



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

## **Vantage REST API 5.5**

Vantage 8.1 Update Pack 2



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

## Table of Contents

Vantage REST API 5.5 .....	1
Introduction .....	3
Changes .....	5
Vantage Terminology .....	11
API Reference .....	14
Domain Operations:.....	15
Category Operations: .....	19
Workflow Operations:.....	21
Job Operations: .....	39
Catalog Operations: .....	49
Service Operations:.....	57
Machine Operations: .....	64
Storage Operations .....	65
Examples .....	69
Submit a Job.....	69
Obtain Job Outputs .....	75
Submit In-Band Material .....	81
Exporting and Importing Workflows .....	85
Submitting Remote Files .....	88
Requesting Versioned Assets from S3.....	93



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

## Introduction

The Vantage REST API is a traditional REST (Representational State Transfer) implementation which uses HTTP as the underlying transfer protocol.

Interaction with the Vantage REST API is performed via a combination of *URIs* and *Methods*. The *URI* typically refers to the Vantage entity or component that is being manipulated. For example: `/Rest/Workflows` refers to the URI which provides information about Workflows in the current domain.

The *Method* corresponds to an HTTP Verb which represents the underlying action that is being performed on the API. The HTTP verbs: *GET*, *PUT*, *POST* and *DELETE* are utilized in conjunction with a URI to indicate the action that the underlying message is performing.

- *GETs* are typically used when obtaining or requesting information about a Vantage component from the API.
- *PUTs* are generally used when modifying or updating a Vantage component.
- *POSTs* are generally utilized when creating new Vantage components.
- *DELETE* is used to indicate the removal of components from Vantage.

As a reference, the following URL can always be utilized to obtain a list of the available API operations along with the URI, Method and a description of the operation:

<http://localhost:8676/Rest/help>

Note: *localhost* is used above; this assumes the Vantage SDK service is running on the local server. If the Vantage SDK service is running on a different server, use the name or IP address of that server in the above URL.

An example of the output that is presented when the help URL is entered into a browser is:



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Operations at <http://localhost:8676/Rest>

This page describes the service operations at this endpoint.

Uri	Method	Description
/Binder/{id}	GET	Gets a specific Binder for the domain.
/Binder/{id}/Content	GET	Gets the content for a specific Binder.
/Catalogs	GET	Gets all of the Catalog(s) for the domain.
/Catalogs/{id}	GET	Gets a specific catalog for the domain.
/Category/{name}/Create	PUT	Create a category with the specified name.
/Category/{name}/Exists	GET	Returns an indication of whether the Category with the provided name exists in this domain.
/Domain/Categories	GET	Gets all categories from this domain.
/Domain/Load/{criteria}	GET	Gets the load for the Vantage domain based upon the provided criteria, eg: /Domain/Load/CPU
/Domain/Online	GET	Returns an indication as to whether the SDK service has successfully attached to the domain database.
/Domain/Version	GET	Returns a string representation of the version of the current Vantage domain (eg: 7.0.0.0).
/Folder/{id}	GET	Gets a specific Folder for the domain.
/Folders/{id}	GET	Gets all of the Folder(s) for a given Catalog or Folder.
/Folders/{id}/Binder/{name}	GET	Gets a Binder in a specific Catalog or Folder by Binder Name.
/Folders/{id}/Binders	GET	Gets all of the Binder(s) for a given Catalog or Folder.
/Jobs/{id}	GET	Returns information about the job with the provided identifier.
/Jobs/{id}/Actions	GET	Return all actions associated with a specific job.
/Jobs/{id}/ErrorMessage	GET	Returns the relevant error message associated with a job which has failed.
/Jobs/{id}/Forwarded	GET	Returns a collection of jobs which were created as a result of being forwarded from the specified job.
/Jobs/{id}/Metrics	GET	Return specific metrics (eg: total run time, total queue time) about a specified job.
/Jobs/{id}/Outputs	GET	Return all outputs (Media Nicknames, Attachments and/or Labels) associated with a specific job.
/Jobs/{id}/Progress	GET	Returns the progress (percent complete) for the specified job.
/Jobs/{id}/Remove	DELETE	Removes a completed job with the specified identifier.
/Jobs/{id}/Restart	PUT	Restarts a job which has previously been stopped. Only a job that is in the Stopped By User state may be restarted.
/Jobs/{id}/Stop	PUT	Stops the job identified by the provided identifier.
/Workflows	GET	Gets all of the vantage workflows for this domain.
/Workflows/{id}	GET	Gets the workflow with the specified identifier from this domain.
/Workflows/{id}/Activate	PUT	Activate the specified workflow.
/Workflows/{id}/Deactivate	PUT	Deactivate the specified workflow.
/Workflows/{id}/Export	GET	Obtain a compressed representation of the specified workflow.
/Workflows/{id}/JobInputs	GET	Gets the inputs which may be submitted to the workflow with the specified identifier.
/Workflows/{id}/Jobs/	GET	Return all jobs for a specified workflow which match the specified filter criteria. The valid values for the filter query parameter are: 'All', 'Active', 'Failed', 'Paused' or 'Complete'
/Workflows/{id}/Rename	PUT	Renames the specified workflow to a new name.
/Workflows/{id}/Signature	GET	Obtain a value representing a unique signature for a workflow.
/Workflows/{id}/Submit	POST	Submit a job to a workflow. The request data should be obtained via the /Workflows/{ID}/JobInputs GET query.
/Workflows/{id}/SubmitBinder	POST	Submit a Binder to a workflow.
/Workflows/Import	PUT	Import a workflow which had been previously exported from a Domain.

*This URL will always list the available list of operations that can be performed via the REST API. This list of operations will update/change as functionality is added or updated.*



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

## Changes

The following changes were added between the 5.5 and 5.4 versions of the SDK:

A new endpoint was introduced to allow an API client to request whether a given Vantage Store was currently in use.

The new endpoint is:

<http://localhost:8676/Rest/Stores/<Store ID>/InUse>

This call will return information about a Vantage Store, including a value which indicates whether Vantage believes the Store is in-use.

For example, if the call is made with the Storage ID of: 13060a84-344a-40cf-9f75-b106b563cc25, ie:

<http://localhost:8676/Rest/Stores/13060a84-344a-40cf-9f75-b106b563cc25/InUse>

and that store was currently in use by Vantage, the following would be returned:

```
{
2  "Store": {
3    "Identifier": "13060a84-344a-40cf-9f75-b106b563cc25",
4    "Aliases": [],
5    "AvailableSpace": 942331715584,
6    "Location": "file:///C:/VantageStore/",
7    "Name": "Local Store",
8    "Offline": false,
9    "StorageMode": 0,
10   "TotalSpace": 999677751296
11  },
12  "StoreInUse": true
13}
```

The new field is highlighted above.

The endpoint: <http://localhost:8676/Rest/Folders/{FOLDERID}/Binders> was updated to include an optional start and end query parameter. These parameters constrain the results that are returned to only binders created after the provided start or before the provided end.

The original query:

<http://localhost:8676/Rest/Folders/{FOLDERID}/Binders>

will return all binders in the folder whose id is: {FOLDERID}, eg:

<http://localhost:8676/Rest/Folders/b364aeed-7089-4821-a896-ed639cb5a021/Binders>

The endpoint now supports the optional query parameters: start and end you can specify something like:

<http://localhost:8676/Rest/Folders/b364aeed-7089-4821-a896-ed639cb5a021/Binders?start=2021-11-1>



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

and this will return all BINDERS in the folder with ID: b364aeed-7089-4821-a896-ed639cb5a021 created after 2021-11-1

if you specify a start query parameter, you can also optionally specify an end query parameter:

<http://localhost:8676/Rest/Folders/b364aeed-7089-4821-a896-ed639cb5a021/Binders?start=2021-11-1&end=2021-11-4>

the above would return all binders in the folder with ID: b364aeed-7089-4821-a896-ed639cb5a021 created AFTER 2021-11-1 AND BEFORE 2021-11-4

The following changes were added between the 5.4 and 5.3 versions of the SDK:

A new call was introduced to allow information about managed Stores defined within the current Vantage domain.

The new call is:

```
http://localhost:8676/Rest/Stores
```

The result of this call will contain information about all managed stores in the Domain. NOTE: Only managed stores (vs. Address Book entries) are returned. More information is included in the details section of the API guide (see below).

A new entry was added to the Label definition that is obtained whenever the outputs of a Job are requested.

The new entry shows the Category name for which the Parameter belongs.

An example of this is illustrated below:

```
{"Identifier":"35814967-6dd5-4809-bde8-65c865edcf14", "Category":"Category1", "DefaultValue":null, "Description":null, "Increment":null, "Maximum":null, "Minimum":null, "Name":"Param2", "Options":[], "TypeCode":"String", "Value":"Val1"}
```

Where the highlighted entry above corresponds to the new entry. The information above is generated when the job output request is made:

```
http://localhost:8676/Rest/Jobs/{ID}/outputs
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

A new call was added to allow the cleanup of Folders from the Catalog which are marked as Transient and are empty. This call is a programmatic way to invoke the same functionality that appears in the VMC->Catalogs pane.

The new call is a POST request to:

<http://localhost:8676/Rest/Catalogs/Cleanup>

The following changes were added between the 5.3 and 5.2 versions of the SDK:

The results of the GetWorkflow call (<http://localhost:8676/Rest/Workflows/{ID}>) now include additional fields which detail the expiration time and the last modified date. This information is also present in the call to retrieve all workflows (<http://localhost:8676/Rest/Workflows>).

An example of the new fields is displayed below:

```
{
  "Identifier": "16b99faa-e31e-4637-893d-093efa77f446",
  "Description": "",
  "ExpirationInHours": 24,
  "Name": "SampleWorkflow",
  "State": 2,
  "UTC_ModifiedTime": "2020-12-07T22:40:44.2700000Z"
}
```

There were no specific changes to the SDK/API between the 5.2 and the 5.1 revision. A section regarding the use of version designator when requesting assets from Amazon S3 storage was added.

There following changes were added between the 5.1 and the 5.0 version of the SDK:

The result of the GetBinderContent call (<http://localhost:8676/Rest/Binder/{ID}/Content>) was updated to include a identifier field for the Content entry associated with a Binder. The following represents the new JSON response:

```
{
  "ContentIdentifier": "1627aea5-8e0a-4371-9022-9b504344e724",
  "Created": "\/Date(928164000000-0400) \/",
  "Name": "String content",
  "Updated": "\/Date(928164000000-0400) \/"
}
```

Where the 'ContentIdentifier' is new.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

A call to retrieve all entries (Media Nicknames, Labels, Attachments) for a particular Content entry was added: <http://localhost:8676/Rest/Content/{ID}/Entries>

A call was added to facilitate looking up information about the services which host actions of a specific type: <http://localhost:8676/Rest/Services/LookupHosting?Type={ACTIONTYPE}>

There were no material changes between the 5.0 and the 4.0 versions.

The following changes were added between the 4.0 version and the 3.0 version (Released with Vantage 7.1):

The following API methods were added which deal with the ability to import/export and synchronize workflows from a location specified by a windows path:

<http://localhost:8676/Rest/Workflows/Synchronize> : Synchronizes a workflow with a new definition from a specified file path.

<http://localhost:8676/Rest/Workflows/ImportFromPath> : Imports a workflow definition from the specified filepath.

<http://localhost:8676/Rest/Workflows/ExportToPath> : Export an existing workflow definition to the specified windows path location.

The following changes were added between the 2.0 version (Vantage 7.0 Update Pack 1) and the 3.0 version (released with Vantage 7.1):

Workflow Signature computation – a new method was included to allow the signature for a workflow to be computed and returned to the client. This signature represents a value that corresponds to the exact structure/configuration at the current time. So long as the workflow remains un-modified (i.e.: no actions removed/added/updated or new connections made or other changes of this nature); the signature shall remain constant. As soon as the workflow changes, the associated signature shall change.

**HTTP Method:** Get

<http://localhost:8676/Rest/Workflows/{ID}/Signature>

The ability to request additional information about Categories within the Vantage domain was added.

In particular, the ability to query whether a Category with a specific name exists in the domain or not was added:

**HTTP Method:** Get

<http://localhost:8676/Rest/Category/{NAME}/Exists>





848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Additionally, the ability to create a specific category was added.

**HTTP Method:** Put

<http://localhost:8676/Rest/Category/{NAME}/Create>

The ability to request information about Vantage services was introduced.

**HTTP Method:** Get

<http://localhost:8676/Rest/Services>

<http://localhost:8676/Rest/Services/{ID}/Metrics>

<http://localhost:8676/Rest/Services/SearchByMachine/{Machine}>

**HTTP Method:** POST

<http://localhost:8676/Rest/Services/{ID}/Preshutdown>

The ability to request information about machines which host services in a Vantage domain was introduced:

**HTTP Method:** GET

<http://localhost:8676/Rest/Machines>

HTTP Method: POST

<http://localhost:8676/Rest/Machines/{ID}/Preshutdown>

Please see below for more information about these new methods.

### *Previous Changes:*

The following changes were added between 1.0 (the initial release in Vantage 7.0 Update Pack 1) and version 2.0:

Introduce Catalog manipulation. Specific methods added for the ability to interact with Vantage Catalogs:

**HTTP Method:** GET

<http://localhost:8676/Rest/Catalogs>

<http://localhost:8676/Rest/Catalogs/{ID}>

<http://localhost:8676/Rest/Folder/{ID}>

<http://localhost:8676/Rest/Folders/{ID}>

<http://localhost:8676/Rest/Folders/{ID}/Binder/{NAME}>

<http://localhost:8676/Rest/Folders/{ID}/Binders>

<http://localhost:8676/Rest/Binder/{ID}>

<http://localhost:8676/Rest/Binder/{ID}/Content>

Please see below for the details of each new call.

Add a call to the Workflows section allowing a Binder to be submitted:

**HTTP Method:** POST

<http://localhost:8676/Rest/Workflows/{ID}/SubmitBinder>

Please see below for the usage details.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Introduce a Domain online indicator:

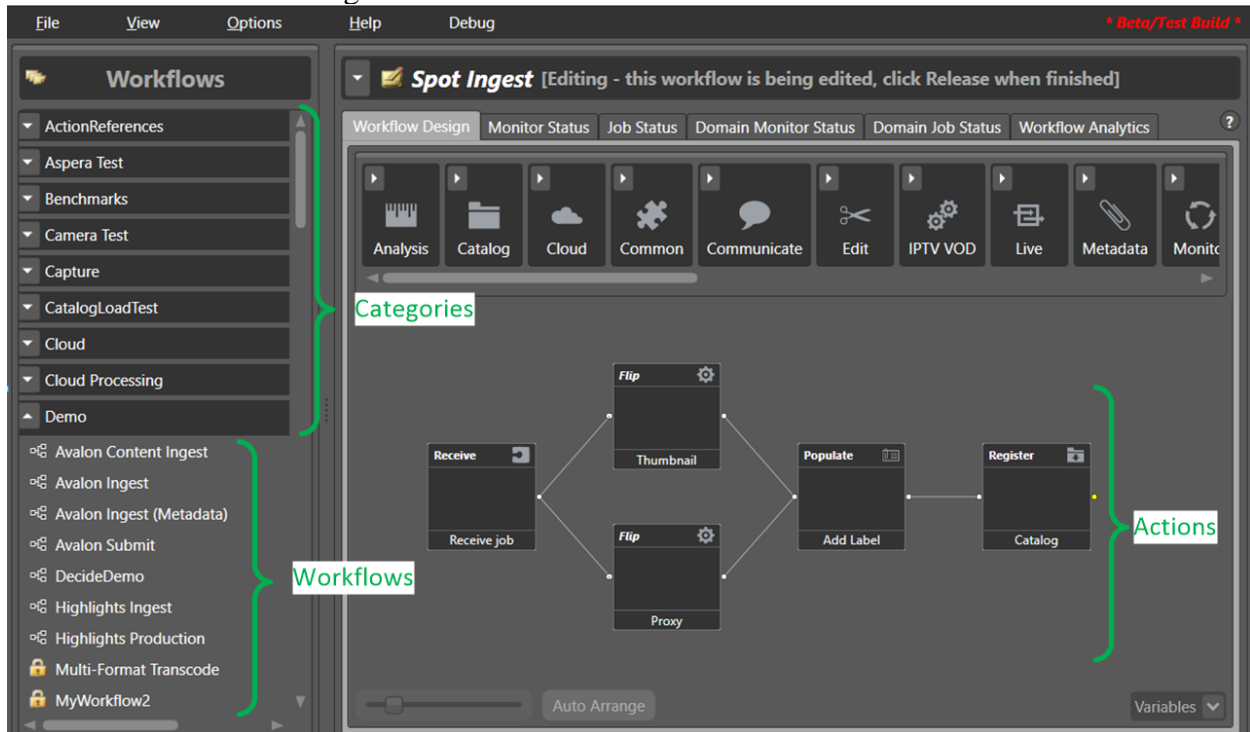
**HTTP Method:** GET

<http://localhost:8676/Rest/Domain/Online>

Please see below for the usage details.

## Vantage Terminology

This API guide will make use of several different terms from Vantage. The following is a brief overview of what Vantage components are referred to by these terms. Additional information is available within the Vantage Users Guide.



**Categories:** These represent groupings which may hold Vantage **Workflows**. In the above screenshot, the Demo category has been expanded to show the **Workflows** that it contains.

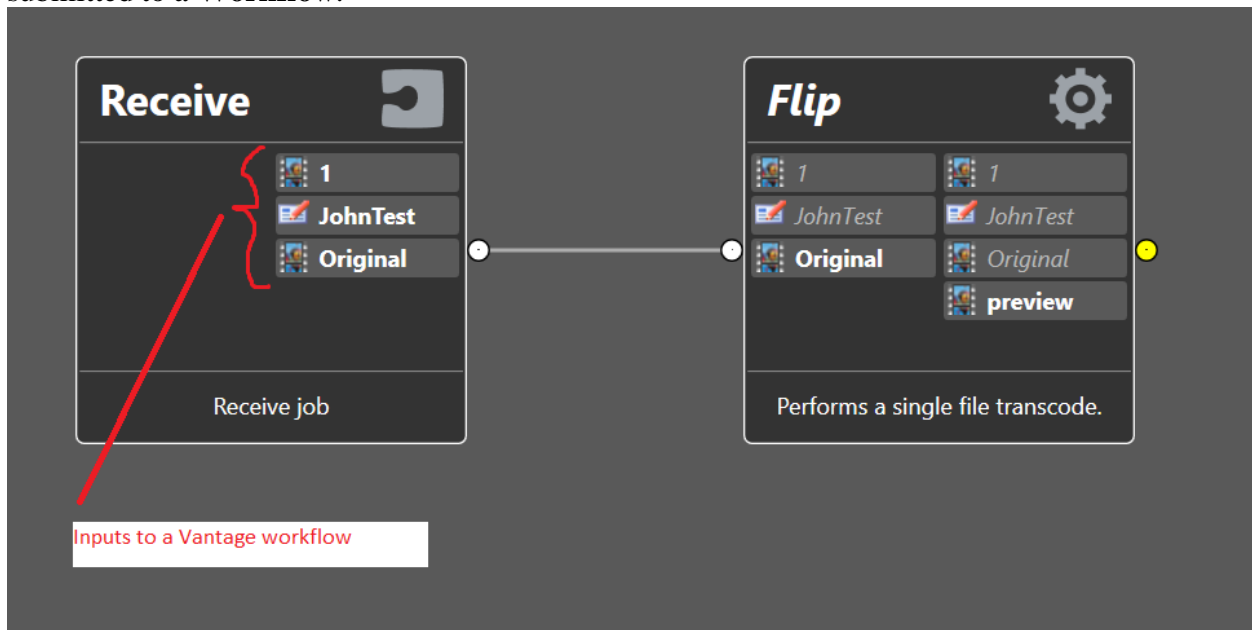
**Workflows:** The discrete set of steps (**Actions**) that a user has constructed to represent the used to process material within Vantage.

**Actions:** A single instance of a task that may be performed within a **Workflow**.

The screenshot shows the 'Spot Ingest' workflow in edit mode. The workflow design consists of six steps: Watch, Copy, Identify, Identify, Compute, and Decide. Below the design is a table of jobs:

Job	Workflow	Category	State
test111.zip	LocalCopy	GIR-12847	Complete
test2.gxf	LocalCopy	GIR-12847	Complete
test1.gxf	LocalCopy	GIR-12847	Complete
test33.mpg	LocalCopy	GIR-12847	Complete
test1.mpg	LocalCopy	GIR-12847	Complete

**Jobs:** Jobs correspond to a running instance of a **Workflow**. A job is created when material is submitted to a **Workflow**.



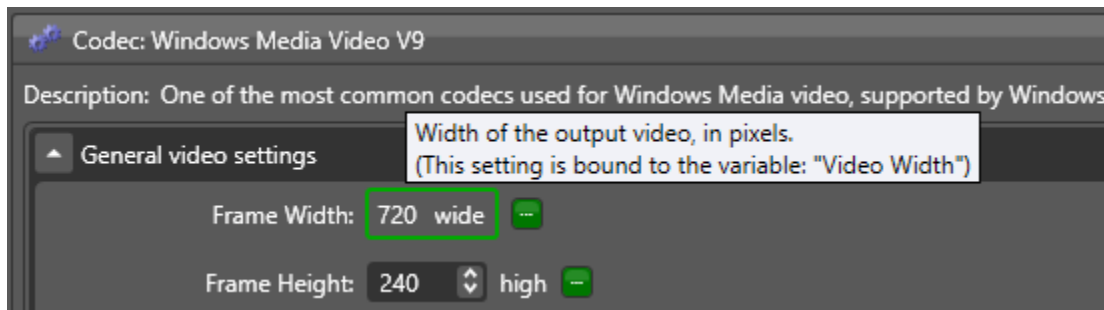
The **'Receive'** action is a special action that must initiate any workflow that is to be the target of a SDK/API submission. The Receive action indicates the **Inputs** that are required to be supplied in order for a **Job** to be submitted to this **workflow**.

## Variables


Variables are utilized within Vantage to allow values to be passed into Vantage via the API as well as passing values from one action to another.

Variables are created/defined within the Vantage Management Console (VMC). Of particular importance is the notion of Variable Type. The Type associated with a Variable defines the interaction it may have with Vantage.

The most important use of Variables is their ability to be BOUND to a parameter of a compatible Type (see below). This binding process can be performed where-ever a green binding button appears in the user interface:



In the image above, the Frame Width parameter has been bound to the Variable ‘Video Width’. The green outline around the parameter value indicates the binding is present.

The green box (  ) is what allows the binding to be defined.

Once a variable is bound to a parameter, it will be advertised via the API as a value which can be submitted as part of a job request.

Certain variable types have very specific rules about their meaning within Vantage.

In particular, Variables defined to hold a PATH have the following rules:

When utilizing a variable to hold a PATH within the API in Vantage, if the variable references a PATH to a FILE; then the string must NOT end in a slash character. If the variable references a PATH to a FOLDER; then the value within the variable MUST end in a trailing slash. If the value does not end in a trailing slash, the variable shall be treated as though it is a PATH to a FILE; where the FILE is the last fragment in the string value.

For example:

[path = //myserver/myfolder/myfile123.mpg](#) → this is a path to a file (the file being myfile123.mpg)

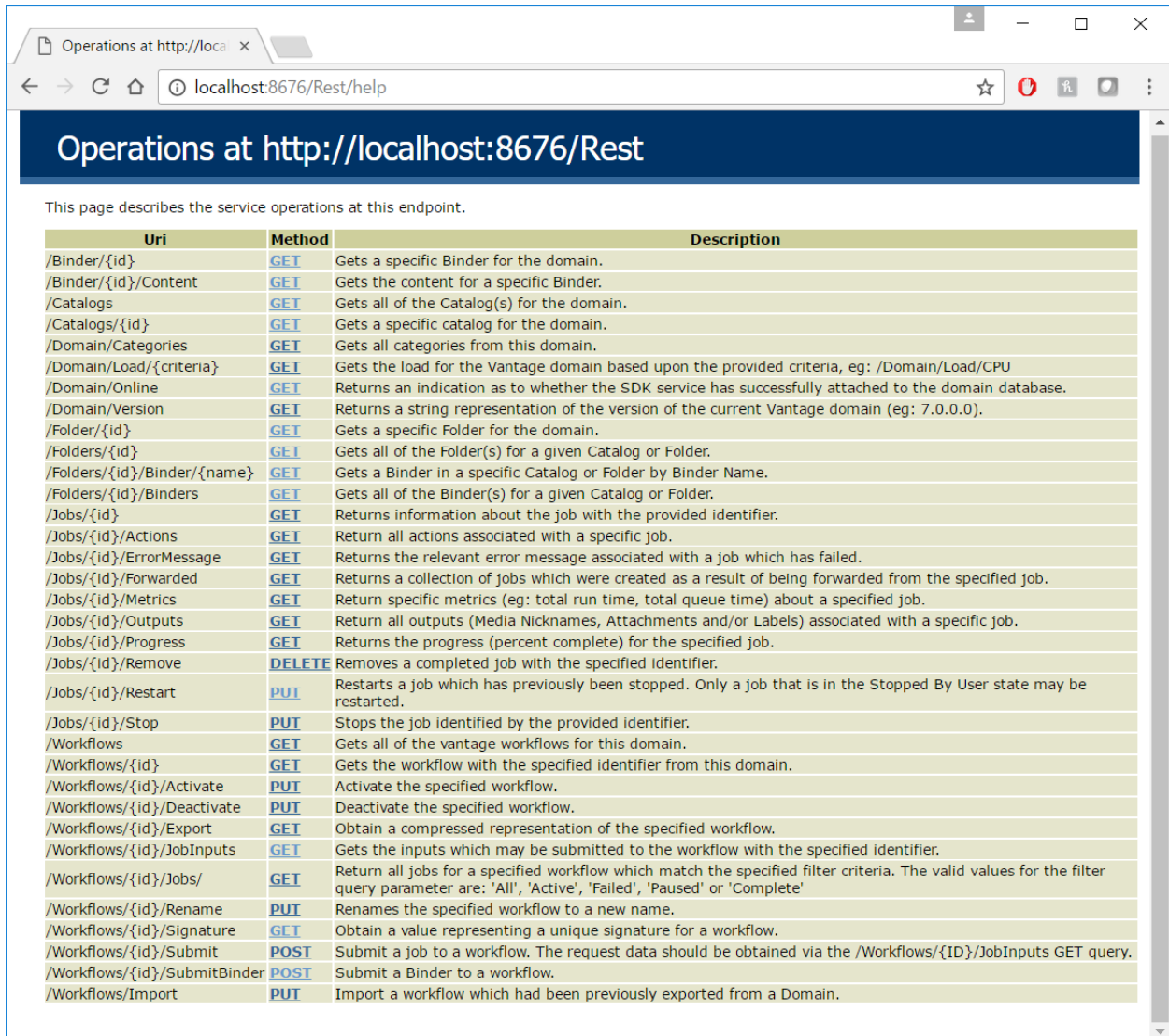
[path = //myserver/myfolder/myfile123](#) → this is a path to a file (the file being myfile123). If this is treated as a folder path, the value is interpreted as: //myserver/myfolder/

[path = //myserver/myfolder/](#) → this is a path to the folder myfolder located on the share myserver

## API Reference

The Vantage REST API uses JSON (JavaScript Object Notation) to provide a serialized context for both request and response structures to the API. The JSON structures for each operation can be accessed via the help URL:

<http://localhost:8676/Rest/help>



This page describes the service operations at this endpoint.

Uri	Method	Description
/Binder/{id}	GET	Gets a specific Binder for the domain.
/Binder/{id}/Content	GET	Gets the content for a specific Binder.
/Catalogs	GET	Gets all of the Catalog(s) for the domain.
/Catalogs/{id}	GET	Gets a specific catalog for the domain.
/Domain/Categories	GET	Gets all categories from this domain.
/Domain/Load/{criteria}	GET	Gets the load for the Vantage domain based upon the provided criteria, eg: /Domain/Load/CPU
/Domain/Online	GET	Returns an indication as to whether the SDK service has successfully attached to the domain database.
/Domain/Version	GET	Returns a string representation of the version of the current Vantage domain (eg: 7.0.0.0).
/Folder/{id}	GET	Gets a specific Folder for the domain.
/Folders/{id}	GET	Gets all of the Folder(s) for a given Catalog or Folder.
/Folders/{id}/Binder/{name}	GET	Gets a Binder in a specific Catalog or Folder by Binder Name.
/Folders/{id}/Binders	GET	Gets all of the Binder(s) for a given Catalog or Folder.
/Jobs/{id}	GET	Returns information about the job with the provided identifier.
/Jobs/{id}/Actions	GET	Return all actions associated with a specific job.
/Jobs/{id}/ErrorMessage	GET	Returns the relevant error message associated with a job which has failed.
/Jobs/{id}/Forwarded	GET	Returns a collection of jobs which were created as a result of being forwarded from the specified job.
/Jobs/{id}/Metrics	GET	Return specific metrics (eg: total run time, total queue time) about a specified job.
/Jobs/{id}/Outputs	GET	Return all outputs (Media Nicknames, Attachments and/or Labels) associated with a specific job.
/Jobs/{id}/Progress	GET	Returns the progress (percent complete) for the specified job.
/Jobs/{id}/Remove	DELETE	Removes a completed job with the specified identifier.
/Jobs/{id}/Restart	PUT	Restarts a job which has previously been stopped. Only a job that is in the Stopped By User state may be restarted.
/Jobs/{id}/Stop	PUT	Stops the job identified by the provided identifier.
/Workflows	GET	Gets all of the vantage workflows for this domain.
/Workflows/{id}	GET	Gets the workflow with the specified identifier from this domain.
/Workflows/{id}/Activate	PUT	Activate the specified workflow.
/Workflows/{id}/Deactivate	PUT	Deactivate the specified workflow.
/Workflows/{id}/Export	GET	Obtain a compressed representation of the specified workflow.
/Workflows/{id}/JobInputs	GET	Gets the inputs which may be submitted to the workflow with the specified identifier.
/Workflows/{id}/Jobs/	GET	Return all jobs for a specified workflow which match the specified filter criteria. The valid values for the filter query parameter are: 'All', 'Active', 'Failed', 'Paused' or 'Complete'
/Workflows/{id}/Rename	PUT	Renames the specified workflow to a new name.
/Workflows/{id}/Signature	GET	Obtain a value representing a unique signature for a workflow.
/Workflows/{id}/Submit	POST	Submit a job to a workflow. The request data should be obtained via the /Workflows/{ID}/JobInputs GET query.
/Workflows/{id}/SubmitBinder	POST	Submit a Binder to a workflow.
/Workflows/Import	PUT	Import a workflow which had been previously exported from a Domain.

**NOTE:** The above image is a representation of the API help and is intended for illustrative purposes only. The image above may not reflect the current methods available in the UI (Mainly due to screen real-estate). Please view the URL <http://localhost:8676/Rest> within a Browser on the machine where the SDK service is running in order to view the most up-to-date and complete graphic for the method listing.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

### Domain Operations:

These operations represent capabilities which are applicable to the entire Vantage domain. They typically manipulate or return information about the entire domain.

## http://localhost:8676/Rest/Domain/Categories

Gets all categories from this domain.

**Uri:** http://localhost:8676/Rest/Domain/Categories

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Categories": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Name": "String content",
    "Workflows": [{
      "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
      "Description": "String content",
      "Name": "String content"
    }]
  }]
}
```

