aws

API Reference

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams: API Reference

Services or capabilities described in AWS documentation might vary by Region. To see the differences applicable to the AWS European Sovereign Cloud Region, see the AWS European Sovereign Cloud User Guide.

Table of Contents

Welcome

Amazon Kinesis Video Streams

Amazon Kinesis Video Streams is a fully managed service that you can use to stream live video from devices to the cloud, or build applications for real-time video processing or batch-oriented video analytics.

Amazon Kinesis Video Streams Media

Amazon Kinesis Video Streams Media provides APIs for reading and writing media data to and from a Kinesis video stream. Use these APIs to send media data from a producer to a Kinesis video stream, or to read media data from a stream consumer.

Amazon Kinesis Video Streams Archived Media

Amazon Kinesis Video Streams Archived Media provides APIs for reading and processing archived media from Kinesis video streams. Use these APIs to retrieve media clips, retrieve HLS and DASH streaming session URLs, retrieve images, and list archived fragments.

Amazon Kinesis Video Signaling Channels

Kinesis Video Streams Signaling Service is a intermediate service that establishes a communication channel for discovering peers, and transmitting offers and answers in order to establish peer-to-peer connection in webRTC technology.

Actions

The following actions are supported by Amazon Kinesis Video Streams:

- CreateSignalingChannel
- CreateStream
- DeleteEdgeConfiguration
- DeleteSignalingChannel
- DeleteStream
- DescribeEdgeConfiguration
- DescribeImageGenerationConfiguration
- DescribeMappedResourceConfiguration
- DescribeMediaStorageConfiguration
- DescribeNotificationConfiguration
- DescribeSignalingChannel
- DescribeStream
- DescribeStreamStorageConfiguration
- GetDataEndpoint
- GetSignalingChannelEndpoint
- ListEdgeAgentConfigurations
- ListSignalingChannels
- ListStreams
- ListTagsForResource
- ListTagsForStream
- StartEdgeConfigurationUpdate
- TagResource
- TagStream
- UntagResource
- UntagStream
- UpdateDataRetention
- UpdateImageGenerationConfiguration

- UpdateMediaStorageConfiguration
- UpdateNotificationConfiguration
- UpdateSignalingChannel
- UpdateStream
- UpdateStreamStorageConfiguration

The following actions are supported by Amazon Kinesis Video Streams Media:

- GetMedia
- PutMedia

The following actions are supported by Amazon Kinesis Video Streams Archived Media:

- GetClip
- GetDASHStreamingSessionURL
- GetHLSStreamingSessionURL
- GetImages
- GetMediaForFragmentList
- ListFragments

The following actions are supported by Amazon Kinesis Video Signaling Channels:

- GetIceServerConfig
- SendAlexaOfferToMaster

Amazon Kinesis Video Streams

The following actions are supported by Amazon Kinesis Video Streams:

- CreateSignalingChannel
- CreateStream
- DeleteEdgeConfiguration
- DeleteSignalingChannel

- DeleteStream
- DescribeEdgeConfiguration
- DescribeImageGenerationConfiguration
- DescribeMappedResourceConfiguration
- DescribeMediaStorageConfiguration
- DescribeNotificationConfiguration
- DescribeSignalingChannel
- DescribeStream
- DescribeStreamStorageConfiguration
- GetDataEndpoint
- GetSignalingChannelEndpoint
- ListEdgeAgentConfigurations
- ListSignalingChannels
- ListStreams
- ListTagsForResource
- ListTagsForStream
- StartEdgeConfigurationUpdate
- TagResource
- TagStream
- UntagResource
- UntagStream
- UpdateDataRetention
- UpdateImageGenerationConfiguration
- UpdateMediaStorageConfiguration
- UpdateNotificationConfiguration
- UpdateSignalingChannel
- UpdateStream
- UpdateStreamStorageConfiguration

CreateSignalingChannel

Service: Amazon Kinesis Video Streams

Creates a signaling channel.

`CreateSignalingChannel` is an asynchronous operation.

Request Syntax

```
POST /createSignalingChannel HTTP/1.1
Content-type: application/json

{
   "ChannelName": "string",
   "ChannelType": "string",
   "SingleMasterConfiguration": {
      "MessageTtlSeconds": number
   },
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelName

A name for the signaling channel that you are creating. It must be unique for each AWS account and AWS Region.

Type: String

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

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

Required: Yes

ChannelType

A type of the signaling channel that you are creating. Currently, `SINGLE_MASTER` is the only supported channel type.

Type: String

Valid Values: `SINGLE_MASTER | FULL_MESH`

Required: No

SingleMasterConfiguration

A structure containing the configuration for the `SINGLE_MASTER` channel type. The default configuration for the channel message's time to live is 60 seconds (1 minute).

Type: SingleMasterConfiguration object

Required: No

Tags

A set of tags (key-value pairs) that you want to associate with this channel.

Type: Array of Tag objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ChannelARN": "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.

ChannelARN

The Amazon Resource Name (ARN) of the created channel.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

AccountChannelLimitExceededException

You have reached the maximum limit of active signaling channels for this AWS account in this region.

HTTP Status Code: 400

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

TagsPerResourceExceededLimitException

You have exceeded the limit of tags that you can associate with the resource. A Kinesis video stream can support up to 50 tags.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

CreateStream

Service: Amazon Kinesis Video Streams

Creates a new Kinesis video stream.

When you create a new stream, Kinesis Video Streams assigns it a version number. When you change the stream's metadata, Kinesis Video Streams updates the version.

`CreateStream` is an asynchronous operation.

For information about how the service works, see How it Works.

You must have permissions for the `KinesisVideo:CreateStream` action.

Request Syntax

```
POST /createStream HTTP/1.1
Content-type: application/json

{
   "DataRetentionInHours": number,
   "DeviceName": "string",
   "KmsKeyId": "string",
   "MediaType": "string",
   "StreamName": "string",
   "StreamStorageConfiguration": {
      "DefaultStorageTier": "string"
   },
   "Tags": {
      "string" : "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

DataRetentionInHours

The number of hours that you want to retain the data in the stream. Kinesis Video Streams retains the data in a data store that is associated with the stream.

The default value is 0, indicating that the stream does not persist data. The minimum is 1 hour.

When the `DataRetentionInHours` value is 0, consumers can still consume the fragments that remain in the service host buffer, which has a retention time limit of 5 minutes and a retention memory limit of 200 MB. Fragments are removed from the buffer when either limit is reached.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

DeviceName

The name of the device that is writing to the stream.

> **Note**
>
> In the current implementation, Kinesis Video Streams doesn't use this name.

Type: String

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

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

Required: No

KmsKeyId

The ID of the AWS Key Management Service (AWS KMS) key that you want Kinesis Video Streams to use to encrypt stream data.

If no key ID is specified, the default, Kinesis Video-managed key (`aws/kinesisvideo`) is used.

For more information, see DescribeKey.

Type: String

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

Pattern: `.+`

Required: No

MediaType

The media type of the stream. Consumers of the stream can use this information when processing the stream. For more information about media types, see Media Types. If you choose to specify the `MediaType`, see Naming Requirements for guidelines.

Example valid values include "video/h264" and "video/h264,audio/aac".

This parameter is optional; the default value is `null` (or empty in JSON).

Type: String

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

Pattern: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Required: No

StreamName

A name for the stream that you are creating.

The stream name is an identifier for the stream, and must be unique for each account and region.

Type: String

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

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

Required: Yes

StreamStorageConfiguration

The configuration for the stream's storage, including the default storage tier for stream data. This configuration determines how stream data is stored and accessed, with different tiers offering varying levels of performance and cost optimization.

If not specified, the stream will use the default storage configuration with HOT tier for optimal performance.

Type: StreamStorageConfiguration object

Required: No

Tags

A list of tags to associate with the specified stream. Each tag is a key-value pair (the value is optional).

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "StreamARN": "string"
}
```

Response Elements

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

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

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Errors

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

AccountStreamLimitExceededException

The number of streams created for the account is too high.

HTTP Status Code: 400

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

DeviceStreamLimitExceededException

Not implemented.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

InvalidDeviceException

Not implemented.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

TagsPerResourceExceededLimitException

You have exceeded the limit of tags that you can associate with the resource. A Kinesis video stream can support up to 50 tags.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteEdgeConfiguration

Service: Amazon Kinesis Video Streams

An asynchronous API that deletes a stream's existing edge configuration, as well as the corresponding media from the Edge Agent.

When you invoke this API, the sync status is set to `DELETING`. A deletion process starts, in which active edge jobs are stopped and all media is deleted from the edge device. The time to delete varies, depending on the total amount of stored media. If the deletion process fails, the sync status changes to `DELETE_FAILED`. You will need to re-try the deletion.

When the deletion process has completed successfully, the edge configuration is no longer accessible.

> **Note**
>
> This API isn't available in the AWS Africa (Cape Town) region, af-south-1.

Request Syntax

```
POST /deleteEdgeConfiguration HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StreamARN

The Amazon Resource Name (ARN) of the stream. Specify either the `StreamName` or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream from which to delete the edge configuration. Specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

StreamEdgeConfigurationNotFoundException

The Exception rendered when the Amazon Kinesis Video Stream can't find a stream's edge configuration that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteSignalingChannel

Service: Amazon Kinesis Video Streams

Deletes a specified signaling channel. `DeleteSignalingChannel` is an asynchronous operation. If you don't specify the channel's current version, the most recent version is deleted.

Request Syntax

```
POST /deleteSignalingChannel HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "CurrentVersion": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelARN

The Amazon Resource Name (ARN) of the signaling channel that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

CurrentVersion

The current version of the signaling channel that you want to delete. You can obtain the current version by invoking the `DescribeSignalingChannel` or `ListSignalingChannels` API operations.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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

Required: No

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

VersionMismatchException

The stream version that you specified is not the latest version. To get the latest version, use the DescribeStream API.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteStream

Service: Amazon Kinesis Video Streams

Deletes a Kinesis video stream and the data contained in the stream.

This method marks the stream for deletion, and makes the data in the stream inaccessible immediately.

To ensure that you have the latest version of the stream before deleting it, you can specify the stream version. Kinesis Video Streams assigns a version to each stream. When you update a stream, Kinesis Video Streams assigns a new version number. To get the latest stream version, use the `DescribeStream` API.

This operation requires permission for the `KinesisVideo:DeleteStream` action.

Request Syntax

```
POST /deleteStream HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

CurrentVersion

Optional: The version of the stream that you want to delete.

Specify the version as a safeguard to ensure that your are deleting the correct stream. To get the stream version, use the `DescribeStream` API.

If not specified, only the `CreationTime` is checked before deleting the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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

Required: No

StreamARN

The Amazon Resource Name (ARN) of the stream that you want to delete.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

VersionMismatchException

The stream version that you specified is not the latest version. To get the latest version, use the DescribeStream API.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeEdgeConfiguration

Service: Amazon Kinesis Video Streams

Describes a stream's edge configuration that was set using the `StartEdgeConfigurationUpdate` API and the latest status of the edge agent's recorder and uploader jobs. Use this API to get the status of the configuration to determine if the configuration is in sync with the Edge Agent. Use this API to evaluate the health of the Edge Agent.

> **Note**
>
> This API isn't available in the AWS Africa (Cape Town) region, af-south-1.

Request Syntax

```
POST /describeEdgeConfiguration HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StreamARN

The Amazon Resource Name (ARN) of the stream. Specify either the `StreamName`or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream whose edge configuration you want to update. Specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "CreationTime": number,
   "EdgeAgentStatus": {
      "LastRecorderStatus": {
         "JobStatusDetails": "string",
         "LastCollectedTime": number,
         "LastUpdatedTime": number,
         "RecorderStatus": "string"
      },
      "LastUploaderStatus": {
         "JobStatusDetails": "string",
         "LastCollectedTime": number,
         "LastUpdatedTime": number,
         "UploaderStatus": "string"
      }
   },
   "EdgeConfig": {
      "DeletionConfig": {
         "DeleteAfterUpload": boolean,
         "EdgeRetentionInHours": number,
         "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": number,
            "StrategyOnFullSize": "string"
         }
```

```
      },
      "HubDeviceArn": "string",
      "RecorderConfig": {
         "MediaSourceConfig": {
            "MediaUriSecretArn": "string",
            "MediaUriType": "string"
         },
         "ScheduleConfig": {
            "DurationInSeconds": number,
            "ScheduleExpression": "string"
         }
      },
      "UploaderConfig": {
         "ScheduleConfig": {
            "DurationInSeconds": number,
            "ScheduleExpression": "string"
         }
      }
   },
   "FailedStatusDetails": "string",
   "LastUpdatedTime": number,
   "StreamARN": "string",
   "StreamName": "string",
   "SyncStatus": "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.

CreationTime

The timestamp at which a stream's edge configuration was first created.

Type: Timestamp

EdgeAgentStatus

An object that contains the latest status details for an edge agent's recorder and uploader jobs. Use this information to determine the current health of an edge agent.

Type: EdgeAgentStatus object

EdgeConfig

A description of the stream's edge configuration that will be used to sync with the Edge Agent IoT Greengrass component. The Edge Agent component will run on an IoT Hub Device setup at your premise.

Type: EdgeConfig object

FailedStatusDetails

A description of the generated failure status.

Type: String

LastUpdatedTime

The timestamp at which a stream's edge configuration was last updated.

Type: Timestamp

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

The name of the stream from which the edge configuration was updated.

Type: String

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

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

SyncStatus

The latest status of the edge configuration update.

Type: String

Valid Values: `SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED`

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

StreamEdgeConfigurationNotFoundException

The Exception rendered when the Amazon Kinesis Video Stream can't find a stream's edge configuration that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeImageGenerationConfiguration

Service: Amazon Kinesis Video Streams

Gets the `ImageGenerationConfiguration` for a given Kinesis video stream.

Request Syntax

```
POST /describeImageGenerationConfiguration HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StreamARN

The Amazon Resource Name (ARN) of the Kinesis video stream from which to retrieve the image generation configuration. You must specify either the `StreamName` or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream from which to retrieve the image generation configuration. You must specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ImageGenerationConfiguration": {
      "DestinationConfig": {
         "DestinationRegion": "string",
         "Uri": "string"
      },
      "Format": "string",
      "FormatConfig": {
         "string" : "string"
      },
      "HeightPixels": number,
      "ImageSelectorType": "string",
      "SamplingInterval": number,
      "Status": "string",
      "WidthPixels": number
   }
}
```

Response Elements

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

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

ImageGenerationConfiguration

The structure that contains the information required for the Kinesis video stream (KVS) images delivery. If this structure is null, the configuration will be deleted from the stream.

Type: ImageGenerationConfiguration object

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeMappedResourceConfiguration

Service: Amazon Kinesis Video Streams

Returns the most current information about the stream. The `streamName` or `streamARN` should be provided in the input.

Request Syntax

```
POST /describeMappedResourceConfiguration HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

MaxResults

The maximum number of results to return in the response.

Type: Integer

Valid Range: Fixed value of 1.

Required: No

NextToken

The token to provide in your next request, to get another batch of results.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "MappedResourceConfigurationList": [
      {
         "ARN": "string",
         "Type": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

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

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

MappedResourceConfigurationList

A structure that encapsulates, or contains, the media storage configuration properties.

Type: Array of MappedResourceConfigurationListItem objects

Array Members: Minimum number of 0 items. Maximum number of 1 item.

NextToken

The token that was used in the `NextToken`request to fetch the next set of results.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeMediaStorageConfiguration

Service: Amazon Kinesis Video Streams

Returns the most current information about the channel. Specify the `ChannelName` or `ChannelARN` in the input.

Request Syntax

```
POST /describeMediaStorageConfiguration HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "ChannelName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelARN

The Amazon Resource Name (ARN) of the channel.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

ChannelName

The name of the channel.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "MediaStorageConfiguration": {
      "Status": "string",
      "StreamARN": "string"
   }
}
```

Response Elements

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

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

MediaStorageConfiguration

A structure that encapsulates, or contains, the media storage configuration properties.

Type: MediaStorageConfiguration object

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeNotificationConfiguration

Service: Amazon Kinesis Video Streams

Gets the `NotificationConfiguration` for a given Kinesis video stream.

Request Syntax

```
POST /describeNotificationConfiguration HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StreamARN

The Amazon Resource Name (ARN) of the Kinesis video stream from where you want to retrieve the notification configuration. You must specify either the `StreamName` or the StreamARN.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream from which to retrieve the notification configuration. You must specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "NotificationConfiguration": {
      "DestinationConfig": {
         "Uri": "string"
      },
      "Status": "string"
   }
}
```

Response Elements

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

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

NotificationConfiguration

The structure that contains the information required for notifications. If the structure is null, the configuration will be deleted from the stream.

Type: NotificationConfiguration object

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeSignalingChannel

Service: Amazon Kinesis Video Streams

Returns the most current information about the signaling channel. You must specify either the name or the Amazon Resource Name (ARN) of the channel that you want to describe.

Request Syntax

```
POST /describeSignalingChannel HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "ChannelName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelARN

The ARN of the signaling channel that you want to describe.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

ChannelName

The name of the signaling channel that you want to describe.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ChannelInfo": {
      "ChannelARN": "string",
      "ChannelName": "string",
      "ChannelStatus": "string",
      "ChannelType": "string",
      "CreationTime": number,
      "SingleMasterConfiguration": {
         "MessageTtlSeconds": number
      },
      "Version": "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.

ChannelInfo

A structure that encapsulates the specified signaling channel's metadata and properties.

Type: ChannelInfo object

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeStream

Service: Amazon Kinesis Video Streams

Returns the most current information about the specified stream. You must specify either the `StreamName` or the `StreamARN`.

Request Syntax

```
POST /describeStream HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "StreamInfo": {
      "CreationTime": number,
      "DataRetentionInHours": number,
      "DeviceName": "string",
      "KmsKeyId": "string",
      "MediaType": "string",
      "Status": "string",
      "StreamARN": "string",
      "StreamName": "string",
      "Version": "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.

StreamInfo

An object that describes the stream.

Type: StreamInfo object

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeStreamStorageConfiguration

Service: Amazon Kinesis Video Streams

Retrieves the current storage configuration for the specified Kinesis video stream.

In the request, you must specify either the `StreamName` or the `StreamARN`.

You must have permissions for the `KinesisVideo:DescribeStreamStorageConfiguration` action.

