits in the *Amazon Kinesis 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. 22) in a loop. Use GetShardIterator (p. 26) to get the shard iterator to specify in the first GetRecords (p. 22) call. GetRecords (p. 22) returns a new shard iterator in `NextShardIterator`. Specify the shard iterator returned in `NextShardIterator` in subsequent calls to GetRecords (p. 22). Note that if the shard has been closed, the shard iterator can't return more data and GetRecords (p. 22) 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 MB in size, and each shard can read up to 2 MB 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. 22) can return. Consider your average record size when determining this limit.

The size of the data returned by GetRecords (p. 22) varies depending on the utilization of the shard. The maximum size of data that GetRecords (p. 22) can return is 10 MB. 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 shard, subsequent calls made within the next 1 second throw `ProvisionedThroughputExceededException`. Note that GetRecords (p. 22) won't return any data when it throws an exception. For this reason, we recommend that you wait one second between calls to GetRecords (p. 22); 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 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 timestamp, whereas a client-side timestamp 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. 43)). The timestamp has millisecond precision. There are no guarantees about the timestamp accuracy, or that the timestamp is always increasing. For example, records in a shard or across a stream might have timestamps that are out of order.

Request Syntax

```
{
   "Limit (p. 22)": number,
   "ShardIterator (p. 23)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

Limit (p. 22)

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. 22) throws `InvalidArgumentException`.

Type: Integer

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

Required: No

ShardIterator (p. 22)

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 (p. 23)": number,
   "NextShardIterator (p. 23)": "string",
   "Records (p. 23)": [
      {
         "ApproximateArrivalTimestamp (p. 58)": number,
         "Data (p. 58)": blob,
         "PartitionKey (p. 58)": "string",
         "SequenceNumber (p. 58)": "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. 23)

The number of milliseconds the GetRecords (p. 22) response is from the tip of the stream, indicating how far behind current time the consumer is. A value of zero indicates 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. 23)

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 (p. 23)

The data records retrieved from the shard.

Type: array of Record (p. 58) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 63).

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

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 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+8e0X7JT
sASDM/Hr2rSwc0X2qz93iuA3udrjTH+ikQvpQk/1ZcMMLzRdAesqwBGPnsthzU0/CBlM/U8/8oEqG
wX3pKw0XyeDNRAAZyXBo3MqkQtCpXhr942BRTjvWKhFz7OmCb2Ncfr8Tl2cB
ktooi6kJhr+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+CN
vMd6uXQ+HL5vAJMznqqC0DLKsIjuoiTi1BpT6nW0LN2M2D56zM5H8anHm30Gbri9ua+qaGgj+3XTyvbh
pERfrezgLHbPB/rIcVpykJbaSj5tmcXYRmFnqZBEyHwtZYFmh6hvWVFkIwLuMZLMrpWhG5r5hzkE=",

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

GetShardIterator

Gets an Amazon Kinesis shard iterator. A shard iterator expires five 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, or right after the sequence number by using the `AFTER_SEQUENCE_NUMBER` shard iterator type, using sequence numbers returned by earlier calls to PutRecord (p. 40), PutRecords (p. 43), GetRecords (p. 22), or DescribeStream (p. 12). 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. 26) request to get the first shard iterator for use in your first GetRecords (p. 22) request and for subsequent reads use the shard iterator returned by the GetRecords (p. 22) request in `NextShardIterator`. A new shard iterator is returned by every GetRecords (p. 22) request in `NextShardIterator`, which you use in the `ShardIterator` parameter of the next GetRecords (p. 22) request.

If a GetShardIterator (p. 26) request is made too often, you receive a `ProvisionedThroughputExceededException`. For more information about throughput limits, see GetRecords (p. 22), and Streams Limits in the *Amazon Kinesis Streams Developer Guide*.

If the shard is closed, GetShardIterator (p. 26) returns a valid iterator for the last sequence number of the shard. Note that a shard can be closed as a result of using SplitShard (p. 50) or MergeShards (p. 37).

GetShardIterator (p. 26) has a limit of 5 transactions per second per account per open shard.

Request Syntax