## http://localhost:8676/Rest/Domain/Load/{CRITERIA}

Gets the load for the Vantage domain based upon the provided criteria, eg: /Domain/Load/CPU

**Uri:** http://localhost:8676/Rest/Domain/Load/{CRITERIA}

**HTTP Method:** GET

Message direction	Format	Body
-------------------	--------	------



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

```
{  
  "Load":50  
}
```

### Additional notes about the Domain Load

Request that the domain load be computed and returned. The domain load shall correspond to a percent value [0-100] which indicates the overall utilization of domain.

The Load may be calculated in one of two ways (based upon what is provided in the `<paramref name="criteria"/>` value of the URI.

If a service name is provided, the utilization of the service is used in the load calculation.

The utilization of the service is defined as:  $\#Active / Capacity$ .

For example, a Transcode service with a capacity of 6 that is currently running 2 jobs would have a utilization of 33% (2 / 6)

If CPU load is requested, this shall correspond to the averaged CPU load for each machine in the domain.

The CPU load is normalized across all machines. For example, if a 32 core machine is running at 80% utilization and a second machine is a 2 core machine running at 100% utilization, Vantage computes the domain load as:

$$(32 * 80\%) + (2 * 100\%) = 2760.  
Final Load = 2760 / 34 cores = \sim 81\% \text{ utilization}$$

Vantage would not simply aggregate each machine load (80% + 100% / 2) == 90% as this would treat all machines as equal when they may be quite different.

The `<paramref name="criteria"/>` will allow the load to be constrained to a specific set of components.

The value for `criteria` may be as follows:

**CPU** --> uses the CPU utilization on all machines as the load value. What is returned is the aggregate value for the CPU utilizations across all machines (running Vantage services) in the domain.

**TRANSCODE** --> uses the Transcode service as the service to compute a load value for. In this scenario, the utilization of each Transcode service in the Domain is computed, averaged and returned.

**ANALYSIS** --> uses the Analysis service as the service to compute the load for.  
**MULTISCREEN** --> uses the Multiscreen service as the service to compute the load for.  
**EDIT** --> uses the Edit service.





848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

**IPTV VOD** --> uses the IPTV VOD service.

Finally, if multiple services are desired to participate in the domain load, the query string can contain several services in a CSV type format, eg:

**TRANSCODE, ANALYSIS, EDIT** -- use the Transcode, Analysis and Edit service  
**MULTISCREEN, IPTV VOD** -- use the Multiscreen and IPTV VOD service

Example Uri: <http://localhost:8676/Rest/Domain/Load/Transcode, Multiscreen>

The above would return the load for the Multiscreen and Transcode Services.

**NOTE:** It is not legal to combine service names and CPU, eg:

**TRANSCODE, CPU** -- this would correspond to an *illegal* value which would result in a value of 0 being returned.

If an illegal query string is provided or if an internal problem precludes the load from being generated, the value of 0 shall be returned.

## http://localhost:8676/Rest/Domain/Version

Returns a string representation of the version of the current Vantage domain (eg: 7.0.0).

**Uri:** <http://localhost:8676/Rest/Domain/Version>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "Version": "7.0.0"  
}
```

## Reference for <http://localhost:8676/Rest/Domain/Online>



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Returns an indication as to whether the SDK service has successfully attached to the domain database.

If the Domain database is not running, is unable to be located or is otherwise inaccessible to the SDK service, then a value of false shall be returned.

**Uri:** <http://localhost:8676/Rest/Domain/Online>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Online":true
}
```

NOTE: The *Online* property in the JSON response shall contain the value of **true** if the SDK service has successfully attached to the target Vantage Domain database. If the SDK service was unable to attach to the target database (for any reason); the *Online* property shall be given a value of **false**.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

### Category Operations:

The following operations focus on the ability to manipulate Categories within Vantage.

## Reference for <http://localhost:8676/Rest/Category/{NAME}/Exists>

Returns an indication of whether the Category with the provided name exists in this domain.

**Uri:** <http://localhost:8676/Rest/Category/{NAME}/Exists>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "CategoryExists":true
}
```

## Reference for <http://localhost:8676/Rest/Category/{NAME}/Create>

Create a category with the specified name.

If a Category already exists with the provided name, then that category shall be returned in the CreateCategoryResponse.

If a Category with the provided name does not already exist, then a new category shall be created and the new category shall be returned in the CreateCategoryResponse.

**Uri:** <http://localhost:8676/Rest/Category/{NAME}/Create>

**HTTP Method:** PUT

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

The following is an example response Json body:

```
{
  "Category": {
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Name": "String content",
    "Workflows": [ {
      "Identifier": "1627aea5-8e0a-4371-9022-
9b504344e724",
      "Description": "String content",
      "Name": "String content"
    } ]
  }
}
```



848 Gold Flat Road  
 Nevada City, CA 95959  
 +1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

### Workflow Operations:

These operations focus on the manipulation of Vantage workflows.

# http://localhost:8676/Rest/Workflows

Gets all of the Vantage workflows for this domain.

**Uri:** http://localhost:8676/Rest/Workflows

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Workflows": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "ExpirationInHours": 2147483647,
    "Name": "String content",
    "State": 0,
    "UTC_ModifiedTime": "String content"
  }]
}
```

NOTE: The string representation of the *Identifier* property returned in the collection of Workflows in the JSON block above is used as the argument to any other Workflow based REST query.

The value for State can correspond to the one the following:

State	Value	Meaning
Idle	0	The workflow is in the idle state. It is not being used nor is it being edited. The workflow may be deleted.
Active	1	The workflow has been activated and is processing jobs. The workflow may be deactivated. The workflow may not be deleted.
Edit	2	The workflow is currently being edited by a user. The workflow may be activated.

NOTE: The call below (Workflows/{ID}) contains additional information regarding other fields returned.

## http://localhost:8676/Rest/Workflows/{ID}

Gets the workflow with the specified identifier from this domain.

**Uri:** http://localhost:8676/Rest/Workflows/{ID}

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Workflow": {
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "ExpirationInHours": 2147483647,
    "Name": "String content",
    "State": 0,
    "UTC_ModifiedTime": "String content"
  }
}
```

The value for State can correspond to the one the following:

State	Value	Meaning
Idle	0	The workflow is in the idle state. It is not being used nor is it being edited. The workflow may be deleted.
Active	1	The workflow has been activated and is processing jobs. The workflow may be deactivated. The workflow may not be deleted.
Edit	2	The workflow is currently being edited by a user. The workflow may be activated.

The field: `ExpirationInHours` corresponds to the number of hours after a job completes (either completes successfully, fails or is stopped); that the workflow shall be cleaned up by the expiration process. For example, if the value of 'ExpirationInHours' is 24, and the job completes at 12PM on Jan 1, 2020; this implies the expiration shall be scheduled for 12PM on Jan 2, 2020.

The field: `UTC_ModifiedTime` corresponds to the last modified timestamp of the workflow (converted to UTC). The modified value of the workflow is converted to UTC time; then this value is stored in an ISO-8601 compatible time format, eg: `"UTC_ModifiedTime": "2020-12-07T22:40:44.2700000Z"`

## http://localhost:8676/Rest/Workflows/{ID}/Activate

Activate the specified workflow.

**Url:** http://localhost:8676/Rest/Workflows/{ID}/Activate

**HTTP Method:** PUT

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "ActivateSuccessful":true
}
```

NOTE: The **{ID}** field in the URL corresponds to a string representation of the Workflow identifier obtained via the */Rest/Workflows* query.

The *ActivateSuccessful* property in the JSON response provides an indication of whether the activation of the workflow with the provided ID was successful or not. If a non-existent workflow is specified, a value of *false* shall be returned. If the workflow is unable to be activated, *false* shall be returned.

## http://localhost:8676/Rest/Workflows/{ID}/Deactivate

Deactivate the specified workflow.

**Url:** http://localhost:8676/Rest/Workflows/{ID}/Deactivate

**HTTP Method:** PUT

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "DeActivateSuccessful":true  
}
```

NOTE: The **{ID}** field in the URL corresponds to a string representation of the Workflow identifier obtained via the [/Rest/Workflows](#) query.

The **DeActivateSuccessful** property in the JSON response provides an indication of whether the deactivation of the workflow with the provided ID was successful or not. A workflow which is not in the active state may not be deactivated (a *false* value shall be returned). Specifying a non-existent workflow shall cause a value of *false* to be returned.

## http://localhost:8676/Rest/Workflows/{ID}/Export

Obtain a compressed representation of the specified workflow.

**Uri:** http://localhost:8676/Rest/Workflows/{ID}/Export

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "CompressedWorkflow":"String content"  
}
```

NOTE: The **{ID}** field in the URL corresponds to a string representation of the Workflow identifier obtained via the [/Rest/Workflows](#) query.

The **CompressedWorkflow** property in the JSON response contains a compressed representation of the desired workflow. *This compressed representation is proprietary to Vantage and the only use of such a value is to provide the string representation of the compressed workflow as input to a subsequent [Import](#) endpoint.*

## http://localhost:8676/Rest/Workflows/Import

Import a workflow which had been previously exported from a Domain.

**IMPORTANT:** The compressed workflow that is specified as a parameter to the Import call MUST be generated by the call: <http://localhost:8676/Rest/Workflows/{ID}/Export>





848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

**Uri:** http://localhost:8676/Rest/Workflows/Import

**HTTP Method:** PUT

Message direction	Format	Body
Request	Json	<a href="#">Example</a>
Response	Json	<a href="#">Example</a>

The following is an example request Json body:

```
{
  "AllowOverwrite":true,
  "CompressedWorkflow":"String content",
  "DestinationCategory":"1627aea5-8e0a-4371-9022-9b504344e724",
  "DuplicateVariables":true
}
```

The *CompressedWorkflow* property in the JSON request must contain the compressed representation of a workflow previously obtained via the [Export](#) endpoint.

The *AllowOverwrite* property indicates whether the import shall allow an overwrite to be performed. If this value is set to true, then an existing workflow which matches the workflow being imported (a workflow with the same ID as the imported workflow) will be overwritten by the Import. **NOTE:** The existing workflow must **NOT** be active. *An active workflow may not be overwritten.*

The *DestinationCategory* value corresponds to the ID of the Vantage Category that should be the target of the import. The Category Identifier may be obtained via the /Categories endpoint. If the *DestinationCategory* is provided the value of **Guid.Empty** (00000000-0000-0000-0000-000000000000); the default import category shall be utilized.

The *DuplicateVariables* property indicates whether the import shall consolidate variables with the same name or generate duplicate variables. The default behavior is to consolidate variables with the same name (and same type) into a single variable and then update references in the workflow to the consolidated variable (i.e.: if the workflow contains a variable named: MYFILE and a variable of the same name (MYFILE) and type already exists in the Domain, when the import is performed and *DuplicateVariables* is FALSE, if the MYFILE variable is identical to the existing variable MYFILE (i.e.: The underlying IDs match), the workflow is simply imported. However, if there is a difference, then when the workflow is imported, the references to the original MYFILE variable are updated in the newly imported workflow (specifically the IDs). If the *DuplicateVariables* property is set to TRUE, then a duplicate variable will be created if the underlying IDs differ (in this case there would be two (2) MYFILE variables (the original one existing in the domain, and the newly created one from the workflow import).



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

The following is an example response Json body:

```
{
  "WorkflowIdentifier": "1627aea5-8e0a-4371-9022-9b504344e724"
}
```

The *WorkflowIdentifier* property in the JSON response represents the Identifier of the newly imported workflow. If the workflow is unable to be imported for any reason (see constraints in the request section); then a **Guid.Empty** (00000000-0000-0000-0000-000000000000) shall be returned in this property. Any other value indicates a successful import, with the value corresponding to the ID of the newly imported workflow.

## http://localhost:8676/Rest/Workflows/{ID}/JobInputs

Gets the inputs which may be submitted to the workflow with the specified identifier.