Request Syntax

```
POST /describeStreamStorageConfiguration HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StreamARN

The Amazon Resource Name (ARN) of the stream for which you want to retrieve the storage configuration.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream for which you want to retrieve the storage configuration.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string",
   "StreamStorageConfiguration": {
      "DefaultStorageTier": "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.

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

The name of the stream.

Type: String

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

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

StreamStorageConfiguration

The current storage configuration for the stream, including the default storage tier and other storage-related settings.

Type: StreamStorageConfiguration object

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetDataEndpoint

Service: Amazon Kinesis Video Streams

Gets an endpoint for a specified stream for either reading or writing. Use this endpoint in your application to read from the specified stream (using the `GetMedia` or `GetMediaForFragmentList` operations) or write to it (using the `PutMedia` operation).

> **Note**
>
> The returned endpoint does not have the API name appended. The client needs to add the API name to the returned endpoint.

In the request, specify the stream either by `StreamName` or `StreamARN`.

Request Syntax

```
POST /getDataEndpoint HTTP/1.1
Content-type: application/json

{
   "APIName": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

APIName

The name of the API action for which to get an endpoint.

Type: String

Valid Values: `PUT_MEDIA | GET_MEDIA | LIST_FRAGMENTS | GET_MEDIA_FOR_FRAGMENT_LIST | GET_HLS_STREAMING_SESSION_URL | GET_DASH_STREAMING_SESSION_URL | GET_CLIP | GET_IMAGES`

Required: Yes

StreamARN

The Amazon Resource Name (ARN) of the stream that you want to get the endpoint for. You must specify either this parameter or a `StreamName` in the request.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream that you want to get the endpoint for. You must specify either this parameter or a `StreamARN` in the request.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "DataEndpoint": "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.

DataEndpoint

The endpoint value. To read data from the stream or to write data to it, specify this endpoint in your application.

Type: String

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetSignalingChannelEndpoint

Service: Amazon Kinesis Video Streams

Provides an endpoint for the specified signaling channel to send and receive messages. This API uses the `SingleMasterChannelEndpointConfiguration` input parameter, which consists of the `Protocols` and `Role` properties.

`Protocols` is used to determine the communication mechanism. For example, if you specify `WSS` as the protocol, this API produces a secure websocket endpoint. If you specify `HTTPS` as the protocol, this API generates an HTTPS endpoint. If you specify `WEBRTC` as the protocol, but the signaling channel isn't configured for ingestion, you will receive the error `InvalidArgumentException`.

`Role` determines the messaging permissions. A `MASTER` role results in this API generating an endpoint that a client can use to communicate with any of the viewers on the channel. A `VIEWER` role results in this API generating an endpoint that a client can use to communicate only with a `MASTER`.

Request Syntax

```
POST /getSignalingChannelEndpoint HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "SingleMasterChannelEndpointConfiguration": {
      "Protocols": [ "string" ],
      "Role": "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelARN

The Amazon Resource Name (ARN) of the signalling channel for which you want to get an endpoint.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

SingleMasterChannelEndpointConfiguration

A structure containing the endpoint configuration for the `SINGLE_MASTER` channel type.

Type: SingleMasterChannelEndpointConfiguration object

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ResourceEndpointList": [
      {
         "Protocol": "string",
         "ResourceEndpoint": "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.

ResourceEndpointList

A list of endpoints for the specified signaling channel.

Type: Array of ResourceEndpointListItem objects

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListEdgeAgentConfigurations

Service: Amazon Kinesis Video Streams

Returns an array of edge configurations associated with the specified Edge Agent.

In the request, you must specify the Edge Agent `HubDeviceArn`.

> **Note**
>
> This API isn't available in the AWS Africa (Cape Town) region, af-south-1.

Request Syntax

```
POST /listEdgeAgentConfigurations HTTP/1.1
Content-type: application/json

{
   "HubDeviceArn": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

HubDeviceArn

The "Internet of Things (IoT) Thing" Arn of the edge agent.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Required: Yes

MaxResults

The maximum number of edge configurations to return in the response. The default is 5.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 10.

Required: No

NextToken

If you specify this parameter, when the result of a `ListEdgeAgentConfigurations` operation is truncated, the call returns the `NextToken` in the response. To get another batch of edge configurations, provide this token in your next request.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "EdgeConfigs": [
      {
         "CreationTime": number,
         "EdgeConfig": {
            "DeletionConfig": {
               "DeleteAfterUpload": boolean,
               "EdgeRetentionInHours": number,
               "LocalSizeConfig": {
                  "MaxLocalMediaSizeInMB": number,
                  "StrategyOnFullSize": "string"
               }
            },
```

```
            "HubDeviceArn": "string",
            "RecorderConfig": {
               "MediaSourceConfig": {
                  "MediaUriSecretArn": "string",
                  "MediaUriType": "string"
               },
               "ScheduleConfig": {
                  "DurationInSeconds": number,
                  "ScheduleExpression": "string"
               }
            },
            "UploaderConfig": {
               "ScheduleConfig": {
                  "DurationInSeconds": number,
                  "ScheduleExpression": "string"
               }
            }
         },
         "FailedStatusDetails": "string",
         "LastUpdatedTime": number,
         "StreamARN": "string",
         "StreamName": "string",
         "SyncStatus": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

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

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

EdgeConfigs

A description of a single stream's edge configuration.

Type: Array of ListEdgeAgentConfigurationsEdgeConfig objects

NextToken

If the response is truncated, the call returns this element with a given token. To get the next batch of edge configurations, use this token in your next request.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListSignalingChannels

Service: Amazon Kinesis Video Streams

Returns an array of `ChannelInfo` objects. Each object describes a signaling channel. To retrieve only those channels that satisfy a specific condition, you can specify a `ChannelNameCondition`.

Request Syntax

```
POST /listSignalingChannels HTTP/1.1
Content-type: application/json

{
   "ChannelNameCondition": {
      "ComparisonOperator": "string",
      "ComparisonValue": "string"
   },
   "MaxResults": number,
   "NextToken": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelNameCondition

Optional: Returns only the channels that satisfy a specific condition.

Type: ChannelNameCondition object

Required: No

MaxResults

The maximum number of channels to return in the response. The default is 500.

Type: Integer

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

Required: No

NextToken

If you specify this parameter, when the result of a `ListSignalingChannels` operation is truncated, the call returns the `NextToken` in the response. To get another batch of channels, provide this token in your next request.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "ChannelInfoList": [
      {
         "ChannelARN": "string",
         "ChannelName": "string",
         "ChannelStatus": "string",
         "ChannelType": "string",
         "CreationTime": number,
         "SingleMasterConfiguration": {
            "MessageTtlSeconds": number
         },
         "Version": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

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

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

ChannelInfoList

An array of `ChannelInfo` objects.

Type: Array of ChannelInfo objects

NextToken

If the response is truncated, the call returns this element with a token. To get the next batch of streams, use this token in your next request.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListStreams

Service: Amazon Kinesis Video Streams

Returns an array of `StreamInfo` objects. Each object describes a stream. To retrieve only streams that satisfy a specific condition, you can specify a `StreamNameCondition`.

Request Syntax

```
POST /listStreams HTTP/1.1
Content-type: application/json

{
   "MaxResults": number,
   "NextToken": "string",
   "StreamNameCondition": {
      "ComparisonOperator": "string",
      "ComparisonValue": "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

MaxResults

The maximum number of streams to return in the response. The default is 10,000.

Type: Integer

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

Required: No

NextToken

If you specify this parameter, when the result of a `ListStreams` operation is truncated, the call returns the `NextToken` in the response. To get another batch of streams, provide this token in your next request.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

StreamNameCondition

Optional: Returns only streams that satisfy a specific condition. Currently, you can specify only the prefix of a stream name as a condition.

Type: StreamNameCondition object

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "NextToken": "string",
   "StreamInfoList": [
      {
         "CreationTime": number,
         "DataRetentionInHours": number,
         "DeviceName": "string",
         "KmsKeyId": "string",
         "MediaType": "string",
         "Status": "string",
         "StreamARN": "string",
         "StreamName": "string",
         "Version": "string"
      }
   ]
}
```

Response Elements

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

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

NextToken

If the response is truncated, the call returns this element with a token. To get the next batch of streams, use this token in your next request.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

StreamInfoList

An array of `StreamInfo` objects.

Type: Array of StreamInfo objects

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2

- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsForResource

Service: Amazon Kinesis Video Streams

Returns a list of tags associated with the specified signaling channel.

Request Syntax

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json

{
   "NextToken": "string",
   "ResourceARN": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

NextToken

If you specify this parameter and the result of a `ListTagsForResource` call is truncated, the response includes a token that you can use in the next request to fetch the next batch of tags.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

ResourceARN

The Amazon Resource Name (ARN) of the signaling channel for which you want to list tags.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

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

Response Elements

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

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

NextToken

If you specify this parameter and the result of a `ListTagsForResource` call is truncated, the response includes a token that you can use in the next request to fetch the next set of tags.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Tags

A map of tag keys and values associated with the specified signaling channel.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsForStream

Service: Amazon Kinesis Video Streams

Returns a list of tags associated with the specified stream.

In the request, you must specify either the `StreamName` or the `StreamARN`.

Request Syntax

```
POST /listTagsForStream HTTP/1.1
Content-type: application/json

{
   "NextToken": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

NextToken

If you specify this parameter and the result of a `ListTagsForStream` call is truncated, the response includes a token that you can use in the next request to fetch the next batch of tags.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Required: No

StreamARN

The Amazon Resource Name (ARN) of the stream that you want to list tags for.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream that you want to list tags for.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

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

Response Elements

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

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

NextToken

If you specify this parameter and the result of a `ListTags` call is truncated, the response includes a token that you can use in the next request to fetch the next set of tags.

Type: String

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

Pattern: `[a-zA-Z0-9+/=]*`

Tags

A map of tag keys and values associated with the specified stream.

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

InvalidResourceFormatException

The format of the `StreamARN` is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartEdgeConfigurationUpdate

Service: Amazon Kinesis Video Streams

An asynchronous API that updates a stream's existing edge configuration. The Kinesis Video Stream will sync the stream's edge configuration with the Edge Agent IoT Greengrass component that runs on an IoT Hub Device, setup at your premise. The time to sync can vary and depends on the connectivity of the Hub Device. The `SyncStatus` will be updated as the edge configuration is acknowledged, and synced with the Edge Agent.

If this API is invoked for the first time, a new edge configuration will be created for the stream, and the sync status will be set to `SYNCING`. You will have to wait for the sync status to reach a terminal state such as: `IN_SYNC`, or `SYNC_FAILED`, before using this API again. If you invoke this API during the syncing process, a `ResourceInUseException` will be thrown. The connectivity of the stream's edge configuration and the Edge Agent will be retried for 15 minutes. After 15 minutes, the status will transition into the `SYNC_FAILED` state.

To move an edge configuration from one device to another, use DeleteEdgeConfiguration to delete the current edge configuration. You can then invoke StartEdgeConfigurationUpdate with an updated Hub Device ARN.

> **Note**
>
> This API isn't available in the AWS Africa (Cape Town) region, af-south-1.

Request Syntax

```
POST /startEdgeConfigurationUpdate HTTP/1.1
Content-type: application/json

{
   "EdgeConfig": {
      "DeletionConfig": {
         "DeleteAfterUpload": boolean,
         "EdgeRetentionInHours": number,
         "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": number,
            "StrategyOnFullSize": "string"
         }
      },
```

```
      "HubDeviceArn": "string",
      "RecorderConfig": {
         "MediaSourceConfig": {
            "MediaUriSecretArn": "string",
            "MediaUriType": "string"
         },
         "ScheduleConfig": {
            "DurationInSeconds": number,
            "ScheduleExpression": "string"
         }
      },
      "UploaderConfig": {
         "ScheduleConfig": {
            "DurationInSeconds": number,
            "ScheduleExpression": "string"
         }
      }
   },
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

EdgeConfig

The edge configuration details required to invoke the update process.

Type: EdgeConfig object

Required: Yes

StreamARN

The Amazon Resource Name (ARN) of the stream. Specify either the `StreamName` or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream whose edge configuration you want to update. Specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "CreationTime": number,
   "EdgeConfig": {
      "DeletionConfig": {
         "DeleteAfterUpload": boolean,
         "EdgeRetentionInHours": number,
         "LocalSizeConfig": {
            "MaxLocalMediaSizeInMB": number,
            "StrategyOnFullSize": "string"
         }
      },
      "HubDeviceArn": "string",
      "RecorderConfig": {
         "MediaSourceConfig": {
            "MediaUriSecretArn": "string",
            "MediaUriType": "string"
         },
         "ScheduleConfig": {
            "DurationInSeconds": number,
```

```
            "ScheduleExpression": "string"
         }
      },
      "UploaderConfig": {
         "ScheduleConfig": {
            "DurationInSeconds": number,
            "ScheduleExpression": "string"
         }
      }
   },
   "FailedStatusDetails": "string",
   "LastUpdatedTime": number,
   "StreamARN": "string",
   "StreamName": "string",
   "SyncStatus": "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.

CreationTime

The timestamp at which a stream's edge configuration was first created.

Type: Timestamp

EdgeConfig

A description of the stream's edge configuration that will be used to sync with the Edge Agent IoT Greengrass component. The Edge Agent component will run on an IoT Hub Device setup at your premise.

Type: EdgeConfig object

FailedStatusDetails

A description of the generated failure status.

Type: String

LastUpdatedTime

The timestamp at which a stream's edge configuration was last updated.

Type: Timestamp

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

The name of the stream from which the edge configuration was updated.

Type: String

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

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

SyncStatus

The current sync status of the stream's edge configuration. When you invoke this API, the sync status will be set to the `SYNCING` state. Use the `DescribeEdgeConfiguration` API to get the latest status of the edge configuration.

Type: String

Valid Values: `SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED`

Errors

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

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NoDataRetentionException

The Stream data retention in hours is equal to zero.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagResource

Service: Amazon Kinesis Video Streams

Adds one or more tags to a signaling channel **only**. **Note :** To tag streams, use the TagStream API instead. A *tag* is a key-value pair (the value is optional) that you can define and assign to AWS resources. If you specify a tag that already exists, the tag value is replaced with the value that you specify in the request. For more information, see Using Cost Allocation Tags in the *AWS Billing and Cost Management and Cost Management User Guide*.

Request Syntax

```
POST /TagResource HTTP/1.1
Content-type: application/json

{
   "ResourceARN": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ResourceARN

The Amazon Resource Name (ARN) of the signaling channel to which you want to add tags.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

Tags

A list of tags to associate with the specified signaling channel. Each tag is a key-value pair.

Type: Array of Tag objects

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

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

TagsPerResourceExceededLimitException

You have exceeded the limit of tags that you can associate with the resource. A Kinesis video stream can support up to 50 tags.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagStream

Service: Amazon Kinesis Video Streams

Adds one or more tags to a stream. A *tag* is a key-value pair (the value is optional) that you can define and assign to AWS resources. If you specify a tag that already exists, the tag value is replaced with the value that you specify in the request. For more information, see Using Cost Allocation Tags in the *AWS Billing and Cost Management and Cost Management User Guide*.

You must provide either the `StreamName` or the `StreamARN`.

This operation requires permission for the `KinesisVideo:TagStream` action.

A Kinesis video stream can support up to 50 tags.

Request Syntax

```
POST /tagStream HTTP/1.1
Content-type: application/json

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StreamARN

The Amazon Resource Name (ARN) of the resource that you want to add the tag or tags to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream that you want to add the tag or tags to.

Type: String

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

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

Required: No

Tags

A list of tags to associate with the specified stream. Each tag is a key-value pair (the value is optional).

Type: String to string map

Map Entries: Maximum number of 50 items.

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

Key Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

InvalidResourceFormatException

The format of the `StreamARN` is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

TagsPerResourceExceededLimitException

You have exceeded the limit of tags that you can associate with the resource. A Kinesis video stream can support up to 50 tags.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagResource

Service: Amazon Kinesis Video Streams

Removes one or more tags from a signaling channel **only**. **Note :** To remove tags from streams, use the UntagStream API instead. In the request, specify only a tag key or keys; don't specify the value. If you specify a tag key that does not exist, it's ignored.

Request Syntax

```
POST /UntagResource HTTP/1.1
Content-type: application/json

{
   "ResourceARN": "string",
   "TagKeyList": [ "string" ]
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ResourceARN

The Amazon Resource Name (ARN) of the signaling channel from which you want to remove tags.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

TagKeyList

A list of the keys of the tags that you want to remove.

Type: Array of strings

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

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

Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagStream

Service: Amazon Kinesis Video Streams

Removes one or more tags from a stream. In the request, specify only a tag key or keys; don't specify the value. If you specify a tag key that does not exist, it's ignored.

In the request, you must provide the `StreamName` or `StreamARN`.

Request Syntax

```
POST /untagStream HTTP/1.1
Content-type: application/json

{
   "StreamARN": "string",
   "StreamName": "string",
   "TagKeyList": [ "string" ]
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StreamARN

The Amazon Resource Name (ARN) of the stream that you want to remove tags from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream that you want to remove tags from.

Type: String

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

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

Required: No

TagKeyList

A list of the keys of the tags that you want to remove.

Type: Array of strings

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

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

Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

InvalidResourceFormatException

The format of the `StreamARN` is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateDataRetention

Service: Amazon Kinesis Video Streams

Increases or decreases the stream's data retention period by the value that you specify. To indicate whether you want to increase or decrease the data retention period, specify the `Operation` parameter in the request body. In the request, you must specify either the `StreamName` or the `StreamARN`.

This operation requires permission for the `KinesisVideo:UpdateDataRetention` action.

Changing the data retention period affects the data in the stream as follows:

- If the data retention period is increased, existing data is retained for the new retention period. For example, if the data retention period is increased from one hour to seven hours, all existing data is retained for seven hours.
- If the data retention period is decreased, existing data is retained for the new retention period. For example, if the data retention period is decreased from seven hours to one hour, all existing data is retained for one hour, and any data older than one hour is deleted immediately.

Request Syntax

```
POST /updateDataRetention HTTP/1.1
Content-type: application/json

{
   "CurrentVersion": "string",
   "DataRetentionChangeInHours": number,
   "Operation": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

CurrentVersion

The version of the stream whose retention period you want to change. To get the version, call either the `DescribeStream` or the `ListStreams` API.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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

Required: Yes

DataRetentionChangeInHours

The number of hours to adjust the current retention by. The value you specify is added to or subtracted from the current value, depending on the `operation`.

The minimum value for data retention is 0 and the maximum value is 87600 (ten years).

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

Operation

Indicates whether you want to increase or decrease the retention period.

Type: String

Valid Values: `INCREASE_DATA_RETENTION | DECREASE_DATA_RETENTION`

Required: Yes

StreamARN

The Amazon Resource Name (ARN) of the stream whose retention period you want to change.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream whose retention period you want to change.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

VersionMismatchException

The stream version that you specified is not the latest version. To get the latest version, use the DescribeStream API.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateImageGenerationConfiguration

Service: Amazon Kinesis Video Streams

Updates the `StreamInfo` and `ImageProcessingConfiguration` fields.