```
{
   "ShardId (p. 26)": "string",
   "ShardIteratorType (p. 26)": "string",
   "StartingSequenceNumber (p. 27)": "string",
   "StreamName (p. 27)": "string",
   "Timestamp (p. 27)": number
}
```

Request Parameters

The request requires the following data in JSON format.

ShardId (p. 26)

The shard ID of the Amazon Kinesis 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. 26)

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 timestamp, 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. 26)

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. 26)

The name of the Amazon Kinesis stream.

Type: String

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

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

Required: Yes

Timestamp (p. 26)

The timestamp of the data record from which to start reading. Used with shard iterator type AT_TIMESTAMP. A timestamp 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 timestamp does not exist, the iterator returned is for the next (later) record. If the timestamp 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 (p. 27)": "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. 27)

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. 63).

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 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+8e0X7JT
sASDM/Hr2rSwc0X2qz93iuA3udrjTH+ikQvpQk/1ZcMMLzRdAesqwBGPnsthzU0/CBlM/U8/8oEqG
wX3pKw0XyeDNRAAZyXBo3MqkQtCpXhr942BRTjvWKhFz7OmCb2Ncfr8Tl2cB
ktooi6kJhr+djN5WYkB38Rr3akRgCl9qaU4dY="
}
```

IncreaseStreamRetentionPeriod

Increases the Amazon Kinesis 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).

Upon choosing a longer stream retention period, this operation will increase the time period records are accessible that have not yet expired. However, it will not make previous data that has expired (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 will remain inaccessible to consumer applications.

Request Syntax

```
{
   "RetentionPeriodHours (p. 30)": number,
   "StreamName (p. 30)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

RetentionPeriodHours (p. 30)

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

Type: Integer

Valid Range: Minimum value of 24. Maximum value of 168.

Required: Yes

StreamName (p. 30)

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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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>
```

ListStreams

Lists your Amazon Kinesis 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, 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 (p. 32) has a limit of 5 transactions per second per account.

Request Syntax

```
{
   "ExclusiveStartStreamName (p. 32)": "string",
   "Limit (p. 32)": number
}
```

Request Parameters

The request requires the following data in JSON format.

ExclusiveStartStreamName (p. 32)

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. 32)

The maximum number of streams to list.

Type: Integer

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

Required: No

Response Syntax

```
{
   "HasMoreStreams (p. 33)": boolean,
   "StreamNames (p. 33)": [ "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. 32)

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

Type: Boolean

StreamNames (p. 32)

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. 63).

LimitExceededException

The requested resource exceeds the maximum number allowed, or the number of concurrent stream requests exceeds the maximum number allowed (5).

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"
  ]
}
```

ListTagsForStream

Lists the tags for the specified Amazon Kinesis stream.

Request Syntax

```
{
   "ExclusiveStartTagKey (p. 34)": "string",
   "Limit (p. 34)": number,
   "StreamName (p. 34)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

ExclusiveStartTagKey (p. 34)

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. 34)

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 10.

Required: No

StreamName (p. 34)

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 (p. 35)": boolean,
   "Tags (p. 35)": [
      {
         "Key (p. 62)": "string",
         "Value (p. 62)": "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. 34)

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. 34)

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. 62) objects

Array Members: Minimum number of 0 items.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 63).

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 (5).

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"
     }
   ]
}
```

MergeShards

Merges two adjacent shards in an Amazon Kinesis 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 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. 12) 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 DescribeStream (p. 12) 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. 5), DeleteStream (p. 10), `MergeShards` or SplitShard (p. 50), you will receive a `LimitExceededException`.

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

Request Syntax

```
{
   "AdjacentShardToMerge (p. 37)": "string",
   "ShardToMerge (p. 37)": "string",
   "StreamName (p. 38)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

AdjacentShardToMerge (p. 37)

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. 37)

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. 37)

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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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>
```

PutRecord

Writes a single data record into an Amazon Kinesis 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 MB 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 Amazon Kinesis to distribute data across shards. Amazon Kinesis segregates the data records that belong to a 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 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 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 Streams 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 a stream.

Request Syntax

```
{
   "Data (p. 40)": blob,
   "ExplicitHashKey (p. 40)": "string",
   "PartitionKey (p. 41)": "string",
   "SequenceNumberForOrdering (p. 41)": "string",
   "StreamName (p. 41)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

Data (p. 40)

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 MB).

Type: Base64-encoded binary data

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

Required: Yes

ExplicitHashKey (p. 40)

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. 40)

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 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. 40)

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 will be coarsely ordered based on arrival time.

Type: String

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

Required: No

StreamName (p. 40)

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

```
{
   "SequenceNumber (p. 41)": "string",
   "ShardId (p. 41)": "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.

