Package: paws.media.services (via r-universe)

September 4, 2024

Title 'Amazon Web Services' Media Services Services

Version 0.7.0

Description Interface to 'Amazon Web Services' media services services, including 'Kinesis' video stream capture and processing, format conversion, and more <https://aws.amazon.com/media-services/>.

License Apache License (>= 2.0)

URL https://github.com/paws-r/paws

BugReports https://github.com/paws-r/paws/issues

Imports paws.common (>= 0.5.4)

Suggests testthat

Encoding UTF-8

Roxygen list(markdown = TRUE, roclets = c(``rd", ``namespace", ``collate"))

RoxygenNote 7.1.1

Collate 'elastictranscoder_service.R' 'elastictranscoder_interfaces.R' 'elastictranscoder_operations.R' 'kinesisvideo_service.R' 'kinesisvideo_interfaces.R' 'kinesisvideo_operations.R' 'kinesisvideoarchivedmedia_service.R' 'kinesisvideoarchivedmedia_interfaces.R' 'kinesisvideoarchivedmedia_operations.R' 'kinesisvideomedia_service.R' 'kinesisvideomedia_interfaces.R' 'kinesisvideomedia_operations.R' 'mediaconnect_service.R' 'mediaconnect_interfaces.R' 'mediaconnect_operations.R' 'mediaconvert_service.R' 'mediaconvert_interfaces.R' 'mediaconvert_operations.R' 'medialive_service.R' 'medialive_interfaces.R' 'medialive_operations.R' 'mediapackage_service.R' 'mediapackage_interfaces.R' 'mediapackage_operations.R' 'mediastore_service.R' 'mediastore_interfaces.R' 'mediastore_operations.R' 'mediastoredata_service.R' 'mediastoredata_interfaces.R' 'mediastoredata_operations.R' 'mediatailor_service.R' 'mediatailor_interfaces.R' 'mediatailor_operations.R'

Repository https://paws-r.r-universe.dev

RemoteUrl https://github.com/paws-r/paws

RemoteRef HEAD

RemoteSha 8ab20b498262e0b343c0153b4f244483aff4987f

Contents

elastictranscoder 2
kinesisvideo 4
kinesisvideoarchivedmedia 5
kinesisvideomedia 6
mediaconnect 8
mediaconvert 9
medialive 11
mediapackage 13
mediastore 15
mediastoredata 16
mediatailor 18

Index **20**

`elastictranscoder` *Amazon Elastic Transcoder*

Description

AWS Elastic Transcoder Service

The AWS Elastic Transcoder Service.

Usage

```
elastictranscoder(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- elastictranscoder(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

cancel_job	The CancelJob operation cancels an unfinished job
create_job	When you create a job, Elastic Transcoder returns JSON data that includes the values that you
create_pipeline	The CreatePipeline operation creates a pipeline with settings that you specify
create_preset	The CreatePreset operation creates a preset with settings that you specify
delete_pipeline	The DeletePipeline operation removes a pipeline
delete_preset	The DeletePreset operation removes a preset that you've added in an AWS region
list_jobs_by_pipeline	The ListJobsByPipeline operation gets a list of the jobs currently in a pipeline
list_jobs_by_status	The ListJobsByStatus operation gets a list of jobs that have a specified status
list_pipelines	The ListPipelines operation gets a list of the pipelines associated with the current AWS accou
list_presets	The ListPresets operation gets a list of the default presets included with Elastic Transcoder an
read_job	The ReadJob operation returns detailed information about a job
read_pipeline	The ReadPipeline operation gets detailed information about a pipeline
read_preset	The ReadPreset operation gets detailed information about a preset
test_role	The TestRole operation tests the IAM role used to create the pipeline
update_pipeline	Use the UpdatePipeline operation to update settings for a pipeline
update_pipeline_notifications	With the UpdatePipelineNotifications operation, you can update Amazon Simple Notification
update_pipeline_status	The UpdatePipelineStatus operation pauses or reactivates a pipeline, so that the pipeline stops

Examples

