

---

# SQUEEZE SERVER

REST API  
Version 2.5



# SQUEEZE SERVER<sup>2.5</sup>

## CONTENTS

Overview .....	3
Service Authentication.....	4
Common Response Status Codes .....	5
Jobs API .....	6
Create Job .....	6
Get Job .....	8
Get Job Collection .....	10
Get Job Status Collection.....	13
Get Job Status .....	15
Delete Job.....	17
Presets API.....	18
Create New Preset.....	18
Get Preset Data .....	20
Get Preset List .....	21
Get Preset Info.....	23
Update Preset Info.....	25
Get Preset Info by Workflow.....	27
Update Preset from Stream.....	29
Delete Preset.....	30
Watch Folders API.....	32
Configuring Watch Folders .....	32
Create Watch Folder.....	32
Get Watch Folders .....	35
Get Watch Folder.....	37
Update Watch Folder.....	39
Get Watch Folder Sources.....	42
Get Watch Folder Source .....	44
Create Watch Folder Source .....	46
Update Watch Folder Source .....	49
Watch Folder Destinations.....	52
Get Watch Folder Destination.....	53
Create Watch Folder Destination.....	54

Delete Watch Folder Destination .....	57
Watch Folder Presets .....	57
Get Watch Folder Presets.....	57
Get Watch Folder Preset .....	59
Create Watch Folder Preset.....	60
Update Watch Folder Preset .....	62
Delete Watch Folder .....	64
Common Data Elements .....	66
StatusInfo.....	66
JobMediaInfo .....	67
CompressionPreset .....	68
DestinationInfo .....	68
DestinationMetadataInfo.....	70
SourceMedia.....	71
MetadataInfo .....	73
JobStatusInfo.....	73
Appendix .....	75
Defining YouTube Destinations .....	75

## Overview

Squeeze Server provides a many REST web service endpoints for management, configuration, and monitoring. All REST services are hosted at the website root of your server. From the local machine where you have Squeeze Server installed, this would be <http://localhost/api/>. This endpoint will display a brief help page with links to individual service help pages.

- **Jobs API:** APIs for working with Jobs, Job Archives, and status
  - Service endpoint: </api/jobs/>
  - Help endpoint: </api/jobs/help>
- **Presets API:** APIs for working with presets and the preset library (Presets stored in the Squeeze Server database).
  - Service endpoint: </api/presets/>
  - Help endpoint: </api/presets/help>
- **Watch Folders API:** APIs for working with watch folders
  - Service endpoint: </api/watchfolders/>
  - Help endpoint: </api/watchfolders/help>

## Service Authentication

If authentication is enabled in the service `web.config` file, each service request must include an authentication information or must be from a trusted IP address if IP Address bypass is enabled. A client may authenticate using any of the following mechanisms

- Query String: Include `uid=username&pwd=password` on the request query string
  - Example cURL call:

```
curl -I -X GET 'http://<host>/api/presets/info/?uid=<userid>&pwd=<password>'
```
- Basic Authentication: See [http://en.wikipedia.org/wiki/Basic\\_access\\_authentication](http://en.wikipedia.org/wiki/Basic_access_authentication)
- Token Authentication: Pass a token returned by any successful authentication in the `x-squeeze-auth-token` header.
- Alternate basic authentication: If you do not wish to use the default basic authentication, the same syntax as for basic authentication can be passed in an `x-Authorization` header. For example: `x-Authorization Basic QWxhZGRpbjpvcmVudlHNlc2FtZQ==`

Successful authentication returns a token that can be used for subsequent requests by including the token in the `x-squeeze-auth-token` header. The token expires after 12 hours by default. You can change the token expiration in the service `web.config` file.

When authentication is enabled, all authentications are role-based. Certain operations can be performed only by members of the appropriate role. Attempting to perform an operation that is not allowed for the authenticated role will return a `401 Unauthorized`.

## Common Response Status Codes

The API returns a variety of HTTP status codes such as **200 OK** or an error code that describe the nature of the error. Error responses also include a string in the response body with a message describing the error. There are cases such as a badly formed payload that may return status codes like **400 Bad Request** or **500 Server Error**, which have been produced outside of the Squeeze Server API handlers. In these cases, an HTML webpage describing the error may be returned.

Status Code	Status	Description
200	OK	The request was accepted and completed successfully
304	Not Modified	The resource was not changed as a result of the request
400	Bad Request	A required parameter was not specified or your request was otherwise improperly formed. See the body of the error response for more details. (For additional information on required parameters, see the documentation for each API.)
401	Unauthorized	The authentication credentials are not valid or expired
403	Forbidden	Insufficient rights to access the resource
404	Not Found	The requested resource does not exist
409	Conflict	An attempt was made to update a resource that would cause it to conflict with another resource.
500	Server Error	The request was successfully understood, but it could not be processed due to a server error. Please try your request again later.

## Jobs API

The REST services for jobs allow you to create, update, and delete jobs, with some restrictions on making changes to jobs that are finished or currently being processed. The "status" calls allow progress to be tracked without getting all of the details associated with a job.

The current implementation sends finished jobs to an archive, where you can view completed jobs but no longer edit them. When a job is completed, whether it succeeds or fails, you can use the same Job ID to find it in the archive. Time values may appear as a default time if they have not been set by the service.

You should always use the queue and paging query parameters be used when getting collections of jobs or job status information to manage the size of a result set.

The job services require authentication. Not all elements are publicly settable and will be ignored if included in the payload rather than returning a **401 Unauthorized** or **403 Forbidden** error message.

## Create Job

Create Job creates a complete job in the desired queue. Create Job returns the ID of the newly created job.

---

<b>Base URI</b>	<a href="/api/jobs/">/api/jobs/</a>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	POST
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/jobs/help/operations/CreateJob">/api/jobs/help/operations/CreateJob</a>

---

## Common Parameters

None

## Request Data Fields

---

<b>Name</b>	<b>Data Type</b>	<b>Required</b>	<b>Description</b>
<a href="#">JobCreateInfo</a>	Element	Yes	Container for the job to create (XML)
<a href="#">InPoint</a>	Float	No	For future use: Time within the source in seconds for encoding to start
<a href="#">Name</a>	String	Recommended	User-friendly name for the job

---

<b>OutPoint</b>	Float	No	For future use: Time within the source in seconds for encoding to end
<b>QueueId</b>	Guid	Yes	The ID of the job queue for this job
<b>Status</b>	Enumeration	Yes	Jobs can be created with a status of Waiting or Hold. If the status is Hold, the job will not be processed until the status is changed to Waiting.
<b>ThumbTime</b>	Float	No	Optional time in seconds within the source to generate the thumbnail. By default, Squeeze will generate a "good" thumbnail within the first few seconds of the source.
<b>WatchFolderName</b>	String	Private	Watch folder name is not user settable. It is only available when the server creates jobs from watch folders
<b>JobMediaInfo</b>	Element	Yes	Container for the job media related objects compression presets, source, and destinations. A valid job must have at least one preset and at least one destination. See <a href="#">JobMediaInfo</a>

## Response Data Fields

Name	Data Type	Required	Description
<b>JobIdInfo</b>	Element	Yes	Element wraps the job ID (XML)
<b>JobId</b>	Guid	Yes	Unique ID for the newly created job

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-type: application/xml" --data "
```

```
<JobCreateInfo>
  <Name>Test</Name>
  <QueueId>00000000-0000-0000-0000-000000000000</QueueId>
  <JobMediaInfo>
    <CompressionPresetList>
      <CompressionPreset>
        <PresetId>758c03b4-7dee-4f58-84a1-e966c95432bf</PresetId>
      </CompressionPreset>
    </CompressionPresetList>
    <DestinationList>
      <DestinationInfo>
        <FileUri>file:///C:///filename.ext</FileUri>
      </DestinationInfo>
    </DestinationList>
  </JobMediaInfo>
</JobCreateInfo>
```



```

    <SourceMedia>
      <FileUri>
        file://C://filename.ext
      </FileUri>
    </SourceMedia>
  </JobMediaInfo>
</JobCreateInfo>” http://<host>/api/jobs/

```

## Get Job

Get Job returns detailed information about a single job by its ID.

<b>Base URI</b>	<code>/api/jobs/{JOBID}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/jobs/help/operations/GetJob">/api/jobs/help/operations/GetJob</a>

## Common Parameters

Name	Required	Description
<code>jobId</code>	Yes	Unique ID for the job

## Request Data Fields

None

## Response Data Fields

Name	Data Type	Required	Description
<code>JobInfo</code>	Element	No	Container for an individual job that contains the elements with information about the job (XML)
<code>InPoint</code>	Float	No	In-point in seconds and fractions of a second for encoding. (Future)
<code>Name</code>	String	No	User-supplied name for the job
<code>OutPoint</code>	Float	No	Out-point in seconds and fractions of a second for encoding (Future)

<code>QueueId</code>	Guid	Yes	Valid queue ID for existing job queue
<code>Status</code>	Enumeration	Yes	Valid states for a job: <b>0: Undefined</b> (Not yet set) <b>1: Waiting</b> <b>2: Downloading</b> <b>3: Transcoding</b> <b>4: Uploading</b> <b>5: Finished</b> <b>6: Error</b> <b>7: Canceled</b> <b>8: Deleted</b> <b>9: Hold</b> <b>10: Incomplete</b>
<code>ThumbTime</code>	Float	No	Time in seconds into the source that a thumbnail should be generated
<code>WatchFolderName</code>	String	No	Name of the watch folder the job was submitted by if the job was submitted by a watch folder
<code>EncodeServerId</code>	String	No	Name of the server that has taken on this job
<code>ErrorCode</code>	String	No	Brief string indicating error type
<code>Id</code>	Guid	Yes	Unique ID for the job
<code>Progress</code>	Float	Yes	Percent completion for current phase of job
<code>StatusList</code>	Element	Yes	Container for array of <b>StatusInfo</b> elements which hold each processional status record for the life of the job
<code>StatusInfo</code>	Element	No	Contains zero or more individual status records for the job (XML) See: <a href="#">StatusInfo</a>
<code>StatusMessage</code>	String	No	String with more detailed compression or configuration error information, if available
<code>StatusName</code>	String	Yes	User-friendly name for the state
<code>TimeFinished</code>	DateTime	No	Time of job completion (compression outputs uploaded or error state).
<code>TimeLastUpdate</code>	DateTime	No	Time when job information was last touched in the database
<code>TimeStarted</code>	DateTime	No	Time when job processing begins (start source download)

<code>TimeSubmitted</code>	DateTime	Yes	Time of job creation, set by the service
<code>JobMediaInfo</code>	Element	Yes	Container for presets, sources, and destinations associated with the job. See: <a href="#">JobMediaInfo</a>
<code>MetadataList</code>	Element	No	Contains zero or more <b>MetadataInfo</b> items to pass to supply meta data to the encoder (For future use). See <a href="#">MetadataInfo</a>

**Example cURL call:**