Request Syntax

```
POST /updateImageGenerationConfiguration HTTP/1.1
Content-type: application/json

{
   "ImageGenerationConfiguration": {
      "DestinationConfig": {
         "DestinationRegion": "string",
         "Uri": "string"
      },
      "Format": "string",
      "FormatConfig": {
         "string" : "string"
      },
      "HeightPixels": number,
      "ImageSelectorType": "string",
      "SamplingInterval": number,
      "Status": "string",
      "WidthPixels": number
   },
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ImageGenerationConfiguration

The structure that contains the information required for the KVS images delivery. If the structure is null, the configuration will be deleted from the stream.

Type: ImageGenerationConfiguration object

Required: No

StreamARN

The Amazon Resource Name (ARN) of the Kinesis video stream from where you want to update the image generation configuration. You must specify either the `StreamName` or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream from which to update the image generation configuration. You must specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NoDataRetentionException

The Stream data retention in hours is equal to zero.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateMediaStorageConfiguration

Service: Amazon Kinesis Video Streams

Associates a `SignalingChannel` to a stream to store the media. There are two signaling modes that you can specify :

- If `StorageStatus` is enabled, the data will be stored in the `StreamARN` provided. In order for WebRTC Ingestion to work, the stream must have data retention enabled.
- If `StorageStatus` is disabled, no data will be stored, and the `StreamARN` parameter will not be needed.

> **Important**
>
> If `StorageStatus` is enabled, direct peer-to-peer (master-viewer) connections no longer occur. Peers connect directly to the storage session. You must call the `JoinStorageSession` API to trigger an SDP offer send and establish a connection between a peer and the storage session.

Request Syntax

```
POST /updateMediaStorageConfiguration HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "MediaStorageConfiguration": {
      "Status": "string",
      "StreamARN": "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelARN

The Amazon Resource Name (ARN) of the channel.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

MediaStorageConfiguration

A structure that encapsulates, or contains, the media storage configuration properties.

Type: MediaStorageConfiguration object

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NoDataRetentionException

The Stream data retention in hours is equal to zero.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateNotificationConfiguration

Service: Amazon Kinesis Video Streams

Updates the notification information for a stream.

Request Syntax

```
POST /updateNotificationConfiguration HTTP/1.1
Content-type: application/json

{
   "NotificationConfiguration": {
      "DestinationConfig": {
         "Uri": "string"
      },
      "Status": "string"
   },
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

NotificationConfiguration

The structure containing the information required for notifications. If the structure is null, the configuration will be deleted from the stream.

Type: NotificationConfiguration object

Required: No

StreamARN

The Amazon Resource Name (ARN) of the Kinesis video stream from where you want to update the notification configuration. You must specify either the `StreamName` or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream from which to update the notification configuration. You must specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NoDataRetentionException

The Stream data retention in hours is equal to zero.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateSignalingChannel

Service: Amazon Kinesis Video Streams

Updates the existing signaling channel. This is an asynchronous operation and takes time to complete.

If the `MessageTtlSeconds` value is updated (either increased or reduced), it only applies to new messages sent via this channel after it's been updated. Existing messages are still expired as per the previous `MessageTtlSeconds` value.

Request Syntax

```
POST /updateSignalingChannel HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "CurrentVersion": "string",
   "SingleMasterConfiguration": {
      "MessageTtlSeconds": number
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelARN

The Amazon Resource Name (ARN) of the signaling channel that you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

CurrentVersion

The current version of the signaling channel that you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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

Required: Yes

SingleMasterConfiguration

The structure containing the configuration for the `SINGLE_MASTER` type of the signaling channel that you want to update. This parameter and the channel message's time-to-live are required for channels with the `SINGLE_MASTER` channel type.

Type: SingleMasterConfiguration object

Required: No

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

VersionMismatchException

The stream version that you specified is not the latest version. To get the latest version, use the DescribeStream API.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateStream

Service: Amazon Kinesis Video Streams

Updates stream metadata, such as the device name and media type.

You must provide the stream name or the Amazon Resource Name (ARN) of the stream.

To make sure that you have the latest version of the stream before updating it, you can specify the stream version. Kinesis Video Streams assigns a version to each stream. When you update a stream, Kinesis Video Streams assigns a new version number. To get the latest stream version, use the `DescribeStream` API.

`UpdateStream` is an asynchronous operation, and takes time to complete.

Request Syntax

```
POST /updateStream HTTP/1.1
Content-type: application/json

{
   "CurrentVersion": "string",
   "DeviceName": "string",
   "MediaType": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

CurrentVersion

The version of the stream whose metadata you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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

Required: Yes

DeviceName

The name of the device that is writing to the stream.

> **Note**
>
> In the current implementation, Kinesis Video Streams does not use this name.

Type: String

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

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

Required: No

MediaType

The stream's media type. Use `MediaType` to specify the type of content that the stream contains to the consumers of the stream. For more information about media types, see Media Types. If you choose to specify the `MediaType`, see Naming Requirements.

To play video on the console, you must specify the correct video type. For example, if the video in the stream is H.264, specify `video/h264` as the `MediaType`.

Type: String

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

Pattern: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Required: No

StreamARN

The ARN of the stream whose metadata you want to update.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream whose metadata you want to update.

The stream name is an identifier for the stream, and must be unique for each account and region.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

VersionMismatchException

The stream version that you specified is not the latest version. To get the latest version, use the DescribeStream API.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateStreamStorageConfiguration

Service: Amazon Kinesis Video Streams

Updates the storage configuration for an existing Kinesis video stream.

This operation allows you to modify the storage tier settings for a stream, enabling you to optimize storage costs and performance based on your access patterns.

`UpdateStreamStorageConfiguration` is an asynchronous operation.

You must have permissions for the `KinesisVideo:UpdateStreamStorageConfiguration` action.

Request Syntax

```
POST /updateStreamStorageConfiguration HTTP/1.1
Content-type: application/json

{
   "CurrentVersion": "string",
   "StreamARN": "string",
   "StreamName": "string",
   "StreamStorageConfiguration": {
      "DefaultStorageTier": "string"
   }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

CurrentVersion

The version of the stream whose storage configuration you want to change. To get the version, call either the `DescribeStream` or the `ListStreams` API.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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

Required: Yes

StreamARN

The Amazon Resource Name (ARN) of the stream for which you want to update the storage configuration.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream for which you want to update the storage configuration.

Type: String

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

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

Required: No

StreamStorageConfiguration

The new storage configuration for the stream. This includes the default storage tier that determines how stream data is stored and accessed.

Different storage tiers offer varying levels of performance and cost optimization to match your specific use case requirements.

Type: StreamStorageConfiguration object

Required: Yes

Response Syntax

```
HTTP/1.1 200
```

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 Error Types.

AccessDeniedException

You do not have required permissions to perform this operation.

HTTP Status Code: 401

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

ResourceInUseException

When the input `StreamARN` or `ChannelARN` in `CLOUD_STORAGE_MODE` is already mapped to a different Kinesis Video Stream resource, or if the provided input `StreamARN` or `ChannelARN` is not in Active status, try one of the following :

1. The `DescribeMediaStorageConfiguration` API to determine what the stream given channel is mapped to.
2. The `DescribeMappedResourceConfiguration` API to determine the channel that the given stream is mapped to.
3. The `DescribeStream` or `DescribeSignalingChannel` API to determine the status of the resource.

HTTP Status Code: 400

ResourceNotFoundException

Amazon Kinesis Video Streams can't find the stream that you specified.

HTTP Status Code: 404

VersionMismatchException

The stream version that you specified is not the latest version. To get the latest version, use the DescribeStream API.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Amazon Kinesis Video Streams Media

The following actions are supported by Amazon Kinesis Video Streams Media:

- GetMedia
- PutMedia

GetMedia

Service: Amazon Kinesis Video Streams Media

Use this API to retrieve media content from a Kinesis video stream. In the request, you identify the stream name or stream Amazon Resource Name (ARN), and the starting chunk. Kinesis Video Streams then returns a stream of chunks in order by fragment number.

> **Note**
>
> You must first call the `GetDataEndpoint` API to get an endpoint. Then send the `GetMedia` requests to this endpoint using the --endpoint-url parameter.

When you put media data (fragments) on a stream, Kinesis Video Streams stores each incoming fragment and related metadata in what is called a "chunk." For more information, see PutMedia. The `GetMedia` API returns a stream of these chunks starting from the chunk that you specify in the request.

The `GetMedia` API is designed to operate as a streaming API over a long-running connection. It is not intended for use in a traditional RESTful manner, where a new HTTP connection is established and closed for each fragment. When you invoke the `GetMedia` API, Kinesis Video Streams delivers fragments continuously through a persistent long-running connection using HTTP chunked transfer encoding.

The following limits apply when using the `GetMedia` API:

- A client can call `GetMedia` up to five times per second per stream.
- Kinesis Video Streams sends media data at a rate of up to 25 megabytes per second (or 200 megabits per second) during a `GetMedia` session.

> **Note**
>
> Use `GetMedia` as a streaming long-running connection to retrieve multiple fragments in a single persistent connection. Don't use the `GetMedia` API in a traditional RESTful manner where you establish and close a new HTTP connection for each fragment. If you attempt more than three concurrent `GetMedia` connections, Kinesis Video Streams throttles the latest connections with a `ConnectionLimitExceededException` error.

Note

The `GetMedia` HTTP response status code will be returned immediately, but the reading of the HTTP response payload will timeout after 3 seconds if there are no ingested fragments available for playback.

Note

If an error is thrown after invoking a Kinesis Video Streams media API, in addition to the HTTP status code and the response body, it includes the following pieces of information:

- `x-amz-ErrorType` HTTP header – contains a more specific error type in addition to what the HTTP status code provides.
- `x-amz-RequestId` HTTP header – if you want to report an issue to AWS, the support team can better diagnose the problem if given the Request Id.

Both the HTTP status code and the ErrorType header can be utilized to make programmatic decisions about whether errors are retry-able and under what conditions, as well as provide information on what actions the client programmer might need to take in order to successfully try again.
For more information, see the **Errors** section at the bottom of this topic, as well as Common Errors.

Request Syntax

```
POST /getMedia HTTP/1.1
Content-type: application/json

{
   "StartSelector": {
      "AfterFragmentNumber": "string",
      "ContinuationToken": "string",
      "StartSelectorType": "string",
      "StartTimestamp": number
   },
   "StreamARN": "string",
```

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

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

StartSelector

Identifies the starting chunk to get from the specified stream.

Type: StartSelector object

Required: Yes

StreamARN

The ARN of the stream from where you want to get the media content. If you don't specify the `streamARN`, you must specify the `streamName`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The Kinesis video stream name from where you want to get the media content. If you don't specify the `streamName`, you must specify the `streamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Response Elements

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

The response returns the following HTTP headers.

ContentType

The content type of the requested media.

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

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

The response returns the following as the HTTP body.

Payload

The payload Kinesis Video Streams returns is a sequence of chunks from the specified stream. For more information about the chunks, see PutMedia. The chunks that Kinesis Video Streams returns in the `GetMedia` call also include the following additional Matroska (MKV) tags:

- AWS_KINESISVIDEO_CONTINUATION_TOKEN (UTF-8 string) - In the event your `GetMedia` call terminates, you can use this continuation token in your next request to get the next chunk where the last request terminated.
- AWS_KINESISVIDEO_MILLIS_BEHIND_NOW (UTF-8 string) - Client applications can use this tag value to determine how far behind the chunk returned in the response is from the latest chunk on the stream.
- AWS_KINESISVIDEO_FRAGMENT_NUMBER - Fragment number returned in the chunk.
- AWS_KINESISVIDEO_SERVER_TIMESTAMP - Server timestamp of the fragment.
- AWS_KINESISVIDEO_PRODUCER_TIMESTAMP - Producer timestamp of the fragment.

The following tags will be present if an error occurs:

- AWS_KINESISVIDEO_ERROR_CODE - String description of an error that caused GetMedia to stop.
- AWS_KINESISVIDEO_ERROR_ID: Integer code of the error.

The error codes are as follows:

- 3002 - Error writing to the stream
- 4000 - Requested fragment is not found
- 4500 - Access denied for the stream's KMS key
- 4501 - Stream's KMS key is disabled
- 4502 - Validation error on the stream's KMS key
- 4503 - KMS key specified in the stream is unavailable
- 4504 - Invalid usage of the KMS key specified in the stream
- 4505 - Invalid state of the KMS key specified in the stream
- 4506 - Unable to find the KMS key specified in the stream
- 5000 - Internal error

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

ConnectionLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client connections.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

InvalidEndpointException

Caller used wrong endpoint to write data to a stream. On receiving such an exception, the user must call `GetDataEndpoint` with `APIName` set to `PUT_MEDIA` and use the endpoint from response to invoke the next `PutMedia` call.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform an operation on the given stream, or the token has expired.

HTTP Status Code: 401

ResourceNotFoundException

Status Code: 404, The stream with the given name does not exist.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutMedia

Service: Amazon Kinesis Video Streams Media

Use this API to send media data to a Kinesis video stream.

> **Note**
>
> You must first call the `GetDataEndpoint` API to get an endpoint. Then send the `PutMedia` requests to this endpoint using the --endpoint-url parameter.

In the request, you use the HTTP headers to provide parameter information, for example, stream name, timestamp, and whether the timestamp value is absolute or relative to when the producer started recording. You use the request body to send the media data. Kinesis Video Streams supports only the Matroska (MKV) container format for sending media data using this API.

You have the following options for sending data using this API:

- Send media data in real time: For example, a security camera can send frames in real time as it generates them. This approach minimizes the latency between the video recording and data sent on the wire. This is referred to as a continuous producer. In this case, a consumer application can read the stream in real time or when needed.
- Send media data offline (in batches): For example, a body camera might record video for hours and store it on the device. Later, when you connect the camera to the docking port, the camera can start a `PutMedia` session to send data to a Kinesis video stream. In this scenario, latency is not an issue.

When using this API, note the following considerations:

- You must specify either `streamName` or `streamARN`, but not both.
- To be able to play the media on the console or via HLS, track 1 of each fragment should contain h.264 encoded video, the CodecID in the fragment metadata should be "V_MPEG/ISO/AVC", and the fragment metadata should include AVCC formatted h.264 codec private data. Optionally, track 2 of each fragment should contain AAC encoded audio, the CodecID in the fragment metadata should be "A_AAC", and the fragment metadata should include AAC codec private data.
- The `PutMedia` API is designed to operate as a streaming API over a long-running connection. It is not intended for use in a traditional RESTful manner, where a new HTTP connection is

established and closed for each fragment. When using the `PutMedia` API, use HTTP chunked transfer encoding to send fragments continuously over a persistent connection.

- For each fragment received in a `PutMedia` session, Kinesis Video Streams sends one or more acknowledgements. Potential client-side network considerations might cause you to not get all these acknowledgements as they are generated.

> **Note**
>
> Use `PutMedia` as a streaming long-running connection to send multiple fragments in a single persistent connection. If you attempt more than one concurrent `PutMedia` connection, Kinesis Video Streams throttles the latest connections with a `ConnectionLimitExceededException` error.

The following limits apply when using the `PutMedia` API:

- A client can call `PutMedia` up to five times per second per stream.
- A client can send up to five fragments per second per stream.
- Kinesis Video Streams reads media data at a rate of up to 12.5 MB/second, or 100 Mbps during a `PutMedia` session.

Note the following constraints. In these cases, Kinesis Video Streams sends the Error acknowledgement in the response.

- Fragments that have time codes spanning longer than the maximum allowed limit and that contain more than 50 MB of data are not allowed.
- Fragments containing more than three tracks are not allowed. Each frame in every fragment must have the same track number as one of the tracks defined in the fragment header. Additionally, every fragment must contain at least one frame for each track defined in the fragment header.
- Each fragment must contain at least one frame for each track defined in the fragment metadata.
- The earliest frame timestamp in a fragment must be after the latest frame timestamp in the previous fragment.
- An MKV stream containing more than one MKV segment or containing disallowed MKV elements (like `track*`) also results in the Error acknowledgement.

Kinesis Video Streams stores each incoming fragment and related metadata in what is called a "chunk." The fragment metadata includes the following:

- The MKV headers provided at the start of the `PutMedia` request
- The following Kinesis Video Streams-specific metadata for the fragment:
  - `server_timestamp` - Timestamp when Kinesis Video Streams started receiving the fragment.
  - `producer_timestamp` - Timestamp, when the producer started recording the fragment. Kinesis Video Streams uses three pieces of information received in the request to calculate this value.
    - The fragment timecode value received in the request body along with the fragment.
    - Two request headers: `producerStartTimestamp` (when the producer started recording) and `fragmentTimeCodeType` (whether the fragment timecode in the payload is absolute or relative).

    Kinesis Video Streams then computes the `producer_timestamp` for the fragment as follows:

    If `fragmentTimeCodeType` is relative, then

    `producer_timestamp` = `producerStartTimeStamp` + fragment timecode

    If `fragmentTimeCodeType` is absolute, then

    `producer_timestamp` = fragment timecode (converted to milliseconds)
- Unique fragment number assigned by Kinesis Video Streams.

> **Note**
>
> When you make the `GetMedia` request, Kinesis Video Streams returns a stream of these chunks. The client can process the metadata as needed.

> **Note**
>
> This operation is only available for the AWS SDK for Java. It is not supported in AWS SDKs for other languages.

Note

Kinesis Video Streams does not parse and validate the codec private data during ingestion and archival via the PutMedia API. KVS extracts and validates the necessary information from the codec private data for MPEG-TS and MP4 fragment packaging when consuming the stream via the HLS APIs.

Note

If an error is thrown after invoking a Kinesis Video Streams media API, in addition to the HTTP status code and the response body, it includes the following pieces of information:

- `x-amz-ErrorType` HTTP header – contains a more specific error type in addition to what the HTTP status code provides.
- `x-amz-RequestId` HTTP header – if you want to report an issue to AWS, the support team can better diagnose the problem if given the Request Id.

Both the HTTP status code and the ErrorType header can be utilized to make programmatic decisions about whether errors are retry-able and under what conditions, as well as provide information on what actions the client programmer might need to take in order to successfully try again.
For more information, see the **Errors** section at the bottom of this topic, as well as Common Errors.

Request Syntax

```
POST /putMedia HTTP/1.1
x-amzn-stream-name: StreamName
x-amzn-stream-arn: StreamARN
x-amzn-fragment-timecode-type: FragmentTimecodeType
x-amzn-producer-start-timestamp: ProducerStartTimestamp

Payload
```

URI Request Parameters

The request uses the following URI parameters.

FragmentTimecodeType

You pass this value as the `x-amzn-fragment-timecode-type` HTTP header.

Indicates whether timecodes in the fragments (payload, HTTP request body) are absolute or relative to `producerStartTimestamp`. Kinesis Video Streams uses this information to compute the `producer_timestamp` for the fragment received in the request, as described in the API overview.