**Url:** http://localhost:8676/Rest/Workflows/{ID}/JobInputs

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Attachments": [ {
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Data": "String content",
    "Description": "String content",
    "File": "String content",
    "Name": "String content"
  } ],
  "JobName": "String content",
  "Labels": [ {
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "Name": "String content",
    "Params": [ {
      "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
        "DefaultValue":"String content",
        "Description":"String content",
        "Increment":"String content",
        "Maximum":"String content",
        "Minimum":"String content",
        "Name":"String content",
        "Options":["String content"],
        "TypeCode":"String content",
        "Value":"String content"
    }
  ],
  "Medias":[{
    "Identifier":"1627aea5-8e0a-4371-9022-9b504344e724",
    "Data":"String content",
    "Description":"String content",
    "Files":["String content"],
    "Name":"String content"
  }],
  "Priority":2147483647,
  "Variables":[{
    "Identifier":"1627aea5-8e0a-4371-9022-9b504344e724",
    "DefaultValue":"String content",
    "Description":"String content",
    "Name":"String content",
    "TypeCode":"String content",
    "Value":"String content"
  }
  ]
}
```

The response JSON body contains a representation of the input that may be provided to a workflow via the Submit method. The Input is comprised of the following:

Property	Type	Description
JobName	String	This field contains the name that the newly created job shall contain.
Priority	Int	A value which indicates the priority of the job. The value of 0 corresponds to no priority being utilized. The greater the value, the more important the job.
Attachments	[]	A collection of the Attachments which will be provided to the workflow. This field may be empty.
Labels	[]	A collection of Labels which will be provided to the workflow. This field may be empty.
Priority	[]	A collection of Media Nicknames which will be provided to the workflow. This field may be empty.

The primary purpose/use of this endpoint is to obtain the 'required input' for a given workflow in order to submit a job to the workflow.

The most general use case would be:

<workflow input> = <http://localhost:8676/Rest/Workflows/{ID}/JobInputs>



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Populate the <workflow input> with the desired values which correspond to the input files/metadata/jobname and priority

Submit the <workflow input> to the desired workflow via the:

<http://localhost:8676/Rest/Workflows/{ID}/Submit> endpoint.

**NOTE:** Both the Media and Attachment elements which are published as part of the Input allow for an embedded Data representation to be specified. If this is desired (eg: you wish to submit an inline representation of the input vs. referencing a file on disk); you would utilize the Data property in lieu of the File property to pass this information. When using the data property, the contents of the Data property must be a UTF8 encoded Base64 representation of the string contents of the file.

**NOTE:** This mechanism should NOT be used for media files. It is primarily present as a way to allow material which is more naturally represented as in-band data (eg: CML submissions or XML metadata) to be supplied.

Please see the example section of this document for a sample of how to send in-band data using the Data property.

## <http://localhost:8676/Rest/Workflows/{ID}/Submit>

Submit a job to a workflow. The request data should be obtained via the /Workflows/{ID}/JobInputs GET query.

**Url:** <http://localhost:8676/Rest/Workflows/{ID}/Submit>

**HTTP Method:** POST

Message direction	Format	Body
Request	Json	<a href="#">Example</a>
Response	Json	<a href="#">Example</a>

The following is an example request Json body:

```
{
  "Attachments": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "File": "String content",
    "Name": "String content"
  }],
  "JobName": "String content",
  "Labels": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "Name": "String content",
    "Params": [{
      "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
        "DefaultValue":"String content",
        "Description":"String content",
        "Increment":"String content",
        "Maximum":"String content",
        "Minimum":"String content",
        "Name":"String content",
        "Options":["String content"],
        "TypeCode":"String content",
        "Value":"String content"
    }
  ]],
  "Medias":[{
    "Identifier":"1627aea5-8e0a-4371-9022-9b504344e724",
    "Description":"String content",
    "Files":["String content"],
    "Name":"String content"
  }],
  "Priority":2147483647,
  "Variables":[{
    "Identifier":"1627aea5-8e0a-4371-9022-9b504344e724",
    "DefaultValue":"String content",
    "Description":"String content",
    "Name":"String content",
    "TypeCode":"String content",
    "Value":"String content"
  }]]
}
```

The following is an example response Json body:

```
{
  "JobIdentifier":"1627aea5-8e0a-4371-9022-9b504344e724"
}
```

The *JobIdentifier* property in the JSON response corresponds to the ID of the newly created job (i.e.: the job created as a result of the submit). If a job is unable to be created (eg: The specified workflow is not active or some other error occurs); then the value of **Guid.Empty** (00000000-0000-0000-0000-000000000000) shall be returned in *JobIdentifier*. Any other value indicates that a job was successfully completed and the value of *JobIdentifier* can be utilized to ask for further information about the Job.

## Reference for <http://localhost:8676/Rest/Workflows/{ID}/SubmitBinder>

Submit a Binder which exists in the Catalog to the workflow identified by the provided ID.

The GET URL: <http://localhost:8676/Rest/Workflows/{ID}/JobInputs> may be utilized to obtain the Variables that the target workflow expects.

**Url:** <http://localhost:8676/Rest/Workflows/{ID}/SubmitBinder>

**HTTP Method:** POST

Message direction	Format	Body
Request	Json	<a href="#">Example</a>
Response	Json	<a href="#">Example</a>

The following is an example request Json body:

```
{
  "BinderIdentifier": "1627aea5-8e0a-4371-9022-9b504344e724",
  "JobName": "String content",
  "Priority": 2147483647,
  "Variables": [
    {
      "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
      "DefaultValue": "String content",
      "Description": "String content",
      "Name": "String content",
      "TypeCode": "String content",
      "Value": "String content"
    }
  ]
}
```

Submits a new job using the binder referenced by *BinderIdentifier* to the workflow specified by the ID fragment in the POST URL.

If the binder referenced by the *BinderIdentifier* value does not exist or if a workflow does not exist for the provided ID; then no job shall be submitted and a value of Guid.Empty shall be returned.

***BinderIdentifier***: The value which identifies the binder that is to be submitted.

***Variables***: The collection of variables that accompanies this job.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

**JobName:** The user facing name by which this job shall be known (non-unique).

**Priority:** The relative importance of this job compared to others (higher values imply a more important job).

The following is an example response Json body:

```
{  
  "JobIdentifier": "1627aea5-8e0a-4371-9022-9b504344e724"  
}
```

The **JobIdentifier** property in the JSON response corresponds to the ID of the newly created job (i.e.: the job created as a result of the submit). If a job is unable to be created (eg: The specified workflow is not active or some other error occurs); then the value of **Guid.Empty** (00000000-0000-0000-0000-000000000000) shall be returned in **JobIdentifier**. Any other value indicates that a job was successfully completed and the value of **JobIdentifier** can be utilized to ask for further information about the Job.

## http://localhost:8676/Rest/Workflows/{ID}/Rename

Renames the specified workflow to a new name.

**Uri:** http://localhost:8676/Rest/Workflows/{ID}/Rename

**HTTP Method:** PUT

Message direction	Format	Body
Request	Json	<a href="#">Example</a>
Response	Json	<a href="#">Example</a>

The following is an example request Xml body:

```
<RenameWorkflowRequest  
  xmlns="http://schemas.datacontract.org/2004/07/Telestream.Soa.Contract.Sdk">  
  <NewWorkflowName>String content</NewWorkflowName>  
</RenameWorkflowRequest>
```

The following is an example request Json body:

```
{  
  "NewWorkflowName": "String content"  
}
```

NOTE: The target workflow (the workflow whose ID is specified in the URL); must be in the IDLE state in order to be renamed (a workflow that is currently **Active** or is being **Edited** may not be renamed).

The following is an example response Json body:

```
{  
  "RenameSuccessful": true  
}
```

NOTE: The *RenameSuccessful* property in the JSON response shall contain the value of **true** if the Rename operation was successfully performed. If the operation was unable to be performed (eg: the target workflow was not idle or the specified workflow did not exist); then *RenameSuccessful* shall contain a value of **false**.

## http://localhost:8676/Rest/Workflows/{ID}/Jobs/?filter={FILTER}

Return all jobs for a specified workflow which match the specified filter criteria. The valid values for the filter query parameter are: **'All'**, **'Active'**, **'Failed'**, **'Paused'** or **'Complete'**

**Url:** http://localhost:8676/Rest/Workflows/{ID}/Jobs/?filter={FILTER}

**HTTP Method:** GET

Message direction	Format	Body
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "Jobs": [{  
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",  
    "IsMonitor": true,  
    "Name": "String content",  
  }]
```





848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
    "Started": "\/Date(928164000000-0400)\/",  
    "State": 0,  
    "Updated": "\/Date(928164000000-0400)\/"  
  } ]  
}
```

## Reference for <http://localhost:8676/Rest/Workflows/{ID}/Signature>

Obtain a value representing a unique signature for a workflow. The signature shall remain constant so long as the workflow is not modified (Actions added or removed or updated, new connections made, workflow renamed or moved).

**NOTE:** Importing the same workflow multiple times will assign each new import a new unique identifier and consequently each copy of the workflow would have a distinct (and different) Signature.

**Uri:** <http://localhost:8676/Rest/Workflows/{ID}/Signature>

**HTTP Method:** GET

Message direction	Format	Body
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "Signature": "String content"  
}
```

## Reference for <http://localhost:8676/Rest/Workflows/ExportToPath>

Create a new XML representation of a workflow at the specified path.

Exports the workflow identified by the WorkflowIdentifier field to the location specified by BasePathForExport field (see JSON below).

If everything is successful, the result of this method shall be a string which corresponds to the complete path of the newly exported file.

If the specified workflow does not exist, or if there is a problem accessing the specified location; the export shall fail and the empty string '' shall be returned.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

**NOTE:** If there is a collision of names in the folder specified in `BasePathForExport` then this method shall unique-ify the output name of the newly exported workflow and return the new unique name.

**Uri:** <http://localhost:8676/Rest/Workflows/ExportToPath>

**HTTP Method:** POST

Message direction	Format	Body
Request	Json	<a href="#">Example</a>
Response	Json	<a href="#">Example</a>

The following is an example request Json body:

```
{
  "BasePathForExport": "String content",
  "WorkflowIdentifier": "1627aea5-8e0a-4371-9022-9b504344e724"
}
```

The following is an example response Json body:

```
{
  "PathToExportedFile": "String content"
}
```

**IMPORTANT:** The XML file which is created as a result of this call is NOT a compressed file (a compressed file is generated from the API call:

<http://localhost:8676/Rest/Workflows/{ID}/Export>). The file generated by this API call is a traditional, readable XML file. The output of this call MAY NOT be utilized in the API call:

<http://localhost:8676/Rest/Workflows/Import>. Only workflows exported with the call:

<http://localhost:8676/Rest/Workflows/{ID}/Export> may be utilized by the

<http://localhost:8676/Rest/Workflows/Import> call.

The workflows created by the call: <http://localhost:8676/Rest/Workflows/ExportToPath> may be utilized in the following API calls: <http://localhost:8676/Rest/Workflows/ImportFromPath> AND <http://localhost:8676/Rest/Workflows/Synchronize>.

The *BasePathForExport* property indicates the base path where the new XML file shall be created. The API shall unique-ify outputs (i.e.: so if a file collision would occur, steps are taken to prevent this and generate a new unique name).

The *WorkflowIdentifier* value corresponds to the ID of the Vantage Workflow that should be the target of the export.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

The result of this call shall be the full path of the file that is created (including any unique-ifying name characters).

## Reference for <http://localhost:8676/Rest/Workflows/ImportFromPath>

Import a workflow specified by a file path.

NOTE: The input to this workflow MUST be a traditional workflow XML document that is either:

Generated by the <http://localhost:8676/Rest/Workflows/ExportToPath> API call.

Created as a result of exporting a workflow from the Workflow Designer.

**Url:** <http://localhost:8676/Rest/Workflows/ImportFromPath>

**HTTP Method:** PUT

Message direction	Format	Body
Request	Json	<a href="#">Example</a>
Response	Json	<a href="#">Example</a>

The following is an example request Json body:

```
{
  "AllowOverwrite":true,
  "DestinationCategory":"1627aea5-8e0a-4371-9022-9b504344e724",
  "DuplicateVariables":true,
  "Path":"String content"
}
```

The **Path** property in the JSON request must contain the path to a valid workflow XML created either via the **ExportToPath** endpoint or via creating a workflow XML by exporting the workflow manually from the Workflow Designer.

The **AllowOverwrite** property indicates whether the import shall allow an overwrite to be performed. If this value is set to true, then an existing workflow which matches the workflow being imported (a workflow with the same ID as the imported workflow) will be overwritten by the Import. **NOTE:** The existing workflow must **NOT** be active. *An active workflow may not be overwritten.*



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

The *DestinationCategory* value corresponds to the ID of the Vantage Category that should be the target of the import. The Category Identifier may be obtained via the /Categories endpoint. If the *DestinationCategory* is provided the value of **Guid.Empty** (00000000-0000-0000-0000-000000000000); the default import category shall be utilized.

The *DuplicateVariables* property indicates whether the import shall consolidate variables with the same name or generate duplicate variables. The default behavior is to consolidate variables with the same name (and same type) into a single variable and then update references in the workflow to the consolidated variable (i.e.: if the workflow contains a variable named: MYFILE and a variable of the same name (MYFILE) and type already exists in the Domain, when the import is performed and *DuplicateVariables* is FALSE, if the MYFILE variable is identical to the existing variable MYFILE (i.e.: The underlying IDs match), the workflow is simply imported. However, if there is a difference, then when the workflow is imported, the references to the original MYFILE variable are updated in the newly imported workflow (specifically the IDs). If the *DuplicateVariables* property is set to TRUE, then a duplicate variable will be created if the underlying IDs differ (in this case there would be two (2) MYFILE variables (the original one existing in the domain, and the newly created one from the workflow import).

The following is an example response Json body:

```
{
  "WorkflowIdentifier": "1627aea5-8e0a-4371-9022-
9b504344e724"
}
```

The *WorkflowIdentifier* property in the JSON response corresponds to the GUID identifier of the workflow created as a result of this call. If the value of Guid.Empty (all 0's) is returned, this indicates that the workflow was unable to be imported (there may be a problem with the workflow, the path cannot be accessed, the workflow already exists and is active).

## Reference for <http://localhost:8676/Rest/Workflows/Synchronize>

Attempt to resynchronize the definition of a workflow using the path to a Vantage workflow, previously generated by the call to **ExportToPath** into this domain or via manually exporting a workflow from the Workflow Designer.

This method will overwrite an existing workflow if the workflow represented by the Identifier of the input workflow already exists in the domain.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

If the existing workflow is active, the workflow shall be placed into an edit state and updated to allow the definitions which are present in the supplied workflow to overwrite the existing definitions.

Any in-process actions shall not be manipulated and shall continue to run with the definitions of the actions which were present when the job was initially created.

If there is no existing workflow which corresponds to the workflow specified as input, then the new workflow is added to the domain without any existing workflows being modified.

**Uri:** <http://localhost:8676/Rest/Workflows/Synchronize>

**HTTP Method:** PUT

Message direction	Format	Body
Request	Json	<a href="#">Example</a>
Response	Json	<a href="#">Example</a>

The following is an example request Json body:

```
{
  "DestinationCategory": "1627aea5-8e0a-4371-9022-9b504344e724",
  "Path": "String content"
}
```

The *Path* property in the JSON request must contain the path to a valid workflow XML created either via the [ExportToPath](#) endpoint or via creating a workflow XML by exporting the workflow manually from the Workflow Designer.

The *DestinationCategory* value corresponds to the ID of the Vantage Category that should be the target of the import. The Category Identifier may be obtained via the [/Categories](#) endpoint. If the *DestinationCategory* is provided the value of **Guid.Empty** (00000000-0000-0000-0000-000000000000); the default import category shall be utilized. NOTE: The *DestinationCategory* value is only relevant if the workflow specified in the *Path* property does not exist.

The following is an example response Json body:

```
{
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
"WorkflowIdentifier": "1627aea5-8e0a-4371-9022-9b504344e724"
```

The *WorkflowIdentifier* property in the JSON response corresponds to the GUID identifier of the workflow created as a result of this call. If the value of `Guid.Empty` (all 0's) is returned, this indicates that the workflow was unable to be imported (there may be a problem with the workflow, the path can not be accessed, the workflow already exists and is active).



848 Gold Flat Road  
 Nevada City, CA 95959  
 +1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

### Job Operations:

These operations focus on the manipulation of Vantage jobs.

## http://localhost:8676/Rest/Jobs/{ID}

Returns information about the job with the provided identifier.

**Uri:** http://localhost:8676/Rest/Jobs/{ID}

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Job": {
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "IsMonitor": true,
    "Name": "String content",
    "Started": "\/Date(928164000000-0400)\/",
    "State": 0,
    "Updated": "\/Date(928164000000-0400)\/"
  }
}
```

The JSON response shall contain the following properties:

Property Name	Type	Description
Identifier	Guid	The unique identifier for the job.
IsMonitor	Boolean	A Boolean value which indicates whether the job corresponds to a Monitor type transaction (i.e.: a Supervisor task which is ultimately responsible for the creation of other jobs). A value of true indicates this is a Monitor job, while a value of false implies that the job in question is a more traditional (normal) Vantage job.
Name	String	The job name (the name which appears in the Vantage Workflow Designer)
Started	DateTime	The timestamp of when the job was created.
State	JobState	The current state of the job (see below)
Updated	DateTime	The timestamp of when the job was last updated.

Job State	Value	Meaning
In Process	0	The job is currently active and in-process (it contains actions which are currently being processed).



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Failed	4	The job has failed.
Complete	5	The job has successfully completed.
Waiting	6	The job is active, but is currently waiting for a resource. More specifically the job contains actions which are all awaiting being allocated to a service for processing.
Stopped by User	7	The job was stopped by a user/Administrator.
Waiting to Retry	8	The job has entered a state where the remaining actions are waiting to be retried. This is typically the result of a Retry rules being applied to one or more actions in a job (eg: retry an FTP transfer after 10 minutes if the target site is not accessible).
Queued for Submission	10	The job was received by the SDK service and has been queued for submission into Vantage.
No Such Job	11	The job with the provided ID does not exist.

## http://localhost:8676/Rest/Jobs/{ID}/Actions

Return all actions associated with a specific job.

**Url:** http://localhost:8676/Rest/Jobs/{ID}/Actions

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Actions": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "Name": "String content",
    "Started": "\/Date(928164000000-0400)\/",
    "State": 0,
    "Updated": "\/Date(928164000000-0400)\/"
  }]
}
```

The JSON response shall contain the following properties:

Property Name	Type	Description
Identifier	Guid	The unique identifier for the instance of the running action.
Description	String	The description of the action. This is the string that appears on the bottom of the action visualization on the Workflow Designer.
Name	String	The name of the action (eg: "Flip")





848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Started	DateTime	The timestamp of when the instance of the action in the job was created.
State	Action State	The current state of the action (see below)
Updated	DateTime	The timestamp of when the action within the job was last updated.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Action State	Value	Meaning
In Process	0	The action is currently being processed by a service. (In the Workflow Designer, this would correspond to a <b>GREEN</b> icon).
Ignored	1	The action has completed but was set to an Ignored state. (In the Workflow Designer, this would correspond to a <b>BLACK</b> icon).
Complete	2	The action has successfully completed. (In the Workflow Designer, this would correspond to a <b>BLUE</b> icon).
Failed	3	The action has failed during processing. (In the Workflow Designer, this would correspond to a <b>RED</b> icon).
Waiting	4	The action has been assigned to a service (i.e.: on the Queue) and is currently waiting for resources (waiting for an execution slot). (In the Workflow Designer, this would correspond to a <b>YELLOW</b> icon).
Created	5	The action has been created, but has not been assigned to a service. (In the Workflow Designer, this would correspond to a <b>WHITE</b> icon).
Paused	6	The action has been paused. (In the Workflow Designer, this would correspond to a <b>GREY</b> icon).
Stopped By User	9	The action was stopped by an Administrator. (In the Workflow Designer, this would correspond to a <b>MAGENTA</b> icon).
Waiting to Retry	10	The action had previously failed and is waiting to retry [based upon the use of Action Retry rules]. (In the Workflow Designer, this would correspond to an <b>ORANGE</b> icon).
Waiting on Service	12	The action is waiting on a service to be available for execution (this state implies that no services are running which may currently host the action) . (In the Workflow Designer, this would correspond to a <b>PURPLE</b> icon).
Waiting on Run On Rule	13	The action is waiting on a Run On Rule. This state implies that Run On Rules are precluding the action from running on a given service. (In the Workflow Designer, this would correspond to a <b>YELLOW-GREEN</b> icon).

## <http://localhost:8676/Rest/Jobs/{ID}/ErrorMessage>

Returns the relevant error message associated with a job which has failed.

**Url:** <http://localhost:8676/Rest/Jobs/{ID}/ErrorMessage>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "JobErrorMessage": "String content"  
}
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

NOTE: The *JobErrorMessage* property in the JSON response shall contain the error message which was generated for the job specified by the provided ID. If the provided ID does not correspond to a valid job or the provided ID does not correspond to a job which has failed, then an Empty string shall be returned in *JobErrorMessage*.

## http://localhost:8676/Rest/Jobs/{ID}/Metrics

Return specific metrics (eg: total run time, total queue time) about a specified job.

**Uri:** http://localhost:8676/Rest/Jobs/{ID}/Metrics

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "TotalQueueTimeInSeconds":2147483647,
  "TotalRunTimeInSeconds":2147483647
}
```

NOTE:

The *TotalQueueTimeInSeconds* property in the JSON response shall contain the total job queued time (the total time the actions in the job spent queued) in Seconds.

The *TotalRunTimeInSeconds* property in the JSON response shall contain the total job run time (the total amount of time the actions in the job spent performing work) in Seconds.

## http://localhost:8676/Rest/Jobs/{ID}/Outputs

Return all outputs (Media Nicknames, Attachments and/or Labels) associated with a specific job.

**Uri:** http://localhost:8676/Rest/Jobs/{ID}/Outputs

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Attachments": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Data": "String content",
    "Description": "String content",
    "File": "String content",
    "Name": "String content"
  }],
  "Labels": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "Name": "String content",
    "Params": [{
      "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
      "DefaultValue": "String content",
      "Description": "String content",
      "Increment": "String content",
      "Maximum": "String content",
      "Minimum": "String content",
      "Name": "String content",
      "Options": ["String content"],
      "TypeCode": "String content",
      "Value": "String content"
    }
  ]
}],
  "Medias": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "Files": ["String content"],
    "Name": "String content"
  }
]
}
```

The response JSON body contains a representation of all of the Outputs which are available when a job has completed. Outputs correspond to any Attachment, Media or Label that is present within the job when it has completed. :

Output	Type	Description
Attachments	[]	A collection of the Attachments which is available at the completion of the job. Attachments typically refer to Metadata files (eg: Sidecar XML files). There will be one attachment entry in the Attachments collection for each entry which is present Attachments section of the Binder view of a completed job in the Vantage Workflow designer.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

		<p><b>NOTE:</b> Attachments have a Data property and a File property. Attachments may be generated internal to Vantage (and not have a representative file on disk). In such a case, the File property will be an empty string and the Data property will contain the string representation of the Attachment.</p> <p><i>In all cases (whether or not a File is present for an attachment); the Data property shall always contain the string representation of the contents of the Attachment.</i></p>
Labels	[ ]	A collection of Metadata Labels which are available at the completion of a job. A Label will contain one (1) or more Parameters which will have a specific value and type. There will be a specific Label in the response collection for each entry in the Metadata Labels section of the Binder view of a completed job in the Vantage Workflow designer.
Medias	[ ]	A collection of Media Nicknames which are available at the completion of a job. Medias refer to the individual Media Nicknames present when a job completes. The entries in the Medias collection shall correspond to the entries which are present in the Media Files section of the Binder view of a completed job in the Vantage Workflow designer.