```
curl -H "Accept: application/json" -H "x-Authorization: Basic YWRtaW46YWRtaW4="
http://<host>/api/jobs/<jobId>
```

**NOTE:** header “x-Authorization: Basic <Base64 Encoded String>” is an other way to authorize a connection right in the header. Use application/xml if you want xml response.

## Get Job Collection

Returns a collection of all jobs active or waiting jobs. Results can be paged and filtered by queue.

<b>Base URI</b>	<code>/api/jobs/? queueId={QUEUEID}</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="#">/api/jobs/help/operations/GetJobs</a>

## Common Parameters

Name	Required	Description
<b>QueueId</b>	No	ID for the queue to query for jobs. If you do not supply a Queue ID, all active jobs will be returned.

## Request Data Fields

None

## Response Data Fields

Name	Data Type	Required	Description
<code>JobInfoList</code>	Element	Yes	Container for jobs returned, if any (XML)
<code>JobInfo</code>	Element	No	Container for an individual job that contains the elements with information about the job (XML).
<code>InPoint</code>	Float	No	In-point in seconds and fractions of a second for encoding (Future)
<code>Name</code>	String	No	User-supplied name for the job
<code>OutPoint</code>	Float	No	Out-point in seconds and fractions of a second for encoding (Future)
<code>QueueId</code>	Guid	Yes	Valid queue ID for existing job queue
<code>Status</code>	Enumeration	Yes	Valid states for a job: <b>0: Undefined</b> (Not yet set) <b>1: Waiting</b> <b>2: Downloading</b> <b>3: Transcoding</b> <b>4: Uploading</b> <b>5: Finished</b> <b>6: Error</b> <b>7: Canceled</b> <b>8: Deleted</b> <b>9: Hold</b> <b>10: Incomplete</b>
<code>ThumbTime</code>	Float	No	Time in seconds into the source that a thumbnail should be generated
<code>WatchFolderName</code>	String	No	Name of the watch folder the job was submitted by if the job was submitted by a watch folder
<code>EncodeServerId</code>	String	No	Name of the server that has taken on this job
<code>ErrorCode</code>	String	No	Brief string indicating error type
<code>Id</code>	Guid	Yes	Unique ID for the job
<code>Progress</code>	Float	Yes	Percent completion for current phase of job

<a href="#">StatusList</a>	Element	Yes	Container for array of <b>StatusInfo</b> elements which hold each processional status record for the life of the job.
<a href="#">StatusInfo</a>	Element	No	Contains zero or more individual status records for the job (XML) See: <a href="#">StatusInfo</a>
<a href="#">StatusMessage</a>	String	No	String with more detailed compression or configuration error information, if available
<a href="#">StatusName</a>	String	Yes	User-friendly name for the state
<a href="#">TimeFinished</a>	DateTime	No	Time of job completion (compression outputs uploaded or error state)
<a href="#">TimeLastUpdate</a>	DateTime	No	Time when job information was last touched in the database
<a href="#">TimeStarted</a>	DateTime	No	Time when job processing begins (start source download)
<a href="#">TimeSubmitted</a>	DateTime	Yes	Time of job creation, set by the service
<a href="#">JobMediaInfo</a>	Element	Yes	Container for presets, sources, and destinations associated with the job. See: <a href="#">JobMediaInfo</a>
<a href="#">MetadataList</a>	Element	No	Contains zero or more <b>MetadataInfo</b> items to pass to supply meta data to the encoder. (For future use). See <a href="#">MetadataIInfo</a>

**Example cURL call:**

```
curl -H "Accept: application/xml" http://<host>/api/jobs/ -u user:password
```

## Get Job Status Collection

Get Job Status collection with paging and queue selection support. Job status is a greatly reduced payload than that of Get Jobs.

<b>Base URI</b>	<code>/api/jobs/status/?queueId={QUEUEID}&amp;page={PAGE}&amp;pageSize={PAGESIZE}&amp;filter={FILTER}&amp;sortBy={SORTBY}</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/jobs/help/operations/GetJobStatusByQueue">/api/jobs/help/operations/GetJobStatusByQueue</a>

## Common Parameters

Name	Required	Description
<b>queueId</b>	No	Unique ID for the queue. If not supplied, status for all jobs in all queues will be returned
<b>page</b>	No	0-based page number to return where 0 (default) is the first page
<b>pageSize</b>	No	Number of job status records to return per page. The default is 10
<b>filter</b>	No	Deprecated
<b>sortBy</b>	No	Sorting for the results. By default, results are sorted by the time the jobs were created in descending order. The syntax for the sort is: <code>fieldname[order]</code> where <code>fieldname</code> must be a valid field from the results, and <code>order</code> must be <code>asc</code> (ascending) or <code>desc</code> (descending)

## Request Data Fields

None

## Response Data Fields

Name	Data Type	Required	Description
<a href="#">JobStatusInfoPage</a>	Element	Yes	Contains the page information and <a href="#">JobStatus</a> collection (XML)
<a href="#">HasNextPage</a>	Boolean	Yes	True if there are one or more additional pages
<a href="#">HasPreviousPage</a>	Boolean	Yes	True if there are one or more previous pages
<a href="#">JobStatus</a>	Element	Yes	Container list of zero or more <a href="#">JobStatusInfo</a> records
<a href="#">JobStatusInfo</a>	Element	No	Contains status information about each job returned in the request. See <a href="#">JobStatusInfo</a>
<a href="#">PageIndex</a>	Int	Yes	Page number (0 is first page)
<a href="#">PageSize</a>	Int	Yes	Number of items per page. <b>Note:</b> the total number of items returned can be less than the <a href="#">PageSize</a> for partial pages (i.e.: the last page).
<a href="#">SortColumn</a>	String	Yes	The column (field) by which the servers are sorted
<a href="#">SortDirection</a>	String	Yes	The sort direction can be ASC, asc, ascending, DESC, desc, descending
<a href="#">TotalCount</a>	Int	Yes	Total number of available results
<a href="#">TotalPages</a>	Int	Yes	The total number of pages for the given <a href="#">PageSize</a>

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4="
http://<host>/api/jobs/status/?queueId =00000000-0000-0000-0000-
000000000000&pagesize=0&sortby=JobStatus[asc]
```

**NOTE:** header "x-Authorization: Basic <Base64 Encoded String>" is an other way to authorize a connection right in the header

## Get Job Status

Get Job Status returns the status summary information for a single job by job ID.

<b>Base URI</b>	<code>/api/jobs/status/{JOBID}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/jobs/help/operations/GetJobStatus">/api/jobs/help/operations/GetJobStatus</a>

## Common Parameters

Name	Required	Description
<b>jobId</b>	Yes	Unique ID for the job to retrieve

## Request Data Fields

None

## Response Data Fields

Name	Type	Required	Description
<code>JobStatusInfo</code>	Element	Yes	Element containing status information for a single job
<code>EncodeServerId</code>	string	No	ID of the server that encoded or is encoding a job
<code>JobId</code>	Guid	No	Unique ID for a job
<code>LastUpdate</code>	DaetTime	No	Date and time (GMT) when the job was last updated
<code>Name</code>	String	Yes	User-friendly job name
<code>QueueId</code>	Guid	Yes	ID of the queue to which the job has been assigned



<code>Status</code>	Element	Yes	Contains a consolidated and summarized status for the job
<code>Created</code>	DateTime	Yes	Date and time (GMT) that the status record was created
<code>DestinationName</code>	String	No	User-friendly name for the job's destination. (Not used in this context)
<code>Duration</code>	DateTime	No	Total time the job has been processing or has been processed
<code>ErrorCode</code>	String	No	Last error encountered during job processing, if any
<code>Id</code>	Guid	No	Unique ID for the status record. Usually not used in this context
<code>Modified</code>	DateTime	No	Date and time (GMT) that the status was last changed
<code>PresetName</code>	String	No	Preset name associated with this status record. Not used in this context
<code>Progress</code>	Float	No	Overall job progress percent
<code>RetryCount</code>	Int	No	Number of retries this job has attempted
<code>Status</code>	Enumeration	Yes	Current status for the job. May be one of the following:  <b>0: Undefined</b> (not yet set) <b>1: Waiting</b> <b>2: Downloading</b> <b>3: Compressing</b> <b>4: Uploading</b> <b>5: Finished</b> <b>6: Error</b> <b>7: Canceled</b> <b>8: Deleted</b> <b>9: Hold</b> <b>10: Incomplete</b>
<code>StatusMessage</code>	String	No	Last status message generated during job processing
<code>TimeFinished</code>	DateTime	No	Date and time (GMT) the job finished processing
<code>TimeStarted</code>	DateTime	No	Date and time the job started processing
<code>TimeSubmitted</code>	DateTime	Yes	Date and time (GMT) when the job was submitted for processing

---

<code>TimeStarted</code>	<code>DateTime</code>	No	Date and time (GMT) when the job started processing
--------------------------	-----------------------	----	---

---

**Example cURL call:**

```
curl -H "Accept: application/json" -H "x-Authorization: Basic YWRtaW46YWRtaW4="
http://<host>/api/jobs/status/<jobId>
```

**NOTE:** header “x-Authorization: Basic <Base64 Encoded String>” is an other way to authorize a connection right in the header. Use application/xml if you want xml response.

## Delete Job

Delete Job removes a single job and its related elements from the database. Jobs should be in a Hold, Error, or Finished state. If a job is in another state, for best results, place the job in a Hold state before deleting.

---

<b>Base URI</b>	<code>/api/jobs/{JOBID}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	DELETE
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/jobs/help/operations/DeleteJob">/api/jobs/help/operations/DeleteJob</a>

---

## Common Parameters

---

Name	Required	Description
<code>jobId</code>	Yes	Unique ID for the job to delete.

---

## Request Data Fields

None

## Response Data Fields

None

**Example cURL call:**

```
curl -H "Accept: application/json" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X DELETE
http://<host>/api/jobs/<jobId>
```

## Presets API

Presets are .spfx XML files created with Sorenson Squeeze. Any preset you create in Squeeze can be used in Squeeze Server. Presets tell the Squeeze compression engine how to encode source files into the desired output. Squeeze Server can use presets from its own database (the Preset Library), the Sorenson Media Preset Exchange, a network file system, FTP site, or other web location.

**Note:** As of version 2.2 the Squeeze Server API accepts an ID or Name to identify a preset, including when creating a new preset. When Name is used to identify a preset, the Name must be unique.

## Create New Preset

Create New Preset allows you to add or replace, and if desired, publish a new preset to the Squeeze Server preset library. If authentication is enabled, a caller must be a member of the [SqueezeServerAdmin](#) or [SqueezeServerLibrarian](#) roles.

<b>Base URI</b>	<code>/api/presets/{preset_id_or_name}?replace={replace}&amp;publish={publish}</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	POST
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/presets/help/operations/CreateNewPreset">/api/presets/help/operations/CreateNewPreset</a>
<b>Authentication</b>	<a href="#">SqueezeServerAdmin</a> or <a href="#">SqueezeServerLibrarian</a>

## Common Parameters

Name	Required	Description
<b>preset_id_or_name</b>	Yes	Unique ID or name of the new preset.  If the ID is used, it must match the ID in the preset's XML. If they do not match, an exception will be thrown and the request will fail.  If the preset name is used, it should be unique to prevent a naming conflict and to easily distinguish presets within lists.
<b>replace</b>	No	<b>true</b> or <b>false</b> (default): Indicates that the preset should replace a preset with the same name. If the preset exists of the same name and <b>replace=false</b> , a 409 Conflict will be returned.
<b>publish</b>	No	<b>true</b> or <b>false</b> (default) indicates that the preset should be made available for use, or "published." Presets are not published by default to allow librarians the opportunity to test presets before making them

available to others.

## Request Data Fields

The request is a byte-stream consisting of the XML data for the new preset.

## Response Data Fields

Name	Data Type	Required	Description
<code>PresetInfo</code>	Element	Yes	Container for the preset information (XML)
<code>Created</code>	DateTime	Yes	Date the preset was created (GMT)
<code>Description</code>	String	Yes	The description pulled from the preset data, if any
<code>Extension</code>	String	Yes	The default file extension the preset will produce (ie: mov, flv, mp4). Note that a preset may produce a different file type in some cases.
<code>Id</code>	Guid	Yes	Unique ID for the new preset
<code>Modified</code>	DateTime	Yes	The date and time the preset was last modified
<code>Name</code>	String	Yes	The preset name extracted from the preset XML data
<code>Published</code>	Boolean	Yes	Indicates if the preset is published (publicly available) or not published (not publicly available)
<code>Uri</code>	String	Yes	The URL for the preset. Squeeze Server and other clients can request the preset data with this URL
<code>Workflow</code>	String	Yes	The workflow string for the preset. Extracted from the preset data if present

### Example CURL call:

Send XML data:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: text/plain" -X POST --data "<!DOCTYPE squeeze_audience><OutputDefaultName=\\"MXF_DNxHD_1455.spfx\\" Description=\\"Create New Test\\" FavWorkflow=\\"\\\" Id=\\"82629120\\" IsLite=\\"False\\" IsMBR=\\"False\\" Name=\\"Preset_Name_Create_New_Preset\\" OutputFormat=\\"MXFOutputFormat\\" SubOutputFormat=\\"OP-Atom\\" UsingCustomFilter=\\"True\\" Version=\\"6.0\\" Workflow=\\"Editing\\DNxHD\\"><CanBeResetToDefaultSettings Value=\\"True\\"/><FileName Value=\\"\\\"/><VideoFileName Value=\\"\\\"/><AudioFileName Value=\\"\\\"/><UsingCustomFileName Value=\\"False\\"/><OutputExt Value=\\"\\\"/><OutputAudExt Value=\\"\\\"/><OutputPrefix Value=\\"\\\"/><OutputSuffix Value=\\"\\\"/><UsePrefix Value=\\"True\\"/><UseSuffix Value=\\"True\\"/><Status Value=\\"Ready\\"/><EncodeVideo Value=\\"True\\"/><EncodeAudio Value=\\"True\\"/><UsePreFilter Value=\\"True\\"/><UsePlaybackConfiguration
```

```
Value="False"/><UseFileSizeConstraint Value="False"/><FileSizeConstraint
Value="0"/><InOutPointsLocallyAssigned Value="False"/><InOutPointsEnable
Value="False"/><Filter Name="Custom"/><FilterSetting Name="Crop"/><AspectRatio
Value="Unconstrained"/><FloatRatio Value="0.000000000000e+00"/><Top Value="0"/><Bottom
Value="0"/><Left Value="0"/><Right
Value="0"/></FilterSetting></Filter><Notifications></Notifications><AudioCompressor
codec="AVIDLPCM"/><DataRate Value="6144000"/><SampleRate Value="48000"/><SampleSize
Value="16"/><ChannelMode Value="ChannelMode7_1"/><SubOutputFormat
Value="None"/><ExtendedPluginSettings></ExtendedPluginSettings></AudioCompressor><VideoCompress
or codec="AVIDVC3 "/><DataRate Value="145000000"/><FrameRate Value="29.97"/><OutputSize
Height="1080" Width="1920"/><LetterBoxOrPillarEnable Value="True"/><BitRateMode
Value="BitRateModeOnePassCBR"/><ExtendedPluginSettings><FieldEncoding
Value="Interlaced"/></ExtendedPluginSettings></VideoCompressor></Output>"
http://localhost/api/presets/any_dummy_string/?replace=true&publish=true -v
```

**NOTE:** The preset name in the URL, any\_dummy\_string from the example, is required to create a new preset, but it is not used. So, it can be any string.

Send a preset file:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type:
text/plain" -X POST -d @"C:\MyPresetFolder\MOV_Wii.spfx"
http://<host>/api/presets/MyPreset/?replace=true&publish=true
```

## Get Preset Data

Get preset data by preset name or Unique ID, and returns with the content type based on format, such as 'xml' (default), 'raw', or 'spfx'.