```
## Not run:
svc <- elastictranscoder()
svc$cancel_job(
  Foo = 123
)

## End(Not run)
```

kinesisvideo *Amazon Kinesis Video Streams*

Description

Amazon Kinesis Video Streams

Usage

```
kinesisvideo(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- kinesisvideo(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

create_signaling_channel	Creates a signaling channel
create_stream	Creates a new Kinesis video stream
delete_signaling_channel	Deletes a specified signaling channel
delete_stream	Deletes a Kinesis video stream and the data contained in the stream
describe_signaling_channel	Returns the most current information about the signaling channel
describe_stream	Returns the most current information about the specified stream
get_data_endpoint	Gets an endpoint for a specified stream for either reading or writing

get_signaling_channel_endpoint	Provides an endpoint for the specified signaling channel to send and receive messages
list_signaling_channels	Returns an array of ChannelInfo objects
list_streams	Returns an array of StreamInfo objects
list_tags_for_resource	Returns a list of tags associated with the specified signaling channel
list_tags_for_stream	Returns a list of tags associated with the specified stream
tag_resource	Adds one or more tags to a signaling channel
tag_stream	Adds one or more tags to a stream
untag_resource	Removes one or more tags from a signaling channel
untag_stream	Removes one or more tags from a stream
update_data_retention	Increases or decreases the stream's data retention period by the value that you specify
update_signaling_channel	Updates the existing signaling channel
update_stream	Updates stream metadata, such as the device name and media type

Examples

```
## Not run:
svc <- kinesisvideo()
svc$create_signaling_channel(
  Foo = 123
)

## End(Not run)
```

`kinesisvideoarchivedmedia` *Amazon Kinesis Video Streams Archived Media*

Description

Amazon Kinesis Video Streams Archived Media

Usage

```
kinesisvideoarchivedmedia(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- kinesisvideoarchivedmedia(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

get_clip	Downloads an MP4 file (clip) containing the archived, on-demand media from the specifiec
get_dash_streaming_session_url	Retrieves an MPEG Dynamic Adaptive Streaming over HTTP (DASH) URL for the stream
get_hls_streaming_session_url	Retrieves an HTTP Live Streaming (HLS) URL for the stream
get_media_for_fragment_list	Gets media for a list of fragments (specified by fragment number) from the archived data in
list_fragments	Returns a list of Fragment objects from the specified stream and timestamp range within th

Examples

```
## Not run:
svc <- kinesisvideoarchivedmedia()
svc$get_clip(
  Foo = 123
)

## End(Not run)
```

---

kinesisvideomedia *Amazon Kinesis Video Streams Media*

Description

Amazon Kinesis Video Streams Media

Usage

```
kinesisvideomedia(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- kinesisvideomedia(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

get_media	Use this API to retrieve media content from a Kinesis video stream

Examples

```
## Not run:
svc <- kinesisvideomedia()
svc$get_media(
  Foo = 123
)

## End(Not run)
```

mediaconnect *AWS MediaConnect*

Description

API for AWS Elemental MediaConnect

Usage

```
mediaconnect(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- mediaconnect(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

add_flow_outputs	Adds outputs to an existing flow
add_flow_sources	Adds Sources to flow
add_flow_vpc_interfaces	Adds VPC interfaces to flow
create_flow	Creates a new flow
delete_flow	Deletes a flow
describe_flow	Displays the details of a flow
describe_offering	Displays the details of an offering

describe_reservation	Displays the details of a reservation
grant_flow_entitlements	Grants entitlements to an existing flow
list_entitlements	Displays a list of all entitlements that have been granted to this account
list_flows	Displays a list of flows that are associated with this account
list_offerings	Displays a list of all offerings that are available to this account in the current AWS Region
list_reservations	Displays a list of all reservations that have been purchased by this account in the current AWS R
list_tags_for_resource	List all tags on an AWS Elemental MediaConnect resource
purchase_offering	Submits a request to purchase an offering
remove_flow_output	Removes an output from an existing flow
remove_flow_source	Removes a source from an existing flow
remove_flow_vpc_interface	Removes a VPC Interface from an existing flow
revoke_flow_entitlement	Revokes an entitlement from a flow
start_flow	Starts a flow
stop_flow	Stops a flow
tag_resource	Associates the specified tags to a resource with the specified resourceArn
untag_resource	Deletes specified tags from a resource
update_flow	Updates flow
update_flow_entitlement	You can change an entitlement's description, subscribers, and encryption
update_flow_output	Updates an existing flow output
update_flow_source	Updates the source of a flow

Examples