Valid Values: `ABSOLUTE | RELATIVE`

Required: Yes

ProducerStartTimestamp

You pass this value as the `x-amzn-producer-start-timestamp` HTTP header.

This is the producer timestamp at which the producer started recording the media (not the timestamp of the specific fragments in the request).

StreamARN

You pass this value as the `x-amzn-stream-arn` HTTP header.

Amazon Resource Name (ARN) of the Kinesis video stream where you want to write the media content. If you don't specify the `streamARN`, you must specify the `streamName`.

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

StreamName

You pass this value as the `x-amzn-stream-name` HTTP header.

Name of the Kinesis video stream where you want to write the media content. If you don't specify the `streamName`, you must specify the `streamARN`.

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

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

Request Body

The request accepts the following binary data.

Payload

The media content to write to the Kinesis video stream. In the current implementation, Kinesis Video Streams supports only the Matroska (MKV) container format with a single MKV segment. A segment can contain one or more clusters.

> **Note**
>
> Each MKV cluster maps to a Kinesis video stream fragment. Whatever cluster duration you choose becomes the fragment duration.

Response Syntax

```
HTTP/1.1 200

Payload
```

Response Elements

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

The response returns the following as the HTTP body.

Payload

After Kinesis Video Streams successfully receives a `PutMedia` request, the service validates the request headers. The service then starts reading the payload and first sends an HTTP 200 response.

The service then returns a stream containing a series of JSON objects (`Acknowledgement` objects) separated by newlines. The acknowledgements are received on the same connection on which the media data is sent. There can be many acknowledgements for a `PutMedia` request. Each `Acknowledgement` consists of the following key-value pairs:

- `AckEventType` - Event type the acknowledgement represents.

- **Buffering:** Kinesis Video Streams has started receiving the fragment. Kinesis Video Streams sends the first Buffering acknowledgement when the first byte of fragment data is received.
- **Received:** Kinesis Video Streams received the entire fragment. If you did not configure the stream to persist the data, the producer can stop buffering the fragment upon receiving this acknowledgement.
- **Persisted:** Kinesis Video Streams has persisted the fragment (for example, to Amazon S3). You get this acknowledgement if you configured the stream to persist the data. After you receive this acknowledgement, the producer can stop buffering the fragment.
- **Error:** Kinesis Video Streams ran into an error while processing the fragment. You can review the error code and determine the next course of action.
- **Idle:** The `PutMedia` session is in-progress. However, Kinesis Video Streams is currently not receiving data. Kinesis Video Streams sends this acknowledgement periodically for up to 30 seconds after the last received data. If no data is received within the 30 seconds, Kinesis Video Streams closes the request.

  > **Note**
  >
  > This acknowledgement can help a producer determine if the `PutMedia` connection is alive, even if it is not sending any data.

- `FragmentTimecode` - Fragment timecode for which acknowledgement is sent.

  The element can be missing if the `AckEventType` is **Idle**.
- `FragmentNumber` - Kinesis Video Streams-generated fragment number for which the acknowledgement is sent.
- `ErrorId` and `ErrorCode` - If the `AckEventType` is `Error`, this field provides corresponding error code. The following is the list of error IDs and their corresponding error codes and error messages:
  - 4000 - STREAM_READ_ERROR - Error reading the data stream.
  - 4001 - MAX_FRAGMENT_SIZE_REACHED - Fragment size is greater than maximum limit, 50 MB, allowed.
  - 4002 - MAX_FRAGMENT_DURATION_REACHED - Fragment duration is greater than maximum allowed limit.

- 4003 - MAX_CONNECTION_DURATION_REACHED - Connection duration is greater than maximum allowed threshold.
- 4004 - FRAGMENT_TIMECODE_LESSER_THAN_PREVIOUS - Fragment timecode is less than the timecode previous time code (within a `PutMedia` call, you cannot send fragments out of order).
- 4005 - MORE_THAN_ALLOWED_TRACKS_FOUND - More than one track is found in MKV. (deprecated)
- 4006 - INVALID_MKV_DATA - Failed to parse the input stream as valid MKV format.
- 4007 - INVALID_PRODUCER_TIMESTAMP - Invalid producer timestamp.
- 4008 - STREAM_NOT_ACTIVE - Stream no longer exists (deleted).
- 4009 - FRAGMENT_METADATA_LIMIT_REACHED - Fragment metadata limit reached. See the Limits section of the developer guide.
- 4010 - TRACK_NUMBER_MISMATCH - The track number in an MKV frame did not match the tracks in the MKV header.
- 4011 - FRAMES_MISSING_FOR_TRACK - The fragment did not contain any frames for at least one of the tracks in the MKV header.
- 4012 - INVALID_FRAGMENT_METADATA - Fragment metadata name cannot begin with the string `AWS_`.
- 4500 - KMS_KEY_ACCESS_DENIED - Access to the stream's specified KMS key is denied.
- 4501 - KMS_KEY_DISABLED - The stream's specified KMS key is disabled.
- 4502 - KMS_KEY_VALIDATION_ERROR - The stream's specified KMS key failed validation.
- 4503 - KMS_KEY_UNAVAILABLE - The stream's specified KMS key is unavailable.
- 4504 - KMS_KEY_INVALID_USAGE - Invalid usage of the stream's specified KMS key.
- 4505 - KMS_KEY_INVALID_STATE - The stream's specified KMS key is in an invalid state.
- 4506 - KMS_KEY_NOT_FOUND - The stream's specified KMS key is not found.
- 5000 - INTERNAL_ERROR - Internal service error.
- 5001 - ARCHIVAL_ERROR - Kinesis Video Streams failed to persist fragments to the data store.

> **Note**
>
> The producer, while sending the payload for a long running `PutMedia` request, should read the response for acknowledgements. A producer might receive chunks of

acknowledgements at the same time, due to buffering on an intermediate proxy server. A producer that wants to receive timely acknowledgements can send fewer fragments in each `PutMedia` request.

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

ConnectionLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded the limit of allowed client connections.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

InvalidEndpointException

Caller used wrong endpoint to write data to a stream. On receiving such an exception, the user must call `GetDataEndpoint` with `APIName` set to `PUT_MEDIA` and use the endpoint from response to invoke the next `PutMedia` call.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform an operation on the given stream, or the token has expired.

HTTP Status Code: 401

ResourceNotFoundException

Status Code: 404, The stream with the given name does not exist.

HTTP Status Code: 404

Examples

Acknowledgement Format

The format of the acknowledgement is as follows:

```
{
       Acknowledgement : {
          "EventType": enum
          "FragmentTimecode": Long,
          "FragmentNumber": Long,
          "ErrorId" : String
      }
}
```

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Amazon Kinesis Video Streams Archived Media

The following actions are supported by Amazon Kinesis Video Streams Archived Media:

- GetClip
- GetDASHStreamingSessionURL
- GetHLSStreamingSessionURL
- GetImages
- GetMediaForFragmentList
- ListFragments

GetClip

Service: Amazon Kinesis Video Streams Archived Media

Downloads an MP4 file (clip) containing the archived, on-demand media from the specified video stream over the specified time range.

Both the StreamName and the StreamARN parameters are optional, but you must specify either the StreamName or the StreamARN when invoking this API operation.

> **Note**
>
> You must first call the `GetDataEndpoint` API to get an endpoint. Then send the `GetClip` requests to this endpoint using the --endpoint-url parameter.

An Amazon Kinesis video stream has the following requirements for providing data through MP4:

- Video playback track requirements.
- Data retention must be greater than 0.
- The video track of each fragment must contain codec private data in the Advanced Video Coding (AVC) for H.264 format and HEVC for H.265 format. For more information, see MPEG-4 specification ISO/IEC 14496-15. For information about adapting stream data to a given format, see NAL Adaptation Flags.
- The audio track (if present) of each fragment must contain codec private data in the AAC format (AAC specification ISO/IEC 13818-7) or the MS Wave format.

You can monitor the amount of outgoing data by monitoring the `GetClip.OutgoingBytes` Amazon CloudWatch metric. For information about using CloudWatch to monitor Kinesis Video Streams, see Monitoring Kinesis Video Streams. For pricing information, see Amazon Kinesis Video Streams Pricing and AWS Pricing. Charges for outgoing AWS data apply.

> **Important**
>
> The codec private data (CPD) contained in each fragment contains codec-specific initialization information, such as frame rate, resolution, and encoding profile, which are necessary to properly decode the fragment. CPD changes aren't supported between the

target fragments of the resulting clip. The CPD must remain consistent through the queried media, otherwise an error will be returned.

Important

Track changes aren't supported. Tracks must remain consistent throughout the queried media. An error is returned if the fragments in the stream change from having only video to having both audio and video, or if an AAC audio track is changed to an A-Law audio track.

Request Syntax

```
POST /getClip HTTP/1.1
Content-type: application/json

{
   "ClipFragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
         "EndTimestamp": number,
         "StartTimestamp": number
      }
   },
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ClipFragmentSelector

The time range of the requested clip and the source of the timestamps.

Type: ClipFragmentSelector object

Required: Yes

StreamARN

The Amazon Resource Name (ARN) of the stream for which to retrieve the media clip.

You must specify either the StreamName or the StreamARN.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream for which to retrieve the media clip.

You must specify either the StreamName or the StreamARN.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Response Elements

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

The response returns the following HTTP headers.

ContentType

The content type of the media in the requested clip.

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

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

The response returns the following as the HTTP body.

Payload

Traditional MP4 file that contains the media clip from the specified video stream. The output will contain the first 100 MB or the first 200 fragments from the specified start timestamp. For more information, see Kinesis Video Streams quotas.

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded a limit. Try making the call later. For information about limits, see Kinesis Video Streams quotas.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used.

HTTP Status Code: 400

InvalidCodecPrivateDataException

The codec private data in at least one of the tracks of the video stream is not valid for this operation.

HTTP Status Code: 400

InvalidMediaFrameException

One or more frames in the requested clip could not be parsed based on the specified codec.

HTTP Status Code: 400

MissingCodecPrivateDataException

No codec private data was found in at least one of tracks of the video stream.

HTTP Status Code: 400

NoDataRetentionException

`GetImages` was requested for a stream that does not retain data (that is, has a `DataRetentionInHours` of 0).

HTTP Status Code: 400

NotAuthorizedException

Status Code: 403, The caller is not authorized to perform an operation on the given stream, or the token has expired.

HTTP Status Code: 401

ResourceNotFoundException

`GetImages` will throw this error when Kinesis Video Streams can't find the stream that you specified.

`GetHLSStreamingSessionURL` and `GetDASHStreamingSessionURL` throw this error if a session with a `PlaybackMode` of `ON_DEMAND` or `LIVE_REPLAY` is requested for a stream that has no fragments within the requested time range, or if a session with a `PlaybackMode` of `LIVE` is requested for a stream that has no fragments within the last 30 seconds.

HTTP Status Code: 404

UnsupportedStreamMediaTypeException

The type of the media (for example, h.264 or h.265 video or AAC or G.711 audio) could not be determined from the codec IDs of the tracks in the first fragment for a playback session. The codec ID for track 1 should be `V_MPEG/ISO/AVC` and, optionally, the codec ID for track 2 should be `A_AAC`.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetDASHStreamingSessionURL

Service: Amazon Kinesis Video Streams Archived Media

Retrieves an MPEG Dynamic Adaptive Streaming over HTTP (DASH) URL for the stream. You can then open the URL in a media player to view the stream contents.

Both the `StreamName` and the `StreamARN` parameters are optional, but you must specify either the `StreamName` or the `StreamARN` when invoking this API operation.

An Amazon Kinesis video stream has the following requirements for providing data through MPEG-DASH:

- Video playback track requirements.
- Data retention must be greater than 0.
- The video track of each fragment must contain codec private data in the Advanced Video Coding (AVC) for H.264 format and HEVC for H.265 format. For more information, see MPEG-4 specification ISO/IEC 14496-15. For information about adapting stream data to a given format, see NAL Adaptation Flags.
- The audio track (if present) of each fragment must contain codec private data in the AAC format (AAC specification ISO/IEC 13818-7) or the MS Wave format.

The following procedure shows how to use MPEG-DASH with Kinesis Video Streams:

1. Call the `GetDataEndpoint` API to get an endpoint. Then send the `GetDASHStreamingSessionURL` requests to this endpoint using the --endpoint-url parameter.
2. Retrieve the MPEG-DASH URL using `GetDASHStreamingSessionURL`. Kinesis Video Streams creates an MPEG-DASH streaming session to be used for accessing content in a stream using the MPEG-DASH protocol. `GetDASHStreamingSessionURL` returns an authenticated URL (that includes an encrypted session token) for the session's MPEG-DASH *manifest* (the root resource needed for streaming with MPEG-DASH).

> **Note**
>
> Don't share or store this token where an unauthorized entity can access it. The token provides access to the content of the stream. Safeguard the token with the same measures that you use with your AWS credentials.

The media that is made available through the manifest consists only of the requested stream, time range, and format. No other media data (such as frames outside the requested window or alternate bitrates) is made available.

3. Provide the URL (containing the encrypted session token) for the MPEG-DASH manifest to a media player that supports the MPEG-DASH protocol. Kinesis Video Streams makes the initialization fragment and media fragments available through the manifest URL. The initialization fragment contains the codec private data for the stream, and other data needed to set up the video or audio decoder and renderer. The media fragments contain encoded video frames or encoded audio samples.
4. The media player receives the authenticated URL and requests stream metadata and media data normally. When the media player requests data, it calls the following actions:
   - **GetDASHManifest:** Retrieves an MPEG DASH manifest, which contains the metadata for the media that you want to playback.
   - **GetMP4InitFragment:** Retrieves the MP4 initialization fragment. The media player typically loads the initialization fragment before loading any media fragments. This fragment contains the "`fytp`" and "`moov`" MP4 atoms, and the child atoms that are needed to initialize the media player decoder.

     The initialization fragment does not correspond to a fragment in a Kinesis video stream. It contains only the codec private data for the stream and respective track, which the media player needs to decode the media frames.
   - **GetMP4MediaFragment:** Retrieves MP4 media fragments. These fragments contain the "`moof`" and "`mdat`" MP4 atoms and their child atoms, containing the encoded fragment's media frames and their timestamps.

     > **Important**
     >
     > The codec private data (CPD) contained in each fragment contains codec-specific initialization information, such as frame rate, resolution, and encoding profile, which are necessary to properly decode the fragment. CPD changes aren't supported during a streaming session. The CPD must remain consistent through the queried media.

> **Important**
>
> Track changes aren't supported. Tracks must remain consistent throughout the queried media. Streaming will fail if the fragments in the stream change from having only video to having both audio and video, or if an AAC audio track is changed to an A-Law audio track.

Data retrieved with this action is billable. See Pricing for details.

> **Note**
>
> For restrictions that apply to MPEG-DASH sessions, see Kinesis Video Streams quotas.

You can monitor the amount of data that the media player consumes by monitoring the `GetMP4MediaFragment.OutgoingBytes` Amazon CloudWatch metric. For information about using CloudWatch to monitor Kinesis Video Streams, see Monitoring Kinesis Video Streams. For pricing information, see Amazon Kinesis Video Streams Pricing and AWS Pricing. Charges for both HLS sessions and outgoing AWS data apply.

For more information about HLS, see HTTP Live Streaming on the Apple Developer site.

> **Important**
>
> If an error is thrown after invoking a Kinesis Video Streams archived media API, in addition to the HTTP status code and the response body, it includes the following pieces of information:
>
> - `x-amz-ErrorType` HTTP header – contains a more specific error type in addition to what the HTTP status code provides.
> - `x-amz-RequestId` HTTP header – if you want to report an issue to AWS the support team can better diagnose the problem if given the Request Id.
>
> Both the HTTP status code and the ErrorType header can be utilized to make programmatic decisions about whether errors are retry-able and under what conditions, as well as

provide information on what actions the client programmer might need to take in order to successfully try again.
For more information, see the **Errors** section at the bottom of this topic, as well as Common Errors.

Request Syntax

```
POST /getDASHStreamingSessionURL HTTP/1.1
Content-type: application/json

{
   "DASHFragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
         "EndTimestamp": number,
         "StartTimestamp": number
      }
   },
   "DisplayFragmentNumber": "string",
   "DisplayFragmentTimestamp": "string",
   "Expires": number,
   "MaxManifestFragmentResults": number,
   "PlaybackMode": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

DASHFragmentSelector

The time range of the requested fragment and the source of the timestamps.

This parameter is required if `PlaybackMode` is `ON_DEMAND` or `LIVE_REPLAY`. This parameter is optional if PlaybackMode is `LIVE`. If `PlaybackMode` is `LIVE`, the `FragmentSelectorType`

can be set, but the `TimestampRange` should not be set. If `PlaybackMode` is `ON_DEMAND` or `LIVE_REPLAY`, both `FragmentSelectorType` and `TimestampRange` must be set.

Type: DASHFragmentSelector object

Required: No

DisplayFragmentNumber

Fragments are identified in the manifest file based on their sequence number in the session. If DisplayFragmentNumber is set to `ALWAYS`, the Kinesis Video Streams fragment number is added to each S element in the manifest file with the attribute name “kvs:fn”. These fragment numbers can be used for logging or for use with other APIs (e.g. `GetMedia` and `GetMediaForFragmentList`). A custom MPEG-DASH media player is necessary to leverage these this custom attribute.

The default value is `NEVER`.

Type: String

Valid Values: `ALWAYS` | `NEVER`

Required: No

DisplayFragmentTimestamp

Per the MPEG-DASH specification, the wall-clock time of fragments in the manifest file can be derived using attributes in the manifest itself. However, typically, MPEG-DASH compatible media players do not properly handle gaps in the media timeline. Kinesis Video Streams adjusts the media timeline in the manifest file to enable playback of media with discontinuities. Therefore, the wall-clock time derived from the manifest file may be inaccurate. If DisplayFragmentTimestamp is set to `ALWAYS`, the accurate fragment timestamp is added to each S element in the manifest file with the attribute name “kvs:ts”. A custom MPEG-DASH media player is necessary to leverage this custom attribute.

The default value is `NEVER`. When DASHFragmentSelector is `SERVER_TIMESTAMP`, the timestamps will be the server start timestamps. Similarly, when DASHFragmentSelector is `PRODUCER_TIMESTAMP`, the timestamps will be the producer start timestamps.

Type: String

Valid Values: `ALWAYS` | `NEVER`

Required: No

Expires

The time in seconds until the requested session expires. This value can be between 300 (5 minutes) and 43200 (12 hours).

When a session expires, no new calls to `GetDashManifest`, `GetMP4InitFragment`, or `GetMP4MediaFragment` can be made for that session.

The default is 300 (5 minutes).

Type: Integer

Valid Range: Minimum value of 300. Maximum value of 43200.