<b>Base URI</b>	<code>/api/presets/{preset_id_or_name}/?format={format}</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	Preset Data (XML or Raw)
<b>Help URL</b>	<a href="/api/presets/help/operations/GetPresetData">/api/presets/help/operations/GetPresetData</a>
<b>Authentication:</b>	None

## Common Parameters

Name	Required	Description
<b>preset_id_or_name</b>	Yes	Unique ID or name of the preset. For best results, use the unique preset ID whenever possible.

---

<b>format</b>	No	<p>The format for the returned data:</p> <ul style="list-style-type: none"> <li>• <b>xml</b> (default): If you leave off format, or supply format=xml, the response type will be text/xml</li> <li>• <b>spfx</b>: if format=spfx, the preset will be returned as a .spfx file with a content disposition of preset-name.spfx and a content type of text/xml</li> <li>• <b>raw</b>: If format=raw, the preset will be returned as an ordinary byte stream.</li> </ul>
---------------	----	--

---

## Request Data Fields

None

## Response Data Fields

The response is returned as a stream with Content-Type of text/xml and Content-disposition set based on the format parameter, or as raw data. It is up to the caller to process the Preset XML data.

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4="
http://<host>/api/presets/0a6b0bc0-ac07-4fd5-9766-dd75e2a2bb7d/?format=xml
```

## Get Preset List

Returns a collection of preset meta-data (descriptive information) about presets in the Squeeze Server preset library. The actual preset data is not returned.

---

<b>Base URI</b>	<a href="/api/presets/info/?filter={filter}&amp;sortby={sortby}&amp;unpub={unpublished}">/api/presets/info/?filter={filter}&amp;sortby={sortby}&amp;unpub={unpublished}</a>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/presets/help/operations/GetLocalCollection">/api/presets/help/operations/GetLocalCollection</a>

---

## Common Parameters

Name	Required	Description
<b>filter</b>	No	Restrict results to presets that produce output files with the desired extension(s) and match given workflow(s), if any. For example, <b>filter=ext=mp4,ext=mov,workflow=%Adaptive%</b> will return all presets that produce mp4 or mov files, and whose workflow has Adaptive in the name. To return all presets, do not supply a filter parameter. Both ext and workflow filter parameters can be specified as a string matching pattern. String pattern syntax is documented at <a href="http://msdn.microsoft.com/en-us/library/ms179859.aspx">http://msdn.microsoft.com/en-us/library/ms179859.aspx</a> .
<b>sortby</b>	No	Sorting for the results. By default, results are sorted by the preset name in ascending order. The syntax for the sort is: fieldname[order] where fieldname must be a valid field from the results, and order must be asc (ascending) or desc (descending).
<b>unpublished</b>	No	Indicate to return both published and unpublished presets. By default or if unpublished is false, only published presets will be returned. <b>unpublished=true</b> will return both published and unpublished presets.

## Response Data Fields

Name	Data Type	Required	Description
<b>PresetInfoList</b>	Element	Yes	Top level container for the list of preset information (XML)
<b>PresetInfo</b>	Element	No	Container of preset information, if any
<b>Created</b>	DateTime	Yes	Date the preset was created (GMT)
<b>Description</b>	String	Yes	The description pulled from the preset data, if any
<b>Extension</b>	String	Yes	The default file extension the preset will produce (ie: mov, flv, mp4). Note that a preset may produce a different file type in some cases.
<b>Id</b>	Guid	Yes	Unique ID for the new preset
<b>Modified</b>	DateTime	Yes	The date and time the preset was last modified
<b>Name</b>	String	Yes	The preset name extracted from the preset XML data
<b>Published</b>	Boolean	Yes	Indicates if the preset is published (publicly available) or not published (not publicly available )

<b>Uri</b>	String	Yes	The URL for the preset. Squeeze Server and other clients can request the preset data with this URL.
<b>Workflow</b>	String	Yes	The workflow string for the preset. Extracted from the preset data if present.

**Example cURL call:**

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4="
http://<host>/api/presets/info/
```

## Get Preset Info

Get location information and description for the named preset.

<b>Base URI</b>	<code>/api/presets/info/{preset_id_or_name}?unpub={unpublished}</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/presets/help/operations/GetPresetInfo">/api/presets/help/operations/GetPresetInfo</a>

## Common Parameters

Name	Required	Value and Notes
<b>preset_id_or_name</b>	Yes	Unique ID or name of the preset. For best results, use the unique preset ID whenever possible.
<b>unpublished</b>	No	Indicate to return both published and unpublished presets. By default or if unpublished is false, only published presets will be returned. <b>unpublished=true</b> will return both published and unpublished presets.

## Request Data Fields

None



## Response Data Fields

Name	Data Type	Required	Description
<a href="#">PresetInfo</a>	Element	Yes	Container for preset information data.
<a href="#">Created</a>	DateTime	Yes	Date the preset was created (GMT)
<a href="#">Description</a>	String	Yes	The description pulled from the preset data, if any
<a href="#">Extension</a>	String	Yes	The default file extension the preset will produce (ie: mov, flv, mp4). Note that a preset may produce a different file type in some cases.
<a href="#">Id</a>	Guid	Yes	Unique ID for the new preset
<a href="#">Modified</a>	DateTime	Yes	The date and time the preset was last modified
<a href="#">Name</a>	String	Yes	The preset name extracted from the preset XML data
<a href="#">Published</a>	Boolean	Yes	Indicates if the preset is published (publicly available) or not published (not publicly available)
<a href="#">Uri</a>	String	Yes	The URL for the preset. Squeeze Server and other clients can request the preset data with this URL
<a href="#">Workflow</a>	String	Yes	The workflow string for the preset. Extracted from the preset data if present

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4="
http://<host>/api/presets/info/<Preset_GUID>/
```

## Update Preset Info

Updates an existing preset with new meta-data. Both the actual preset XML and the data record are updated with new information. If changing the preset name and there is already a preset with that name, the service will return a **409 Conflict** status code.

<b>Base URI</b>	<code>/api/presets/info/{preset_id_or_name }/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	PUT
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/presets/help/operations/UpdatePresetFromInfo">/api/presets/help/operations/UpdatePresetFromInfo</a>

## Common Parameters

Name	Required	Description
<b>preset_id_or_name</b>	Yes	Unique ID or name of the preset. For best results, use the unique preset ID whenever possible.

## Request Data Fields

Name	Data Type	Required	Description
<code>PresetInfo</code>	Element	Yes	Container for preset data
<code>Created</code>	DateTime	Private	Private - cannot be set by user
<code>Description</code>	String	No	The new description for the preset, if any
<code>Extension</code>	String	No	Private - cannot be set by user. This will be supplied by the preset data.
<code>Id</code>	Guid	Private	Private - cannot be set by the user
<code>Modified</code>	DateTime	Private	Private - cannot be set by the user

<b>Name</b>	String	No	The new name for the preset, if any. If the new name clashes with an existing preset, the service will return a <b>409 Conflict</b>
<b>Published</b>	Boolean	No	Indicates if the preset should be published or not published
<b>Uri</b>	String	Private	Private-- cannot be set by the user
<b>Workflow</b>	String	No	New workflow string for the preset, if any

### Response Data Fields

<b>Name</b>	<b>Data Type</b>	<b>Required</b>	<b>Description</b>
<b>PresetInfo</b>	Element	Yes	Contains preset information
<b>Created</b>	DateTime	Yes	Date the preset was created (GMT)
<b>Description</b>	String	No	The description pulled from the preset data, if any
<b>Extension</b>	String	No	The default file extension the preset will produce (ie: mov, flv, mp4). Note that a preset may produce a different file type in some cases.
<b>Id</b>	GUID	Yes	Unique ID for the new preset
<b>Modified</b>	DateTime	Yes	The date and time the preset was last modified
<b>Name</b>	String	Yes	The preset name extracted from the preset XML data
<b>Published</b>	Boolean	Yes	Indicates if the preset is published (publicly available) or not published (not publicly available)
<b>Uri</b>	String	Yes	The URL for the preset. Squeeze Server and other clients can request the preset data with this URL
<b>Workflow</b>	String	Yes	The workflow string for the preset. Extracted from the preset data if present

#### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -X PUT --data "
<PresetInfo>
<Created>2012-05-31T11:20:00</Created>
```

<Description>SampleDescription</Description>

<Modified>2012-05-31T11:20:00</Modified>

<Name>SampleName</Name></PresetInfo>" http://<host>/api/presets/info/Preset\_GUID/

## Get Preset Info by Workflow

Returns a collection of preset meta-data (descriptive information) about presets in the Squeeze Server preset library based on the workflow pattern. The actual preset data is not returned. You can specify as much or as little of a workflow string as you wish. For example, if you specify just the workflow `/web/` then information about all web presets will be returned. If you specify the root workflow and a sub category, such as `/web/flash 8/` then all presets that have a workflow that starts with `/web/flash 8/` will be returned, such as `/web/flash 8/streaming/` and `/web/flash 8/progressive/`.

<b>Base URI</b>	<code>/api/presets/info/workflow/{workflow}?unpub={unpublished}</code> <code>/api/presets/info/workflow/{workflow}/{subcategory}?unpub={unpublished}</code> <code>/api/presets/info/workflow/{workflow}/{subcategory}/{subsub}?unpub={unpublished}</code> <code>/api/presets/info/workflow/{workflow}/{subcategory}/{subsub}/{subsubsub}?unpub={unpublished}</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/presets/help/operations/GetLocalCollectionByMatching4Strings">/api/presets/help/operations/GetLocalCollectionByMatching4Strings</a>

## Common Parameters

Name	Required	Description
<b>workflow</b>	Yes	The root workflow to return
<b>subcategory</b>	No	workflow sub-category to match
<b>subsub</b>	No	Next level in the workflow hierarchy.
<b>subsubsub</b>	No	Next level in the workflow hierarchy.
<b>unpublished</b>	No	Indicate to return both published and unpublished presets. By default or if unpublished is false, only published presets will be returned. <code>unpublished=true</code> will return both published and unpublished presets.