```
## Not run:
svc <- mediaconnect()
svc$add_flow_outputs(
  Foo = 123
)

## End(Not run)
```

`mediaconvert` *AWS Elemental MediaConvert*

Description

AWS Elemental MediaConvert

Usage

```
mediaconvert(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- mediaconvert(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

associate_certificate	Associates an AWS Certificate Manager (ACM) Amazon Resource Name (ARN) with AWS Elemen
cancel_job	Permanently cancel a job
create_job	Create a new transcoding job
create_job_template	Create a new job template
create_preset	Create a new preset
create_queue	Create a new transcoding queue
delete_job_template	Permanently delete a job template you have created
delete_preset	Permanently delete a preset you have created
delete_queue	Permanently delete a queue you have created
describe_endpoints	Send an request with an empty body to the regional API endpoint to get your account API endpoint
disassociate_certificate	Removes an association between the Amazon Resource Name (ARN) of an AWS Certificate Manag
get_job	Retrieve the JSON for a specific completed transcoding job
get_job_template	Retrieve the JSON for a specific job template
get_preset	Retrieve the JSON for a specific preset
get_queue	Retrieve the JSON for a specific queue
list_jobs	Retrieve a JSON array of up to twenty of your most recently created jobs
list_job_templates	Retrieve a JSON array of up to twenty of your job templates
list_presets	Retrieve a JSON array of up to twenty of your presets
list_queues	Retrieve a JSON array of up to twenty of your queues
list_tags_for_resource	Retrieve the tags for a MediaConvert resource
tag_resource	Add tags to a MediaConvert queue, preset, or job template
untag_resource	Remove tags from a MediaConvert queue, preset, or job template
update_job_template	Modify one of your existing job templates
update_preset	Modify one of your existing presets

update_queue	Modify one of your existing queues

Examples

```
## Not run:
svc <- mediaconvert()
svc$associate_certificate(
  Foo = 123
)

## End(Not run)
```

---

medialive *AWS Elemental MediaLive*

Description

API for AWS Elemental MediaLive

Usage

```
medialive(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- medialive(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

accept_input_device_transfer	Accept an incoming input device transfer
batch_delete	Starts delete of resources
batch_start	Starts existing resources
batch_stop	Stops running resources
batch_update_schedule	Update a channel schedule
cancel_input_device_transfer	Cancel an input device transfer that you have requested
create_channel	Creates a new channel
create_input	Create an input
create_input_security_group	Creates a Input Security Group
create_multiplex	Create a new multiplex
create_multiplex_program	Create a new program in the multiplex
create_tags	Create tags for a resource
delete_channel	Starts deletion of channel
delete_input	Deletes the input end point
delete_input_security_group	Deletes an Input Security Group
delete_multiplex	Delete a multiplex
delete_multiplex_program	Delete a program from a multiplex
delete_reservation	Delete an expired reservation
delete_schedule	Delete all schedule actions on a channel
delete_tags	Removes tags for a resource
describe_channel	Gets details about a channel
describe_input	Produces details about an input
describe_input_device	Gets the details for the input device
describe_input_device_thumbnail	Get the latest thumbnail data for the input device
describe_input_security_group	Produces a summary of an Input Security Group
describe_multiplex	Gets details about a multiplex
describe_multiplex_program	Get the details for a program in a multiplex
describe_offering	Get details for an offering
describe_reservation	Get details for a reservation
describe_schedule	Get a channel schedule
list_channels	Produces list of channels that have been created
list_input_devices	List input devices
list_input_device_transfers	List input devices that are currently being transferred
list_inputs	Produces list of inputs that have been created
list_input_security_groups	Produces a list of Input Security Groups for an account
list_multiplexes	Retrieve a list of the existing multiplexes
list_multiplex_programs	List the programs that currently exist for a specific multiplex
list_offerings	List offerings available for purchase
list_reservations	List purchased reservations
list_tags_for_resource	Produces list of tags that have been created for a resource
purchase_offering	Purchase an offering and create a reservation

reject_input_device_transfer	Reject the transfer of the specified input device to your AWS account
start_channel	Starts an existing channel
start_multiplex	Start (run) the multiplex
stop_channel	Stops a running channel
stop_multiplex	Stops a running multiplex
transfer_input_device	Start an input device transfer to another AWS account
update_channel	Updates a channel
update_channel_class	Changes the class of the channel
update_input	Updates an input
update_input_device	Updates the parameters for the input device
update_input_security_group	Update an Input Security Group's Whilelists
update_multiplex	Updates a multiplex
update_multiplex_program	Update a program in a multiplex
update_reservation	Update reservation

Examples