Required: No

MaxManifestFragmentResults

The maximum number of fragments that are returned in the MPEG-DASH manifest.

When the `PlaybackMode` is `LIVE`, the most recent fragments are returned up to this value. When the `PlaybackMode` is `ON_DEMAND`, the oldest fragments are returned, up to this maximum number.

When there are a higher number of fragments available in a live MPEG-DASH manifest, video players often buffer content before starting playback. Increasing the buffer size increases the playback latency, but it decreases the likelihood that rebuffering will occur during playback. We recommend that a live MPEG-DASH manifest have a minimum of 3 fragments and a maximum of 10 fragments.

The default is 5 fragments if `PlaybackMode` is `LIVE` or `LIVE_REPLAY`, and 1,000 if `PlaybackMode` is `ON_DEMAND`.

The maximum value of 1,000 fragments corresponds to more than 16 minutes of video on streams with 1-second fragments, and more than 2 1/2 hours of video on streams with 10-second fragments.

Type: Long

Valid Range: Minimum value of 1. Maximum value of 5000.

Required: No

PlaybackMode

Whether to retrieve live, live replay, or archived, on-demand data.

Features of the three types of sessions include the following:

- **`LIVE`** : For sessions of this type, the MPEG-DASH manifest is continually updated with the latest fragments as they become available. We recommend that the media player retrieve a new manifest on a one-second interval. When this type of session is played in a media player, the user interface typically displays a "live" notification, with no scrubber control for choosing the position in the playback window to display.

  > **Note**
  >
  > In `LIVE` mode, the newest available fragments are included in an MPEG-DASH manifest, even if there is a gap between fragments (that is, if a fragment is missing). A gap like this might cause a media player to halt or cause a jump in playback. In this mode, fragments are not added to the MPEG-DASH manifest if they are older than the newest fragment in the playlist. If the missing fragment becomes available after a subsequent fragment is added to the manifest, the older fragment is not added, and the gap is not filled.

- **`LIVE_REPLAY`** : For sessions of this type, the MPEG-DASH manifest is updated similarly to how it is updated for `LIVE` mode except that it starts by including fragments from a given start time. Instead of fragments being added as they are ingested, fragments are added as the duration of the next fragment elapses. For example, if the fragments in the session are two seconds long, then a new fragment is added to the manifest every two seconds. This mode is useful to be able to start playback from when an event is detected and continue live streaming media that has not yet been ingested as of the time of the session creation. This mode is also useful to stream previously archived media without being limited by the 1,000 fragment limit in the `ON_DEMAND` mode.
- **`ON_DEMAND`** : For sessions of this type, the MPEG-DASH manifest contains all the fragments for the session, up to the number that is specified in `MaxManifestFragmentResults`. The manifest must be retrieved only once for each session. When this type of session is played in a media player, the user interface typically displays a scrubber control for choosing the position in the playback window to display.

In all playback modes, if `FragmentSelectorType` is `PRODUCER_TIMESTAMP`, and if there are multiple fragments with the same start timestamp, the fragment that has the larger fragment

number (that is, the newer fragment) is included in the MPEG-DASH manifest. The other fragments are not included. Fragments that have different timestamps but have overlapping durations are still included in the MPEG-DASH manifest. This can lead to unexpected behavior in the media player.

The default is `LIVE`.

Type: String

Valid Values: `LIVE | LIVE_REPLAY | ON_DEMAND`

Required: No

StreamARN

The Amazon Resource Name (ARN) of the stream for which to retrieve the MPEG-DASH manifest URL.

You must specify either the `StreamName` or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream for which to retrieve the MPEG-DASH manifest URL.

You must specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "DASHStreamingSessionURL": "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.

DASHStreamingSessionURL

The URL (containing the session token) that a media player can use to retrieve the MPEG-DASH manifest.

Type: String

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded a limit. Try making the call later. For information about limits, see Kinesis Video Streams quotas.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used.

HTTP Status Code: 400

InvalidCodecPrivateDataException

The codec private data in at least one of the tracks of the video stream is not valid for this operation.

HTTP Status Code: 400

MissingCodecPrivateDataException

No codec private data was found in at least one of tracks of the video stream.

HTTP Status Code: 400

NoDataRetentionException

`GetImages` was requested for a stream that does not retain data (that is, has a `DataRetentionInHours` of 0).

HTTP Status Code: 400

NotAuthorizedException

Status Code: 403, The caller is not authorized to perform an operation on the given stream, or the token has expired.

HTTP Status Code: 401

ResourceNotFoundException

`GetImages` will throw this error when Kinesis Video Streams can't find the stream that you specified.

`GetHLSStreamingSessionURL` and `GetDASHStreamingSessionURL` throw this error if a session with a `PlaybackMode` of `ON_DEMAND` or `LIVE_REPLAY` is requested for a stream that has no fragments within the requested time range, or if a session with a `PlaybackMode` of `LIVE` is requested for a stream that has no fragments within the last 30 seconds.

HTTP Status Code: 404

UnsupportedStreamMediaTypeException

The type of the media (for example, h.264 or h.265 video or AAC or G.711 audio) could not be determined from the codec IDs of the tracks in the first fragment for a playback session. The codec ID for track 1 should be `V_MPEG/ISO/AVC` and, optionally, the codec ID for track 2 should be `A_AAC`.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetHLSStreamingSessionURL

Service: Amazon Kinesis Video Streams Archived Media

Retrieves an HTTP Live Streaming (HLS) URL for the stream. You can then open the URL in a browser or media player to view the stream contents.

Both the `StreamName` and the `StreamARN` parameters are optional, but you must specify either the `StreamName` or the `StreamARN` when invoking this API operation.

An Amazon Kinesis video stream has the following requirements for providing data through HLS:

- Video playback track requirements.
- Data retention must be greater than 0.
- The video track of each fragment must contain codec private data in the Advanced Video Coding (AVC) for H.264 format or HEVC for H.265 format (MPEG-4 specification ISO/IEC 14496-15). For information about adapting stream data to a given format, see NAL Adaptation Flags.
- The audio track (if present) of each fragment must contain codec private data in the AAC format (AAC specification ISO/IEC 13818-7).

Kinesis Video Streams HLS sessions contain fragments in the fragmented MPEG-4 form (also called fMP4 or CMAF) or the MPEG-2 form (also called TS chunks, which the HLS specification also supports). For more information about HLS fragment types, see the HLS specification.

The following procedure shows how to use HLS with Kinesis Video Streams:

1. Call the `GetDataEndpoint` API to get an endpoint. Then send the `GetHLSStreamingSessionURL` requests to this endpoint using the --endpoint-url parameter.
2. Retrieve the HLS URL using `GetHLSStreamingSessionURL`. Kinesis Video Streams creates an HLS streaming session to be used for accessing content in a stream using the HLS protocol. `GetHLSStreamingSessionURL` returns an authenticated URL (that includes an encrypted session token) for the session's HLS *master playlist* (the root resource needed for streaming with HLS).

> **Note**
>
> Don't share or store this token where an unauthorized entity could access it. The token provides access to the content of the stream. Safeguard the token with the same measures that you would use with your AWS credentials.

The media that is made available through the playlist consists only of the requested stream, time range, and format. No other media data (such as frames outside the requested window or alternate bitrates) is made available.

3. Provide the URL (containing the encrypted session token) for the HLS master playlist to a media player that supports the HLS protocol. Kinesis Video Streams makes the HLS media playlist, initialization fragment, and media fragments available through the master playlist URL. The initialization fragment contains the codec private data for the stream, and other data needed to set up the video or audio decoder and renderer. The media fragments contain H.264-encoded video frames or AAC-encoded audio samples.
4. The media player receives the authenticated URL and requests stream metadata and media data normally. When the media player requests data, it calls the following actions:
   - **GetHLSMasterPlaylist:** Retrieves an HLS master playlist, which contains a URL for the `GetHLSMediaPlaylist` action for each track, and additional metadata for the media player, including estimated bitrate and resolution.
   - **GetHLSMediaPlaylist:** Retrieves an HLS media playlist, which contains a URL to access the MP4 initialization fragment with the `GetMP4InitFragment` action, and URLs to access the MP4 media fragments with the `GetMP4MediaFragment` actions. The HLS media playlist also contains metadata about the stream that the player needs to play it, such as whether the `PlaybackMode` is `LIVE` or `ON_DEMAND`. The HLS media playlist is typically static for sessions with a `PlaybackType` of `ON_DEMAND`. The HLS media playlist is continually updated with new fragments for sessions with a `PlaybackType` of `LIVE`. There is a distinct HLS media playlist for the video track and the audio track (if applicable) that contains MP4 media URLs for the specific track.
   - **GetMP4InitFragment:** Retrieves the MP4 initialization fragment. The media player typically loads the initialization fragment before loading any media fragments. This fragment contains the "`fytp`" and "`moov`" MP4 atoms, and the child atoms that are needed to initialize the media player decoder.

The initialization fragment does not correspond to a fragment in a Kinesis video stream. It contains only the codec private data for the stream and respective track, which the media player needs to decode the media frames.

- **GetMP4MediaFragment:** Retrieves MP4 media fragments. These fragments contain the "`moof`" and "`mdat`" MP4 atoms and their child atoms, containing the encoded fragment's media frames and their timestamps.

  > **Note**
  >
  > The codec private data (CPD) contained in each fragment contains codec-specific initialization information, such as frame rate, resolution, and encoding profile, which are necessary for properly decoding the fragment. For both TS and MP4, CPD changes are supported during a streaming session. Therefore, the fragments in a session can have a different information in the CPD without interrupting playback. For each streaming session, only 500 CPD changes are allowed.

  > **Important**
  >
  > Track changes aren't supported. Tracks must remain consistent throughout the queried media. Streaming will fail if the fragments in the stream change from having only video to having both audio and video, or if an AAC audio track is changed to an A-Law audio track.

  Data retrieved with this action is billable. For information, see Pricing.

- **GetTSFragment:** Retrieves MPEG TS fragments containing both initialization and media data for all tracks in the stream.

  > **Note**
  >
  > If the `ContainerFormat` is `MPEG_TS`, this API is used instead of `GetMP4InitFragment` and `GetMP4MediaFragment` to retrieve stream media.

Data retrieved with this action is billable. For more information, see Kinesis Video Streams pricing.

A streaming session URL must not be shared between players. The service might throttle a session if multiple media players are sharing it. For connection limits, see Kinesis Video Streams quotas.

You can monitor the amount of data that the media player consumes by monitoring the `GetMP4MediaFragment.OutgoingBytes` Amazon CloudWatch metric. For information about using CloudWatch to monitor Kinesis Video Streams, see Monitoring Kinesis Video Streams. For pricing information, see Amazon Kinesis Video Streams Pricing and AWS Pricing. Charges for both HLS sessions and outgoing AWS data apply.

See the video playback examples in the documentation guide: Use the CLI to retrieve an HLS streaming session URL and Use HLS in HTML and JavaScript.

For more information about HLS, see HTTP Live Streaming on the Apple Developer site.

Important

If an error is thrown after invoking a Kinesis Video Streams archived media API, in addition to the HTTP status code and the response body, it includes the following pieces of information:

- `x-amz-ErrorType` HTTP header – contains a more specific error type in addition to what the HTTP status code provides.
- `x-amz-RequestId` HTTP header – if you want to report an issue to AWS, the support team can better diagnose the problem if given the Request Id.

Both the HTTP status code and the ErrorType header can be utilized to make programmatic decisions about whether errors are retry-able and under what conditions, as well as provide information on what actions the client programmer might need to take in order to successfully try again.
For more information, see the **Errors** section at the bottom of this topic, as well as Common Errors.

Request Syntax

```
POST /getHLSStreamingSessionURL HTTP/1.1
Content-type: application/json

{
   "ContainerFormat": "string",
   "DiscontinuityMode": "string",
   "DisplayFragmentTimestamp": "string",
   "Expires": number,
   "HLSFragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
         "EndTimestamp": number,
         "StartTimestamp": number
      }
   },
   "MaxMediaPlaylistFragmentResults": number,
   "PlaybackMode": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ContainerFormat

Specifies which format should be used for packaging the media. Specifying the FRAGMENTED_MP4 container format packages the media into MP4 fragments (fMP4 or CMAF). This is the recommended packaging because there is minimal packaging overhead. The other container format option is MPEG_TS. HLS has supported MPEG TS chunks since it was released and is sometimes the only supported packaging on older HLS players. MPEG TS typically has a 5-25 percent packaging overhead. This means MPEG TS typically requires 5-25 percent more bandwidth and cost than fMP4.

The default is FRAGMENTED_MP4.

Type: String

Valid Values: `FRAGMENTED_MP4 | MPEG_TS`

Required: No

DiscontinuityMode

Specifies when flags marking discontinuities between fragments are added to the media playlists.

Media players typically build a timeline of media content to play, based on the timestamps of each fragment. This means that if there is any overlap or gap between fragments (as is typical if HLSFragmentSelector is set to `SERVER_TIMESTAMP`), the media player timeline will also have small gaps between fragments in some places, and will overwrite frames in other places. Gaps in the media player timeline can cause playback to stall and overlaps can cause playback to be jittery. When there are discontinuity flags between fragments, the media player is expected to reset the timeline, resulting in the next fragment being played immediately after the previous fragment.

The following modes are supported:

- `ALWAYS`: a discontinuity marker is placed between every fragment in the HLS media playlist. It is recommended to use a value of `ALWAYS` if the fragment timestamps are not accurate.
- `NEVER`: no discontinuity markers are placed anywhere. It is recommended to use a value of `NEVER` to ensure the media player timeline most accurately maps to the producer timestamps.
- `ON_DISCONTINUITY`: a discontinuity marker is placed between fragments that have a gap or overlap of more than 50 milliseconds. For most playback scenarios, it is recommended to use a value of `ON_DISCONTINUITY` so that the media player timeline is only reset when there is a significant issue with the media timeline (e.g. a missing fragment).

The default is `ALWAYS` when HLSFragmentSelector is set to `SERVER_TIMESTAMP`, and `NEVER` when it is set to `PRODUCER_TIMESTAMP`.

Type: String

Valid Values: `ALWAYS | NEVER | ON_DISCONTINUITY`

Required: No

DisplayFragmentTimestamp

Specifies when the fragment start timestamps should be included in the HLS media playlist. Typically, media players report the playhead position as a time relative to the start of the first fragment in the playback session. However, when the start timestamps are included in the HLS media playlist, some media players might report the current playhead as an absolute time based on the fragment timestamps. This can be useful for creating a playback experience that shows viewers the wall-clock time of the media.

The default is `NEVER`. When HLSFragmentSelector is `SERVER_TIMESTAMP`, the timestamps will be the server start timestamps. Similarly, when HLSFragmentSelector is `PRODUCER_TIMESTAMP`, the timestamps will be the producer start timestamps.

Type: String

Valid Values: `ALWAYS | NEVER`

Required: No

Expires

The time in seconds until the requested session expires. This value can be between 300 (5 minutes) and 43200 (12 hours).

When a session expires, no new calls to `GetHLSMasterPlaylist`, `GetHLSMediaPlaylist`, `GetMP4InitFragment`, `GetMP4MediaFragment`, or `GetTSFragment` can be made for that session.

The default is 300 (5 minutes).

Type: Integer

Valid Range: Minimum value of 300. Maximum value of 43200.

Required: No

HLSFragmentSelector

The time range of the requested fragment and the source of the timestamps.

This parameter is required if `PlaybackMode` is `ON_DEMAND` or `LIVE_REPLAY`. This parameter is optional if PlaybackMode is `LIVE`. If `PlaybackMode` is `LIVE`, the `FragmentSelectorType` can be set, but the `TimestampRange` should not be set. If `PlaybackMode` is `ON_DEMAND` or `LIVE_REPLAY`, both `FragmentSelectorType` and `TimestampRange` must be set.

Type: HLSFragmentSelector object

Required: No

MaxMediaPlaylistFragmentResults

The maximum number of fragments that are returned in the HLS media playlists.

When the `PlaybackMode` is `LIVE`, the most recent fragments are returned up to this value. When the `PlaybackMode` is `ON_DEMAND`, the oldest fragments are returned, up to this maximum number.

When there are a higher number of fragments available in a live HLS media playlist, video players often buffer content before starting playback. Increasing the buffer size increases the playback latency, but it decreases the likelihood that rebuffering will occur during playback. We recommend that a live HLS media playlist have a minimum of 3 fragments and a maximum of 10 fragments.

The default is 5 fragments if `PlaybackMode` is `LIVE` or `LIVE_REPLAY`, and 1,000 if `PlaybackMode` is `ON_DEMAND`.

The maximum value of 5,000 fragments corresponds to more than 80 minutes of video on streams with 1-second fragments, and more than 13 hours of video on streams with 10-second fragments.

Type: Long

Valid Range: Minimum value of 1. Maximum value of 5000.

Required: No

PlaybackMode

Whether to retrieve live, live replay, or archived, on-demand data.

Features of the three types of sessions include the following:

- **`LIVE`** : For sessions of this type, the HLS media playlist is continually updated with the latest fragments as they become available. We recommend that the media player retrieve a new playlist on a one-second interval. When this type of session is played in a media player, the user interface typically displays a "live" notification, with no scrubber control for choosing the position in the playback window to display.

> **Note**
>
> In LIVE mode, the newest available fragments are included in an HLS media playlist, even if there is a gap between fragments (that is, if a fragment is missing). A gap like this might cause a media player to halt or cause a jump in playback. In this mode, fragments are not added to the HLS media playlist if they are older than the newest fragment in the playlist. If the missing fragment becomes available after a subsequent fragment is added to the playlist, the older fragment is not added, and the gap is not filled.

- **LIVE_REPLAY** : For sessions of this type, the HLS media playlist is updated similarly to how it is updated for LIVE mode except that it starts by including fragments from a given start time. Instead of fragments being added as they are ingested, fragments are added as the duration of the next fragment elapses. For example, if the fragments in the session are two seconds long, then a new fragment is added to the media playlist every two seconds. This mode is useful to be able to start playback from when an event is detected and continue live streaming media that has not yet been ingested as of the time of the session creation. This mode is also useful to stream previously archived media without being limited by the 1,000 fragment limit in the ON_DEMAND mode.
- **ON_DEMAND** : For sessions of this type, the HLS media playlist contains all the fragments for the session, up to the number that is specified in MaxMediaPlaylistFragmentResults. The playlist must be retrieved only once for each session. When this type of session is played in a media player, the user interface typically displays a scrubber control for choosing the position in the playback window to display.

In all playback modes, if FragmentSelectorType is PRODUCER_TIMESTAMP, and if there are multiple fragments with the same start timestamp, the fragment that has the largest fragment number (that is, the newest fragment) is included in the HLS media playlist. The other fragments are not included. Fragments that have different timestamps but have overlapping durations are still included in the HLS media playlist. This can lead to unexpected behavior in the media player.

The default is LIVE.

Type: String

Valid Values: LIVE | LIVE_REPLAY | ON_DEMAND

Required: No

StreamARN

The Amazon Resource Name (ARN) of the stream for which to retrieve the HLS master playlist URL.

You must specify either the `StreamName` or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream for which to retrieve the HLS master playlist URL.

You must specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "HLSStreamingSessionURL": "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.

HLSStreamingSessionURL

The URL (containing the session token) that a media player can use to retrieve the HLS master playlist.

Type: String

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded a limit. Try making the call later. For information about limits, see Kinesis Video Streams quotas.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used.

HTTP Status Code: 400

InvalidCodecPrivateDataException

The codec private data in at least one of the tracks of the video stream is not valid for this operation.

HTTP Status Code: 400

MissingCodecPrivateDataException

No codec private data was found in at least one of tracks of the video stream.

HTTP Status Code: 400

NoDataRetentionException

`GetImages` was requested for a stream that does not retain data (that is, has a `DataRetentionInHours` of 0).

HTTP Status Code: 400

NotAuthorizedException

Status Code: 403, The caller is not authorized to perform an operation on the given stream, or the token has expired.

HTTP Status Code: 401

ResourceNotFoundException

`GetImages` will throw this error when Kinesis Video Streams can't find the stream that you specified.

`GetHLSStreamingSessionURL` and `GetDASHStreamingSessionURL` throw this error if a session with a `PlaybackMode` of `ON_DEMAND` or `LIVE_REPLAY` is requested for a stream that has no fragments within the requested time range, or if a session with a `PlaybackMode` of `LIVE` is requested for a stream that has no fragments within the last 30 seconds.

HTTP Status Code: 404

UnsupportedStreamMediaTypeException

The type of the media (for example, h.264 or h.265 video or AAC or G.711 audio) could not be determined from the codec IDs of the tracks in the first fragment for a playback session. The codec ID for track 1 should be `V_MPEG/ISO/AVC` and, optionally, the codec ID for track 2 should be `A_AAC`.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetImages

Service: Amazon Kinesis Video Streams Archived Media

Managed support for images provides a fully managed way to get images from the video data streamed and stored in Kinesis Video Streams. You can use images to run machine learning (ML) workloads such as person, pet, or vehicle detection. Images can also be used to add interactive elements to playback, such as image previews for motion events and scrubbing for a video clip.

GetImages also retrieves a list of images corresponding to each timestamp for a given time range, sampling interval, and image format configuration.

> **Note**
>
> You must first call the `GetDataEndpoint` API to get an endpoint. Then send the `GetImages` requests to this endpoint using the --endpoint-url parameter.

Video playback track requirements.

Request Syntax