## Request Data Fields

None

## Response Data Fields

Name	Data Type	Required	Description
<code>PresetInfoList</code>	Element	Yes	Top level container for the list of preset information (XML)
<code>PresetInfo</code>	Element	No	Container of preset information, if any
<code>Created</code>	DateTime	Yes	Date the preset was created (GMT)
<code>Description</code>	String	Yes	The description pulled from the preset data, if any
<code>Extension</code>	String	Yes	The default file extension the preset will produce (ie: mov, flv, mp4). Note that a preset may produce a different file type in some cases
<code>Id</code>	Guid	Yes	Unique ID for the new preset
<code>Modified</code>	DateTime	Yes	The date and time the preset was last modified
<code>Name</code>	String	Yes	The preset name extracted from the preset XML data
<code>Published</code>	Boolean	Yes	Indicates if the preset is published (publicly available) or not published (not publicly available yet)
<code>Uri</code>	String	Yes	The URL for the preset. Squeeze Server and other clients can request the preset data with this URL
<code>Workflow</code>	String	Yes	The workflow string for the preset. Extracted from the preset data if present

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4" -X GET http://<host>/api/presets/info/workflow/<workflowString>
```

## Update Preset from Stream

Updates an existing preset (will create a new preset if not found) with new preset data.

<b>Base URI</b>	<code>/api/presets/{preset_id_or_name}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	PUT
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<code>/api/presets/help/operations/UpdatePresetFromStream</code>
<b>Authentication</b>	If authentication is enabled, caller must be in the <code>SqueezeServerAdmin</code> or <code>SqueezeServerLibrarian</code> roles

### Common Parameters

Name	Required	Description
<code>preset_id_or_name</code>	Yes	Unique ID or name of the preset. For best results, use the unique preset ID whenever possible.

### Request Data Fields

The request body is a byte-stream consisting of the XML preset data.

### Response Data Fields

Name	Data Type	Required	Description
<code>PresetInfo</code>	Element	Yes	Container for the preset information (XML)
<code>Created</code>	DateTime	Yes	Date the preset was created (GMT)
<code>Description</code>	String	Yes	The description pulled from the preset data, if any
<code>Extension</code>	String	Yes	The default file extension the preset will produce (ie: mov, flv, mp4). Note that a preset may produce a different file type in some cases.
<code>Id</code>	Guid	Private	Unique ID for the new preset

<b>Modified</b>	DateTime	Yes	The date and time the preset was last modified
<b>Name</b>	String	Yes	The preset name extracted from the preset XML data
<b>Published</b>	Boolean	Yes	Indicates if the preset is published (publicly available) or not published (not publicly available)
<b>Uri</b>	String	Yes	The URL for the preset. Squeeze Server and other clients can request the preset data with this URL
<b>Workflow</b>	String	Yes	The workflow string for the preset. Extracted from the preset data if present

#### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: text/plain" -X PUT --data "<!DOCTYPE squeeze_audience><Output DefaultName=\"MXF_DNxHD_1455.spfx\" Description=\"Update Test\" FavWorkflow=\"\" Id=\"82629120\" IsLite=\"False\" IsMBR=\"False\" Name=\" Preset_Name_Update_Preset\" OutputFormat=\"MXFOutputFormat\" SubOutputFormat=\"OP-Atom\" UsingCustomFilter=\"True\" Version=\"6.0\" Workflow=\"Editing\DNxHD\"><CanBeResetToDefaultSettings Value=\"True\"/><FileName Value=\"\"/><VideoFileName Value=\"\"/><AudioFileName Value=\"\"/><UsingCustomFileName Value=\"False\"/><OutputExt Value=\"\"/><OutputAudExt Value=\"\"/><OutputPrefix Value=\"\"/><OutputSuffix Value=\"\"/><UsePrefix Value=\"True\"/><UseSuffix Value=\"True\"/><Status Value=\"Ready\"/><EncodeVideo Value=\"True\"/><EncodeAudio Value=\"True\"/><UsePreFilter Value=\"True\"/><UsePlaybackConfiguration Value=\"False\"/><UseFileSizeConstraint Value=\"False\"/><FileSizeConstraint Value=\"0\"/><InOutPointsLocallyAssigned Value=\"False\"/><InOutPointsEnable Value=\"False\"/><Filter Name=\"Custom\"><FilterSetting Name=\"Crop\"><AspectRatio Value=\"Unconstrained\"/><FloatRatio Value=\"0.000000000000e+00\"/><Top Value=\"0\"/><Bottom Value=\"0\"/><Left Value=\"0\"/><Right Value=\"0\"/></FilterSetting></Filter><Notifications></Notifications><AudioCompressor codec=\"AVIDLPCM\"><DataRate Value=\"6144000\"/><SampleRate Value=\"48000\"/><SampleSize Value=\"16\"/><ChannelMode Value=\"ChannelMode7_1\"/><SubOutputFormat Value=\"None\"/><ExtendedPluginSettings></ExtendedPluginSettings></AudioCompressor><VideoCompressor codec=\"AVIDVC3 \"><DataRate Value=\"145000000\"/><FrameRate Value=\"29.97\"/><OutputSize Height=\"1080\" Width=\"1920\"/><LetterBoxOrPillarEnable Value=\"True\"/><BitRateMode Value=\"BitRateModeOnePassCBR\"/><ExtendedPluginSettings><FieldEncoding Value=\"Interlaced\"/></ExtendedPluginSettings></VideoCompressor></Output>" http://localhost/api/presets/Preset_Name_Create_New_Preset/
```

## Delete Preset

Removes a preset from the library.

<b>Base URI</b>	<code>/api/presets/{preset_id_or_name}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	DELETE

<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/presets/help/operations/DeletePreset">/api/presets/help/operations/DeletePreset</a>
<b>Authentication</b>	If authentication is enabled, caller must be in the <a href="#">SqueezeServerAdmin</a> or <a href="#">SqueezeServerLibrarian</a> roles

### Common Parameters

Name	Required	Description
<b>preset_id_or_name</b>	Yes	Unique ID or name of the preset. For best results, use the unique preset ID whenever possible.

### Request Data Fields

None

### Response Data Fields

None



## Watch Folders API

Watch folders allow you to place source media files in a storage location such as an FTP site or Amazon S3 and have Squeeze Server automatically detect new and changed files, compress these files and place the results where you want them.

When Squeeze Server sees new files in a watch folder, it creates jobs for each of the files that it finds. Your Squeeze Server installations will then process these jobs normally. Your original files remain where they are. Squeeze Server will not move or delete them, nor will Squeeze Server process the same file again. You do not need to manually clean out your watch folders, and you can easily re-encode a file by “touching,” or updating its date and time to the current date and time. Squeeze Server will see it as a new or updated file and create a new job for it.

## Configuring Watch Folders

You configure watch folders just like a compression job. Like a compression job, a watch folder has a source -in this case a file folder or location visible to squeeze server, one or more destinations- where squeeze server will publish the output files, one or more presets that tell squeeze server how to transcode the media files, and a queue that squeeze server will use to place jobs it creates from the watch folders.

Unlike jobs, watch folders do not need a preset assigned. Watch folders will use any presets the server finds in the watch folder. This makes it very easy to change your encoding options without having to do anything other than copy a file. Watch folders still support the same preset options as the job-- getting presets from the library, the Internet, or with the preset data attached.

## Create Watch Folder

Create a new watch folder configuration.

---

<b>Base URI</b>	<code>/api/watchfolders/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	POST
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/CreateWatchFolder">/api/watchfolders/help/operations/CreateWatchFolder</a>

---

### Common Parameters

None

### Request Data Fields

Name	Data Type	Required	Description
<a href="#">WatchFolderCreateInfo</a>	Element	Yes	Container for an individual watch folder (XML)
<a href="#">DeepScan</a>	Boolean	No	Scan the watch folder and sub folders if true (default). If false, sub folders are not scanned.
<a href="#">Enabled</a>	Boolean	Yes	If true, the watch folder is available for server processing. If false, the watch folder will not be processed
<a href="#">Name</a>	String	Yes	Name given to the watch folder
<a href="#">QueueId</a>	Guid	Yes	Queue that jobs will be assigned to when created
<a href="#">JobMediaInfo</a>	Element	Yes	Container for presets, sources, and destinations associated with the job. See <a href="#">JobMediaInfo</a> . A watch folder must have one <a href="#">SourceMedia</a> , at least one <a href="#">DestinationInfo</a> and at least one <a href="#">CompressionPreset</a> of the preset is not in the watch folder itself.

### Response Data Fields

Name	Data Type	Required	Description
<a href="#">WatchFolderGetInfo</a>	Element	No	Container for an individual watch folder
<a href="#">DeepScan</a>	Boolean	No	Scan the watch folder and sub folders if true (default). If false, sub folders are not scanned.
<a href="#">Enabled</a>	Boolean	Yes	If true, the watch folder is available for server processing. If false, the watch folder will not be processed
<a href="#">Name</a>	String	Yes	Name given to the watch folder
<a href="#">QueueId</a>	Guid	Yes	Queue that jobs will be assigned to when created
<a href="#">CurrentScanServerId</a>	Guid	No	If supplied, the watch folder is currently being scanned by the server indicated by the ID

<code>CurrentScanServerStart</code>	DateTime	No	If supplied, this is the time (GMT) that the current scan was started
<code>LastScanEnd</code>	DateTime	No	If supplied, this is the time (GMT) that the last scan completed
<code>LastScanNewestFileTimeTicks</code>	Long	No	Time in ticks of the newest file found n the last scan that found files
<code>LastScanNumberJobsCreated</code>	Int	No	Number of jobs that were created the last time the folder was scanned and created jobs
<code>LastScanOldestFileTimeTicks</code>	Long	No	Time in ticks of the oldest file found in the last scan that found files
<code>LastScanServerName</code>	String	No	Name of the server that last scanned the watch folder
<code>LastScanStart</code>	DateTime	No	If supplied, the time the last watch folder scan started
<code>Message</code>	String	No	Any additional information message generated by the server when scanning the watch folder
<code>WatchFolderId</code>	Guid	Yes	Unique ID for the watch folder
<code>Created</code>	DateTime	No	Private-- supplied automatically
<code>JobMediaInfo</code>	Element	Yes	Container for presets, sources, and destinations associated with the job. See <a href="#">JobMediaInfo</a>

**Example cURL call:**

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -X POST -d "
<WatchFolderCreateInfo>
<DeepScan>true</DeepScan>
<Enabled>>false</Enabled>
<Name>My New Watch Folder</Name>
<QueueId>00000000-0000-0000-0000-000000000000</QueueId>
<JobMediaInfo>
  <CompressionPresetList>
    <CompressionPreset>
      <PresetId>0ecdccb9-a80e-4648-9467-c98c62787523</PresetId>
```

```

    </CompressionPreset>
    <CompressionPreset>
    <UriLocation>file:///C:/MyPresets/MP4\_x264\_512Kbps.spfx</UriLocation>
    </CompressionPreset>
</CompressionPresetList>
<DestinationList>
  <DestinationInfo>
    <ExtensionNamingMethod>Default</ExtensionNamingMethod>
    <FileNamingMethod>Default</FileNamingMethod>
    <FileUri>file:///C:/output/</FileUri>
  </DestinationInfo>
</DestinationList>
<SourceMedia>
  <FileUri>file:///C:/source/</FileUri>
</SourceMedia>
</JobMediaInfo>
</WatchFolderCreateInfo>" http://localhost/api/watchfolders/

```

## Get Watch Folders

Get a list of watch folders.

<b>Base URI</b>	<a href="/api/watchfolders/">/api/watchfolders/</a>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/GetWatchFolders">/api/watchfolders/help/operations/GetWatchFolders</a>

### Common Parameters

None

### Request Data Fields

None

### Response Data Fields

Name	Data Type	Required	Description
<code>WatchFolderGetInfoList</code>	Element	Yes	Container for the set of watch folders (XML)
<code>WatchFolderGetInfo</code>	Element	No	Container for an individual watch folder (XML)
<code>DeepScan</code>	Boolean	No	Scan the watch folder and sub folders if true (default). If false, sub folders are not scanned.
<code>Enabled</code>	Boolean	Yes	If true, the watch folder is available for server processing. If false, the watch folder will not be processed
<code>Name</code>	String	Yes	Name given to the watch folder
<code>QueueId</code>	Guid	Yes	Queue that jobs will be assigned to when created
<code>CurrentScanServerId</code>	Guid	No	If supplied, the watch folder is currently being scanned by the server indicated by the ID.
<code>CurrentScanServerStart</code>	DateTime	No	If supplied, this is the time (GMT) that the current scan was started.
<code>LastScanEnd</code>	DateTime	No	If supplied, this is the time (GMT) that the last scan completed.
<code>LastScanNewestFileTimeTicks</code>	Long	No	Time in ticks of the newest file found in the last scan that found files
<code>LastScanNumberJobsCreated</code>	Int	No	Number of jobs that were created the last time the folder was scanned and created jobs
<code>LastScanOldestFileTimeTicks</code>	Long	No	Time in ticks of the oldest file found in the last scan that found files
<code>LastScanServerName</code>	String	No	Name of the server that last scanned the watch folder
<code>LastScanStart</code>	DateTime	No	If supplied, the time the last watch folder scan started.
<code>Message</code>	String	No	Any additional information message generated by the server when scanning the watch folder.

<code>WatchFolderId</code>	Guid	Yes	Unique ID for the watch folder
<code>Created</code>	DateTime	Yes	Date and time the watch folder was created
<code>JobMediaInfo</code>	Element	Yes	Container for the <b>JobMediaInfo</b> object. See <a href="#">JobMediaInfo</a> , <a href="#">CompressionPreset</a> , <a href="#">DestinationInfo</a> , and <a href="#">SourceMedia</a> . A watch folder must have at least one destination in the <b>JobMediaInfo</b> , only one <b>SourceMedia</b> , but <b>CompressionPreset</b> is optional if at least one preset is in the watch folder.

**Example cURL call:**

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X Get http://<host>/api/watchfolders/
```

## Get Watch Folder

Get information about a watch folder selected by the watch folder's unique ID.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/GetWatchFolder">/api/watchfolders/help/operations/GetWatchFolder</a>

## Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder to retrieve.