```
## Not run:
svc <- medialive()
svc$accept_input_device_transfer(
  Foo = 123
)

## End(Not run)
```

`mediapackage` *AWS Elemental MediaPackage*

Description

AWS Elemental MediaPackage

Usage

```
mediapackage(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- mediapackage(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

configure_logs	Changes the Channel's properities to configure log subscription
create_channel	Creates a new Channel
create_harvest_job	Creates a new HarvestJob record
create_origin_endpoint	Creates a new OriginEndpoint record
delete_channel	Deletes an existing Channel
delete_origin_endpoint	Deletes an existing OriginEndpoint
describe_channel	Gets details about a Channel
describe_harvest_job	Gets details about an existing HarvestJob
describe_origin_endpoint	Gets details about an existing OriginEndpoint
list_channels	Returns a collection of Channels
list_harvest_jobs	Returns a collection of HarvestJob records
list_origin_endpoints	Returns a collection of OriginEndpoint records
list_tags_for_resource	List tags for resource
rotate_channel_credentials	Changes the Channel's first IngestEndpoint's username and password
rotate_ingest_endpoint_credentials	Rotate the IngestEndpoint's username and password, as specified by the IngestEndpoint'
tag_resource	Tag resource
untag_resource	Untag resource
update_channel	Updates an existing Channel
update_origin_endpoint	Updates an existing OriginEndpoint

Examples

```
## Not run:
svc <- mediapackage()
svc$configure_logs(
  Foo = 123
)
```

```
## End(Not run)
```

mediastore	*AWS Elemental MediaStore*

Description

An AWS Elemental MediaStore container is a namespace that holds folders and objects. You use a container endpoint to create, read, and delete objects.

Usage

```
mediastore(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- mediastore(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

create_container	Creates a storage container to hold objects
delete_container	Deletes the specified container
delete_container_policy	Deletes the access policy that is associated with the specified container

delete_cors_policy	Deletes the cross-origin resource sharing (CORS) configuration information that is set for the conta
delete_lifecycle_policy	Removes an object lifecycle policy from a container
delete_metric_policy	Deletes the metric policy that is associated with the specified container
describe_container	Retrieves the properties of the requested container
get_container_policy	Retrieves the access policy for the specified container
get_cors_policy	Returns the cross-origin resource sharing (CORS) configuration information that is set for the conta
get_lifecycle_policy	Retrieves the object lifecycle policy that is assigned to a container
get_metric_policy	Returns the metric policy for the specified container
list_containers	Lists the properties of all containers in AWS Elemental MediaStore
list_tags_for_resource	Returns a list of the tags assigned to the specified container
put_container_policy	Creates an access policy for the specified container to restrict the users and clients that can access it
put_cors_policy	Sets the cross-origin resource sharing (CORS) configuration on a container so that the container can
put_lifecycle_policy	Writes an object lifecycle policy to a container
put_metric_policy	The metric policy that you want to add to the container
start_access_logging	Starts access logging on the specified container
stop_access_logging	Stops access logging on the specified container
tag_resource	Adds tags to the specified AWS Elemental MediaStore container
untag_resource	Removes tags from the specified container

Examples