```
POST /getImages HTTP/1.1
Content-type: application/json

{
   "EndTimestamp": number,
   "Format": "string",
   "FormatConfig": {
      "string" : "string"
   },
   "HeightPixels": number,
   "ImageSelectorType": "string",
   "MaxResults": number,
   "NextToken": "string",
   "SamplingInterval": number,
   "StartTimestamp": number,
   "StreamARN": "string",
   "StreamName": "string",
   "WidthPixels": number
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

EndTimestamp

The end timestamp for the range of images to be generated. If the time range between `StartTimestamp` and `EndTimestamp` is more than 300 seconds above `StartTimestamp`, you will receive an `IllegalArgumentException`.

Type: Timestamp

Required: Yes

Format

The format that will be used to encode the image.

Type: String

Valid Values: `JPEG | PNG`

Required: Yes

FormatConfig

The list of a key-value pair structure that contains extra parameters that can be applied when the image is generated. The `FormatConfig` key is the `JPEGQuality`, which indicates the JPEG quality key to be used to generate the image. The `FormatConfig` value accepts ints from 1 to 100. If the value is 1, the image will be generated with less quality and the best compression. If the value is 100, the image will be generated with the best quality and less compression. If no value is provided, the default value of the `JPEGQuality` key will be set to 80.

Type: String to string map

Map Entries: Maximum number of 1 item.

Valid Keys: `JPEGQuality`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^[a-zA-Z_0-9]+`

Required: No

HeightPixels

The height of the output image that is used in conjunction with the `WidthPixels` parameter. When both `HeightPixels` and `WidthPixels` parameters are provided, the image will be stretched to fit the specified aspect ratio. If only the `HeightPixels` parameter is provided, its original aspect ratio will be used to calculate the `WidthPixels` ratio. If neither parameter is provided, the original image size will be returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 2160.

Required: No

ImageSelectorType

The origin of the Server or Producer timestamps to use to generate the images.

Type: String

Valid Values: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Required: Yes

MaxResults

The maximum number of images to be returned by the API.

> **Note**
>
> The default limit is 25 images per API response. Providing a `MaxResults` greater than this value will result in a page size of 25. Any additional results will be paginated.

Type: Long

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

NextToken

A token that specifies where to start paginating the next set of Images. This is the `GetImages:NextToken` from a previously truncated response.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Pattern: `[a-zA-Z0-9+/]+={0,2}`

Required: No

SamplingInterval

The time interval in milliseconds (ms) at which the images need to be generated from the stream. The minimum value that can be provided is 200 ms (5 images per second). If the timestamp range is less than the sampling interval, the image from the `startTimestamp` will be returned if available.

Type: Integer

Required: No

StartTimestamp

The starting point from which the images should be generated. This `StartTimestamp` must be within an inclusive range of timestamps for an image to be returned.

Type: Timestamp

Required: Yes

StreamARN

The Amazon Resource Name (ARN) of the stream from which to retrieve the images. You must specify either the `StreamName` or the `StreamARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream from which to retrieve the images. You must specify either the `StreamName` or the `StreamARN`.

Type: String

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

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

Required: No

WidthPixels

The width of the output image that is used in conjunction with the `HeightPixels` parameter. When both `WidthPixels` and `HeightPixels` parameters are provided, the image will be stretched to fit the specified aspect ratio. If only the `WidthPixels` parameter is provided or if only the `HeightPixels` is provided, a `ValidationException` will be thrown. If neither parameter is provided, the original image size from the stream will be returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 3840.

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Images": [
      {
         "Error": "string",
         "ImageContent": "string",
         "TimeStamp": number
      }
   ],
   "NextToken": "string"
```

```
}
```

Response Elements

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

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

Images

The list of images generated from the video stream. If there is no media available for the given timestamp, the `NO_MEDIA` error will be listed in the output. If an error occurs while the image is being generated, the `MEDIA_ERROR` will be listed in the output as the cause of the missing image.

Type: Array of Image objects

NextToken

The encrypted token that was used in the request to get more images.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Pattern: `[a-zA-Z0-9+/]+={0,2}`

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded a limit. Try making the call later. For information about limits, see Kinesis Video Streams quotas.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used.

HTTP Status Code: 400

NoDataRetentionException

`GetImages` was requested for a stream that does not retain data (that is, has a `DataRetentionInHours` of 0).

HTTP Status Code: 400

NotAuthorizedException

Status Code: 403, The caller is not authorized to perform an operation on the given stream, or the token has expired.

HTTP Status Code: 401

ResourceNotFoundException

`GetImages` will throw this error when Kinesis Video Streams can't find the stream that you specified.

`GetHLSStreamingSessionURL` and `GetDASHStreamingSessionURL` throw this error if a session with a `PlaybackMode` of `ON_DEMAND` or `LIVE_REPLAY` is requested for a stream that has no fragments within the requested time range, or if a session with a `PlaybackMode` of `LIVE` is requested for a stream that has no fragments within the last 30 seconds.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

GetMediaForFragmentList

Service: Amazon Kinesis Video Streams Archived Media

Gets media for a list of fragments (specified by fragment number) from the archived data in an Amazon Kinesis video stream.

Note

You must first call the `GetDataEndpoint` API to get an endpoint. Then send the `GetMediaForFragmentList` requests to this endpoint using the --endpoint-url parameter.

For limits, see Kinesis Video Streams quotas.

Important

If an error is thrown after invoking a Kinesis Video Streams archived media API, in addition to the HTTP status code and the response body, it includes the following pieces of information:

- `x-amz-ErrorType` HTTP header – contains a more specific error type in addition to what the HTTP status code provides.
- `x-amz-RequestId` HTTP header – if you want to report an issue to AWS, the support team can better diagnose the problem if given the Request Id.

Both the HTTP status code and the ErrorType header can be utilized to make programmatic decisions about whether errors are retry-able and under what conditions, as well as provide information on what actions the client programmer might need to take in order to successfully try again.
For more information, see the **Errors** section at the bottom of this topic, as well as Common Errors.

Request Syntax

```
POST /getMediaForFragmentList HTTP/1.1
```

```
Content-type: application/json

{
   "Fragments": [ "string" ],
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

Fragments

A list of the numbers of fragments for which to retrieve media. You retrieve these values with ListFragments.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 1000 items.

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

Pattern: `^[0-9]+$`

Required: Yes

StreamARN

The Amazon Resource Name (ARN) of the stream from which to retrieve fragment media. Specify either this parameter or the `StreamName` parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream from which to retrieve fragment media. Specify either this parameter or the `StreamARN` parameter.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-Type: ContentType

Payload
```

Response Elements

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

The response returns the following HTTP headers.

ContentType

The content type of the requested media.

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

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

The response returns the following as the HTTP body.

Payload

The payload that Kinesis Video Streams returns is a sequence of chunks from the specified stream. For information about the chunks, see PutMedia. The chunks that Kinesis Video Streams

returns in the `GetMediaForFragmentList` call also include the following additional Matroska (MKV) tags:

- AWS_KINESISVIDEO_FRAGMENT_NUMBER - Fragment number returned in the chunk.
- AWS_KINESISVIDEO_SERVER_SIDE_TIMESTAMP - Server-side timestamp of the fragment.
- AWS_KINESISVIDEO_PRODUCER_SIDE_TIMESTAMP - Producer-side timestamp of the fragment.

The following tags will be included if an exception occurs:

- AWS_KINESISVIDEO_FRAGMENT_NUMBER - The number of the fragment that threw the exception.
- AWS_KINESISVIDEO_EXCEPTION_ERROR_CODE - The integer code of the error.
- AWS_KINESISVIDEO_EXCEPTION_MESSAGE - A text description of the exception.

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded a limit. Try making the call later. For information about limits, see Kinesis Video Streams quotas.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used.

HTTP Status Code: 400

NotAuthorizedException

Status Code: 403, The caller is not authorized to perform an operation on the given stream, or the token has expired.

HTTP Status Code: 401

ResourceNotFoundException

`GetImages` will throw this error when Kinesis Video Streams can't find the stream that you specified.

`GetHLSStreamingSessionURL` and `GetDASHStreamingSessionURL` throw this error if a session with a `PlaybackMode` of `ON_DEMAND` or `LIVE_REPLAY` is requested for a stream that has no fragments within the requested time range, or if a session with a `PlaybackMode` of `LIVE` is requested for a stream that has no fragments within the last 30 seconds.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListFragments

Service: Amazon Kinesis Video Streams Archived Media

Returns a list of Fragment objects from the specified stream and timestamp range within the archived data.

Listing fragments is eventually consistent. This means that even if the producer receives an acknowledgment that a fragment is persisted, the result might not be returned immediately from a request to `ListFragments`. However, results are typically available in less than one second.

Note

You must first call the `GetDataEndpoint` API to get an endpoint. Then send the `ListFragments` requests to this endpoint using the --endpoint-url parameter.

Important

If an error is thrown after invoking a Kinesis Video Streams archived media API, in addition to the HTTP status code and the response body, it includes the following pieces of information:

- `x-amz-ErrorType` HTTP header – contains a more specific error type in addition to what the HTTP status code provides.
- `x-amz-RequestId` HTTP header – if you want to report an issue to AWS, the support team can better diagnose the problem if given the Request Id.

Both the HTTP status code and the ErrorType header can be utilized to make programmatic decisions about whether errors are retry-able and under what conditions, as well as provide information on what actions the client programmer might need to take in order to successfully try again.
For more information, see the **Errors** section at the bottom of this topic, as well as Common Errors.

Request Syntax

```
POST /listFragments HTTP/1.1
```

```
Content-type: application/json

{
   "FragmentSelector": {
      "FragmentSelectorType": "string",
      "TimestampRange": {
         "EndTimestamp": number,
         "StartTimestamp": number
      }
   },
   "MaxResults": number,
   "NextToken": "string",
   "StreamARN": "string",
   "StreamName": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

FragmentSelector

Describes the timestamp range and timestamp origin for the range of fragments to return.

> **Note**
>
> This is only required when the `NextToken` isn't passed in the API.

Type: FragmentSelector object

Required: No

MaxResults

The total number of fragments to return. If the total number of fragments available is more than the value specified in `max-results`, then a ListFragments:NextToken is provided in the output that you can use to resume pagination.

The default value is 100.

Type: Long

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

NextToken

A token to specify where to start paginating. This is the ListFragments:NextToken from a previously truncated response.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Pattern: `[a-zA-Z0-9+/]+={0,2}`

Required: No

StreamARN

The Amazon Resource Name (ARN) of the stream from which to retrieve a fragment list. Specify either this parameter or the `StreamName` parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream from which to retrieve a fragment list. Specify either this parameter or the `StreamARN` parameter.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Fragments": [
      {
         "FragmentLengthInMilliseconds": number,
         "FragmentNumber": "string",
         "FragmentSizeInBytes": number,
         "ProducerTimestamp": number,
         "ServerTimestamp": number
      }
   ],
   "NextToken": "string"
}
```

Response Elements

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

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

Fragments

A list of archived Fragment objects from the stream that meet the selector criteria. Results are in no specific order, even across pages.

If there are no fragments in the stream that meet the selector criteria, an empty list is returned.

Type: Array of Fragment objects

NextToken

If the returned list is truncated, the operation returns this token to use to retrieve the next page of results. This value is `null` when there are no more results to return.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4096.

Pattern: `[a-zA-Z0-9+/]+={0,2}`

Errors

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

ClientLimitExceededException

Kinesis Video Streams has throttled the request because you have exceeded a limit. Try making the call later. For information about limits, see Kinesis Video Streams quotas.

HTTP Status Code: 400

InvalidArgumentException

A specified parameter exceeds its restrictions, is not supported, or can't be used.

HTTP Status Code: 400

NotAuthorizedException

Status Code: 403, The caller is not authorized to perform an operation on the given stream, or the token has expired.

HTTP Status Code: 401

ResourceNotFoundException

`GetImages` will throw this error when Kinesis Video Streams can't find the stream that you specified.

`GetHLSStreamingSessionURL` and `GetDASHStreamingSessionURL` throw this error if a session with a `PlaybackMode` of `ON_DEMAND` or `LIVE_REPLAY` is requested for a stream that has no fragments within the requested time range, or if a session with a `PlaybackMode` of `LIVE` is requested for a stream that has no fragments within the last 30 seconds.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Amazon Kinesis Video Signaling Channels

The following actions are supported by Amazon Kinesis Video Signaling Channels:

- GetIceServerConfig
- SendAlexaOfferToMaster

GetIceServerConfig

Service: Amazon Kinesis Video Signaling Channels

Note: Before using this API, you must call the `GetSignalingChannelEndpoint` API to request the HTTPS endpoint. You then specify the endpoint and region in your `GetIceServerConfig` API request.

Gets the Interactive Connectivity Establishment (ICE) server configuration information, including URIs, user name, and password which can be used to configure the WebRTC connection. The ICE component uses this configuration information to set up the WebRTC connection, including authenticating with the Traversal Using Relays around NAT (TURN) relay server.

TURN is a protocol that is used to improve the connectivity of peer-to-peer applications. By providing a cloud-based relay service, TURN ensures that a connection can be established even when one or more peers are incapable of a direct peer-to-peer connection. For more information, see A REST API For Access To TURN Services.

You can invoke this API to establish a fallback mechanism in case either of the peers is unable to establish a direct peer-to-peer connection over a signaling channel. You must specify the Amazon Resource Name (ARN) of your signaling channel in order to invoke this API.

Request Syntax

```
POST /v1/get-ice-server-config HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "ClientId": "string",
   "Service": "string",
   "Username": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelARN

The ARN of the signaling channel to be used for the peer-to-peer connection between configured peers.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

ClientId

Unique identifier for the viewer. Must be unique within the signaling channel.

Type: String

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

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

Required: No

Service

Specifies the desired service. Currently, TURN is the only valid value.

Type: String

Valid Values: TURN

Required: No

Username

An optional user ID to be associated with the credentials.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "IceServerList": [
      {
         "Password": "string",
         "Ttl": number,
         "Uris": [ "string" ],
         "Username": "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.

IceServerList

The list of ICE server information objects.

Type: Array of IceServer objects

Errors

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

ClientLimitExceededException

Your request was throttled because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

InvalidClientException

The specified client is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceNotFoundException

The specified resource is not found.

HTTP Status Code: 404

SessionExpiredException

If the client session is expired. Once the client is connected, the session is valid for 45 minutes. Client should reconnect to the channel to continue sending/receiving messages.

HTTP Status Code: 400

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

SendAlexaOfferToMaster

Service: Amazon Kinesis Video Signaling Channels

> **Note**
>
> Before using this API, you must call the `GetSignalingChannelEndpoint` API to get an endpoint. You then specify the endpoint and region in your `SendAlexaOfferToMaster` API request.

This API allows you to connect WebRTC-enabled devices with Alexa display devices. When invoked, it sends the Alexa Session Description Protocol (SDP) offer to the master peer. The offer is delivered as soon as the master is connected to the specified signaling channel. This API returns the SDP answer from the connected master. If the master is not connected to the signaling channel, redelivery requests are made until the message expires.