## Request Data Fields

None

## Response Data Fields

Name	Data Type	Required	Description
<code>WatchFolderGetInfo</code>	Element	No	Container for an individual watch folder (XML)
<code>DeepScan</code>	Boolean	No	Scan the watch folder and sub folders if true (default). If false, sub folders are not scanned.
<code>Enabled</code>	Boolean	Yes	If true, the watch folder is available for server processing. If false, the watch folder will not be processed
<code>Name</code>	String	Yes	Name given to the watch folder
<code>QueueId</code>	Guid	Yes	Queue that jobs will be assigned to when created
<code>CurrentScanServerId</code>	Guid	No	If supplied, the watch folder is currently being scanned by the server indicated by the ID
<code>CurrentScanServerStart</code>	DateTime	No	If supplied, this is the time (GMT) that the current scan was started.
<code>LastScanEnd</code>	DateTime	No	If supplied, this is the time (GMT) that the last scan completed
<code>LastScanNewestFileTimeTicks</code>	Long	No	Time in ticks of the newest file found in the last scan that found files
<code>LastScanNumberJobsCreated</code>	Int	No	Number of jobs that were created the last time the folder was scanned and created jobs
<code>LastScanOldestFileTimeTicks</code>	Long	No	Time in ticks of the oldest file found in the last scan that found files
<code>LastScanServerName</code>	String	No	Name of the server that last scanned the watch folder
<code>LastScanStart</code>	DateTime	No	If supplied, the time the last watch folder scan started
<code>Message</code>	String	No	Any additional information message generated by the server when scanning the watch folder

<code>WatchFolderId</code>	Guid	Yes	Unique ID for the watch folder
<code>Created</code>	DateTime	No	Private-- supplied automatically
<code>JobMediaInfo</code>	Element	Yes	Container for presets, sources, and destinations associated with the watch folder. See <a href="#">JobMediaInfo</a>

**Example cURL call:**

```
curl -H "Accep: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4" -X GET
http://<host>//api/watchfolders/<WatchFolder_GUID>
```

## Update Watch Folder

Update an existing watch folder configuration. This updates basic watch folder configuration. To update individual sources, destinations, or presets, update these with the services provided for that purpose.

Some fields are private and can only be updated by the server.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	PUT
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<code>/api/watchfolders/help/operations/UpdateWatchFolder</code>

## Common Parameters

Name	Required	Description
<code>folderId</code>	Yes	Unique ID of the watch folder to update.

## Request Data Fields

Name	Data Type	Required	Description
------	-----------	----------	-------------



<code>WatchFolderUpdateInfo</code>	Element	Yes	Container for an individual watch folder (XML)
<code>DeepScan</code>	Boolean	No	Scan the watch folder and sub folders if true (default). If false, sub folders are not scanned.
<code>Enabled</code>	Boolean	Yes	If true, the watch folder is available for server processing. If false, the watch folder will not be processed
<code>Name</code>	String	Yes	Name given to the watch folder
<code>QueueId</code>	Guid	Yes	Queue that jobs will be assigned to when created
<code>CurrentScanServerId</code>	Guid	Private	Set by squeeze server when a server scans a watch folder
<code>CurrentScanStart</code>	DateTime	Private	Set by squeeze server when a server scans a watch folder
<code>LastScanEnd</code>	DateTime	Private	Set by squeeze server when a server scans a watch folder
<code>LastScanNewestFileTimeTicks</code>	Long	Private	Set by squeeze server when a server scans a watch folder
<code>LastScanNumberJobsCreated</code>	Int	Private	Set by squeeze server when a server scans a watch folder
<code>LastScanOldestFileTimeTicks</code>	Long	Private	Set by squeeze server when a server scans a watch folder
<code>LastScanServerName</code>	String	Private	Set by squeeze server when a server scans a watch folder
<code>LastScanStart</code>	DateTime	Private	Set by squeeze server when a server scans a watch folder
<code>Message</code>	String	Private	Set by squeeze server when a server scans a watch folder
<code>WatchFolderId</code>	Guid	No	Unique ID for the watch folder, same as the folder ID on the request. May be omitted.

## Response Data Fields

Name	Data Type	Required	Description
<code>WatchFolderGetInfo</code>	Element	No	Container for an individual watch folder (XML)
<code>DeepScan</code>	Boolean	No	Scan the watch folder and sub folders if true (default). If false, sub folders are not scanned.
<code>Enabled</code>	Boolean	Yes	If true, the watch folder is available for server processing. If false, the watch folder will not be processed
<code>Name</code>	String	Yes	Name given to the watch folder
<code>QueueId</code>	Guid	Yes	Queue that jobs will be assigned to when created
<code>CurrentScanServerId</code>	Guid	No	If supplied, the watch folder is currently being scanned by the server indicated by the ID
<code>CurrentScanServerStart</code>	DateTime	No	If supplied, this is the time (GMT) that the current scan was started
<code>LastScanEnd</code>	DateTime	No	If supplied, this is the time (GMT) that the last scan completed
<code>LastScanNewestFileTimeTicks</code>	Long	No	Time in ticks of the newest file found in the last scan that found files
<code>LastScanNumberJobsCreated</code>	Int	No	Number of jobs that were created the last time the folder was scanned and created jobs
<code>LastScanOldestFileTimeTicks</code>	Long	No	Time in ticks of the oldest file found in the last scan that found files
<code>LastScanServerName</code>	String	No	Name of the server that last scanned the watch folder
<code>LastScanStart</code>	DateTime	No	If supplied, the time the last watch folder scan started
<code>Message</code>	String	No	Any additional information message generated by the server when scanning the watch folder

<code>WatchFolderId</code>	Guid	Yes	Unique ID for the watch folder
<code>Created</code>	DateTime	No	Private - supplied automatically
<code>JobMediaInfo</code>	Element	Yes	Container for presets, sources, and destinations associated with the watch folder. See <a href="#">JobMediaInfo</a>

#### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -X PUT -d "
<WatchFolderUpdateInfo>
  <DeepScan>>false</DeepScan>
  <Enabled>>true</Enabled>
  <Name>testWatchFolder</Name>
  <QueueId>00000000-0000-0000-0000-000000000000</QueueId>
</WatchFolderUpdateInfo>" http://<host>/api/watchfolders/<WatchFolder_GUID>/
```

## Watch Folder Sources

A watch folder source specifies the location for source files. Sources can be Amazon S3 buckets (including a prefix or artificial folder within a bucket), FTP site, network file system. Presets created with Sorenson Squeeze placed in a watch folder will be processed for each media file. To re-process a file in a watch folder, update the file's date and time so Squeeze Server will think the file is new.

## Get Watch Folder Sources

Returns a list of the sources locations for a watch folder. A watch folder can have only one source location. This service will return at most a list of one, but exists to keep REST URIs consistent and to allow for more than one source in the future.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/sources/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="#">/api/watchfolders/help/operations/GetWatchFolderSources</a>

## Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder

## Request Data Fields

None

## Response Data Fields

Name	Data Type	Required	Description
<code>SourceMediaInfoList</code>	Element	Yes	Contains one <code>SourceMediaInfo</code> element (XML). A watch folder can have only one source
<code>SourceMediaInfo</code>	Element	Yes	Element contains data describing a job source or a watch folder source location
<code>Created</code>	DateTime	Private	Date and time (GMT) that the source record was created
<code>CredentialId</code>	Guid	No	Optional ID of a stored credential set to use to access the source.
<code>FileName</code>	String	No	The file name will be treated as a folder if it ends with a slash (/) for watch folders
<code>FileSize</code>	Long	No	Not used with watch folders
<code>FileUri</code>	String	No	File Uri is the fully specified URI to the source file or location. Supported sources are <code>ftp://</code> , <code>sftp://</code> , <code>s3://</code> , <code>azureblob://</code> , and <code>file://</code>
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>Modified</code>	DateTime	Private	Date and time the source has been modified, if any
<code>Password</code>	String	No	Password to access secure source if the <code>CredentialId</code> is not used
<code>S3BucketName</code>	String	Obsolete	Name of the S3 bucket with the source. Required if the type is S3
<code>SourceId</code>	Guid	Private	Unique ID for the source record

<b>Type</b>	Enumeration	Obsolete	Type of source. The source can be: <b>0: Undefined</b> – Not Set <b>1: Uri</b> – Source is defined by a URI <b>2: S3</b> – Source is on Amazon S3
<b>UserName</b>	String	No	Username for accessing a secured source location or file and when not using <b>CredentialId</b>
<b>WatchFolderId</b>	Guid	Private	Unique ID of the watch folder this source is attached to

**Example cURL call:**

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X DELETE -d "" http://<host>/api/watchfolders/<watchfolder_GUID>/
```

## Get Watch Folder Source

Returns details for the specified source associated with the specified watch folder.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/sources/{sourceId}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<code>/api/watchfolders/help/operations/GetWatchFolderSource</code>

## Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder
<b>sourceId</b>	Yes	Unique Source ID for the source

## Request Data Fields

None

## Response Data Fields

Name	Data Type	Required	Description
<code>SourceMediaInfo</code>	Element	Yes	Element contains data describing a watch folder source location (XML)
<code>Created</code>	DateTime	Private	Date and time (GMT) that the source record was created
<code>CredentialId</code>	GUID	No	Optional ID of a stored credential set to use to access the source
<code>FileName</code>	String	No	The file name will be treated as a folder if it ends with a slash (/) for watch folders.
<code>FileSize</code>	Long	No	Not used with watch folders
<code>FileUri</code>	String	No	File Uri is the fully specified URI to the source file or location. Supported sources are <code>ftp://</code> , <code>sftp://</code> , <code>s3://</code> , <code>azureblob://</code> and <code>file://</code>
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>Modified</code>	DateTime	Private	Date and time the source has been modified, if any
<code>Password</code>	String	No	Password to access secure source if the <code>CredentialId</code> is not used
<code>S3BucketName</code>	String	Obsolete	Name of the S3 bucket with the source. Required if the type is S3
<code>SourceId</code>	Guid	Private	Unique ID for the source record
<code>Type</code>	Enumeration	Obsolete	Type of source. The source can be: <b>0: Undefined</b> – Not Set <b>1: Uri</b> – Source is defined by a URI <b>2: S3</b> – Source is on Amazon S3
<code>UserName</code>	String	No	Username for accessing a secured source location or file and when not using <code>CredentialId</code>
<code>WatchFolderId</code>	Guid	Private	Unique ID of the watch folder to which this source belongs

**Example cURL call:**

```
curl -H "Accept: application/xml" -H "Content-Type: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X GET http://<host>/api/watchfolders/<WatchFolder_GUID>/sources/<Source_GUID>/
```

## Create Watch Folder Source

Adds a source to the specified watch folder. If the watch folder has a source assigned already, this will replace the existing source with the new source.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/sources/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	POST
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/CreateWatchFolderSource">/api/watchfolders/help/operations/CreateWatchFolderSource</a>

### Common Parameters

Name	Required	Description
<code>folderId</code>	Yes	Unique ID for the watch folder

### Request Data Fields

Name	Data Type	Required	Description
<code>SourceMediaInfo</code>	Element	Yes	Element contains data describing a watch folder source location
<code>Created</code>	DateTime	Private	Date and time (GMT) that the source record was created
<code>CredentialId</code>	Guid	No	Optional ID of a stored credential set to use to access the source
<code>FileName</code>	String	No	The file name will be treated as a folder if it ends with a slash (/) for watch folders.
<code>FileSize</code>	Long	No	Not used with watch folders

<code>FileUri</code>	String	No	File Uri is the fully specified URI to the source file or location. Supported sources <code>ftp://</code> , <code>sftp://</code> , <code>s3://</code> , <code>azureblob://</code> , and <code>file://</code>
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>Modified</code>	DateTime	Private	Date and time the source has been modified, if any
<code>Password</code>	String	No	Password to access secure source if the <code>CredentialId</code> is not used
<code>S3BucketName</code>	String	Obsolete	Name of the S3 bucket with the source. Required if the type is S3
<code>SourceId</code>	Guid	Private	Unique ID for the source record
<code>Type</code>	Enumeration	Obsolete	Type of source. The source can be: <b>0: Undefined</b> – Not Set <b>1: Uri</b> – Source is defined by a URI <b>2: S3</b> – Source is on Amazon S3
<code>UserName</code>	String	No	Username for accessing a secured source location or file and when not using <code>CredentialId</code> .
<code>WatchFolderId</code>	Guid	Private	Watch folder ID if the source is attached to a watch folder.