```
## Not run:
svc <- mediastore()
svc$create_container(
  Foo = 123
)

## End(Not run)
```

mediastoredata   *AWS Elemental MediaStore Data Plane*

Description

An AWS Elemental MediaStore asset is an object, similar to an object in the Amazon S3 service. Objects are the fundamental entities that are stored in AWS Elemental MediaStore.

Usage

```
mediastoredata(config = list())
```

Arguments

config	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- mediastoredata(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

delete_object	Deletes an object at the specified path
describe_object	Gets the headers for an object at the specified path
get_object	Downloads the object at the specified path
list_items	Provides a list of metadata entries about folders and objects in the specified folder
put_object	Uploads an object to the specified path

Examples

```
## Not run:
svc <- mediastoredata()
svc$delete_object(
  Foo = 123
)

## End(Not run)
```

mediatailor *AWS MediaTailor*

Description

Use the AWS Elemental MediaTailor SDK to configure scalable ad insertion for your live and VOD content. With AWS Elemental MediaTailor, you can serve targeted ads to viewers while maintaining broadcast quality in over-the-top (OTT) video applications. For information about using the service, including detailed information about the settings covered in this guide, see the AWS Elemental MediaTailor User Guide.

Through the SDK, you manage AWS Elemental MediaTailor configurations the same as you do through the console. For example, you specify ad insertion behavior and mapping information for the origin server and the ad decision server (ADS).

Usage

```
mediatailor(config = list())
```

Arguments

`config`	Optional configuration of credentials, endpoint, and/or region.

Value

A client for the service. You can call the service's operations using syntax like `svc$operation(...)`, where `svc` is the name you've assigned to the client. The available operations are listed in the Operations section.

Service syntax

```
svc <- mediatailor(
  config = list(
    credentials = list(
      creds = list(
        access_key_id = "string",
        secret_access_key = "string",
        session_token = "string"
      ),
      profile = "string"
    ),
    endpoint = "string",
    region = "string"
  )
)
```

Operations

delete_playback_configuration	Deletes the playback configuration for the specified name
get_playback_configuration	Returns the playback configuration for the specified name
list_playback_configurations	Returns a list of the playback configurations defined in AWS Elemental MediaTailor
list_tags_for_resource	Returns a list of the tags assigned to the specified playback configuration resource
put_playback_configuration	Adds a new playback configuration to AWS Elemental MediaTailor
tag_resource	Adds tags to the specified playback configuration resource
untag_resource	Removes tags from the specified playback configuration resource

Examples

```
## Not run:
svc <- mediatailor()
svc$delete_playback_configuration(
  Foo = 123
)