Request Syntax

```
POST /v1/send-alexa-offer-to-master HTTP/1.1
Content-type: application/json

{
   "ChannelARN": "string",
   "MessagePayload": "string",
   "SenderClientId": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

ChannelARN

The Amazon Resource Name (ARN) of the signaling channel by which Alexa and the master peer communicate.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: Yes

MessagePayload

The base64-encoded SDP offer content.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10000.

Pattern: `[a-zA-Z0-9+/=]+`

Required: Yes

SenderClientId

The unique identifier for the sender client.

Type: String

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

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

Required: Yes

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
   "Answer": "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.

Answer

The base64-encoded SDP answer content.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 10000.

Errors

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

ClientLimitExceededException

Your request was throttled because you have exceeded the limit of allowed client calls. Try making the call later.

HTTP Status Code: 400

InvalidArgumentException

The value for this input parameter is invalid.

HTTP Status Code: 400

NotAuthorizedException

The caller is not authorized to perform this operation.

HTTP Status Code: 401

ResourceNotFoundException

The specified resource is not found.

HTTP Status Code: 404

See Also

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

- AWS Command Line Interface V2
- AWS SDK for .NET V4
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

The following data types are supported by Amazon Kinesis Video Streams:

- ChannelInfo
- ChannelNameCondition
- DeletionConfig
- EdgeAgentStatus
- EdgeConfig
- ImageGenerationConfiguration
- ImageGenerationDestinationConfig
- LastRecorderStatus
- LastUploaderStatus
- ListEdgeAgentConfigurationsEdgeConfig
- LocalSizeConfig
- MappedResourceConfigurationListItem
- MediaSourceConfig
- MediaStorageConfiguration
- NotificationConfiguration
- NotificationDestinationConfig
- RecorderConfig
- ResourceEndpointListItem
- ScheduleConfig
- SingleMasterChannelEndpointConfiguration
- SingleMasterConfiguration
- StreamInfo
- StreamNameCondition
- StreamStorageConfiguration
- Tag
- UploaderConfig

The following data types are supported by Amazon Kinesis Video Streams Media:

- StartSelector

The following data types are supported by Amazon Kinesis Video Streams Archived Media:

- ClipFragmentSelector
- ClipTimestampRange
- DASHFragmentSelector
- DASHTimestampRange
- Fragment
- FragmentSelector
- HLSFragmentSelector
- HLSTimestampRange
- Image
- TimestampRange

The following data types are supported by Amazon Kinesis Video Signaling Channels:

- IceServer

Amazon Kinesis Video Streams

The following data types are supported by Amazon Kinesis Video Streams:

- ChannelInfo
- ChannelNameCondition
- DeletionConfig
- EdgeAgentStatus
- EdgeConfig
- ImageGenerationConfiguration
- ImageGenerationDestinationConfig
- LastRecorderStatus

- LastUploaderStatus
- ListEdgeAgentConfigurationsEdgeConfig
- LocalSizeConfig
- MappedResourceConfigurationListItem
- MediaSourceConfig
- MediaStorageConfiguration
- NotificationConfiguration
- NotificationDestinationConfig
- RecorderConfig
- ResourceEndpointListItem
- ScheduleConfig
- SingleMasterChannelEndpointConfiguration
- SingleMasterConfiguration
- StreamInfo
- StreamNameCondition
- StreamStorageConfiguration
- Tag
- UploaderConfig

ChannelInfo

Service: Amazon Kinesis Video Streams

A structure that encapsulates a signaling channel's metadata and properties.

Contents

ChannelARN

The Amazon Resource Name (ARN) of the signaling channel.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

ChannelName

The name of the signaling channel.

Type: String

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

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

Required: No

ChannelStatus

Current status of the signaling channel.

Type: String

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

Required: No

ChannelType

The type of the signaling channel.

Type: String

Valid Values: `SINGLE_MASTER | FULL_MESH`

Required: No

CreationTime

The time at which the signaling channel was created.

Type: Timestamp

Required: No

SingleMasterConfiguration

A structure that contains the configuration for the `SINGLE_MASTER` channel type.

Type: SingleMasterConfiguration object

Required: No

Version

The current version of the signaling channel.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ChannelNameCondition

Service: Amazon Kinesis Video Streams

An optional input parameter for the `ListSignalingChannels` API. When this parameter is specified while invoking `ListSignalingChannels`, the API returns only the channels that satisfy a condition specified in `ChannelNameCondition`.

Contents

ComparisonOperator

A comparison operator. Currently, you can only specify the `BEGINS_WITH` operator, which finds signaling channels whose names begin with a given prefix.

Type: String

Valid Values: `BEGINS_WITH`

Required: No

ComparisonValue

A value to compare.

Type: String

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

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

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeletionConfig

Service: Amazon Kinesis Video Streams

The configuration details required to delete the connection of the stream from the Edge Agent.

Contents

DeleteAfterUpload

The `boolean` value used to indicate whether or not you want to mark the media for deletion, once it has been uploaded to the Kinesis Video Stream cloud. The media files can be deleted if any of the deletion configuration values are set to `true`, such as when the limit for the `EdgeRetentionInHours`, or the `MaxLocalMediaSizeInMB`, has been reached.

Since the default value is set to `true`, configure the uploader schedule such that the media files are not being deleted before they are initially uploaded to the AWS cloud.

Type: Boolean

Required: No

EdgeRetentionInHours

The number of hours that you want to retain the data in the stream on the Edge Agent. The default value of the retention time is 720 hours, which translates to 30 days.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 720.

Required: No

LocalSizeConfig

The value of the local size required in order to delete the edge configuration.

Type: LocalSizeConfig object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EdgeAgentStatus

Service: Amazon Kinesis Video Streams

An object that contains the latest status details for an edge agent's recorder and uploader jobs. Use this information to determine the current health of an edge agent.

Contents

LastRecorderStatus

The latest status of a stream’s edge recording job.

Type: LastRecorderStatus object

Required: No

LastUploaderStatus

The latest status of a stream’s edge to cloud uploader job.

Type: LastUploaderStatus object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EdgeConfig

Service: Amazon Kinesis Video Streams

A description of the stream's edge configuration that will be used to sync with the Edge Agent IoT Greengrass component. The Edge Agent component will run on an IoT Hub Device setup at your premise.

Contents

HubDeviceArn

The "**Internet of Things (IoT) Thing**" Arn of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:iot:[a-z0-9-]+:[0-9]+:thing/[a-zA-Z0-9_.-]+`

Required: Yes

RecorderConfig

The recorder configuration consists of the local `MediaSourceConfig` details, that are used as credentials to access the local media files streamed on the camera.

Type: RecorderConfig object

Required: Yes

DeletionConfig

The deletion configuration is made up of the retention time (`EdgeRetentionInHours`) and local size configuration (`LocalSizeConfig`) details that are used to make the deletion.

Type: DeletionConfig object

Required: No

UploaderConfig

The uploader configuration contains the `ScheduleExpression` details that are used to schedule upload jobs for the recorded media files from the Edge Agent to a Kinesis Video Stream.

Type: UploaderConfig object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ImageGenerationConfiguration

Service: Amazon Kinesis Video Streams

The structure that contains the information required for the KVS images delivery. If null, the configuration will be deleted from the stream.

Contents

DestinationConfig

The structure that contains the information required to deliver images to a customer.

Type: ImageGenerationDestinationConfig object

Required: Yes

Format

The accepted image format.

Type: String

Valid Values: `JPEG | PNG`

Required: Yes

ImageSelectorType

The origin of the Server or Producer timestamps to use to generate the images.

Type: String

Valid Values: `SERVER_TIMESTAMP | PRODUCER_TIMESTAMP`

Required: Yes

SamplingInterval

The time interval in milliseconds (ms) at which the images need to be generated from the stream. The minimum value that can be provided is 200 ms. If the timestamp range is less than the sampling interval, the Image from the `StartTimestamp` will be returned if available.

Type: Integer

Required: Yes

Status

Indicates whether the `ContinuousImageGenerationConfigurations` API is enabled or disabled.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: Yes

FormatConfig

The list of a key-value pair structure that contains extra parameters that can be applied when the image is generated. The `FormatConfig` key is the `JPEGQuality`, which indicates the JPEG quality key to be used to generate the image. The `FormatConfig` value accepts ints from 1 to 100. If the value is 1, the image will be generated with less quality and the best compression. If the value is 100, the image will be generated with the best quality and less compression. If no value is provided, the default value of the `JPEGQuality` key will be set to 80.

Type: String to string map

Map Entries: Maximum number of 1 item.

Valid Keys: `JPEGQuality`

Value Length Constraints: Minimum length of 0. Maximum length of 256.

Value Pattern: `^[a-zA-Z_0-9]+`

Required: No

HeightPixels

The height of the output image that is used in conjunction with the `WidthPixels` parameter. When both `HeightPixels` and `WidthPixels` parameters are provided, the image will be stretched to fit the specified aspect ratio. If only the `HeightPixels` parameter is provided, its original aspect ratio will be used to calculate the `WidthPixels` ratio. If neither parameter is provided, the original image size will be returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 2160.

Required: No

WidthPixels

The width of the output image that is used in conjunction with the `HeightPixels` parameter. When both `WidthPixels` and `HeightPixels` parameters are provided, the image will be stretched to fit the specified aspect ratio. If only the `WidthPixels` parameter is provided, its original aspect ratio will be used to calculate the `HeightPixels` ratio. If neither parameter is provided, the original image size will be returned.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 3840.

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ImageGenerationDestinationConfig

Service: Amazon Kinesis Video Streams

The structure that contains the information required to deliver images to a customer.

Contents

DestinationRegion

The AWS Region of the S3 bucket where images will be delivered. This `DestinationRegion` must match the Region where the stream is located.

Type: String

Length Constraints: Minimum length of 9. Maximum length of 14.

Pattern: `^[a-z]+(-[a-z]+)?-[a-z]+-[0-9]$`

Required: Yes

Uri

The Uniform Resource Identifier (URI) that identifies where the images will be delivered.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[a-zA-Z_0-9]+:(//)?([^/]+)/?([^*]*)$`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LastRecorderStatus

Service: Amazon Kinesis Video Streams

The latest status of a stream's edge recording job.

Contents

JobStatusDetails

A description of a recorder job’s latest status.

Type: String

Required: No

LastCollectedTime

The timestamp at which the recorder job was last executed and media stored to local disk.

Type: Timestamp

Required: No

LastUpdatedTime

The timestamp at which the recorder status was last updated.

Type: Timestamp

Required: No

RecorderStatus

The status of the latest recorder job.

Type: String

Valid Values: `SUCCESS | USER_ERROR | SYSTEM_ERROR`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LastUploaderStatus

Service: Amazon Kinesis Video Streams

The latest status of a stream's edge to cloud uploader job.

Contents

JobStatusDetails

A description of an uploader job's latest status.

Type: String

Required: No

LastCollectedTime

The timestamp at which the uploader job was last executed and media collected to the cloud.

Type: Timestamp

Required: No

LastUpdatedTime

The timestamp at which the uploader status was last updated.

Type: Timestamp

Required: No

UploaderStatus

The status of the latest uploader job.

Type: String

Valid Values: `SUCCESS | USER_ERROR | SYSTEM_ERROR`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListEdgeAgentConfigurationsEdgeConfig

Service: Amazon Kinesis Video Streams

A description of a single stream's edge configuration.

Contents

CreationTime

The timestamp when the stream first created the edge config.

Type: Timestamp

Required: No

EdgeConfig

A description of the stream's edge configuration that will be used to sync with the Edge Agent IoT Greengrass component. The Edge Agent component will run on an IoT Hub Device setup at your premise.

Type: EdgeConfig object

Required: No

FailedStatusDetails

A description of the generated failure status.

Type: String

Required: No

LastUpdatedTime

The timestamp when the stream last updated the edge config.

Type: Timestamp

Required: No

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream.

Type: String

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

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

Required: No

SyncStatus

The current sync status of the stream's edge configuration.

Type: String

Valid Values: `SYNCING | ACKNOWLEDGED | IN_SYNC | SYNC_FAILED | DELETING | DELETE_FAILED | DELETING_ACKNOWLEDGED`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LocalSizeConfig

Service: Amazon Kinesis Video Streams

The configuration details that include the maximum size of the media (`MaxLocalMediaSizeInMB`) that you want to store for a stream on the Edge Agent, as well as the strategy that should be used (`StrategyOnFullSize`) when a stream's maximum size has been reached.

Contents

MaxLocalMediaSizeInMB

The overall maximum size of the media that you want to store for a stream on the Edge Agent.

Type: Integer

Valid Range: Minimum value of 64. Maximum value of 2000000.

Required: No

StrategyOnFullSize

The strategy to perform when a stream's `MaxLocalMediaSizeInMB` limit is reached.

Type: String

Valid Values: `DELETE_OLDEST_MEDIA | DENY_NEW_MEDIA`

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MappedResourceConfigurationListItem

Service: Amazon Kinesis Video Streams

A structure that encapsulates, or contains, the media storage configuration properties.

Contents

ARN

The Amazon Resource Name (ARN) of the Kinesis Video Stream resource, associated with the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

Type

The type of the associated resource for the kinesis video stream.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaSourceConfig

Service: Amazon Kinesis Video Streams

The configuration details that consist of the credentials required (`MediaUriSecretArn` and `MediaUriType`) to access the media files that are streamed to the camera.

Contents

MediaUriSecretArn

The AWS Secrets Manager ARN for the username and password of the camera, or a local media file location.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 2048.

Pattern: `arn:[a-z\d-]+:secretsmanager:[a-z0-9-]+:[0-9]+:secret:[a-zA-Z0-9_.-]+`

Required: Yes

MediaUriType

The Uniform Resource Identifier (URI) type. The `FILE_URI` value can be used to stream local media files.

> **Note**
>
> Preview only supports the `RTSP_URI` media source URI format .

Type: String

Valid Values: `RTSP_URI | FILE_URI`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

MediaStorageConfiguration

Service: Amazon Kinesis Video Streams

A structure that encapsulates, or contains, the media storage configuration properties.

- If `StorageStatus` is enabled, the data will be stored in the `StreamARN` provided. In order for WebRTC Ingestion to work, the stream must have data retention enabled.
- If `StorageStatus` is disabled, no data will be stored, and the `StreamARN` parameter will not be needed.

Contents

Status

The status of the media storage configuration.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: Yes

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

See Also

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

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

NotificationConfiguration

Service: Amazon Kinesis Video Streams

Use this API to configure Amazon Simple Notification Service (Amazon SNS) notifications for when fragments become available in a stream. If this parameter is null, the configuration will be deleted from the stream.

See Notifications in Kinesis Video Streams for more information.

Contents

DestinationConfig

The destination information required to deliver a notification to a customer.

Type: NotificationDestinationConfig object

Required: Yes

Status

Indicates if a notification configuration is enabled or disabled.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NotificationDestinationConfig

Service: Amazon Kinesis Video Streams

The structure that contains the information required to deliver a notification to a customer.

Contents

Uri

The Uniform Resource Identifier (URI) that identifies where the images will be delivered.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `^[a-zA-Z_0-9]+:(//)?([^/]+)/?([^*]*)$`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

RecorderConfig

Service: Amazon Kinesis Video Streams

The recorder configuration consists of the local `MediaSourceConfig` details that are used as credentials to access the local media files streamed on the camera.

Contents

MediaSourceConfig

The configuration details that consist of the credentials required (`MediaUriSecretArn` and `MediaUriType`) to access the media files streamed to the camera.

Type: MediaSourceConfig object

Required: Yes

ScheduleConfig

The configuration that consists of the `ScheduleExpression` and the `DurationInMinutes` details that specify the scheduling to record from a camera, or local media file, onto the Edge Agent. If the `ScheduleExpression` attribute is not provided, then the Edge Agent will always be set to recording mode.

Type: ScheduleConfig object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ResourceEndpointListItem

Service: Amazon Kinesis Video Streams

An object that describes the endpoint of the signaling channel returned by the `GetSignalingChannelEndpoint` API.

The media server endpoint will correspond to the `WEBRTC` Protocol.

Contents

Protocol

The protocol of the signaling channel returned by the `GetSignalingChannelEndpoint` API.

Type: String

Valid Values: `WSS | HTTPS | WEBRTC`

Required: No

ResourceEndpoint

The endpoint of the signaling channel returned by the `GetSignalingChannelEndpoint` API.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ScheduleConfig

Service: Amazon Kinesis Video Streams

This API enables you to specify the duration that the camera, or local media file, should record onto the Edge Agent. The `ScheduleConfig` consists of the `ScheduleExpression` and the `DurationInMinutes` attributes.

If the `ScheduleConfig` is not provided in the `RecorderConfig`, then the Edge Agent will always be set to recording mode.

If the `ScheduleConfig` is not provided in the `UploaderConfig`, then the Edge Agent will upload at regular intervals (every 1 hour).

Contents

DurationInSeconds

The total duration to record the media. If the `ScheduleExpression` attribute is provided, then the `DurationInSeconds` attribute should also be specified.

Type: Integer

Valid Range: Minimum value of 60. Maximum value of 3600.

Required: Yes

ScheduleExpression

The Quartz cron expression that takes care of scheduling jobs to record from the camera, or local media file, onto the Edge Agent. If the `ScheduleExpression` is not provided for the `RecorderConfig`, then the Edge Agent will always be set to recording mode.

For more information about Quartz, refer to the *Cron Trigger Tutorial* page to understand the valid expressions and its use.

Type: String

Length Constraints: Minimum length of 11. Maximum length of 100.

Pattern: `[^\n]{11,100}`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SingleMasterChannelEndpointConfiguration

Service: Amazon Kinesis Video Streams

An object that contains the endpoint configuration for the `SINGLE_MASTER` channel type.

Contents

Protocols

This property is used to determine the nature of communication over this `SINGLE_MASTER` signaling channel. If `WSS` is specified, this API returns a websocket endpoint. If `HTTPS` is specified, this API returns an `HTTPS` endpoint.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 5 items.

Valid Values: `WSS | HTTPS | WEBRTC`

Required: No

Role

This property is used to determine messaging permissions in this `SINGLE_MASTER` signaling channel. If `MASTER` is specified, this API returns an endpoint that a client can use to receive offers from and send answers to any of the viewers on this signaling channel. If `VIEWER` is specified, this API returns an endpoint that a client can use only to send offers to another `MASTER` client on this signaling channel.

Type: String

Valid Values: `MASTER | VIEWER`

Required: No

See Also

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

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

SingleMasterConfiguration

Service: Amazon Kinesis Video Streams

A structure that contains the configuration for the `SINGLE_MASTER` channel type.

Contents

MessageTtlSeconds

The period of time (in seconds) a signaling channel retains undelivered messages before they are discarded. Use UpdateSignalingChannel to update this value.

Type: Integer

Valid Range: Minimum value of 5. Maximum value of 120.

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamInfo

Service: Amazon Kinesis Video Streams

An object describing a Kinesis video stream.

Contents

CreationTime

A time stamp that indicates when the stream was created.

Type: Timestamp

Required: No

DataRetentionInHours

How long the stream retains data, in hours.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

DeviceName

The name of the device that is associated with the stream.

Type: String

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

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

Required: No

KmsKeyId

The ID of the AWS Key Management Service (AWS KMS) key that Kinesis Video Streams uses to encrypt data on the stream.

Type: String

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

Pattern: `.+`

Required: No

MediaType

The `MediaType` of the stream.

Type: String

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

Pattern: `[\w\-\.\+]+/[\w\-\.\+]+(,[\w\-\.\+]+/[\w\-\.\+]+)*`

Required: No

Status

The status of the stream.

Type: String

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

Required: No

StreamARN