## Response Data Fields

Name	Data Type	Required	Description
<code>SourceMediaInfo</code>	Element	Yes	Element contains data describing a watch folder source location (XML)
<code>Created</code>	DateTime	Private	Date and time (GMT) that the source record was created.
<code>CredentialId</code>	Guid	No	Optional ID of a stored credential set to use to access the source.
<code>FileName</code>	String	No	The file name will be treated as a folder if it ends with a slash (/) for watch folders.
<code>FileSize</code>	Long	No	Not used with watch folders.



<code>FileUri</code>	String	No	File Uri is the fully specified URI to the source file or location. Supported sources are <code>ftp://</code> , <code>sftp://</code> , <code>s3://</code> , <code>azureblob://</code> , and <code>file://</code>
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>Modified</code>	DateTime	Private	Date and time the source has been modified, if any.
<code>Password</code>	String	No	Password to access secure source if the <code>CredentialId</code> is not used.
<code>S3BucketName</code>	String	Obsolete	Name of the S3 bucket with the source. Required if the type is S3.
<code>SourceId</code>	Guid	Private	Unique ID for the source record.
<code>Type</code>	Enumeration	Obsolete	Type of source. The source can be: <b>0: Undefined</b> – Not Set <b>1: Uri</b> – Source is defined by a URI <b>2: S3</b> – Source is on Amazon S3
<code>UserName</code>	String	No	Username for accessing a secured source location or file and when not using <code>CredentialId</code> .
<code>WatchFolderId</code>	Guid	Private	Unique ID of the watch folder to which this source belongs.

**Example cURL call:**

```
curl -H "Accept: application/xml" -H "Content-Type: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X POST -d "<SourceMediaInfo><FileUri>file://C://testFolder</FileUri></SourceMediaInfo>" http://<host>/api/watchfolders/<WatchFolder_GUID>/sources/
```

## Update Watch Folder Source

This operation allows the user to change the details for a job source. No fields are required; update is sparse. Include only the elements that are to be changed.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/sources/{sourceId}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	PUT
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/UpdateWatchFolderSource">/api/watchfolders/help/operations/UpdateWatchFolderSource</a>

### Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the Watch Folder
<b>sourceId</b>	Yes	Unique ID for the watch folder source

### Request Data Fields

Name	Data Type	Required	Description
<code>SourceMediaInfo</code>	Element	Yes	Element contains data describing a watch folder source location
<code>Created</code>	DateTime	Private	Date and time (GMT) that the source record was created
<code>CredentialId</code>	Guid	No	Optional ID of a stored credential set to use to access the source
<code>FileName</code>	String	No	The file name will be treated as a folder if it ends with a slash (/) for watch folders.
<code>FileSize</code>	Long	No	Not used with watch folders
<code>FileUri</code>	String	No	File Uri is the fully specified URI to the source file or location. Supported sources are <b>ftp://</b> ,

---

<code>sftp://</code> , <code>s3://</code> , <code>azureblob://</code> , and <code>file://</code>			
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>Modified</code>	DateTime	Private	Date and time the source has been modified, if any
<code>Password</code>	String	No	Password to access secure source if the <code>CredentialId</code> is not used
<code>S3BucketName</code>	String	Obsolete	Name of the S3 bucket with the source. Required if the type is S3
<code>SourceId</code>	Guid	Private	Unique ID for the source record
<code>Type</code>	Enumeration	Obsolete	Type of source. The source can be: <b>0: Undefined</b> – Not Set <b>1: Uri</b> – Source is defined by a URI <b>2: S3</b> – Source is on Amazon S3
<code>UserName</code>	String	No	Username for accessing a secured source location or file and when not using <code>CredentialId</code>
<code>WatchFolderId</code>	Guid	Private	Watch folder ID if the source is attached to a watch folder

---

### Response Data Fields

---

Name	Data Type	Required	Description
<code>SourceMediaInfo</code>	Element	Yes	Element contains data describing a watch folder source location (XML)
<code>Created</code>	DateTime	Private	Date and time (GMT) that the source record was created
<code>CredentialId</code>	GUID	No	Optional ID of a stored credential set to use to access the source
<code>FileName</code>	String	No	The file name will be treated as a folder if it ends with a slash (/) for watch folders.
<code>FileSize</code>	Long	No	Not used with watch folders
<code>FileUri</code>	String	No	File Uri is the fully specified URI to the source file or location. Supported sources are <code>ftp://</code> ,

---

---

<code>sftp://</code> , <code>s3://</code> , <code>azureblob://</code> , and <code>file://</code>			
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>Modified</code>	DateTime	Private	Date and time the source has been modified, if any
<code>Password</code>	String	No	Password to access secure source if the <code>CredentialId</code> is not used
<code>S3BucketName</code>	String	Obsolete	Name of the S3 bucket with the source. Required if the type is S3
<code>SourceId</code>	Guid	Private	Unique ID for the source record
<code>Type</code>	Enumeration	Obsolete	Type of source. The source can be: <b>0: Undefined</b> – Not Set <b>1: Uri</b> – Source is defined by a URI <b>2: S3</b> – Source is on Amazon S3
<code>UserName</code>	String	No	Username for accessing a secured source location or file and when not using <code>CredentialId</code>
<code>WatchFolderId</code>	Guid	Private	Unique ID of the watch folder to which this source belongs

---

**Example cURL call:**

```
curl -H "Accept: application/xml" -H "Content-Type: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X PUT -d "
```

```
<SourceMediaInfo>
  <FileUri>file:///C://newSource//</FileUri>
  <SourceId>aea33fcf-303c-4c52-aacc-031c8beeea70</SourceId>
  <WatchFolderId>d6a3c54e-e0cf-482f-91f9-c6b0cf077c28</WatchFolderId>
</SourceMediaInfo>" http://<host>/api/watchfolders/<WatchFolderId>/sources/<SourceId>/
```

## Delete Watch Folder Source

Delete Watch Folder Source deletes the specified source from the specified job. A watch folder without a valid source will be skipped by the servers until a valid source is supplied.

---

<b>Base URI</b>	<code>/api/watchfolder/{folderId}/sources/{sourceId}/</code>
<b>Protocol</b>	HTTP

---

---

<b>HTTP Method</b>	DELETE
<b>Response Formats</b>	None
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/DeleteWatchFolderSource">/api/watchfolders/help/operations/DeleteWatchFolderSource</a>

---

### Common Parameters

---

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch older
<b>sourceId</b>	Yes	Unique Source ID for the source

---

### Request Data Fields

None

### Response Data Fields

None

#### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X DELETE
http://<host>/api/watchfolders/<watchFolder_GUID>/sources/SourceId/
```

## Watch Folder Destinations

Just like a job, a watch folder can have any number of destinations. Destinations can be Amazon S3 buckets with optional artificial directory structure, FTP, or network file system. Squeeze Server will copy output files to each of the watch folder destinations.

### Get Watch Folder Destinations

Returns a list of the destinations associated with a watch folder.

---

<b>Base URI</b>	<a href="/api/watchfolders/{folderId}/destinations/">/api/watchfolders/{folderId}/destinations/</a>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET

---

<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/GetWatchFolderDestinations">/api/watchfolders/help/operations/GetWatchFolderDestinations</a>

### Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder

### Request Data Fields

None

### Response Data Fields

Name	Data Type	Required	Description
<a href="#">DestinationInfoList</a>	Element	Yes	Contains one or more <a href="#">DestinationInfo</a> elements (XML).
<a href="#">DestinationInfo</a>	Element	Yes	Contains a DestinationInfo record describing a watch folder publishing destination (XML). See <a href="#">DestinationInfo</a>

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -X GET http://<host>/api/watchfolders/<WatchFolderGUID>/destinations/
```

## Get Watch Folder Destination

Returns details for the specified destination associated with the specified watch folder.

<b>Base URI</b>	<a href="/api/watchfolders/{folderId}/destinations/{destinationId}/">/api/watchfolders/{folderId}/destinations/{destinationId}/</a>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON

---

**Help URL**                    </api/jobs/help/operations/GetWatchFolderDestination>

---

### Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder
<b>destinationId</b>	Yes	Unique Destination ID for the destination

### Request Data Fields

None

### Response Data Fields

Name	Data Type	Required	Description
<a href="#">DestinationInfo</a>	Element	Yes	Contains a <b>DestinationInfo</b> record describing a watch folder publishing destination (XML). See <a href="#">DestinationInfo</a>

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -X GET
http://<host>/api/watchfolders/<WatchFolderGUID>/destinations/<DestinationID>/
```

## Create Watch Folder Destination

Adds a destination to the specified watch folder.

<b>Base URI</b>	<a href="/api/watchfolders/{folderId}/destinations/">/api/watchfolders/{folderId}/destinations/</a>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	POST
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/CreateWatchFoldersDestination">/api/watchfolders/help/operations/CreateWatchFoldersDestination</a>

---

## Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder

## Request Data Fields

Name	Data Type	Required	Description
<a href="#">DestinationInfo</a>	Element	Yes	Container for destination information. See <a href="#">DestinationInfo</a>

## Response Data Fields

Name	Data Type	Required	Description
<a href="#">DestinationInfo</a>	Element	Yes	Container for destination information. See <a href="#">DestinationInfo</a>

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -d "
```

```
<DestinationInfo>
```

```
  <DestinationName>myNewDestination</DestinationName>
```

```
  <ExtensionNamingMethod>Default</ExtensionNamingMethod>
```

```
  <FileNamingMethod>Default</FileNamingMethod>
```

```
  <FileUri>file:///C:/myNewDestination/</FileUri>
```

```
</DestinationInfo>" -X POST http://<host>/api/watchfolders/<WatchFolder_ID>/destinations/
```

## Update Watch Folder Destination

Allows the user to change the details for a watch folder destination. No fields are required, update is sparse. Include only the elements that are to be changed.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/destinations/{destinationId}/</code>
<b>Protocol</b>	HTTP



---

<b>HTTP Method</b>	PUT
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/UpdateWatchFolderDestination">/api/watchfolders/help/operations/UpdateWatchFolderDestination</a>

---

### Common Parameters

---

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder
<b>destinationId</b>	Yes	Unique ID for the watch folder destination

---

### Request Data Fields

---

Name	Data Type	Required	Description
<a href="#">DestinationInfo</a>	Element	Yes	Container for destination information. See <a href="#">DestinationInfo</a>

---

### Response Data Fields

---

Name	Data Type	Required	Description
<a href="#">DestinationInfo</a>	Element	Yes	Container for destination information. See <a href="#">DestinationInfo</a>

---

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -d "
<DestinationInfo>
  <DestinationName>myNewDestination</DestinationName>
  <ExtensionNamingMethod>Default</ExtensionNamingMethod>
  <FileNamingMethod>Default</FileNamingMethod>
  <FileUri>file:///C://myUpdatedDestination/</FileUri>
</DestinationInfo>" -X PUT
http://<host>/api/watchfolders/<WatchFolder_ID>/destinations/<Destination_Id>
```

## Delete Watch Folder Destination

Deletes the specified destination from the specified watch folder.

---

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/destinations/{destinationId}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	DELETE
<b>Response Formats</b>	None
<b>Help URL</b>	<a href="/api/jobs/help/operations/DeleteWatchFolderDestination">/api/jobs/help/operations/DeleteWatchFolderDestination</a>

---

### Common Parameters

---

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder
<b>destinationId</b>	Yes	Unique Destination ID for the destination

---

Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X DELETE http://<host>/api/watchfolders/<WatchFolder_ID>/destinations/<Destination_Id>
```

## Watch Folder Presets

A Watch Folder Preset is not the preset itself, but a locator associating the watch folder with a preset file. The Watch Folder Preset can include a PresetId for a stored preset, or a Uri pointing to a remote preset file. At least one of these must be set, with PresetId taking precedence when processing.

Watch folders do not need presets assigned in this way as jobs do. Watch folders will process preset files storied in the watch folder with media files as well as any presets assigned to the watch folder in this way.

## Get Watch Folder Presets

Returns a list of the presets associated with a particular watch folder except for any preset files stored in watch folders.