## End(Not run)
```

Index

accept_input_device_transfer, 12
add_flow_outputs, 8
add_flow_sources, 8
add_flow_vpc_interfaces, 8
associate_certificate, 10

batch_delete, 12
batch_start, 12
batch_stop, 12
batch_update_schedule, 12

cancel_input_device_transfer, 12
cancel_job, 3, 10
configure_logs, 14
create_channel, 12, 14
create_container, 15
create_flow, 8
create_harvest_job, 14
create_input, 12
create_input_security_group, 12
create_job, 3, 10
create_job_template, 10
create_multiplex, 12
create_multiplex_program, 12
create_origin_endpoint, 14
create_pipeline, 3
create_preset, 3, 10
create_queue, 10
create_signaling_channel, 4
create_stream, 4
create_tags, 12

delete_channel, 12, 14
delete_container, 15
delete_container_policy, 15
delete_cors_policy, 16
delete_flow, 8
delete_input, 12
delete_input_security_group, 12
delete_job_template, 10
delete_lifecycle_policy, 16
delete_metric_policy, 16
delete_multiplex, 12
delete_multiplex_program, 12
delete_object, 17
delete_origin_endpoint, 14
delete_pipeline, 3
delete_playback_configuration, 19
delete_preset, 3, 10
delete_queue, 10
delete_reservation, 12
delete_schedule, 12
delete_signaling_channel, 4
delete_stream, 4
delete_tags, 12
describe_channel, 12, 14
describe_container, 16
describe_endpoints, 10
describe_flow, 8
describe_harvest_job, 14
describe_input, 12
describe_input_device, 12
describe_input_device_thumbnail, 12
describe_input_security_group, 12
describe_multiplex, 12
describe_multiplex_program, 12
describe_object, 17
describe_offering, 8, 12
describe_origin_endpoint, 14
describe_reservation, 9, 12
describe_schedule, 12
describe_signaling_channel, 4
describe_stream, 4
disassociate_certificate, 10

elastictranscoder, 2

get_clip, 6
get_container_policy, 16
get_cors_policy, 16

get_dash_streaming_session_url, *6*
get_data_endpoint, *4*
get_hls_streaming_session_url, *6*
get_job, *10*
get_job_template, *10*
get_lifecycle_policy, *16*
get_media, *7*
get_media_for_fragment_list, *6*
get_metric_policy, *16*
get_object, *17*
get_playback_configuration, *19*
get_preset, *10*
get_queue, *10*
get_signaling_channel_endpoint, *5*
grant_flow_entitlements, *9*

kinesisvideo, 4
kinesisvideoarchivedmedia, 5
kinesisvideomedia, 6

list_channels, *12*, *14*
list_containers, *16*
list_entitlements, *9*
list_flows, *9*
list_fragments, *6*
list_harvest_jobs, *14*
list_input_device_transfers, *12*
list_input_devices, *12*
list_input_security_groups, *12*
list_inputs, *12*
list_items, *17*
list_job_templates, *10*
list_jobs, *10*
list_jobs_by_pipeline, *3*
list_jobs_by_status, *3*
list_multiplex_programs, *12*
list_multiplexes, *12*
list_offerings, *9*, *12*
list_origin_endpoints, *14*
list_pipelines, *3*
list_playback_configurations, *19*
list_presets, *3*, *10*
list_queues, *10*
list_reservations, *9*, *12*
list_signaling_channels, *5*
list_streams, *5*
list_tags_for_resource, *5*, *9*, *10*, *12*, *14*, *16*, *19*
list_tags_for_stream, *5*

mediaconnect, 8
mediaconvert, 9
medialive, 11
mediapackage, 13
mediastore, 15
mediastoredata, 16
mediatailor, 18

purchase_offering, *9*, *12*
put_container_policy, *16*
put_cors_policy, *16*
put_lifecycle_policy, *16*
put_metric_policy, *16*
put_object, *17*
put_playback_configuration, *19*

read_job, *3*
read_pipeline, *3*
read_preset, *3*
reject_input_device_transfer, *13*
remove_flow_output, *9*
remove_flow_source, *9*
remove_flow_vpc_interface, *9*
revoke_flow_entitlement, *9*
rotate_channel_credentials, *14*
rotate_ingest_endpoint_credentials, *14*

start_access_logging, *16*
start_channel, *13*
start_flow, *9*
start_multiplex, *13*
stop_access_logging, *16*
stop_channel, *13*
stop_flow, *9*
stop_multiplex, *13*

tag_resource, *5*, *9*, *10*, *14*, *16*, *19*
tag_stream, *5*
test_role, *3*
transfer_input_device, *13*

untag_resource, *5*, *9*, *10*, *14*, *16*, *19*
untag_stream, *5*
update_channel, *13*, *14*
update_channel_class, *13*
update_data_retention, *5*
update_flow, *9*
update_flow_entitlement, *9*
update_flow_output, *9*

update_flow_source, *9*
update_input, *13*
update_input_device, *13*
update_input_security_group, *13*
update_job_template, *10*
update_multiplex, *13*
update_multiplex_program, *13*
update_origin_endpoint, *14*
update_pipeline, *3*
update_pipeline_notifications, *3*
update_pipeline_status, *3*
update_preset, *10*
update_queue, *11*
update_reservation, *13*
update_signaling_channel, *5*
update_stream, *5*