**NOTE:** Please see the **Examples** section at the end of this document for a more in-depth discussion about the **Outputs** endpoint.

## http://localhost:8676/Rest/Jobs/{ID}/Progress

Returns the progress (percent complete) for the specified job.

**Uri:** http://localhost:8676/Rest/Jobs/{ID}/Progress

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "JobProgress": 50
}
```

**NOTE:** The **JobProgress** property in the JSON response shall contain a value which represents the progress of the job. The value shall fall between 0 and 100. If the supplied ID does not correspond to a job, the value of 0 shall be returned.

## http://localhost:8676/Rest/Jobs/{ID}/Remove

Removes a completed job with the specified identifier. **NOTE:** Only a completed job (Completed Successfully, Failed, or a job that has been stopped by a user) may be removed.

**Uri:** http://localhost:8676/Rest/Jobs/{ID}/Remove



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

**HTTP Method:** DELETE

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "RemoveJobSuccessful":true  
}
```

NOTE: The *RemoveJobSuccessful* property in the JSON response shall contain the value of **true** if the job with the specified ID was successfully removed. If the operation was unable to be performed (for any reason); the *RemoveJobSuccessful* property shall be given a value of **false**.

## http://localhost:8676/Rest/Jobs/{ID}/Stop

Stops the job identified by the provided identifier. NOTE: Only a job that is in-process (i.e.: not complete) may be stopped.

**Uri:** http://localhost:8676/Rest/Jobs/{ID}/Stop

**HTTP Method:** PUT

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "JobStopped":true  
}
```

NOTE: The *JobStopped* property in the JSON response shall contain the value of **true** if the job with the specified ID was successfully stopped. If the operation was unable to be performed (for any reason); the *JobStopped* property shall be given a value of **false**.

## http://localhost:8676/Rest/Jobs/{ID}/Restart



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Restarts a job which had previously been stopped. Only a job that is in the **Stopped By User** state may be restarted.

**Uri:** <http://localhost:8676/Rest/Jobs/{ID}/Restart>

**HTTP Method:** PUT

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "RestartJobSuccessful": true
}
```

NOTE: The *RestartJobSuccessful* property in the JSON response shall contain the value of **true** if the job with the specified ID was successfully restarted. If the operation was unable to be performed (for any reason); the *RestartJobSuccessful* property shall be given a value of **false**.

## <http://localhost:8676/Rest/Jobs/{ID}/Forwarded>

Returns a collection of jobs which were created as a result of being forwarded from the specified job.

This endpoint allows a client to receive information about new jobs which are created as a result of Forward actions being present in a workflow. The presence of a Forward action in a workflow implies that a new job is being created. This endpoint method provides the ability to track a new job that is created via a current job containing one (1) or more Forward actions (each which results in a new job being created). This method allows a reference to those newly created jobs to be returned (so that each new job may be tracked by the client).

**Uri:** <http://localhost:8676/Rest/Jobs/{ID}/Forwarded>

**HTTP Method:** GET

Message direction	Format	Body
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Jobs": [ {
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
  } ]
}
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
    "IsMonitor":true,  
    "Name":"String content",  
    "Started":"\\/Date(928164000000-0400)\\/",  
    "State":0,  
    "Updated":"\\/Date(928164000000-0400)\\/"  
  }]  
}
```



### Catalog Operations:

Methods which manipulate Vantage Catalogs, the Folders which they contain and any Binders which reside in these folders.

**NOTE:** A Catalog is akin to a top-level folder. API methods which provide information about a specific folder may be provided with either a specific Folder's identifier or a specific Catalog's identifier (since a Catalog is a top-level folder).

## Reference for <http://localhost:8676/Rest/Catalogs>

Gets all of the Catalog(s) for the Domain.

**Uri:** <http://localhost:8676/Rest/Catalogs>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Catalogs": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Name": "String content"
  }]
}
```

The Catalogs response shall contain an array of entries. Each entry shall contain an *Identifier* field and a *Name* field. The *Identifier* field corresponds to the unique identifier for the catalog with the designation indicated in the *Name* field. The Identifier value can be utilized in subsequent invocations of other REST methods which accept a Catalog identifier as input.

## Reference for <http://localhost:8676/Rest/Catalogs/{ID}>

Returns the catalog entry from the current Domain which corresponds to the provided value of ID.

**NOTE:** If the provided ID does not correspond to a Catalog which exists in the current Domain, a NULL entry shall be returned in the Catalog response.

**Uri:** <http://localhost:8676/Rest/Catalogs/{ID}>

**HTTP Method:** GET



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Catalog": {
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Name": "String content"
  }
}
```

NOTE: If the value of ID in the request URL does not correspond to a Catalog which exists within the current Vantage domain, then a null entry shall be returned.

```
{
  "Catalog": null
}
```

## Reference for <http://localhost:8676/Rest/Catalogs/Cleanup>

Clean up (remove from Vantage) all folders which are marked as Transient and are empty.

**Uri:** <http://localhost:8676/Rest/Catalogs/Cleanup>

**HTTP Method:** POST

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	N/A	The Response body is empty.

The `/Rest/Catalogs/Cleanup` shall instruct Vantage to remove ALL folders from the Catalog which have been marked as Transient and which are empty. This call is a programmatic way of invoking the same cleanup functionality which is available in the VMC->Catalogs pane for cleaning up empty folders in the catalog.

## Reference for <http://localhost:8676/Rest/Content/{ID}/Entries>

Returns all outputs (Media Nicknames, Attachments and/or Labels) associated with the specified content entry.

**Url:** <http://localhost:8676/Rest/Content/{ID}/Entries>

**HTTP Method:** GET

This operation supports JSONP responses. The callback function can be specified using the "**callback**" Url query parameter.

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Attachments": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Data": "String content",
    "Description": "String content",
    "File": "String content",
    "Name": "String content"
  }],
  "Labels": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Description": "String content",
    "Name": "String content",
    "Params": [{
      "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
      "DefaultValue": "String content",
      "Description": "String content",
      "Increment": "String content",
      "Maximum": "String content",
      "Minimum": "String content",
      "Name": "String content",
      "Options": ["String content"],
      "TypeCode": "String content",
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
        "Value":"String content"
      }
    }],
    "Medias":[{
      "Identifier":"1627aea5-8e0a-4371-9022-9b504344e724",
      "Data":"String content",
      "Description":"String content",
      "Files":["String content"],
      "Name":"String content"
    }]
  }
}
```

The intent of this call is to allow a client application to request information about what is present in a particular content entry (the content entry may be the result of traditional submit or the content entry associated with a Binder). All of the Attachments, Labels and Media outputs shall be returned.

NOTE: If the provided ID does not correspond to a Content entry which exists in the domain, the following JSON structure shall be returned:

```
{
  "Attachments":null,
  "Labels":null,
  "Medias":null
}
```

## Reference for <http://localhost:8676/Rest/Folders/{ID}>

Returns all of the Folder entries which exist in either the Catalog or the Folder whose Identifier corresponds to the provided ID.

**Url:** <http://localhost:8676/Rest/Folders/{ID}>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
"Folders": [{
  "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
  "Name": "String content"
}]
}
```

NOTE: If the value of ID in the request URL does not correspond to a Folder or a Catalog which exists within the current Vantage domain, then a null entry shall be returned.

```
{
  "Folder": null
}
```

## Reference for <http://localhost:8676/Rest/Folder/{ID}>

Returns the specific Folder entry which is identified by the provided ID.

**Uri:** <http://localhost:8676/Rest/Folder/{ID}>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Folder": {
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Name": "String content"
  }
}
```

NOTE: If the value of ID in the request URL does not correspond to a Folder or a Catalog which exists within the current Vantage domain, then a null entry shall be returned.

```
{
  "Folder": null
}
```

## Reference for <http://localhost:8676/Rest/Folders/{ID}/Binders>

Gets all of the Binder(s) for a given Catalog or Folder. This method returns all of the Binders which exist within the Folder with the provided ID. This method does not traverse the entire Folder hierarchy to discover all Binders which may exist in all sub-Folders of the specified Folder. It shall only return the Binder entries which reside within the Folder with the specified ID.

**Url:** <http://localhost:8676/Rest/Folders/{ID}/Binders>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Binders": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Content": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Name": "String content"
  }]
}
```

**NOTE:** If the value of ID in the request URL does not correspond to a Folder or a Catalog which exists within the current Vantage domain, then a null entry shall be returned.

```
{
  "Binders": null
}
```

This endpoint also supports optional start and end query parameters. These parameters constrain the results that are returned to only binders created after the provided start or before the provided end.

The original query:

<http://localhost:8676/Rest/Folders/{FOLDERID}/Binders>

will return all binders in the folder whose id is: {FOLDERID}, eg:

<http://localhost:8676/Rest/Folders/b364aeed-7089-4821-a896-ed639cb5a021/Binders>



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

The endpoint now supports the optional query parameters: start and end you can specify something like:

<http://localhost:8676/Rest/Folders/b364aeed-7089-4821-a896-ed639cb5a021/Binders?start=2021-11-1>

and this will return all BINDERS in the folder with ID: b364aeed-7089-4821-a896-ed639cb5a021 created after 2021-11-1

if you specify a start query parameter, you can also optionally specify an end query parameter:

<http://localhost:8676/Rest/Folders/b364aeed-7089-4821-a896-ed639cb5a021/Binders?start=2021-11-1&end=2021-11-4>

the above would return all binders in the folder with ID: b364aeed-7089-4821-a896-ed639cb5a021 created AFTER 2021-11-1 AND BEFORE 2021-11-4

## Reference for <http://localhost:8676/Rest/Folders/{ID}/Binder/{NAME}>

Obtain the specified Binder object that is associated with the provided NAME and that is located in the folder identified by ID.

**NOTE:** The first binder that exactly matches the specified name is returned. Since a catalog folder may contain Binders with names that are duplicates of one another, it is advised that when binders are registered in a catalog; that steps be taken to give them a unique name (see the Register action for more information about this).

If a Binder with the specified name Name does not exist in the specified folder, a NULL value shall be returned.

**Url:** <http://localhost:8676/Rest/Folders/{ID}/Binder/{NAME}>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Response    Json    [Example](#)

The following is an example response Json body:

```
{
  "Binder":{
    "Identifier":"1627aea5-8e0a-4371-9022-9b504344e724",
    "Content":"1627aea5-8e0a-4371-9022-9b504344e724",
    "Name":"String content"
  }
}
```

**NOTE:** If the value of ID in the request URL does not correspond to a Folder or a Catalog which exists within the current Vantage domain OR if the provided NAME does not correspond to a matching Binder name within this Folder, then a null entry shall be returned.

```
{
  "Binder": null
}
```

## Reference for <http://localhost:8676/Rest/Binder/{ID}>

Gets a specific Binder for the domain.

**Url:** <http://localhost:8676/Rest/Binder/{ID}>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Binder":{
    "Identifier":"1627aea5-8e0a-4371-9022-9b504344e724",
    "Content":"1627aea5-8e0a-4371-9022-9b504344e724",
    "Name":"String content"
  }
}
```

**NOTE:** If the value of ID in the request URL does not correspond to a Binder which exists within the current Vantage domain, then a null entry shall be returned.





848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
{  
  "Binder": null  
}
```

## Reference for <http://localhost:8676/Rest/Binder/{ID}/Content>

Gets the content for a specific Binder.

**Url:** <http://localhost:8676/Rest/Binder/{ID}/Content>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{  
  "ContentIdentifier": "1627aea5-8e0a-4371-9022-  
9b504344e724",  
  "Created": "\/Date(928164000000-0400) \/",  
  "Name": "String content",  
  "Updated": "\/Date(928164000000-0400) \/"  
}
```

NOTE: If the provided ID does not correspond to a Binder which exists within the Vantage Domain, then a default entry shall be returned. In particular, within this entry, the *Name* field shall be Null and the *Created* and *Updated* fields shall be set to **DateTime.Min**. The client should use the presence of a non-null *Name* field as an indication of whether the result contains valid data.

Here is an example of the result that is returned when a non-existent Binder ID is supplied:

```
{  
  "ContentIdentifier": "00000000-0000-0000-0000-000000000000",  
  "Created": "\/Date(-62135578800000-0500) \/",  
  "Name": null,  
  "Updated": "\/Date(-62135578800000-0500) \/"  
}
```

### Service Operations:

Methods which allow for the retrieval of information about Vantage services.

## Reference for <http://localhost:8676/Rest/Services>

Get all online services present in this Vantage domain. NOTE: Only online services are returned when this call is invoked. Services which are in maintenance mode (or any other state) shall NOT be returned.

**Uri:** <http://localhost:8676/Rest/Services>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Services": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Machine": "1627aea5-8e0a-4371-9022-9b504344e724",
    "MachineType": 0,
    "ServiceKind": "1627aea5-8e0a-4371-9022-9b504344e724",
    "ServiceTypeName": "String content",
    "State": 0
  }]
}
```

**NOTE:**

The *Identifier* property in the JSON response corresponds to the identifier for the service. This value would be used as the Id property for any service specific query (eg: [/Services/{Id}/Metrics](#))

The *Machine* property in the JSON response shall contain the identifier of the machine that the service is running on.

The *MachineType* property in the JSON response corresponds to whether the service is a Transient (1) or Permanent (0) service. Certain operations (eg: Preshutdown) are only applicable to Transient services.

The *ServiceKind* property in the JSON response corresponds to the unique type identifier for this service.

The *ServiceTypeName* property in the JSON response corresponds to the friendly name of the service type (eg: Transcode).

The *State* property in the JSON response corresponds to the state of the service. Since this query only returns online services, this value shall always be 0.

## Reference for <http://localhost:8676/Rest/Services/{ID}/Metrics>

Gets the metric information associated with the specified service.

**Url:** <http://localhost:8676/Rest/Services/{ID}/Metrics>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Metric":{
    "AcceptingWork":true,
    "ActiveSessionCount":2147483647,
    "AverageQueueLength":2147483647,
    "AverageQueueWaitInSeconds":2147483647,
    "CurrentResourceUsage":2147483647,
    "FailedLicenseCount":2147483647,
    "FailedSessionCount":2147483647,
    "LightspeedFallbackCount":2147483647,
    "QueuedSessionCount":2147483647,
    "SendQueueSize":2147483647,
    "ServiceUtilization":2147483647,
    "SuccessfulSessionCount":2147483647,
    "TargetResourceUsage":2147483647,
    "TotalSessionCount":2147483647
  }
}
```

**NOTE:**

The *AcceptingWork* property in the JSON response shall indicate whether the service is accepting work. (Services which have been instructed to enter the preshutdown mode shall indicate that they are NOT accepting work (i.e.: this value would be false)).

The *ActiveSessionCount* property in the JSON response shall contain the current number of running (active) actions that the service is hosting.



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

The *AverageQueueLength* property in the JSON response shall contain the average size of the waiting queue for this service since it entered the online state.

The *AverageQueueWaitTimeInSeconds* property in the JSON response shall contain the average amount of time (in seconds) that requests have remained on the waiting queue.

The *CurrentResourceUsage* property in the JSON response shall contain the current cost that the service is consuming. This value only has meaning in Cost Based Load Balancing mode.

The *FailedLicenseCount* property in the JSON response shall contain the total number of times that this service has been unable to acquire a required license feature resulting in an action failure.

The *FailedSessionCount* property in the JSON response shall contain the total number of times that an action resulted in a failure (for any reason).

The *LightspeedFallbackCount* property in the JSON response shall contain the total number of times that a fallback from Lightspeed GPU mode to regular mode was encountered on this service. (This is only meaningful for services which make use of the Lightspeed technology).

The *QueuedSessionCount* property in the JSON response shall contain the current number actions which are queued in the waiting queue.

The *SendQueueSize* property in the JSON response shall contain the current number of requests that the service is waiting to send to another service.

The *ServiceUtilization* property in the JSON response shall contain the utilization percentage which corresponds to the number of active actions (jobs) over the total number of allowable jobs.

The *SuccessfulSessionCount* property in the JSON response shall contain the total number of successful actions (jobs) which were run on this service.

The *TargetResourceUsage* property in the JSON response shall contains the resource usage defined for this service with respect to Cost Based Load Balancing. This value only has meaning when in Cost Based Load Balancing mode.

The *TotalSessionCount* property in the JSON response shall contain the total number of sessions (actions) which have been hosted on this service.

## Reference for <http://localhost:8676/Rest/Services/SearchByMachine/{MACHINE}>

Gets all online services in the domain which reside on the specified machine. Note: Only Online services are returned. Services in maintenance mode (or any other state besides Online) shall not be returned.

**Url:** <http://localhost:8676/Rest/Services/SearchByMachine/{MACHINE}>

**HTTP Method:** GET

Message direction	Format	Body
-------------------	--------	------



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Services": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Machine": "1627aea5-8e0a-4371-9022-9b504344e724",
    "MachineType": 0,
    "ServiceKind": "1627aea5-8e0a-4371-9022-9b504344e724",
    "ServiceTypeName": "String content",
    "State": 0
  }]
}
```

**NOTE:**

The *Identifier* property in the JSON response corresponds to the identifier for the service. This value would be used as the Id property for any service specific query (eg:

/Services/{Id}/Metrics)

The *Machine* property in the JSON response shall contain the identifier of the machine that the service is running on.

The *MachineType* property in the JSON response corresponds to whether the service is a Transient (1) or Permanent (0) service. Certain operations (eg: Preshutdown) are only applicable to Transient services.

The *ServiceKind* property in the JSON response corresponds to the unique type identifier for this service.

The *ServiceTypeName* property in the JSON response corresponds to the friendly name of the service type (eg: Transcode).

The *State* property in the JSON response corresponds to the state of the service. Since this query only returns online services, this value shall always be 0.

## Reference for

<http://localhost:8676/Rest/Services/LookupHosting?Type={ACTIONTYPE}>

Returns information about services hosting a specific type of action.

**Uri:** <http://localhost:8676/Rest/Services/LookupHosting?Type={ACTIONTYPE}>



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Services": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Machine": "1627aea5-8e0a-4371-9022-9b504344e724",
    "MachineType": 0,
    "ServiceKind": "1627aea5-8e0a-4371-9022-9b504344e724",
    "ServiceTypeName": "String content",
    "State": 0,
    "Endpoint": "String content"
  }]
}
```

The primary intent of this call is to allow client applications who wish to interact with specific Vantage services (primarily the Capture service) to obtain information about all of the services hosting specific action types. Calling this method with the `actionType` of 'Capture' would result in information about all of the services currently hosting an in process instance of the Capture action (allowing the client application to interact with the Capture service for additional capability regarding the capture actions – see the Capture API guide for details about this specific capability).

## Reference for <http://localhost:8676/Rest/Services/{ID}/Preshutdown>

Request that the specified service enter the preshutdown state. This is only applicable to services of the **Transient** type. When this request is made, the service shall enter a state where it shall no longer accept new work. The service shall complete all active and queued work. Once this has completed, the service shall shutdown. This call is an asynchronous call. The invoking entity will have no indication that the service has completed this request other than to monitor the service, noting that the service will eventually no longer be available (due to the shutdown).

**Url:** <http://localhost:8676/Rest/Services/{ID}/Preshutdown>

**HTTP Method:** POST



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	N/A	The Response body is empty.

### Machine Operations:

Methods which allow for the retrieval of information about machines in the Vantage domain.

## Reference for <http://localhost:8676/Rest/Machines>

Return all machines, which host services, in the domain.

**Url:** <http://localhost:8676/Rest/Machines>

**HTTP Method:** GET

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	Json	<a href="#">Example</a>

The following is an example response Json body:

```
{
  "Machines": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Name": "String content",
    "Processors": 2147483647,
    "MachineType": 0
  }]
}
```

#### NOTE:

The *Identifier* property in the JSON response corresponds to the identifier for the machine.

The *Name* property in the JSON response shall contain the name of the machine.

The *Processors* property in the JSON response corresponds to the number of detected processors on this machine.

The *MachineType* property in the JSON response corresponds to whether this machine is a Transient (a Virtual node) or a Permanent (physical) node. Note: Certain operations (Preshutdown) are only applicable to Transient machines.

## Reference for <http://localhost:8676/Rest/Machines/{ID}/Preshutdown>

Instruct all services on the specified machine to begin the preshutdown state. This is only applicable to machines of the **Transient** type. When this request is made, all onlines services on the machine with the





848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

specified identifier (assuming this is a **Transient** machine) shall enter a state where it shall no longer accept new work. The services shall complete all active and queued work. Once this has completed, the services shall shutdown. This call is an asynchronous call. The invoking entity will have no indication that the services have completed this request other than to monitor the service, noting that the service will eventually no longer be available (due to the shutdown).

**Url:** `http://localhost:8676/Rest/Machines/{ID}/Preshutdown`

**HTTP Method:** POST

Message direction	Format	Body
Request	N/A	The Request body is empty.
Response	N/A	The Response body is empty.

## Storage Operations

The following calls are associated with obtaining information about Managed Stores within a Vantage domain. See the Vantage Domain Administrators guide for more information about creating and utilizing Managed Stores.

Reference for  
`http://localhost:8676/Rest/Stores?IncludeOffline={OFFLINE}`

Returns information about Vantage Managed Stores. An optional qualifier which indicates whether to include offline Stores is available as well.

**Url:** `http://localhost:8676/Rest/Stores?IncludeOffline={OFFLINE}`

**HTTP Method:** GET

The following is an example response Json body:

```
{
  "Stores": [{
    "Identifier": "1627aea5-8e0a-4371-9022-9b504344e724",
    "Aliases": ["String content"],
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
        "AvailableSpace":9223372036854775807,  
        "Location":"String content",  
        "Name":"String content",  
        "Offline":true,  
        "StorageMode":0,  
        "TotalSpace":9223372036854775807  
    }]  
}
```

This call will return a list of all Managed Stores currently defined within the Vantage domain. The OPTIONAL qualifier ?IncludeOffline (which accepts a TRUE/FALSE value) can be used to indicate whether stores which are currently marked as being offline are returned. For example:

```
http://localhost:8676/Rest/Stores?IncludeOffline=True
```

would return information about ALL Managed Stores (which include BOTH Online and Offline stores).