---

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/presets/</code>
-----------------	--

---

<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/GetWatchFolderPresets">/api/watchfolders/help/operations/GetWatchFolderPresets</a>

### Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder

### Request data Fields

None

### Response Data Fields

Name	Data Type	Required	Description
<a href="#">JobPresetInfoList</a>	Element	Yes	Contains zero or more <a href="#">JobPreset</a> Elements (XML)
<a href="#">JobPreset</a>	Element	No	Contains data for an individual preset (XML)
<a href="#">JobId</a>	Guid	Private	Not used with watch folders
<a href="#">JobPresetId</a>	Guid	Private	Unique Id of this Job Preset record
<a href="#">PresetId</a>	Guid	No	ID of a preset from the preset library. A Job Preset record must have only one of <a href="#">PresetId</a> , <a href="#">UriLocation</a> , or <a href="#">PresetXmlBase64Data</a>
<a href="#">PresetXmlBase64Data</a>	String	No	Base-64 encoded preset data
<a href="#">UriLocation</a>	String	No	URI to a preset. Presets may referenced by <a href="#">http://</a> or <a href="#">file://</a>
<a href="#">WatchFolderId</a>	Guid	No	Unique ID of a watch folder this preset is associated with

**Example cURL call:**

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -X GET http://<host>/api/watchfolders/<WatchFolder_Id>/presets/
```

## Get Watch Folder Preset

Returns information about the specified preset associated with the specified watch folder.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/presets/{presetId}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	GET
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/GetWatchFolderPreset">/api/watchfolders/help/operations/GetWatchFolderPreset</a>

### Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the
<b>presetId</b>	Yes	Unique ID for the preset

### Request Data Fields

None

### Response Data Fields

Name	Data Type	Required	Description
<a href="#">JobPreset</a>	Element	No	Contains data for an individual job preset (XML)
<a href="#">JobId</a>	Guid	Private	Not used with watch folders
<a href="#">JobPresetId</a>	Guid	Private	Unique Id of this Job Preset record
<a href="#">PresetId</a>	Guid	No	ID of a preset from the preset library. A Job Preset record must have only one of <a href="#">PresetId</a> , <a href="#">UriLocation</a> , or <a href="#">PresetXmlBase64Data</a>

<code>PresetXmlBase64Data</code>	String	No	Base-64 encoded preset data
<code>UriLocation</code>	String	No	URI to a preset. Presets may be referenced by <a href="#">http://</a> , or <a href="#">file://</a>
<code>WatchFolderId</code>	Guid	No	Unique ID of a watch folder this preset is associated with

#### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -X GET http://<host>/api/watchfolders/<WatchFolder_Id>/presets/<JobPresetId>/
```

## Create Watch Folder Preset

Adds a job preset to the specified watch folder.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/presets/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	POST
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<a href="#">/api/watchfolders/help/operations/CreateWatchFolderPreset</a>

### Common Parameters

Name	Required	Description
<code>folderId</code>	Yes	Unique ID for the watch folder

### Request Data Fields

Name	Data Type	Required	Description
<code>JobPreset</code>	Element	No	Contains data for an individual job preset (XML)
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>JobPresetId</code>	Guid	Private	Unique Id of this Job Preset record

<code>PresetId</code>	Guid	No	ID of a preset from the preset library. A Job Preset record must have only one of <b><code>PresetId</code></b> , <b><code>UriLocation</code></b> , or <b><code>PresetXmlBase64Data</code></b>
<code>PresetXmlBase64Data</code>	String	No	Base-64 encoded preset data
<code>UriLocation</code>	String	No	URI to a preset. Presets may be referenced by <b><code>http://</code></b> or <b><code>file://</code></b>
<code>WatchFolderId</code>	Guid	No	Unique ID of a watch folder this preset is associated with

### Response Data Fields

Name	Data Type	Required	Description
<code>JobPreset</code>	Element	No	Contains data for an individual job preset (XML)
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>JobPresetId</code>	Guid	Private	Unique Id of this Job Preset record
<code>PresetId</code>	Guid	No	ID of a preset from the preset library. A Job Preset record must have only one of <b><code>PresetId</code></b> , <b><code>UriLocation</code></b> , or <b><code>PresetXmlBase64Data</code></b>
<code>PresetXmlBase64Data</code>	String	No	Base-64 encoded preset data
<code>UriLocation</code>	String	No	URI to a preset. Presets may be referenced by <b><code>http://</code></b> or <b><code>file://</code></b>
<code>WatchFolderId</code>	Guid	No	Unique ID of a watch folder this preset is associated with

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -d "
```

```
<JobPreset>
```

```
  <PresetId>PresetGUID</PresetId>
```

```
  <WatchFolderId>WatchFolderID</WatchFolderId>
```

```
</JobPreset>" -X POST http://<host>/api/watchfolders/<watchfolderId>/presets/
```

## Update Watch Folder Preset

Allows the user to change the preset information for a watch folder. No fields are required, update is sparse. Include only the elements that are to be changed.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/presets/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	PUT
<b>Response Formats</b>	XML, JSON
<b>Help URL</b>	<code>/api/watchfolders/{folderId}/presets/{presetId}/</code>

### Common Parameters

Name	Required	Description
<b>folderId</b>	Yes	Unique ID for the watch folder
<b>presetId</b>	Yes	Unique ID for the preset

### Request Data Fields

Name	Data Type	Required	Description
<code>JobPreset</code>	Element	No	Contains data for an individual job preset (XML)
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>JobPresetId</code>	Guid	Private	Unique Id of this Job Preset record
<code>PresetId</code>	Guid	No	ID of a preset from the preset library. A Job Preset record must have only one of <b>PresetId</b> , <b>UriLocation</b> , or <b>PresetXmlBase64Data</b>
<code>PresetXmlBase64Data</code>	String	No	Base-64 encoded preset data
<code>UriLocation</code>	String	No	URI to a preset. Presets may referenced by <b>http://</b> or <b>file://</b>
<code>WatchFolderId</code>	Guid	No	Unique ID of a watch folder this preset is associated with

## Response Data Fields

Name	Data Type	Required	Description
<code>JobPreset</code>	Element	No	Contains data for an individual job preset (XML)
<code>JobId</code>	Guid	Private	Not used with watch folders
<code>JobPresetId</code>	Guid	Private	Unique Id of this Job Preset record
<code>PresetId</code>	Guid	No	ID of a preset from the preset library. A Job Preset record must have only one of <b><code>PresetId</code></b> , <b><code>UriLocation</code></b> , or <b><code>PresetXmlBase64Data</code></b>
<code>PresetXmlBase64Data</code>	String	No	Base-64 encoded preset data
<code>UriLocation</code>	String	No	URI to a preset. Presets may referenced by <b><code>http://</code></b> or <b><code>file://</code></b>
<code>WatchFolderId</code>	Guid	No	Unique ID of a watch folder this preset is associated with

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -H "Content-Type: application/xml" -d "
```

```
<JobPreset>
  <PresetId>PrestGUID</PresetId>
  <PresetXmlBase64Data>Data </PresetXmlBase64Data>
  <WatchFolderId>d6a3c54e-e0cf-482f-91f9-c6b0cf077c28</WatchFolderId>
</JobPreset>" -X PUT http://<host>/api/watchfolders/<watchfolderId>/presets/<JobPresetId>/
```

## Delete Watch Folder Preset

Deletes the specified preset from the specified watch folder.

<b>Base URI</b>	<code>/api/watchfolders/{folderId}/presets/{presetId}/</code>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	DELETE



<b>Response Formats</b>	None
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/DeleteWatchFolderPreset">/api/watchfolders/help/operations/DeleteWatchFolderPreset</a>

### Common Parameters

Name and Description	Required	Value and Notes
<b>folderId</b>	Yes	Unique ID for the watch folder
<b>presetId</b>	Yes	Unique ID for the preset

### Request Data Fields

None

### Response Data Fields

None

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X DELETE http://<host>/api/watchfolders/<watchfolderGUID>/presets/<JobPresetIdGUID/
```

## Delete Watch Folder

Deletes a watch folder with the specified unique watch folder ID.

<b>Base URI</b>	<a href="/api/watchfolders/{folderId}/">/api/watchfolders/{folderId}/</a>
<b>Protocol</b>	HTTP
<b>HTTP Method</b>	DELETE
<b>Response Formats</b>	None
<b>Help URL</b>	<a href="/api/watchfolders/help/operations/DeleteWatchFolder">/api/watchfolders/help/operations/DeleteWatchFolder</a>

### Common Parameters

Name	Required	Description
------	----------	-------------

<b>folderId</b>	Yes	Unique ID of the watch folder to update.
-----------------	-----	--

### Request Data Fields

None

### Response Data Fields

None

### Example cURL call:

```
curl -H "Accept: application/xml" -H "x-Authorization: Basic YWRtaW46YWRtaW4=" -X DELETE -d ""  
http://<host>/api/watchfolders/<watchfolder_GUID>/
```

## Common Data Elements

Many of the Squeeze Server API endpoints use common data structures in requests and responses. For the sake of brevity, these common elements have been extracted from the Data Request Fields and Data Response Fields above and described below.

### StatusInfo

**StatusInfo** contains individual status records for the lifetime of a job. **StatusInfo** is contained in a **StatusList** element.

Name	Type	Required	Description
<b>StatusInfo</b>	Element	Yes	Root element of the <b>StatusInfo</b> record (XML)
<b>Created</b>	DateTime	yes	Date and time (GMT) the status record was created
<b>DestinationName</b>	String	No	Name of the destination associated with the status record
<b>Duration</b>	TimeSpan	No	Total duration for the status record's operation (time to complete)
<b>ErrorCode</b>	String	No	Any error code that was generated during processing
<b>Id</b>	Guid	Yes	Unique ID for the status record
<b>Modified</b>	DateTime	No	Date and time (GMT) that the status record changed
<b>PresetName</b>	String	No	Name of any preset associated with the status record
<b>Progress</b>	Float	No	Percent progress for the status record
<b>RetryCount</b>	String	No	Number of times that the operation associated with the status record has been retried
<b>Status</b>	Enumeration	Yes	Any of the following: 0: <b>Undefined</b> (not yet set) 1: <b>Waiting</b> 2: <b>Downloading</b> 3: <b>Compressing</b> 4: <b>Uploading</b> 5: <b>Finished</b> 6: <b>Error</b> 7: <b>Canceled</b> 8: <b>Deleted</b> 9: <b>Hold</b> 10: <b>Incomplete</b>

<code>StatusMessage</code>	String	No	System-generated informational message
<code>TimeFinished</code>	DateTime	No	Date and time (GMT) that the process completed
<code>TimeStarted</code>	DateTime	Yes	Date and time (GMT) that the process started

## JobMediaInfo

**JobMediaInfo** contains a list of presets the job will use to encode a source file, a list of destinations the job will publish output files to, and a description of the source file to encode.

Name	Data Type	Required	Description
<code>JobMediaInfo</code>	Element	Yes	Root element of the <b>JobMediaInfo</b> structure
<code>CompressionPresetList</code>	Element	Yes	Element containing the list of compression presets used by a job. A job must have at least one preset. A watch folder can have zero or more.
<code>CompressionPreset</code>	Element	Yes if Job No if Watch Folder	Element containing a compression preset information for a job or watch folder. See <a href="#">CompressionPreset</a>
<code>DestinationList</code>	Element	Yes	List of destinations to which Squeeze Server will publish outputs for a job. A job or watch folder can have more than one destination.
<code>DestinationInfo</code>	Element	Yes	Container for the elements that describe a publishing destination. See <a href="#">DestinationInfo</a>
<code>SourceMedia</code>	Element	Yes	Container for the elements that describe a job source file or watch folder source location. See <a href="#">SourceMedia</a>

## CompressionPreset

**CompressionPreset** contains the elements that describe a compression preset for a job or watch folder. A preset record must reference one and only one of a preset by URI, by ID for the preset library, or contain the actual preset data.

Name	Type	Required	Description
<code>CompressionPreset</code>	Element	Yes	Root element of a compression preset
<code>JobId</code>	Guid	Yes if job	ID of the job the preset is associated with, or empty for watch folders
<code>JobPresetId</code>	Guid	No	Unique ID of this preset record
<code>PresetId</code>	Guid	No	If of a preset from the Squeeze Server preset library, if any
<code>PresetXmlBase64Data</code>	String	No	Base-64 data for a preset submitted with a job or watch folder
<code>UriLocation</code>	String	No	URI of a preset stored on the network or Internet
<code>WatchFolderId</code>	Guid	Yes if watch folder	ID of the watch folder to which the preset is associated or empty for a preset associated with a job

## DestinationInfo

**DestinationInfo** is an element that contains data describing a job publishing destination.

Name	Data Type	Required	Description
<code>DestinationInfo</code>	Element	Yes	Root element for the <b>DestinationInfo</b> record
<code>Created</code>	DateTime	Private	System-supplied date and time (GMT) when the destination record was created
<code>CredentialId</code>	Guid	No	ID of stored credentials needed to access the destination
<code>DestinationId</code>	Guid	Private	Unique ID for the destination, supplied by the system
<code>DestinationMetadataList</code>	Element	Yes	List of destination metadata. This is currently only used for YouTube destinations.

<code>DestinationMetadataInfo</code>	Element	Yes	Container for the elements that describe a destination metadata. See <a href="#">DestinationMetadataInfo</a>
<code>ExtensionNamingMethod</code>	Enumerated String	Yes	<p>May be any of the following:</p> <ul style="list-style-type: none"> <li>• <b>Default</b> - Obey system setting</li> <li>• <b>Preserve</b> - Preserve user-supplied extension</li> <li>• <b>Auto</b> - Automatically supply extension based on output file type.</li> </ul> <p>See “Output File Naming Methods” in the Squeeze Server – Operation guide for more information</p>
<code>FileName</code>	String	No	File name prefix or token string or output filename. See “Output File Naming Methods” in the Squeeze Server – Operation guide for more information
<code>FileNamingMethod</code>	Enumerated String	yes	<p>May be any of the following:</p> <ul style="list-style-type: none"> <li>• <b>Default</b> - Obey system settings</li> <li>• <b>Preserve</b> - Preserve user-supplied output file name</li> <li>• <b>Unique</b> - Generate unique file name</li> <li>• <b>Wildcard</b> - The FileName supplied contains wildcard tokens that should be replaced at encode time.</li> </ul> <p>See “Output File Naming Methods” in the Squeeze Server – Operation guide for more information.</p>
<code>FileUri</code>	String	Yes	May be <code>file://</code> , <code>sftp://</code> , <code>s3://</code> , <code>azureblob://</code> or <code>ftp://</code>
<code>JobId</code>	Guid	No	Job ID for the job with which the destination is associated
<code>Modified</code>	DateTime	No	Date and time (GMT) the destination was last modified
<code>Password</code>	String	No	Password, if required and when not using the CredentialId for accessing the destination

<code>S3BucketName</code>	String	Obsolete	Name of the S3 bucket for output files
<code>S3ThumbBucket</code>	String	Obsolete	Name of the thumbnail bucket if generating thumbnails
<code>ThumbFilePattern</code>	String	No	If generating thumbnails, this is the filename pattern for the thumbnails of the format <code>filename%VERSION%.jpg</code> where <code>%VERSION%</code> will represent the resolution of the produced thumbnail. Your system administrator defines these resolutions.
<code>ThumbUri</code>	String	No	Full URI with the supported protocols <code>file://</code> , <code>sftp://</code> , <code>s3://</code> , <code>azureblob://</code> , <code>ftp://</code> , or <code>youtube://</code> ) for the thumbnails following the thumbnail file pattern. The thumbnail URI will use the same credentials as the output file destination.
<code>Type</code>	Enumeration	Obsolete	May be S3, URI
<code>Username</code>	String	No	Username is required to access the destination if not using the <code>CredentialId</code> .
<code>WatchFolderId</code>	Guid	No	If of the watch folder using the destination if the destination is associated with a watch folder, otherwise this is empty.

## DestinationMetadataInfo

**DestinationMetadataInfo** is an element that contains metadata associated with a job publishing destination.

*This element type is currently only used for YouTube destinations. See [Defining YouTube Destinations](#) for more information.*

Name	Data Type	Required	Description
<code>DestinationMetadataInfo</code>	Element	Yes	Root element for the <b>DestinationMetadataInfo</b> record
<code>DestinationId</code>	Guid	No	ID of stored destination needed to access the destination metadatum
<code>MetadataId</code>	Guid	Private	Unique ID for the destination metadatum, supplied by the system

Name	String	Yes	Name of the destination metadatum. This must be one of the valid metadata names. See <a href="#">Defining YouTube Destinations</a> .
Value	String	Yes	Value for the destination metadatum

## SourceMedia

**SourceMedia** is an element that contains data describing the source media for a Job or Watch Folder.

Name	Data Type	Required	Description
<b>SourceMedia</b>	Element	Yes	Root element of a <b>SourceMedia</b> record
Created	DateTime	No	System-supplied date and time (GMT) when the source record was created
CredentialId	Guid	No	ID of stored credentials, if needed to access the source media
FileName	String	Yes if source is part of a job. No if source is a watch folder	File name to encode, or if a watch folder, may be an S3 artificial folder prefix or empty
FileSize	Long	No	Optional but recommended file size of the source file in bytes
FileUri	String	No	URI with protocol <b>file://</b> , <b>http://</b> , <b>sftp://</b> , <b>s3://</b> , <b>azureblob://</b> or <b>ftp://</b> ), to the file or location for a set of files for watch folders
JobId	Guid	No	Unique ID for the job with which the source is associated, or empty for watch folders
Modified	DateTime	No	System-supplied date and time (GMT) the source record was last modified
Password	String	No	Password if required and not using <b>CredentialId</b> to access the source media
S3bucketName	String	Obsolete	Name of the S3 bucket for the source if



---

			using S3
<code>SourceId</code>	Guid	No	System-supplied unique ID for the source record.
<code>Type</code>	Enumerated String	Obsolete	May be "S3" or "URI"
<code>UserName</code>	String	No	Username if required and not using <code>CredentialId</code> to access the source media.
<code>WatchFolderId</code>	Guid	No	Unique ID of the watch folder with which the source is associated or empty if not associated with a watch folder
<code>WatermarkInfo</code>	Element	No	Container for watermark source image file information
<code>WatermarkImageUri</code>	String	No	URI to the watermark image used if a preset contains watermark information. The image source location should be the same URI scheme as the source file and will use the same credentials or username/password as the source media file. This image will replace the image in any watermark preset in the job or watch folder and will have no effect for presets that do not contain the watermark filter.

---

## MetadataInfo

**MetadataInfo** contained in a **MetadataList** element is an optional XML element containing elements that provide metadata name/value pairs to a job. Metadata will be supported in a future release of Squeeze Server.

Name	Data Type	Required	Description
<a href="#">MetadataList</a>	Element	Yes	Element containing name/value pairs of metadata to be passed to the encoder
<a href="#">Created</a>	DateTime	No	System-supplied date and time (GMT) the metadata record was created
<a href="#">JobId</a>	Guid	No	ID of the job with which the metadata is associated.
<a href="#">MetadataId</a>	Guid	No	System-supplied unique ID for the metadata record
<a href="#">MetadataName</a>	String	Yes	Name for the metadata item
<a href="#">MetadataValue</a>	String	Yes	Name for the metadata value associated with the name
<a href="#">Modified</a>	DateTime	No	System-supplied date and time the metadata record was last modified

## JobStatusInfo

JobStatusInfo contains status about a job in a more concise summarized form than is provided in a job.

Name	Type	Required	Description
<a href="#">JobStatusInfo</a>	Element	Yes	Element containing status information for a single job.
<a href="#">EncodeServerId</a>	string	No	ID of the server that encoded or is encoding a job.
<a href="#">JobId</a>	Guid	No	Unique ID for the job, or empty if a watch folder
<a href="#">LastUpdate</a>	DateTime	No	Date and time (GMT) when the job was last updated.
<a href="#">Name</a>	String	Yes	User-supplied friendly name for the job
<a href="#">QueueId</a>	Guid	Yes	Unique ID of the queue to which the job has been assigned
<a href="#">Status</a>	Element	Yes	Contains a consolidated and summarized status

				for the job
<code>Created</code>	DateTime	Yes	Date and time (GMT) that the status record was created	
<code>DestinationName</code>	String	No	User-friendly name for the job's destination. (Not used in this context)	
<code>Duration</code>	DateTime	No	Total time the job has been processing or has been processed	
<code>ErrorCode</code>	String	No	Last error encountered during job processing, if any	
<code>Id</code>	Guid	No	Unique ID for the status record. Usually not used in this context	
<code>Modified</code>	DateTime	No	Date and time (GMT) that the status was last changed	
<code>PresetName</code>	String	No	The Preset name associated with this status record. <code>PresetName</code> is not used in this context.	
<code>Progress</code>	Float	No	Overall job progress percent	
<code>RetryCount</code>	Int	No	Number of retries this job has attempted	
<code>Status</code>	Enumeration	Yes	Current status for the job. May be one of the following: 0: <code>Undefined</code> (not yet set) 1: <code>Waiting</code> 2: <code>Downloading</code> 3: <code>Compressing</code> 4: <code>Uploading</code> 5: <code>Finished</code> 6: <code>Error</code> 7: <code>Canceled</code> 8: <code>Deleted</code> 9: <code>Hold</code> 10: <code>Incomplete</code>	
<code>StatusMessage</code>	String	No	Last status message generated during job processing	
<code>TimeFinished</code>	DateTime	No	Date and time (GMT) the job finished processing	
<code>TimeStarted</code>	DateTime	No	Date and time the job started processing	
<code>TimeSubmitted</code>	DateTime	Yes	Date and time (GMT) when the job was submitted for processing	

---

<code>TimeStarted</code>	DateTime	No	Date and time (GMT) when the job started processing
--------------------------	----------	----	---

---

## Appendix

### Defining YouTube Destinations

When defining a YouTube destination, it is necessary to include a `DestinationMetadataList` element containing the appropriate `DestinationMetadataInfo` elements. Below is a table showing the allowed values for the name attribute for `DestinationMetadataInfo` elements as well as a description of what the value indicates.

---

Name	Data Type	Description
<code>youtube_allow_syndicate</code>	Boolean	Indicates whether to make this video available on mobile phones and televisions.
<code>youtube_privacy</code>	Enumerated String	Specifies privacy level for the video. May be any of the following: <ul style="list-style-type: none"> <li>• <code>youtube_privacy_public</code> – Anyone can search and view</li> <li>• <code>youtube_privacy_unlisted</code> – Anyone with the link can view</li> <li>• <code>youtube_privacy_private</code> – Only specific YouTube users can view</li> </ul>
<code>youtube_allow_embed</code>	Boolean	Indicates whether external sites may embed the video.
<code>youtube_allow_rating</code>	Boolean	Indicates whether to allow this video to be rated by others.
<code>youtube_allow_comments</code>	Enumerated String	Specifies which users may comment on the video. May be any of the following: <ul style="list-style-type: none"> <li>• <code>youtube_allow_comments_auto</code> – Allow comments to all</li> <li>• <code>youtube_allow_comments_with_approval</code> – Allow comments with approval only</li> <li>• <code>youtube_allow_comments_do_not_allow</code> – Do not allow comments</li> </ul>
<code>youtube_allow_vote_on_comments</code>	Boolean	Indicates whether to allow users to vote on comments.
<code>youtube_description</code>	String	Description of the video.

---

---

<code>youtube_keywords</code>	String	A comma-separated list of tags to apply to the video. Example: "snorkeling, surf, beach"
<code>youtube_allow_response</code>	Enumerated String	Specifies when to allow video responses. May be any of the following: <ul style="list-style-type: none"><li>• <code>youtube_allow_response_auto</code> – Allow video responses to all</li><li>• <code>youtube_allow_response_with_approval</code> – Allow video responses with approval only</li><li>• <code>youtube_allow_response_do_not_allow</code> – Do not allow video responses</li></ul>
<code>youtube_category</code>	Enumerated String	The category for the video. May be any one of the following: <ul style="list-style-type: none"><li>• <b>Autos</b> – Autos &amp; Vehicles</li><li>• <b>Comedy</b> – Comedy</li><li>• <b>Education</b> – Education</li><li>• <b>Entertainment</b> – Entertainment</li><li>• <b>Film</b> – Film &amp; Animation</li><li>• <b>Games</b> – Gaming</li><li>• <b>Howto</b> – Howto &amp; Style</li><li>• <b>Music</b> – Music</li><li>• <b>News</b> – News &amp; Politics</li><li>• <b>Nonprofit</b> – Nonprofits &amp; Activism</li><li>• <b>People</b> – People &amp; Blogs</li><li>• <b>Animals</b> – Pets &amp; Animals</li><li>• <b>Tech</b> – Science &amp; Technology</li><li>• <b>Sports</b> – Sports</li><li>• <b>Travel</b> – Travel &amp; Events</li></ul>

---