SequenceNumber (p. 41)

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. 41)

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. 63).

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 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"
}
```

PutRecords

Writes multiple data records into an Amazon Kinesis 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 MB, up to a limit of 5 MB 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 MB 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 Amazon Kinesis 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 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 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. Amazon Kinesis attempts to process all records in each `PutRecords` request. A single record failure does not stop the processing of subsequent records.

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 Streams Developer Guide*.

By default, data records are accessible for only 24 hours from the time that they are added to an Amazon Kinesis stream. This retention period can be modified using the DecreaseStreamRetentionPeriod (p. 8) and IncreaseStreamRetentionPeriod (p. 30) operations.

Request Syntax

```
{
   "Records (p. 44)": [
      {
         "Data (p. 56)": blob,
         "ExplicitHashKey (p. 56)": "string",
         "PartitionKey (p. 56)": "string"
      }
   ],
   "StreamName (p. 44)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

Records (p. 43)

The records associated with the request.

Type: array of PutRecordsRequestEntry (p. 56) objects

Array Members: Minimum number of 1 item. Maximum number of 500 items.

Required: Yes

StreamName (p. 43)

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

```
{
   "FailedRecordCount (p. 44)": number,
   "Records (p. 44)": [
      {
         "ErrorCode (p. 57)": "string",
         "ErrorMessage (p. 57)": "string",
         "SequenceNumber (p. 57)": "string",
         "ShardId (p. 57)": "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.

FailedRecordCount (p. 44)

The number of unsuccessfully processed records in a `PutRecords` request.

Type: Integer

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

Records (p. 44)

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. 57) 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. 63).

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 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."
        }
    ]
}
```

RemoveTagsFromStream

Removes tags from the specified Amazon Kinesis 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.

Request Syntax

```
{
   "StreamName (p. 48)": "string",
   "TagKeys (p. 48)": [ "string" ]
}
```

Request Parameters

The request requires the following data in JSON format.

StreamName (p. 48)

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. 48)

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 10 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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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>
```

SplitShard

Splits a shard into two new shards in the Amazon Kinesis 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 Amazon Kinesis 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 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 Split a Shard in the *Amazon Kinesis Streams Developer Guide*.

You can use DescribeStream (p. 12) 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, `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 Streams Limits in the *Amazon Kinesis Streams Developer Guide*. If you need to increase this limit, contact AWS Support.

If you try to operate on too many streams simultaneously using CreateStream (p. 5), DeleteStream (p. 10), MergeShards (p. 37), and/or SplitShard (p. 50), you receive a `LimitExceededException`.

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

Request Syntax

```
{
   "NewStartingHashKey (p. 50)": "string",
   "ShardToSplit (p. 51)": "string",
   "StreamName (p. 51)": "string"
}
```

Request Parameters

The request requires the following data in JSON format.

NewStartingHashKey (p. 50)

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. 50)

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. 50)

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. 63).

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 (5).

HTTP Status Code: 400

ResourceInUseException

The resource is not available for this operation. For successful operation, the resource needs to 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>
```

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:

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 Streams Service with Amazon CloudWatch in the *Amazon Kinesis 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

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

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 MB).

Type: Base64-encoded binary data

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 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

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

Record

The unit of data of the Amazon Kinesis 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 the Amazon Kinesis service, 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 MB).

Type: Base64-encoded binary data

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

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 of the record in the stream.

Type: String

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

Required: Yes

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

Shard

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.

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. 55) 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. 59) 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

StreamDescription

Represents the output for DescribeStream (p. 12).

Contents

EnhancedMonitoring

Represents the current enhanced monitoring settings of the stream.

Type: array of EnhancedMetrics (p. 54) objects

Required: Yes

HasMoreShards

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

Type: Boolean

Required: Yes

RetentionPeriodHours

The current retention period, in hours.

Type: Integer

Valid Range: Minimum value of 24. Maximum value of 168.

Required: Yes

Shards

The shards that comprise the stream.

Type: array of Shard (p. 60) 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.

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. Amazon Kinesis immediately returns and sets `StreamStatus` to `CREATING`.
- `DELETING` - The stream is being deleted. 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

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

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