The Amazon Resource Name (ARN) of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Pattern: `arn:[a-z\d-]+:kinesisvideo:[a-z0-9-]+:[0-9]+:[a-z]+/[a-zA-Z0-9_.-]+/[0-9]+`

Required: No

StreamName

The name of the stream.

Type: String

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

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

Required: No

Version

The version of the stream.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

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

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamNameCondition

Service: Amazon Kinesis Video Streams

Specifies the condition that streams must satisfy to be returned when you list streams (see the `ListStreams` API). A condition has a comparison operation and a value. Currently, you can specify only the `BEGINS_WITH` operator, which finds streams whose names start with a given prefix.

Contents

ComparisonOperator

A comparison operator. Currently, you can specify only the `BEGINS_WITH` operator, which finds streams whose names start with a given prefix.

Type: String

Valid Values: `BEGINS_WITH`

Required: No

ComparisonValue

A value to compare.

Type: String

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

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

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamStorageConfiguration

Service: Amazon Kinesis Video Streams

The configuration for stream storage, including the default storage tier for stream data. This configuration determines how stream data is stored and accessed, with different tiers offering varying levels of performance and cost optimization.

Contents

DefaultStorageTier

The default storage tier for the stream data. This setting determines the storage class used for stream data, affecting both performance characteristics and storage costs.

Available storage tiers:

- `HOT` - Optimized for frequent access with the lowest latency and highest performance. Ideal for real-time applications and frequently accessed data.
- `WARM` - Balanced performance and cost for moderately accessed data. Suitable for data that is accessed regularly but not continuously.

Type: String

Valid Values: `HOT` | `WARM`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Tag

Service: Amazon Kinesis Video Streams

A key and value pair that is associated with the specified signaling channel.

Contents

Key

The key of the tag that is associated with the specified signaling channel.

Type: String

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

Pattern: `^([\p{L}\p{Z}\p{N}_.:/=+\-@]*)$`

Required: Yes

Value

The value of the tag that is associated with the specified signaling channel.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Pattern: `[\p{L}\p{Z}\p{N}_.:/=+\-@]*`

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UploaderConfig

Service: Amazon Kinesis Video Streams

The configuration that consists of the `ScheduleExpression` and the `DurationInMinutes` details that specify the scheduling to record from a camera, or local media file, onto the Edge Agent. If the `ScheduleConfig` is not provided in the `UploaderConfig`, then the Edge Agent will upload at regular intervals (every 1 hour).

Contents

ScheduleConfig

The configuration that consists of the `ScheduleExpression` and the `DurationInMinutes` details that specify the scheduling to record from a camera, or local media file, onto the Edge Agent. If the `ScheduleConfig` is not provided in this `UploaderConfig`, then the Edge Agent will upload at regular intervals (every 1 hour).

Type: ScheduleConfig object

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon Kinesis Video Streams Media

The following data types are supported by Amazon Kinesis Video Streams Media:

- StartSelector

StartSelector

Service: Amazon Kinesis Video Streams Media

Identifies the chunk on the Kinesis video stream where you want the `GetMedia` API to start returning media data. You have the following options to identify the starting chunk:

- Choose the latest (or oldest) chunk.
- Identify a specific chunk. You can identify a specific chunk either by providing a fragment number or timestamp (server or producer).
- Each chunk's metadata includes a continuation token as a Matroska (MKV) tag (`AWS_KINESISVIDEO_CONTINUATION_TOKEN`). If your previous `GetMedia` request terminated, you can use this tag value in your next `GetMedia` request. The API then starts returning chunks starting where the last API ended.

Contents

StartSelectorType

Identifies the fragment on the Kinesis video stream where you want to start getting the data from.

- NOW - Start with the latest chunk on the stream.
- EARLIEST - Start with earliest available chunk on the stream.
- FRAGMENT_NUMBER - Start with the chunk after a specific fragment. You must also specify the `AfterFragmentNumber` parameter.
- PRODUCER_TIMESTAMP or SERVER_TIMESTAMP - Start with the chunk containing a fragment with the specified producer or server timestamp. You specify the timestamp by adding `StartTimestamp`.
- CONTINUATION_TOKEN - Read using the specified continuation token.

> **Note**
>
> If you choose the NOW, EARLIEST, or CONTINUATION_TOKEN as the `startSelectorType`, you don't provide any additional information in the `startSelector`.

Type: String

Valid Values: `FRAGMENT_NUMBER | SERVER_TIMESTAMP | PRODUCER_TIMESTAMP | NOW | EARLIEST | CONTINUATION_TOKEN`

Required: Yes

AfterFragmentNumber

Specifies the fragment number from where you want the `GetMedia` API to start returning the fragments.

Type: String

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

Pattern: `^[0-9]+$`

Required: No

ContinuationToken

Continuation token that Kinesis Video Streams returned in the previous `GetMedia` response. The `GetMedia` API then starts with the chunk identified by the continuation token.

Type: String

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

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

Required: No

StartTimestamp

A timestamp value. This value is required if you choose the PRODUCER_TIMESTAMP or the SERVER_TIMESTAMP as the `startSelectorType`. The `GetMedia` API then starts with the chunk containing the fragment that has the specified timestamp.

Type: Timestamp

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon Kinesis Video Streams Archived Media

The following data types are supported by Amazon Kinesis Video Streams Archived Media:

- ClipFragmentSelector
- ClipTimestampRange
- DASHFragmentSelector
- DASHTimestampRange
- Fragment
- FragmentSelector
- HLSFragmentSelector
- HLSTimestampRange
- Image
- TimestampRange

ClipFragmentSelector

Service: Amazon Kinesis Video Streams Archived Media

Describes the timestamp range and timestamp origin of a range of fragments.

Fragments that have duplicate producer timestamps are deduplicated. This means that if producers are producing a stream of fragments with producer timestamps that are approximately equal to the true clock time, the clip will contain all of the fragments within the requested timestamp range. If some fragments are ingested within the same time range and very different points in time, only the oldest ingested collection of fragments are returned.

Contents

FragmentSelectorType

The origin of the timestamps to use (Server or Producer).

Type: String

Valid Values: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Required: Yes

TimestampRange

The range of timestamps to return.

Type: ClipTimestampRange object

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ClipTimestampRange

Service: Amazon Kinesis Video Streams Archived Media

The range of timestamps for which to return fragments.

Contents

EndTimestamp

The end of the timestamp range for the requested media.

This value must be within 24 hours of the specified `StartTimestamp`, and it must be later than the `StartTimestamp` value. If `FragmentSelectorType` for the request is `SERVER_TIMESTAMP`, this value must be in the past.

This value is inclusive. The `EndTimestamp` is compared to the (starting) timestamp of the fragment. Fragments that start before the `EndTimestamp` value and continue past it are included in the session.

Type: Timestamp

Required: Yes

StartTimestamp

The starting timestamp in the range of timestamps for which to return fragments.

Only fragments that start exactly at or after `StartTimestamp` are included in the session. Fragments that start before `StartTimestamp` and continue past it aren't included in the session. If `FragmentSelectorType` is `SERVER_TIMESTAMP`, the `StartTimestamp` must be later than the stream head.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

DASHFragmentSelector

Service: Amazon Kinesis Video Streams Archived Media

Contains the range of timestamps for the requested media, and the source of the timestamps.

Contents

FragmentSelectorType

The source of the timestamps for the requested media.

When `FragmentSelectorType` is set to `PRODUCER_TIMESTAMP` and GetDASHStreamingSessionURL:PlaybackMode is `ON_DEMAND` or `LIVE_REPLAY`, the first fragment ingested with a producer timestamp within the specified FragmentSelector:TimestampRange is included in the media playlist. In addition, the fragments with producer timestamps within the `TimestampRange` ingested immediately following the first fragment (up to the GetDASHStreamingSessionURL:MaxManifestFragmentResults value) are included.

Fragments that have duplicate producer timestamps are deduplicated. This means that if producers are producing a stream of fragments with producer timestamps that are approximately equal to the true clock time, the MPEG-DASH manifest will contain all of the fragments within the requested timestamp range. If some fragments are ingested within the same time range and very different points in time, only the oldest ingested collection of fragments are returned.

When `FragmentSelectorType` is set to `PRODUCER_TIMESTAMP` and GetDASHStreamingSessionURL:PlaybackMode is `LIVE`, the producer timestamps are used in the MP4 fragments and for deduplication. But the most recently ingested fragments based on server timestamps are included in the MPEG-DASH manifest. This means that even if fragments ingested in the past have producer timestamps with values now, they are not included in the HLS media playlist.

The default is `SERVER_TIMESTAMP`.

Type: String

Valid Values: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Required: No

TimestampRange

The start and end of the timestamp range for the requested media.

This value should not be present if `PlaybackType` is `LIVE`.

Type: DASHTimestampRange object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DASHTimestampRange

Service: Amazon Kinesis Video Streams Archived Media

The start and end of the timestamp range for the requested media.

This value should not be present if `PlaybackType` is `LIVE`.

The values in `DASHimestampRange` are inclusive. Fragments that start exactly at or after the start time are included in the session. Fragments that start before the start time and continue past it are not included in the session.

Contents

EndTimestamp

The end of the timestamp range for the requested media. This value must be within 24 hours of the specified `StartTimestamp`, and it must be later than the `StartTimestamp` value.

If `FragmentSelectorType` for the request is `SERVER_TIMESTAMP`, this value must be in the past.

The `EndTimestamp` value is required for `ON_DEMAND` mode, but optional for `LIVE_REPLAY` mode. If the `EndTimestamp` is not set for `LIVE_REPLAY` mode then the session will continue to include newly ingested fragments until the session expires.

> **Note**
>
> This value is inclusive. The `EndTimestamp` is compared to the (starting) timestamp of the fragment. Fragments that start before the `EndTimestamp` value and continue past it are included in the session.

Type: Timestamp

Required: No

StartTimestamp

The start of the timestamp range for the requested media.

If the `DASHTimestampRange` value is specified, the `StartTimestamp` value is required.

Only fragments that start exactly at or after `StartTimestamp` are included in the session. Fragments that start before `StartTimestamp` and continue past it aren't included in the session. If `FragmentSelectorType` is `SERVER_TIMESTAMP`, the `StartTimestamp` must be later than the stream head.

Type: Timestamp

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Fragment

Service: Amazon Kinesis Video Streams Archived Media

Represents a segment of video or other time-delimited data.

Contents

FragmentLengthInMilliseconds

The playback duration or other time value associated with the fragment.

Type: Long

Required: No

FragmentNumber

The unique identifier of the fragment. This value monotonically increases based on the ingestion order.

Type: String

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

Pattern: `^[0-9]+$`

Required: No

FragmentSizeInBytes

The total fragment size, including information about the fragment and contained media data.

Type: Long

Required: No

ProducerTimestamp

The timestamp from the producer corresponding to the fragment, in milliseconds.

Type: Timestamp

Required: No

ServerTimestamp

The timestamp from the AWS server corresponding to the fragment, in milliseconds.

Type: Timestamp

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FragmentSelector

Service: Amazon Kinesis Video Streams Archived Media

Describes the timestamp range and timestamp origin of a range of fragments.

Only fragments with a start timestamp greater than or equal to the given start time and less than or equal to the end time are returned. For example, if a stream contains fragments with the following start timestamps:

- 00:00:00
- 00:00:02
- 00:00:04
- 00:00:06

A fragment selector range with a start time of 00:00:01 and end time of 00:00:04 would return the fragments with start times of 00:00:02 and 00:00:04.

Contents

FragmentSelectorType

The origin of the timestamps to use (Server or Producer).

Type: String

Valid Values: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Required: Yes

TimestampRange

The range of timestamps to return.

Type: TimestampRange object

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HLSFragmentSelector

Service: Amazon Kinesis Video Streams Archived Media

Contains the range of timestamps for the requested media, and the source of the timestamps.

Contents

FragmentSelectorType

The source of the timestamps for the requested media.

When `FragmentSelectorType` is set to `PRODUCER_TIMESTAMP` and GetHLSStreamingSessionURL:PlaybackMode is `ON_DEMAND` or `LIVE_REPLAY`, the first fragment ingested with a producer timestamp within the specified FragmentSelector:TimestampRange is included in the media playlist. In addition, the fragments with producer timestamps within the `TimestampRange` ingested immediately following the first fragment (up to the GetHLSStreamingSessionURL:MaxMediaPlaylistFragmentResults value) are included.

Fragments that have duplicate producer timestamps are deduplicated. This means that if producers are producing a stream of fragments with producer timestamps that are approximately equal to the true clock time, the HLS media playlists will contain all of the fragments within the requested timestamp range. If some fragments are ingested within the same time range and very different points in time, only the oldest ingested collection of fragments are returned.

When `FragmentSelectorType` is set to `PRODUCER_TIMESTAMP` and GetHLSStreamingSessionURL:PlaybackMode is `LIVE`, the producer timestamps are used in the MP4 fragments and for deduplication. But the most recently ingested fragments based on server timestamps are included in the HLS media playlist. This means that even if fragments ingested in the past have producer timestamps with values now, they are not included in the HLS media playlist.

The default is `SERVER_TIMESTAMP`.

Type: String

Valid Values: `PRODUCER_TIMESTAMP | SERVER_TIMESTAMP`

Required: No

TimestampRange

The start and end of the timestamp range for the requested media.

This value should not be present if `PlaybackType` is `LIVE`.

Type: HLSTimestampRange object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

HLSTimestampRange

Service: Amazon Kinesis Video Streams Archived Media

The start and end of the timestamp range for the requested media.

This value should not be present if `PlaybackType` is `LIVE`.

Contents

EndTimestamp

The end of the timestamp range for the requested media. This value must be within 24 hours of the specified `StartTimestamp`, and it must be later than the `StartTimestamp` value.

If `FragmentSelectorType` for the request is `SERVER_TIMESTAMP`, this value must be in the past.

The `EndTimestamp` value is required for `ON_DEMAND` mode, but optional for `LIVE_REPLAY` mode. If the `EndTimestamp` is not set for `LIVE_REPLAY` mode then the session will continue to include newly ingested fragments until the session expires.

> **Note**
>
> This value is inclusive. The `EndTimestamp` is compared to the (starting) timestamp of the fragment. Fragments that start before the `EndTimestamp` value and continue past it are included in the session.

Type: Timestamp

Required: No

StartTimestamp

The start of the timestamp range for the requested media.

If the `HLSTimestampRange` value is specified, the `StartTimestamp` value is required.

Only fragments that start exactly at or after `StartTimestamp` are included in the session. Fragments that start before `StartTimestamp` and continue past it aren't included in the session. If `FragmentSelectorType` is `SERVER_TIMESTAMP`, the `StartTimestamp` must be later than the stream head.

Type: Timestamp

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Image

Service: Amazon Kinesis Video Streams Archived Media

A structure that contains the `Timestamp`, `Error`, and `ImageContent`.

Contents

Error

The error message shown when the image for the provided timestamp was not extracted due to a non-tryable error. An error will be returned if:

- There is no media that exists for the specified `Timestamp`.
- The media for the specified time does not allow an image to be extracted. In this case the media is audio only, or the incorrect media has been ingested.

Type: String

Valid Values: `NO_MEDIA | MEDIA_ERROR`

Required: No

ImageContent

An attribute of the `Image` object that is Base64 encoded.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 6291456.

Required: No

TimeStamp

An attribute of the `Image` object that is used to extract an image from the video stream. This field is used to manage gaps on images or to better understand the pagination window.

Type: Timestamp

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TimestampRange

Service: Amazon Kinesis Video Streams Archived Media

The range of timestamps for which to return fragments.

Contents

EndTimestamp

The ending timestamp in the range of timestamps for which to return fragments.

Type: Timestamp

Required: Yes

StartTimestamp

The starting timestamp in the range of timestamps for which to return fragments.

Type: Timestamp

Required: Yes

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon Kinesis Video Signaling Channels

The following data types are supported by Amazon Kinesis Video Signaling Channels:

- IceServer

IceServer

Service: Amazon Kinesis Video Signaling Channels

A structure for the ICE server connection data.

Contents

Password

A password to login to the ICE server.

Type: String

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

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

Required: No

Ttl

The period of time, in seconds, during which the user name and password are valid.

Type: Integer

Valid Range: Fixed value of 300.

Required: No

Uris

An array of URIs, in the form specified in the I-D.petithuguenin-behave-turn-uris spec. These URIs provide the different addresses and/or protocols that can be used to reach the TURN server.

Type: Array of strings

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

Required: No

Username

A user name to login to the ICE server.

Type: String

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

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

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signing AWS API requests in the *IAM User Guide*.

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 Create a signed AWS API request in the *IAM User Guide*.

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 Elements of an AWS API request signature in the *IAM User Guide*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS STS, see AWS services that work with IAM in the *IAM User Guide*.

Condition: If you're using temporary security credentials from AWS STS, 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 Create a signed AWS API request in the *IAM User Guide*.

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 Error Types

This section lists common error types that this AWS service may return. Not all services return all error types listed here. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You don't have permission to perform this action. Verify that your IAM policy includes the required permissions.

HTTP Status Code: 403

ExpiredTokenException

The security token included in the request has expired. Request a new security token and try again.

HTTP Status Code: 403

IncompleteSignature

The request signature doesn't conform to AWS standards. Verify that you're using valid AWS credentials and that your request is properly formatted. If you're using an SDK, ensure it's up to date.

HTTP Status Code: 403

InternalFailure

The request can't be processed right now because of an internal server issue. Try again later. If the problem persists, contact AWS Support.

HTTP Status Code: 500

MalformedHttpRequestException

The request body can't be processed. This typically happens when the request body can't be decompressed using the specified content encoding algorithm. Verify that the content encoding header matches the compression format used.

HTTP Status Code: 400

NotAuthorized

You don't have permissions to perform this action. Verify that your IAM policy includes the required permissions.

HTTP Status Code: 401

OptInRequired

Your AWS account needs a subscription for this service. Verify that you've enabled the service in your account.

HTTP Status Code: 403

RequestAbortedException

The request was aborted before a response could be returned. This typically happens when the client closes the connection.

HTTP Status Code: 400

RequestEntityTooLargeException

The request entity is too large. Reduce the size of the request body and try again.

HTTP Status Code: 413

RequestTimeoutException

The request timed out. The server didn't receive the complete request within the expected time frame. Try again.

HTTP Status Code: 408

ServiceUnavailable

The service is temporarily unavailable. Try again later.

HTTP Status Code: 503

ThrottlingException

Your request rate is too high. The AWS SDKs automatically retry requests that receive this exception. Reduce the frequency of requests.

HTTP Status Code: 400

UnknownOperationException

The action or operation isn't recognized. Verify that the action name is spelled correctly and that it's supported by the API version you're using.

HTTP Status Code: 404

UnrecognizedClientException

The X.509 certificate or AWS access key ID you provided doesn't exist in our records. Verify that you're using valid credentials and that they haven't expired.

HTTP Status Code: 403

ValidationError

The input doesn't meet the required format or constraints. Check that all required parameters are included and that values are valid.

HTTP Status Code: 400