The call:

```
http://localhost:8676/Rest/Stores?IncludeOffline=False
```

would ONLY return information about Managed Stores which are currently in the Online state.

## Reference for <http://localhost:8676/Rest/Stores/{ID}>

Return information about the specified Vantage Managed Store.

**Url:** <http://localhost:8676/Rest/Stores/{ID}>

**HTTP Method:** GET

The following is an example response Json body:

```
{  
  "Store":{  
    "Identifier":"1627aea5-8e0a-4371-9022-9b504344e724",  
    "Aliases":["String content"],  
    "AvailableSpace":9223372036854775807,  
    "Location":"String content",  
    "Name":"String content",
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
        "Offline":true,  
        "StorageMode":0,  
        "TotalSpace":9223372036854775807  
    }  
}
```

This method returns information about the Vantage Managed Store with the specified ID. An example invocation would be:

<http://localhost:8676/Rest/Stores/41FA638D-FEC6-432E-8841-13B5CAEEDCC4>

NOTE: If the ID that is provided to this call does not exist within the Vantage domain, the following shall be returned:

```
{"Store":null}
```

This call will return the store regardless of the state of the store being Online or Offline.

An optional query exists to allow an API client to request whether a given Vantage Store was currently in use.

The new endpoint is:

<http://localhost:8676/Rest/Stores/<Store ID>/InUse>

This call will return information about a Vantage Store, including a value which indicates whether Vantage believes the Store is in-use.

For example, if the call is made with the Storage ID of: 13060a84-344a-40cf-9f75-b106b563cc25, ie:

<http://localhost:8676/Rest/Stores/13060a84-344a-40cf-9f75-b106b563cc25/InUse>

and that store was currently in use by Vantage, the following would be returned:

```
{  
2   "Store": {  
3       "Identifier": "13060a84-344a-40cf-9f75-b106b563cc25",  
4       "Aliases": [],  
5       "AvailableSpace": 942331715584,  
6       "Location": "file:///C:/VantageStore/",  
7       "Name": "Local Store",  
8       "Offline": false,  
9       "StorageMode": 0,  
10      "TotalSpace": 999677751296  
11  },  
}
```



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
12 "StoreInUse": true  
13 }
```

The new field is highlighted above.

## Examples

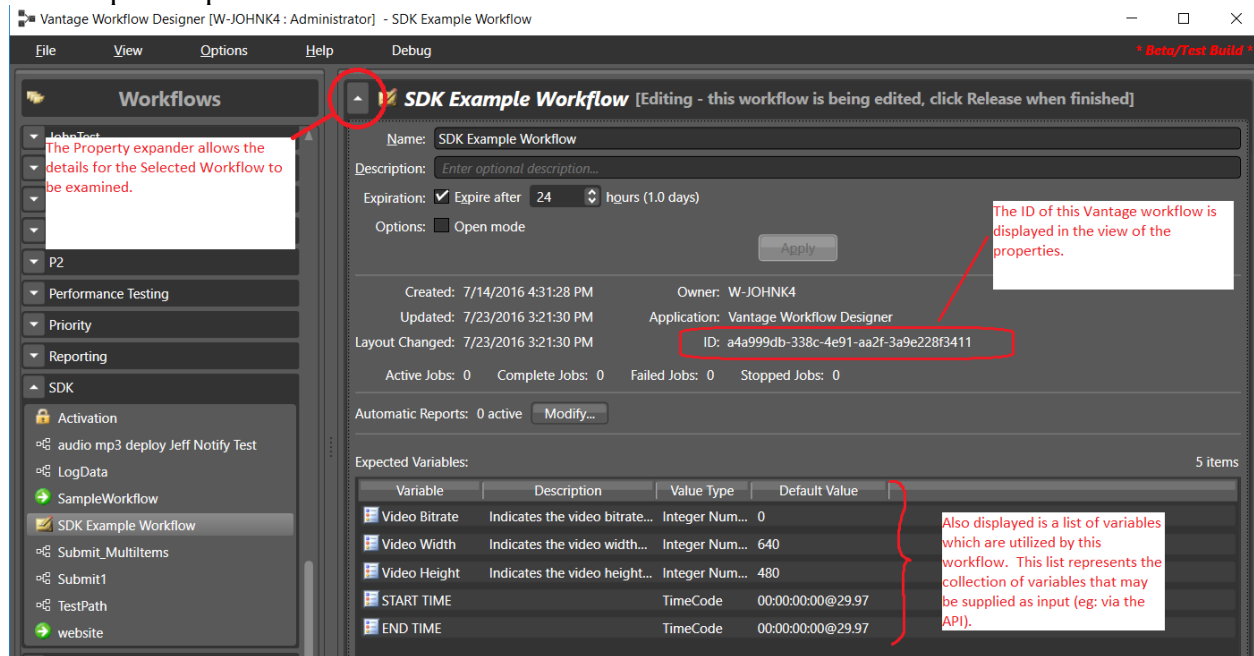
The postman utility () is utilized as the primary mechanism for exercising the Rest API. The sample below uses postman to obtain information about a workflow and submit a job to a workflow.

As mentioned in the introduction, workflows in Vantage are comprised of different Actions. Workflows may have many different actions depending upon the desire of the workflow creator. Each action is comprised of different Parameters. Parameters hold values for individual properties within an action. A Parameter may contain a static value (entered directly via the Vantage workflow designer) or a Parameter may be bound to a Vantage variable. If a Parameter is bound to a Variable, then the value of that Parameter may be supplied at runtime (via an API submit for example).

See the Vantage User’s Guide for more information about binding variables within a Vantage Action.

The Vantage workflow designer makes it easy to visual inspect a workflow to determine what it will indicate may be provided as input.

For a given workflow within the Workflow Designer, the Properties view may be used to obtain specific pieces of information:

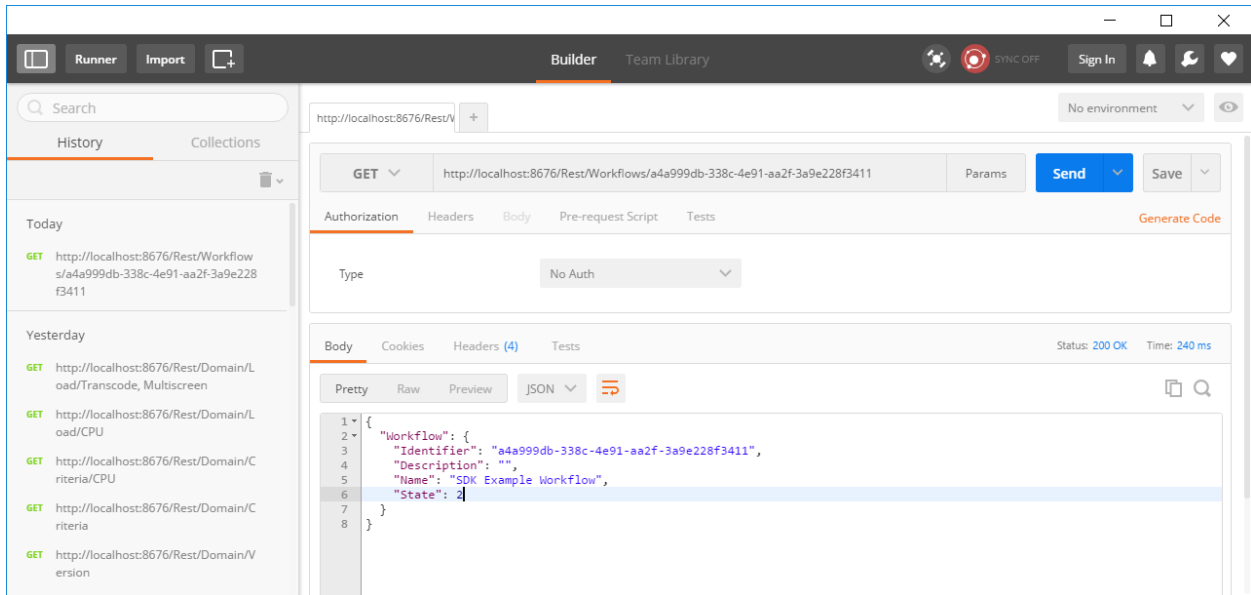


Of particular interest for this example is the workflow ID: [a4a999db-338c-4e91-aa2f-3a9e228f3411](#)

This is the workflow that will be utilized within this example.

## Submit a Job

Using postman, the first task is to ask for information about the workflow:

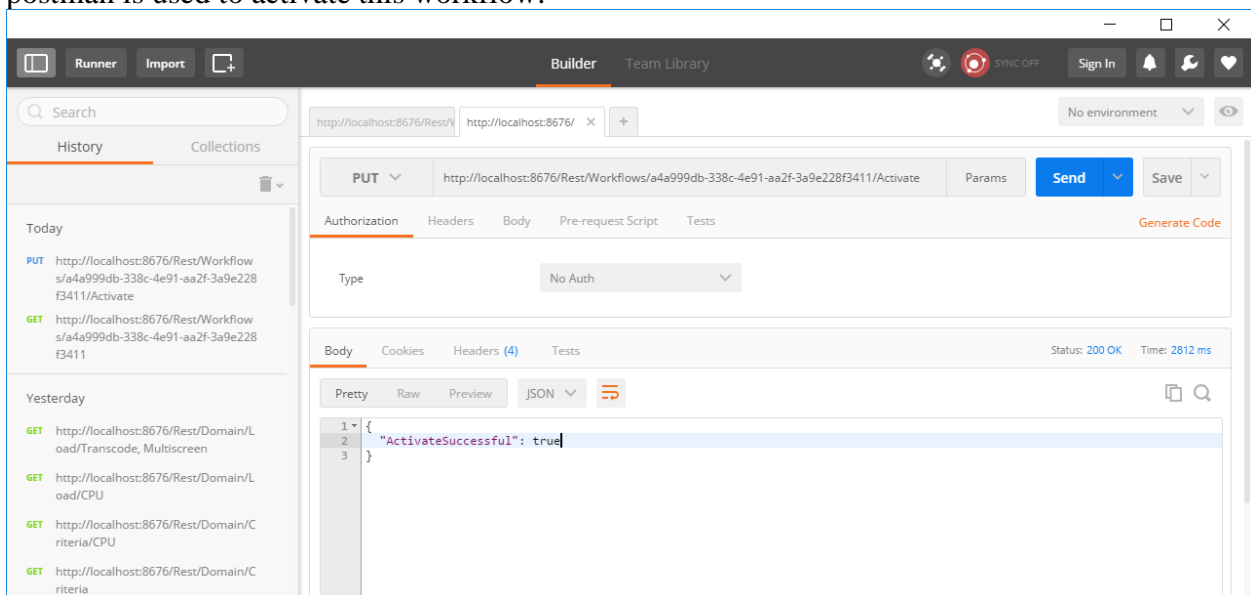


The screen above represents what is observed in postman when the endpoint: <http://localhost:8676/Rest/Workflows/a4a999db-338c-4e91-aa2f-3a9e228f3411> is used to obtain information about the workflow.

NOTE: The ‘State’ of the workflow is 2. From the information in the /Workflows endpoint in the previous section, the value of 2 corresponds to:

Edit 2 The workflow is currently being edited by a user. The workflow may be activated.

Since a workflow in the Edit state must be ‘Activated’ before it may be ‘Submitted to’; postman is used to activate this workflow:

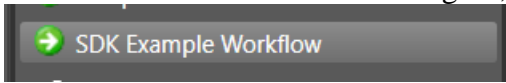


NOTE: The ‘*ActivateSuccessful*’ property above has returned a **true** value.

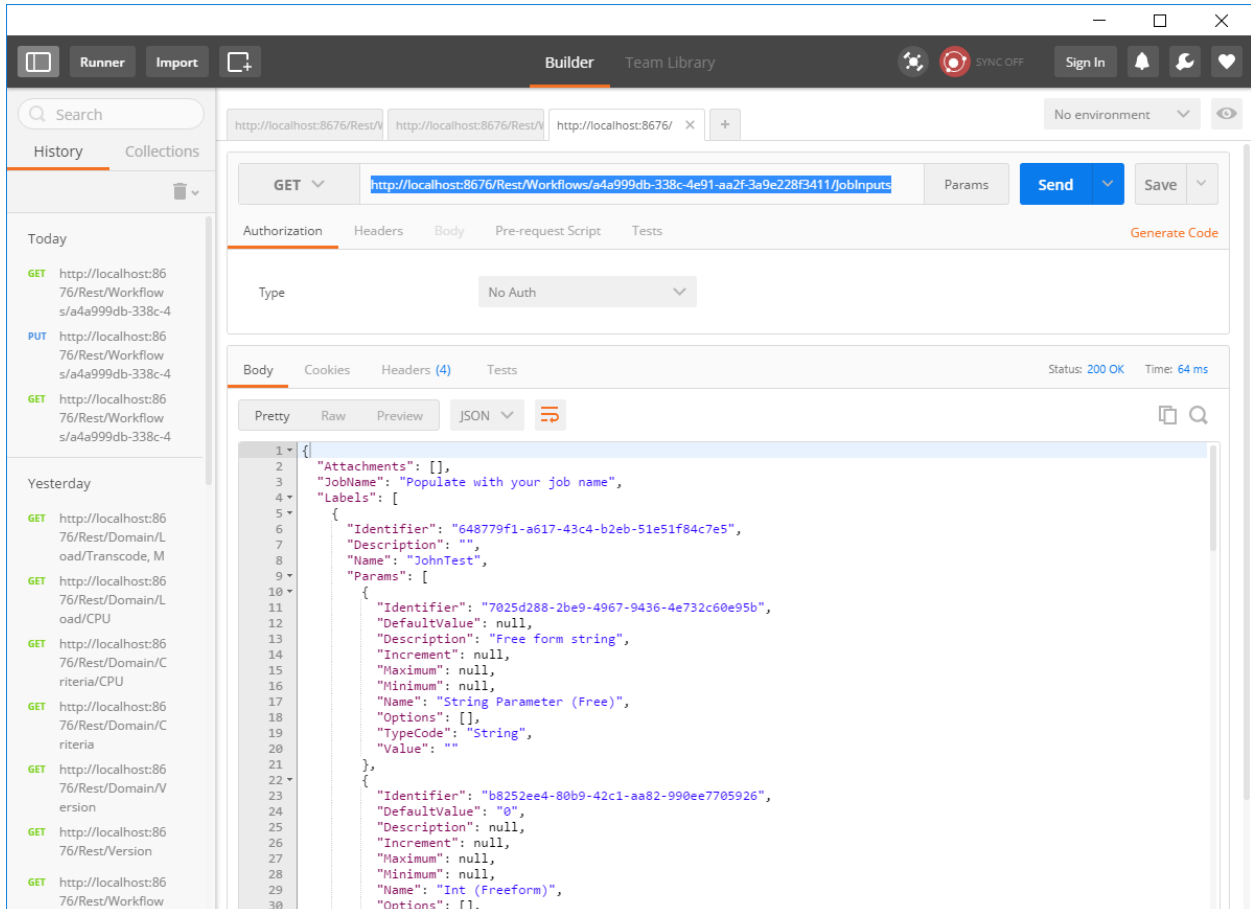


848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

NOTE: Within the workflow designer, the workflow is now activate as well:



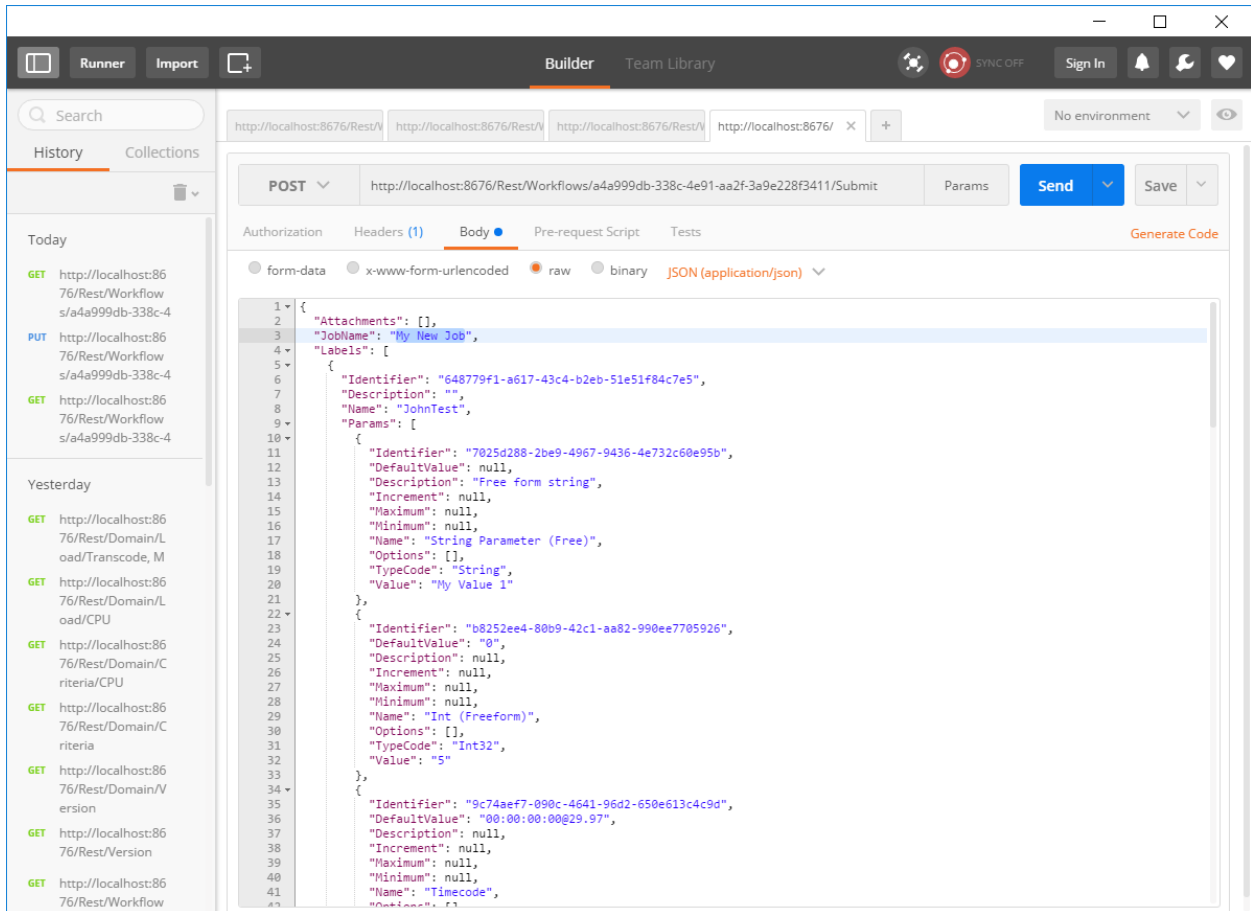
At this point, the endpoint: <http://localhost:8676/Rest/Workflows/a4a999db-338c-4e91-aa2f-3a9e228f3411/JobInputs> is used to obtain the Inputs that must be supplied to start a job within this workflow:



Postman is now used to generate a POST request to the /Submit endpoint. The JSON data from the JobInputs GET output above is cut/paste into the new postman tab, select information is then updated (to correspond to the new information that is desired to be included as part of the new job):



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)



NOTE: JobName was set to 'My New Job'

Scrolling down in the data that defines the new job, the desired input files (C:\Media\demo.mpg and C:\Media\demo1.mpg) are specified:

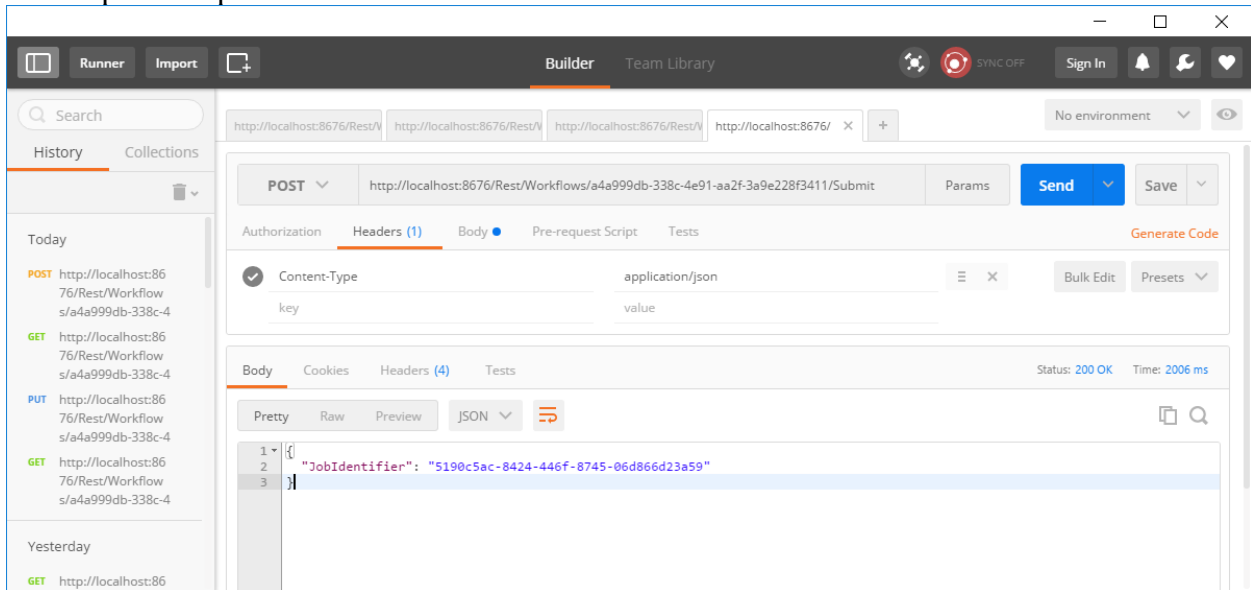
```
130 "Medias": [
131   {
132     "Identifier": "abb3f7e5-8d7e-49ba-84c5-674a55edb700",
133     "Description": "The original version of content encountered or created by Vantage.",
134     "Files": [
135       "C:\\Media\\demo.mpg"
136     ],
137     "Name": "Original"
138   },
139   {
140     "Identifier": "56079dff-e070-4aeb-9e47-a7c1831d586a",
141     "Description": "",
142     "Files": [
143       "C:\\Media\\demo1.mpg"
144     ],
145     "Name": "1"
146   }
147 ],
```

Additionally, a value of 2,000,000 was used to specify the desired Bitrate (via the Video Bitrate Variable):

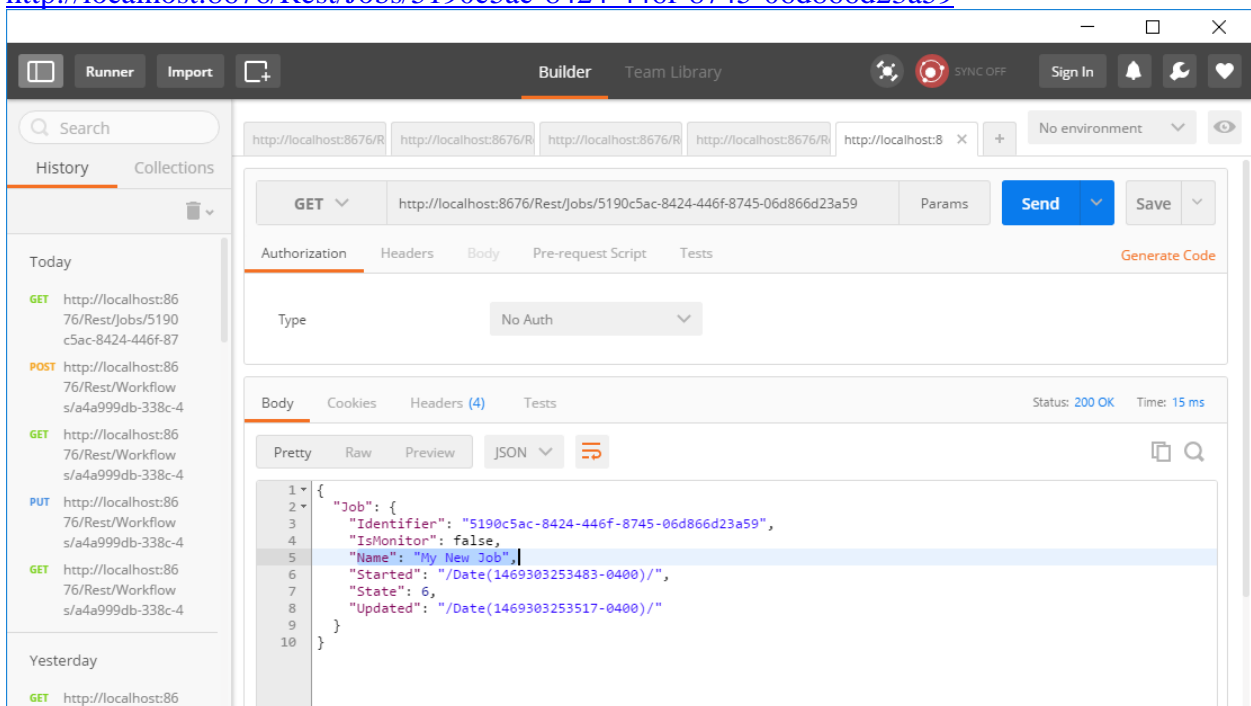
```
149 "Variables": [
150   {
151     "Identifier": "f54ac224-a295-4ff8-b0cc-b021c7650bca",
152     "DefaultValue": "0",
153     "Description": "Indicates the video bitrate of a content version, as found by the Identify service.",
154     "Name": "Video Bitrate",
155     "TypeCode": "Int32",
156     "Value": "2000000"
157   },
158 ]
```



Once the remaining data is updated to reflect the desired values for this submit; the ‘Send’ button on postman is pressed (to POST the new job):  
The response in postman can then be viewed:

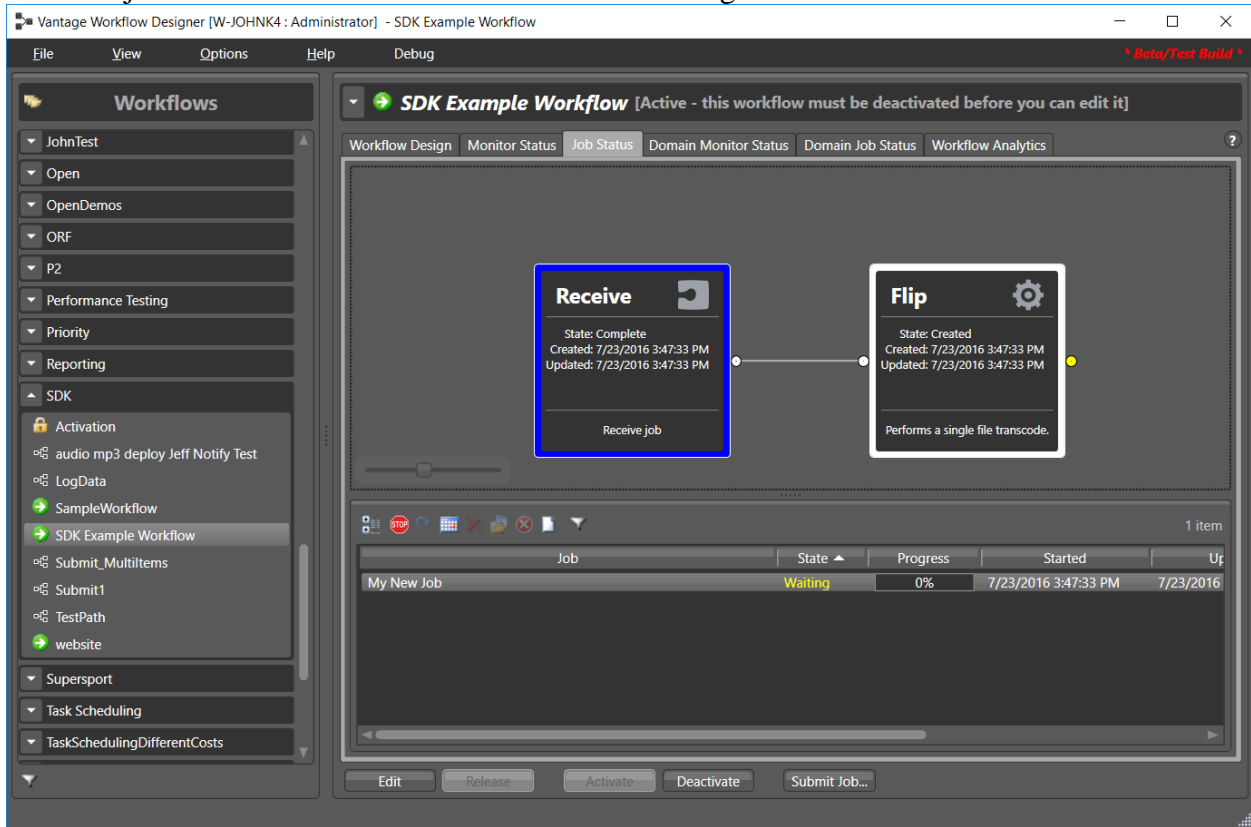


This corresponds to the identifier for the new job: [5190c5ac-8424-446f-8745-06d866d23a59](http://localhost:8676/Rest/Jobs/5190c5ac-8424-446f-8745-06d866d23a59)  
This value can then be utilized to query information about the newly created job:  
<http://localhost:8676/Rest/Jobs/5190c5ac-8424-446f-8745-06d866d23a59>



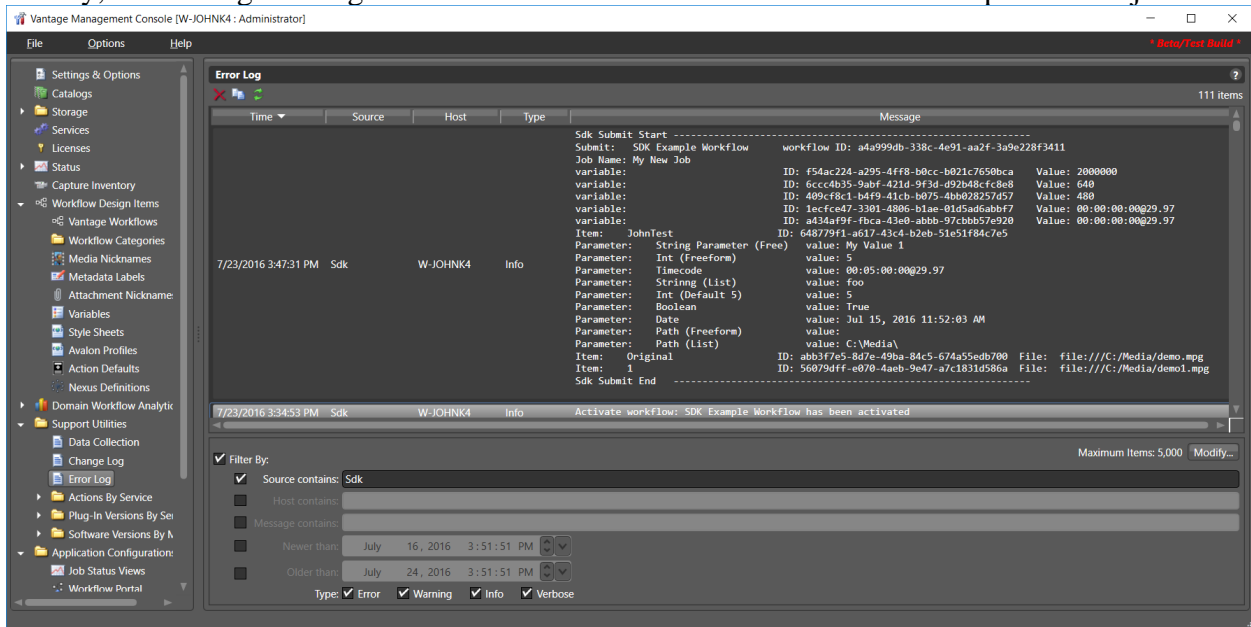
NOTE: The ‘Name’ of the job corresponds to the value specified in the preceding postman screen ‘My New Job’.

The new job can also be examined in the workflow designer:



NOTE: The name of the Job is: 'My New Job'.

Finally, the Vantage Management Console will also contain a LOG of the input for this job.





848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

The Log which is accessible via the VMC shall contain all inputs that were provided to the job. The following contains a focus on the information about this specific job:

```
111 it
Message
Sdk Submit Start -----
Submit: SDK Example Workflow      workflow ID: a4a999db-338c-4e91-aa2f-3a9e228f3411
Job Name: My New Job
variable:                          ID: f54ac224-a295-4ff8-b0cc-b021c7650bca      Value: 2000000
variable:                          ID: 6ccc4b35-9abf-421d-9f3d-d92b48cfc8e8      Value: 640
variable:                          ID: 409cf8c1-b4f9-41cb-b075-4bb028257d57      Value: 480
variable:                          ID: 1ecfce47-3301-4806-b1ae-01d5ad6abbf7      Value: 00:00:00:00@29.97
variable:                          ID: a434af9f-fbca-43e0-abb-97cbbb57e920      Value: 00:00:00:00@29.97
Item: JohnTest                      ID: 648779f1-a617-43c4-b2eb-51e51f84c7e5
Parameter: String Parameter (Free)  value: My Value 1
Parameter: Int (Freeform)           value: 5
Parameter: Timecode                 value: 00:05:00:00@29.97
Parameter: String (List)            value: foo
Parameter: Int (Default 5)          value: 5
Parameter: Boolean                   value: True
Parameter: Date                      value: Jul 15, 2016 11:52:03 AM
Parameter: Path (Freeform)           value:
Parameter: Path (List)              value: C:\Media\
Item: Original                      ID: abb3f7e5-8d7e-49ba-84c5-674a55edb700      File: file:///C:/Media/demo.mpg
Item: 1                             ID: 56079dff-e070-4aeb-9e47-a7c1831d586a      File: file:///C:/Media/demo1.mpg
Sdk Submit End -----
```

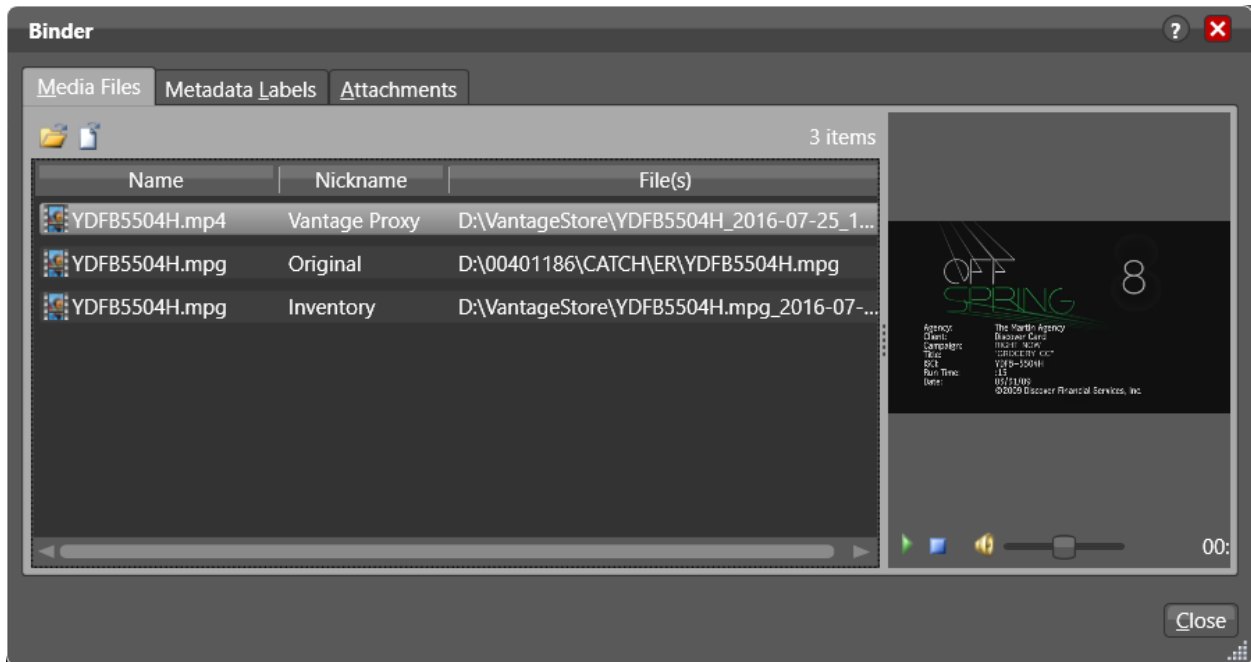
NOTE: the Job Name: 'My New Job', the value of 2,000,000 as part of the variables, and the two (2) inputs: C:/Media/demo.mpg and C:/Media/demo1.mpg (which had previously been specified as inputs during the submit).

### Obtain Job Outputs

This example describes what is entailed with respect to collecting job Outputs following a successful job. In this example, a job has been previously submitted to Vantage and has successfully completed. The intent of the example is to illustrate how a completed job appears in Vantage and how the API may obtain information about it.

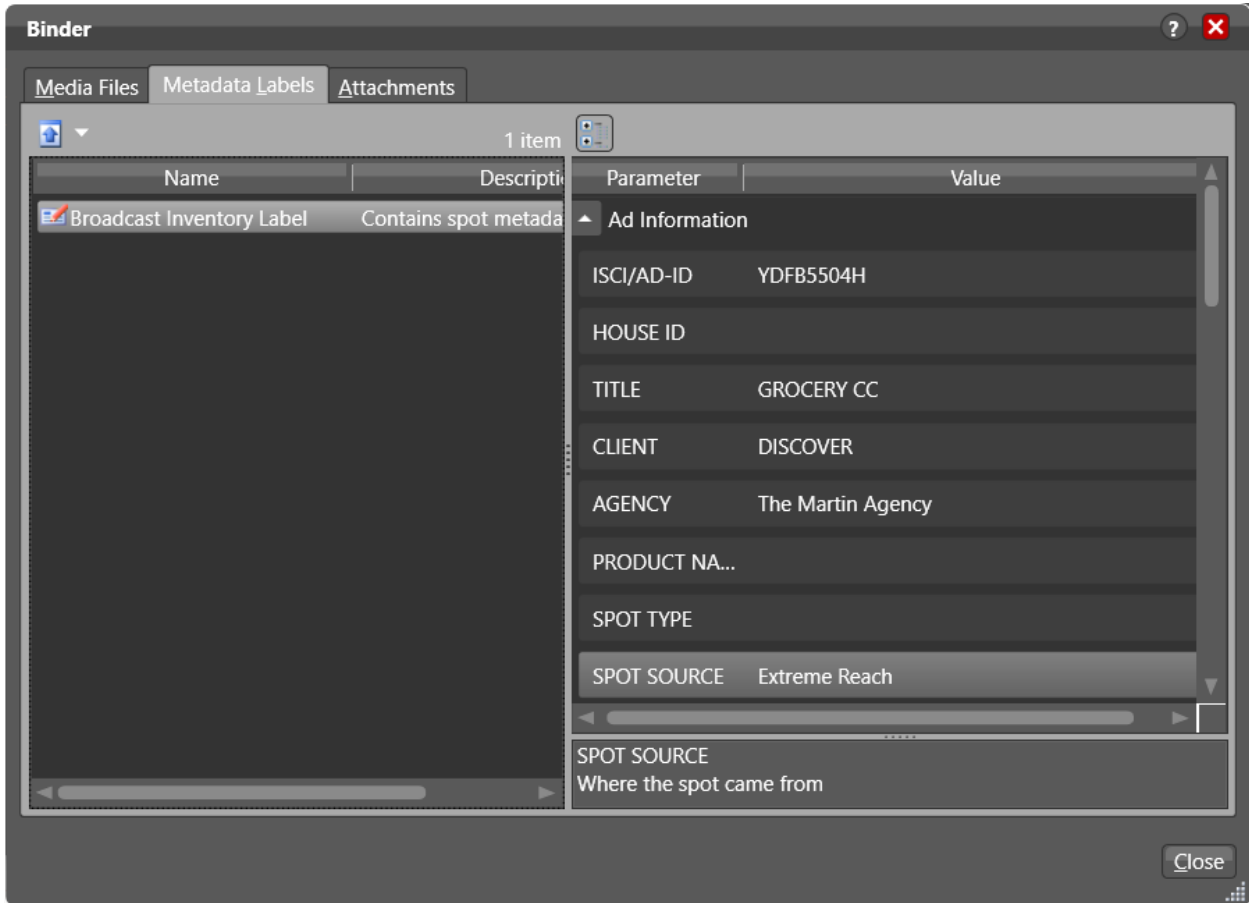
As a reference, here is a completed job within the Vantage Workflow Designer:





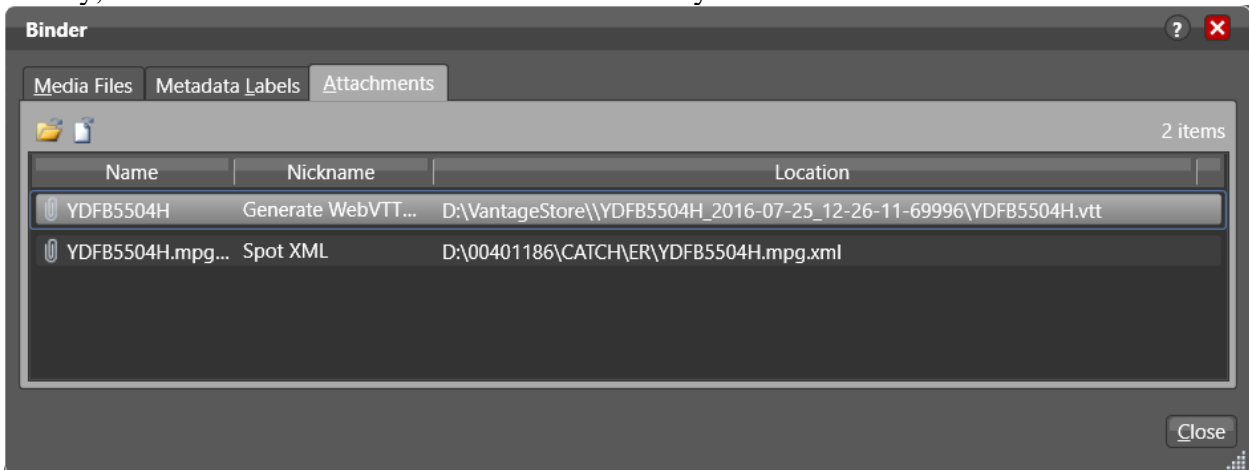
This screen illustrates the three (3) Media Nicknames present in this job.

The second tab allows an examination of the Metadata Labels associated with this job:



NOTE: There is a single Metadata Label ('Broadcast Inventory Label') that is present in this job. The right side of the view allows a user to inspect the various Parameters that are present in the selected Label.

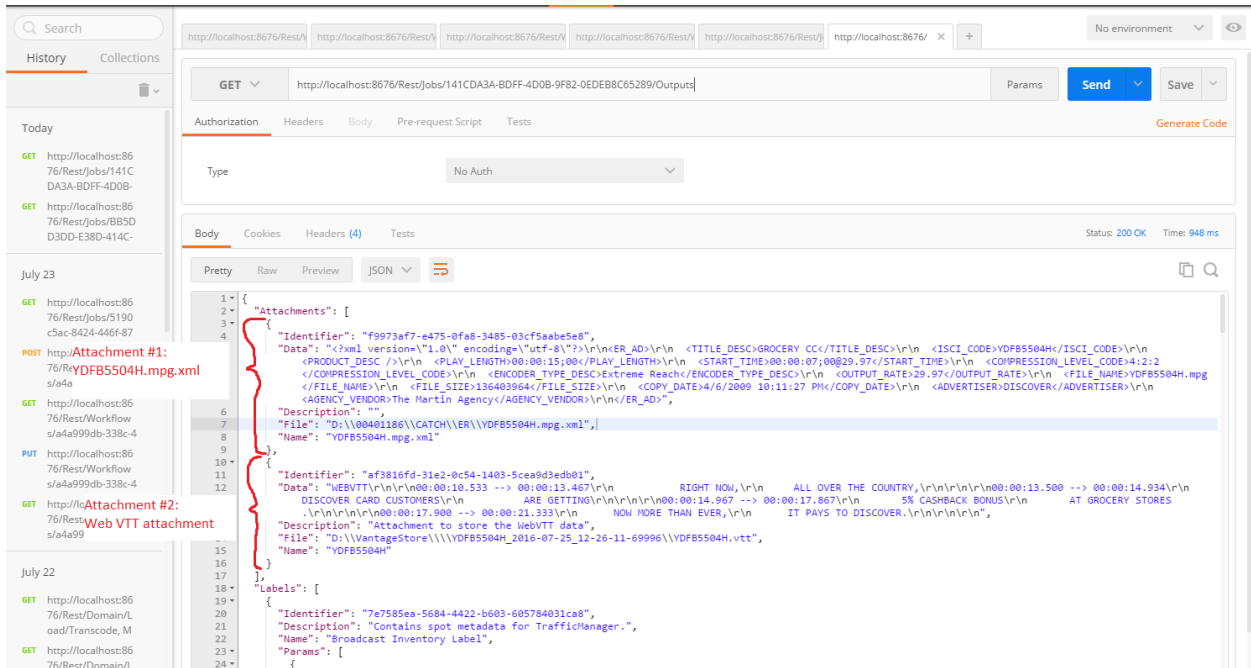
Finally, the Attachment view allows a user to view any Sidecar XML or other Metadata files:



In this case, there are two (2) metadata attachments (Spot XML and a Web VTT script). These attachments each have a distinct file on disk that is referenced.

When the postman utility is utilized to ask for the Output for this job (the job id is: [141CDA3A-BDFF-4D0B-9F82-0EDEB8C65289](http://localhost:8676/Rest/Jobs/141CDA3A-BDFF-4D0B-9F82-0EDEB8C65289) -- this is assumed to be known ahead of time), the expectations (based upon what was present in the Binder view of the workflow designer); is to receive three (3) Medias, 1 Metadata label and 2 Attachments.

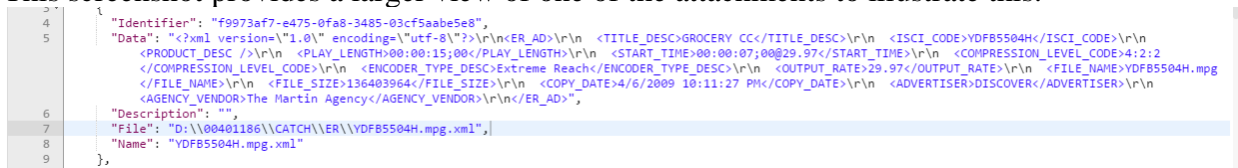
Here is the postman generated output when the <http://localhost:8676/Rest/Jobs/141CDA3A-BDFF-4D0B-9F82-0EDEB8C65289/Outputs> :



The above screen illustrates the outputs (focused on the attachments).

**NOTE:** Each attachment includes the path to the attachment and the data that is present in the file that is referred to (The 'Data' property contains the information within the file that is pointed to).

This screenshot provides a larger view of one of the attachments to illustrate this:



The Labels section of the output in postman corresponds to the Metadata Labels which are available in the output:

```

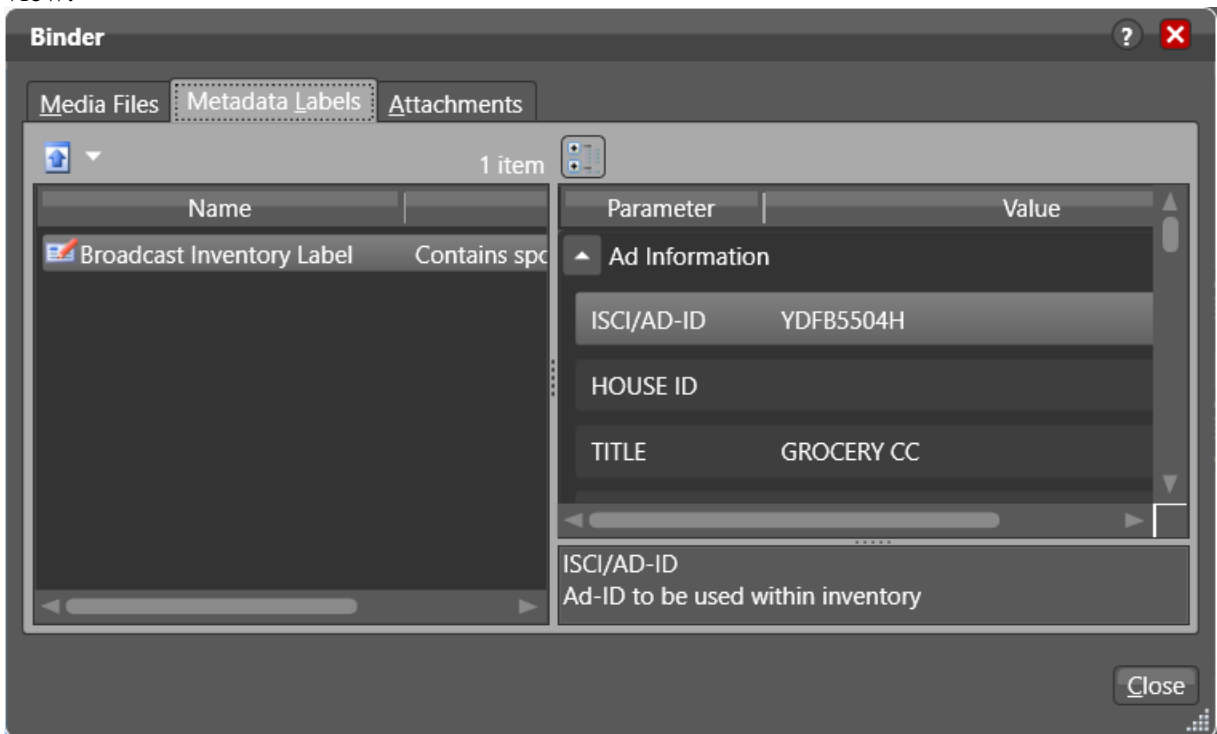
15     "Name": "YDFB5504H"
16   },
17 ],
18 "Labels": [
19   {
20     "Identifier": "7e7585ea-5684-4422-b603-605784031ca8",
21     "Description": "Contains spot metadata for TrafficManager.",
22     "Name": "Broadcast Inventory Label",
23     "Params": [
24       {
25         "Identifier": "374c8a7d-a72d-4cbe-aeec-5f33b6d35478",
26         "DefaultValue": "",
27         "Description": "Ad-ID to be used within inventory",
28         "Increment": null,
29         "Maximum": null,
30         "Minimum": null,
31         "Name": "ISCI/AD-ID",
32         "Options": [],
33         "TypeCode": "String",
34         "Value": "YDFB5504H"
35       },
36     ],
37     "Identifier": "4679c0fe-e6be-4d67-a0e3-bcc34250c67f",
38     "DefaultValue": "",
39     "Description": "House ID of the commercial",
40     "Increment": null,
41     "Maximum": null,
42     "Minimum": null,
43     "Name": "HOUSE ID",
44     "Options": [],

```

Label Name

Value of the parameter named:  
 ISCI/AD-ID is: 'YDFB5504H'

NOTE: The parameters that are present in the Label above (Broadcast Inventory Label), correspond to the value of the parameters listed in the Metadata Label section of the binder view:



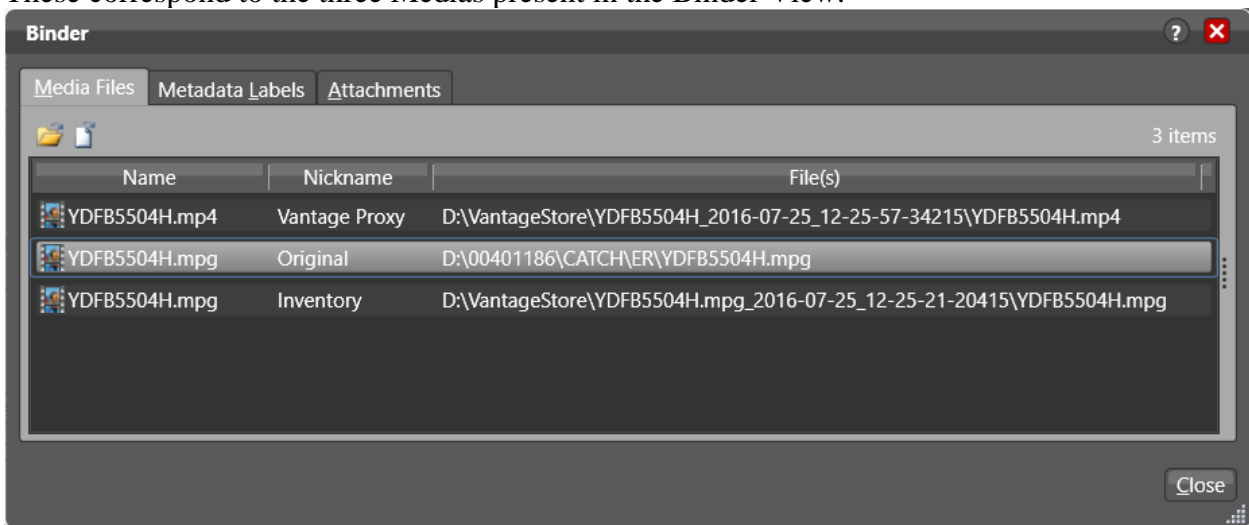
Finally, the output of postman includes references to the three (3) Medias which are present:



```

454 "Medias": [
455 {
456   "Identifier": "abb3f7e5-8d7e-49ba-84c5-674a55edb700",
457   "Description": "The original version of content encountered or created by Vantage.",
458   "Files": [
459     "D:\\00401186\\CATCH\\ER\\YDFB5504H.mpg"
460   ],
461   "Name": "Original"
462 },
463 {
464   "Identifier": "21ed0d15-7473-4c83-bed6-324ca44a7c5f",
465   "Description": "",
466   "Files": [
467     "D:\\VantageStore\\YDFB5504H.mpg_2016-07-25_12-25-21-20415\\YDFB5504H.mpg"
468   ],
469   "Name": "Inventory"
470 },
471 {
472   "Identifier": "f1a4e2d6-1848-4512-87b7-b82078b8bc6b",
473   "Description": "A proxy version capable of being rendered by the Vantage preview component.",
474   "Files": [
475     "D:\\VantageStore\\YDFB5504H_2016-07-25_12-25-57-34215\\YDFB5504H.mp4"
476   ],
477   "Name": "Vantage Proxy"
478 }
479 ]
480 }
  
```

These correspond to the three Medias present in the Binder View:



### Submit In-Band Material

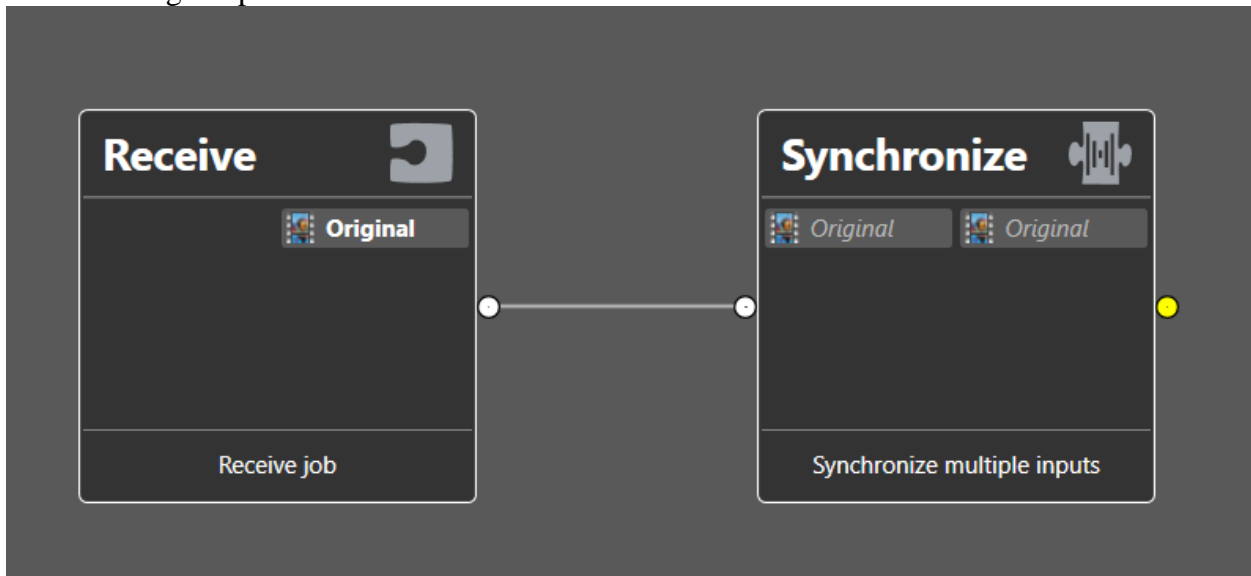
This example illustrates how to transmit in-band data vs. referencing a file which resides on a filesystem during a submission.

In this example, the desire is to submit a CML content representation to a workflow for processing. The CML data is desired to be send in-band so that an on-disk file need not be created.

The sample CML file is very simple:

```
<Composition created="2013-06-25T11:06:59.0960428-07:00"  
xmlns="Telestream.Soa.Facility.Playlist">  
  <Source identifier="1">  
    <File location="D:\\Watchmen.mov" />  
    <Subtitle>  
    </Subtitle>  
  </Source>  
  
  <Sequence layer="0">  
    <Segment>  
      <Video align="head" adjust="edge" fill="none" source="1" layer="0" >  
  
      </Video >  
    </Segment>  
  </Sequence>  
</Composition>
```

The workflow that is the target of this submission is a very simple workflow that is set to receive a single input:



When this workflow is asked for its input requirements, the following is observed:



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

The screenshot shows a REST client interface. At the top, a GET request is defined for the URL `http://localhost:8676/Rest/Workflows/c79ec36f-4deb-4d8e-a48c-25d5e3f4e098/JobInputs`. The response status is 200 OK and the time taken is 755 ms. The response body is displayed in JSON format, with the following structure:

```
1 {
2   "Attachments": [],
3   "JobName": "Populate with your job name",
4   "Labels": [],
5   "Medias": [
6     {
7       "Identifier": "abb3f7e5-8d7e-49ba-84c5-674a55edb700",
8       "Data": "If sending in-band data (eg: CML); place a BASE64 encoded version of the data in this field and do not
9         send a file path.",
10      "Description": "The original version of content encountered or created by Vantage.",
11      "Files": [
12        "full pathname to file"
13      ],
14      "Name": "Original"
15    }
16  ],
17  "Priority": 0,
18  "Variables": []
19 }
```

The response to the <http://localhost:8676/Rest/Workflows/c79ec36f-4deb-4d8e-a48c-25d5e3f4e098/JobInputs> endpoint indicates the structure of the input that this workflow expects.

The 'Data' property for the single Media Input expected by this workflow (the Original nickname) is how the embedded CML shall be submitted.

The first step to transmit the CML as in-band data is to convert the String representation of the CML into a UTF8 Base64 encoded string.

This was done via the website: <https://www.base64encode.org/>

The following allows an easy transformation from the original CML string to a base64 encoded representation:

The screenshot shows a web browser window with the URL <https://www.base64encode.org>. The page has a green background with a circular pattern. At the top, it says "BASE64 Decode and Encode". Below this, there are two buttons: "Decode" and "Encode". To the right, there is a "Do You Like us?" section with a "Like 816" button and a "Thanks... In advance!" message. The main content area is titled "Encode to Base64 format" and includes a text input field with the following XML code:

```
<Composition created="2013-06-25T11:06:59.0960428-07:00"
xmlns="Telestream.Soa.Facility.Playlist">
<Source identifier="1">
<File location="D:\Watchmen.mov" />
<Subtitle>
</Subtitle>
</Source>

<Sequence layer="0">
<Segment>
<Video align="head" adjust="edge" fill="none" source="1" layer="0" >
```

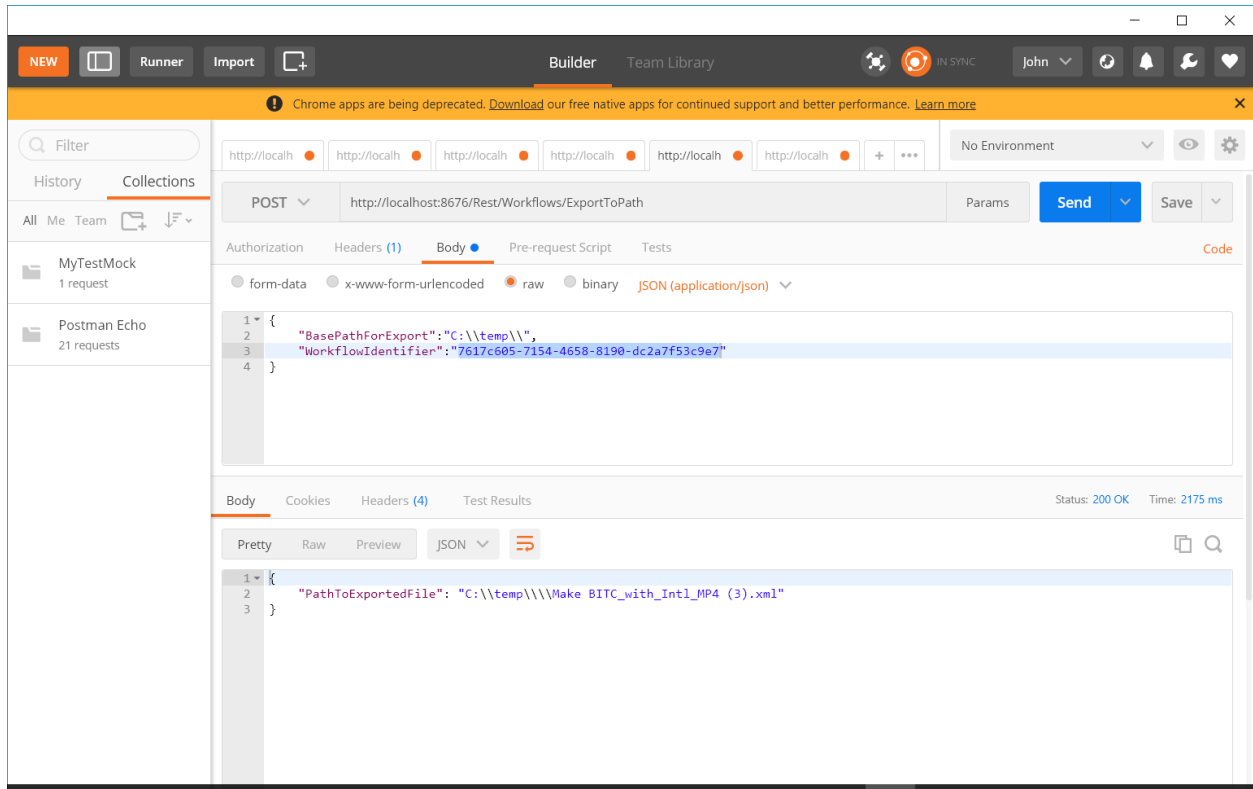
Below the input field, there is a green button labeled "> ENCODE <" and a dropdown menu set to "UTF-8". Below the dropdown, there is a text area containing the encoded Base64 output:

```
PENvbXBvc2l0aW9uIGNyZWZ0ZWQ9IjIwMTM0MjU1MTY1MDY0MDY6NTkuMDk2
MDQyOC0wNzowMCIgeG1sbnM9IIRlbGVzdHJlYXUyZ29hLkZyY2lsaXR5LIBsYXIsa
XN0Ij4NCiAgPFNvdXJjZSBpZGVudGlnaWVvPSixlj4NCiAgIjCA8RmlsZSBsb2NhdGI
vbj0iRDpcXFdhZGNobWVuLm1vdilgLz4NCgk8U3VidGl0bGU+DQoJPC9TdWJ0aXRz
ZT4NCiAgPC9Tb3VyY2U+DQoNCiAgPFNlcXVlbnNIIiGxheWVvPSlwlj4NCiAgIjCA8U
2VnbWVudD4NCgkJPFFpZGVvIjGFsaWduPSJoZWFKIiIBhZGp1c3Q9ImVkZ2UuIGZp
bGw9Im5vbmUiIHVvdXJjZT0iMSIjgbGF5Zl91IjAid4NCgk8L1ZpZGVvIjD4NCiA
glCA8L1NIZ21bnQ+DQoqLDwyU2VxdWVvY2U+DQo8L0NvbXBvc2l0aW9uPg==
```

Blue brackets on the right side of the screenshot point to the original XML code and the encoded Base64 output.

The Encoded representation is then cut/paste from the website above into the job input data in postman:

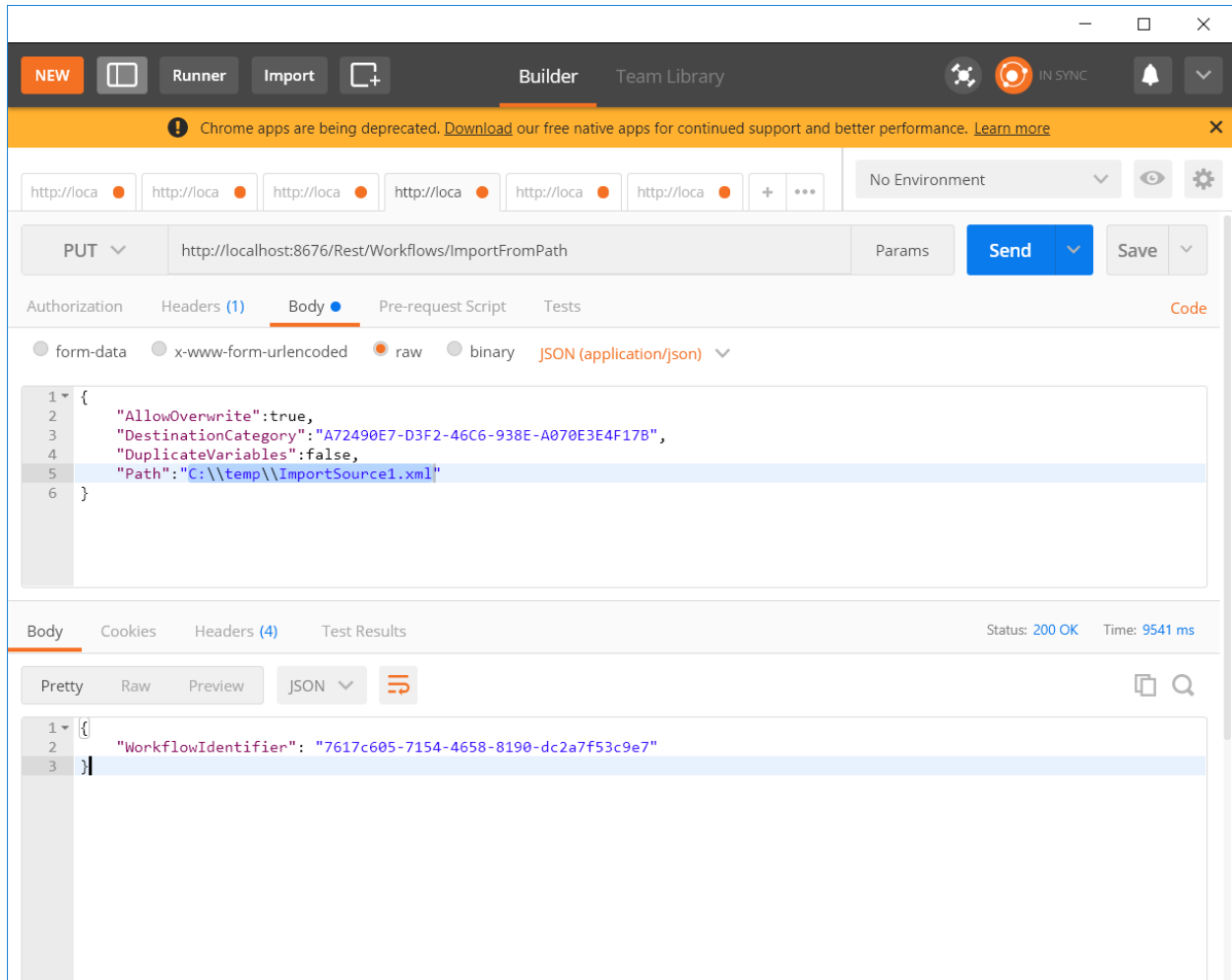




The result of this POST command is a JSON response which indicates the complete path of the XML file created as a result of the Export. NOTE: The filename includes characters required to ensure a unique name on the filesystem.

The corresponding ImportFromPath call allows a workflow which was previously exported via the mechanism above (ExportToPath) OR via a manual export of a workflow from the Workflow Designer.

The ImportFromPath call is illustrated below:



In this scenario; the source of the Import was the workflow at the path:  
C:\Temp\ImportSource1.xml

NOTE: This path must be accessible to the SDK service that is the target of this method. If the path is not accessible; the import shall NOT succeed.

Additional parameters may be specified which govern the ability to overwrite an existing workflow, whether variables should be duplicated or not and if a specific category should be used (in the event that the specified workflow does not already exist in the domain). See the documentation for this method for more information regarding these parameters.

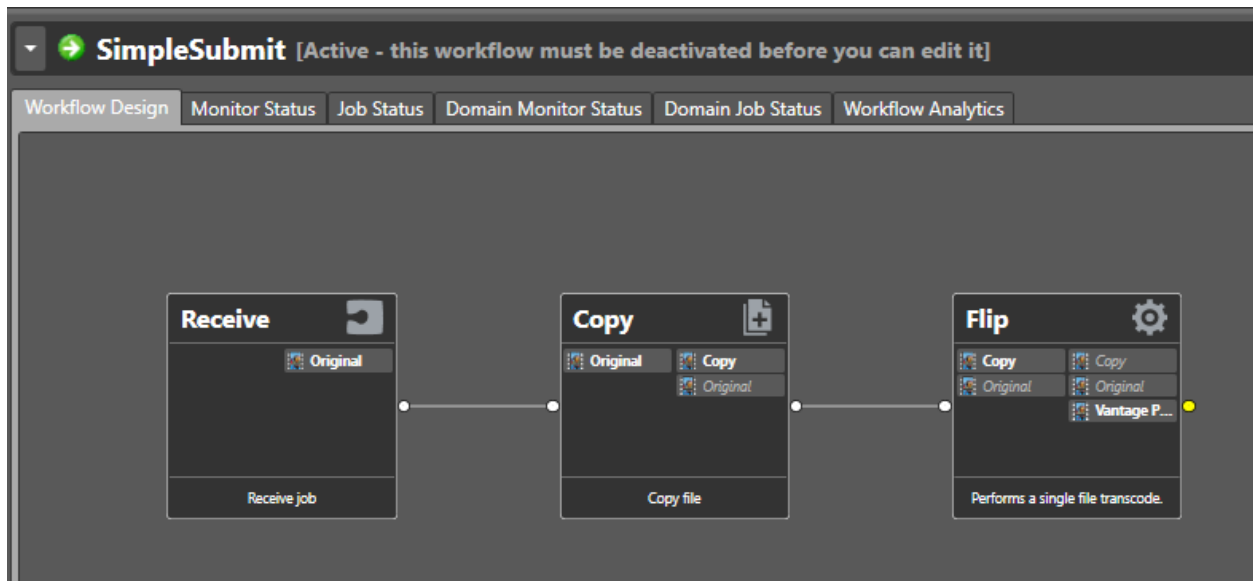
The result of this call (the JSON response) shall contain a value which corresponds to the Identifier of the newly imported workflow. If the value that is returned is Guid.Empty (i.e.: all 0's); this implies an error. Otherwise the GUID value shall correspond to the identifier of the newly imported workflow (which may be used in subsequent calls).

NOTE: The state of newly imported workflows shall be the Edit state.

## Submitting Remote Files

This section details changes introduced in Vantage 7.2 which facilitate the submission of files which reside on remote locations (e.g.: S3 filesystems, FTP sites, etc..). The challenge with submitting files from these locations has always been centered around the construction of the properly formatted URL that Vantage can understand. Vantage expects very specific URL schemes and formats for references to material which exist on remote location. This section discusses a new feature introduced in Vantage 7.2 which greatly simplifies this process.

In this example, a complete job submission shall be examined starting with the specification of a target workflow. The sample workflow is illustrated below:



This workflow will take any file it receives, it will copy it to a local Vantage store and then transcode the local version into a new format.

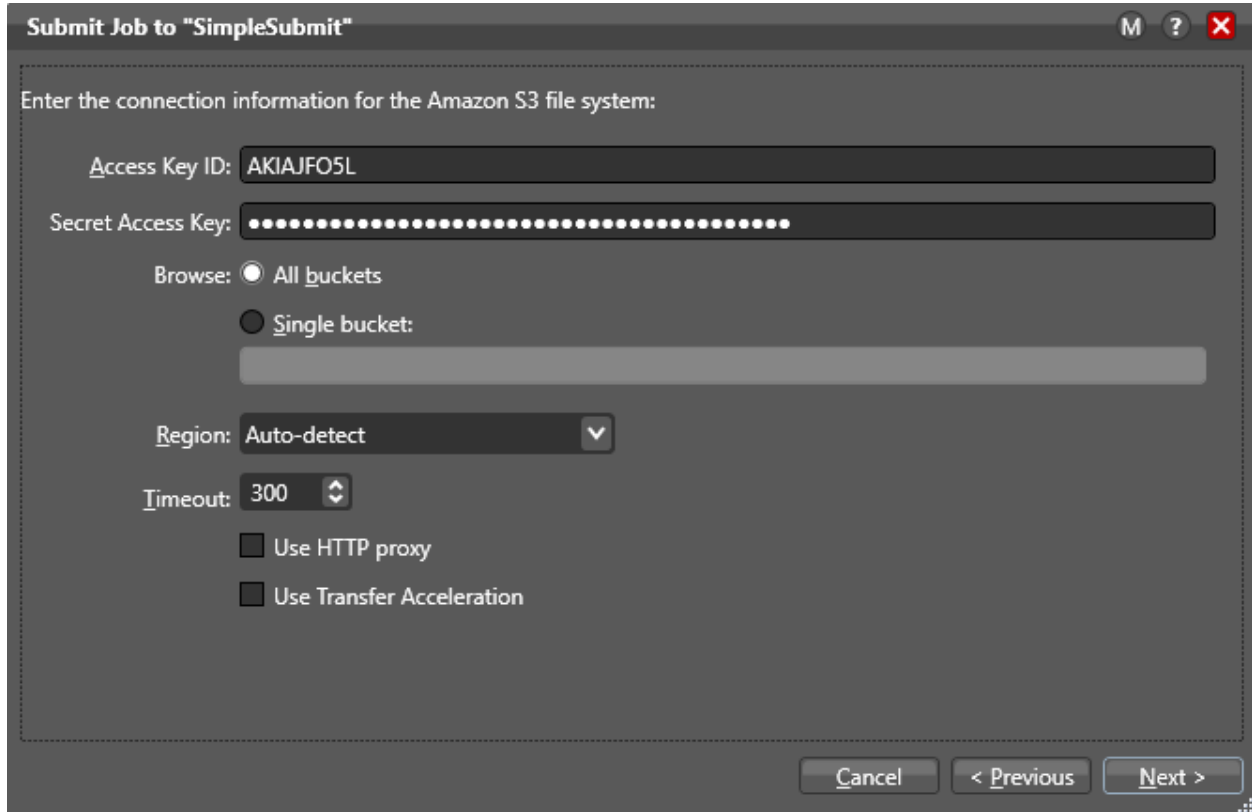
The challenge for most API users is how to properly specify the syntax to the source file, when the source file resides on a remote (non-windows) filesystem (e.g.: FTP, S3, etc.)

The new capability which has been introduced in Vantage 7.2, allows an Administrative user to request the full Vantage URL for a given file reference. (**NOTE:** Administrative privileges are required within Vantage since this capability will display any passwords/keys associated with the remote file).

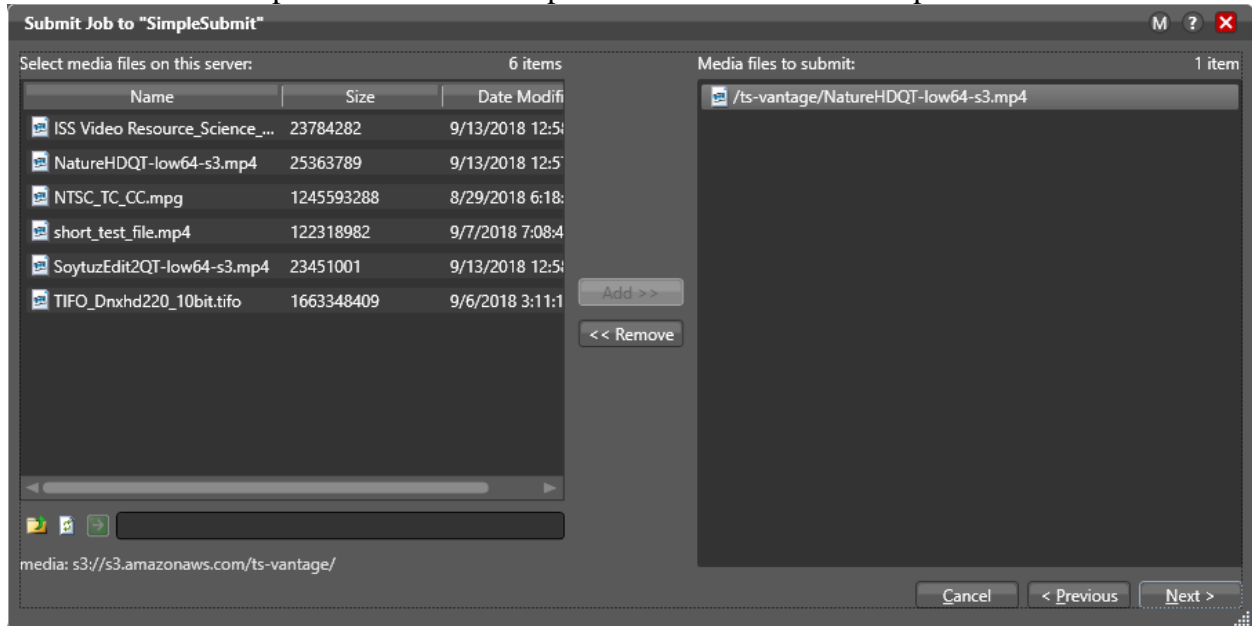
The easiest way to get a starting file is to simply use Vantage to perform a manual submit of a file which is representative of files you ultimately wish to submit via the API. For example, if you have a large collection of files in an S3 bucket that you wish to submit via the API, we will perform a manual submission of one of these files to test the Vantage interaction with the S3 location (i.e.: verify that Vantage can reach the location with the specified credentials).



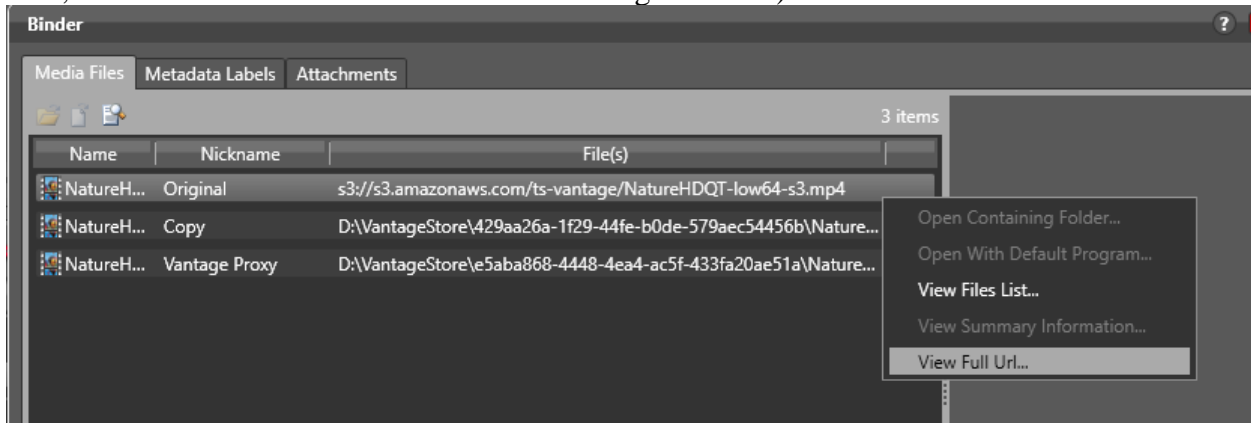
In the manual submission dialog, the S3 filesystem is selected and the appropriate credentials are entered:



The Next> button is pressed to allow the specification of the desired input file:

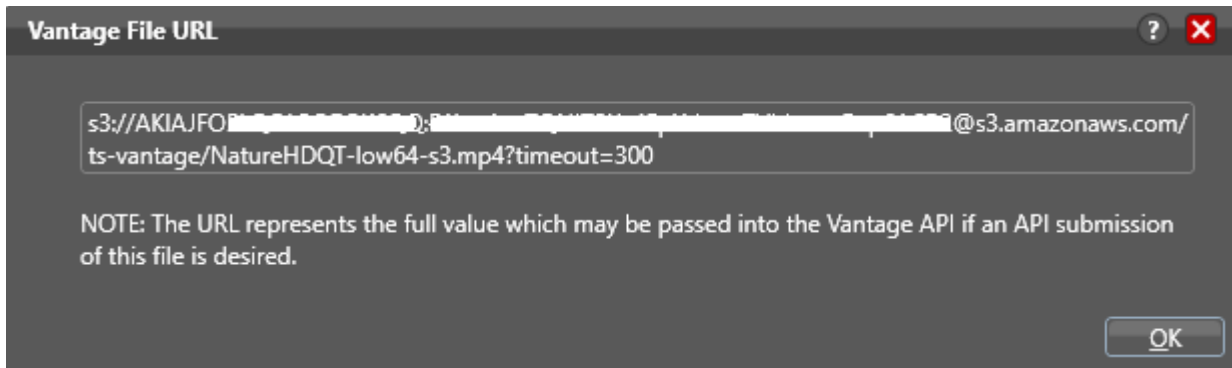


Once the job has been submitted and completed, the Binder Details view may be opened, and the new feature is exposed when the context menu is displayed for the desired media file (in this case, the media file which resides on the S3 storage location):



Again, the new option: View Full URL... , is **ONLY** available to an account logged into the workflow designer which has Administrator privileges. If the account does not have Administrator privileges, this option shall be disabled, and the feature will not be accessible. Further, if user administration is disabled, the feature is not enabled. User Administration must be enabled, and the account must be an Administrator in order to access this capability.

Once the option is selected, the full URL is displayed in a new dialog window:



NOTE: The keys above are obscured in this graphic only, on your system they would be visible.

This value can then be copied and pasted into a REST submission query. It represents the full URL to the file on the specified storage. Further, any custom parameters which were controlled by the user interface are also present (in this case the timeout value was specified. Other scenarios may be passive mode for an FTP file transfer. All of these parameters would be visible in the full URL above.)



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

This URL would represent that structure that Vantage expects to receive for any material located in the selected bucket for this S3 endpoint. If a different file is desired to be selected, the API client simply needs to replace the filename in the text above, with the new filename. For example, a manual submission of the file named: NatureHDQT-low64-s3.mp4 was performed above.

If an API submission of a 2<sup>nd</sup> file located in this bucket (NTSC\_TC\_CC.mpg) was desired; this could be accomplished by copying and pasting the URL above into an API submission query (after replacing the old name with the new name).

This is illustrated below (using PostMan to send a new job to this same workflow):



Once this request is sent to Vantage, a new job is started in which the specified file (NTSC\_TC\_CC.mpg) is processed:

The screenshot shows the SimpleSubmit workflow interface. At the top, the workflow is titled "SimpleSubmit [Active - this workflow must be deactivated before you can edit it]". Below the title are tabs for "Workflow Design", "Monitor Status", "Job Status", "Domain Monitor Status", "Domain Job Status", and "Workflow Analytics". The main area displays a workflow diagram with three steps: "Receive", "Copy", and "Flip". Each step is a box with a title, a state indicator (State: Complete), creation and update timestamps (Created: 9/26/2018 9:43:20 AM, Updated: 9/26/2018 9:43:20 AM), and a machine name (Machine: WE-JOHNK). The "Receive" step has a "Receive job" button, the "Copy" step has a "Copy file" button, and the "Flip" step has a description "Performs a single file transcode." Below the workflow is a table with columns for Job, State, Progress, Started, Updated, Compute Time, and Elapsed Time. The table shows two jobs: "Sample Job Submission -- URL" and "Manual Submit: NatureHDQT-low64-s3...". At the bottom, a "Binder" window is open, showing a list of files with columns for Name, Nickname, and File(s).

Job	State	Progress	Started	Updated	Compute Time	Elapsed
Sample Job Submission -- URL	Complete		9/26/2018 9:43:20 AM	9/26/2018 9:50:29 AM	7 min. 7 sec.	7 min. 8
Manual Submit: NatureHDQT-low64-s3...	Complete		9/26/2018 9:31:08 AM	9/26/2018 9:31:53 AM	18 seconds	45 seconds

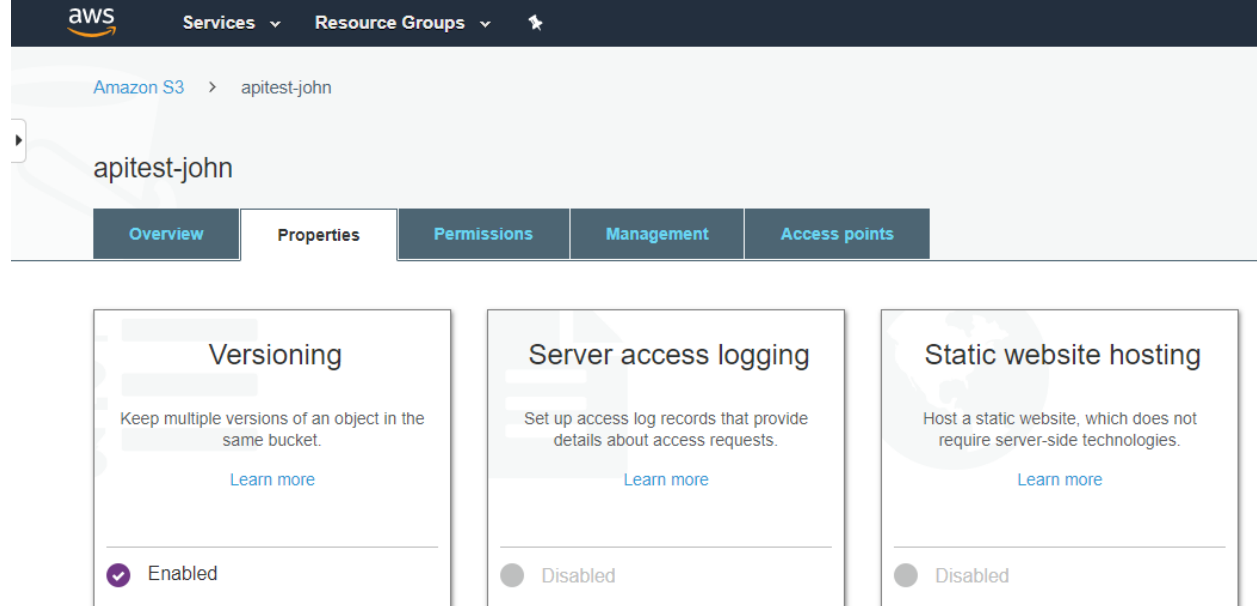
Name	Nickname	File(s)
NTSC_TC...	Copy	D:\VantageStore\554bf991-5dfd-42c3-98ed-3df22b6079a2\NTSC_...
NTSC_TC...	Original	s3://ts-vantage.s3.amazonaws.com/NTSC_TC_CC.mpg
NTSC_TC...	Vantage Proxy	D:\VantageStore\0aecab74-ba83-4af0-afa7-ab9200b6e058\NTSC_T...

Using this mechanism should make it much simpler to obtain a starting URL which represents the input to Vantage for remote files when submitting via the API.

## Requesting Versioned Assets from S3

The Amazon S3 cloud storage allows multiple versions of a given asset to be maintained. Vantage supports requesting specific revisions of a given asset IF AND ONLY IF a version designator is supplied during an API submission.

As a reference, within a Particular S3 bucket, the Versioning property for that bucket may be set:



See the following page for more information/restrictions of S3 versioning:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

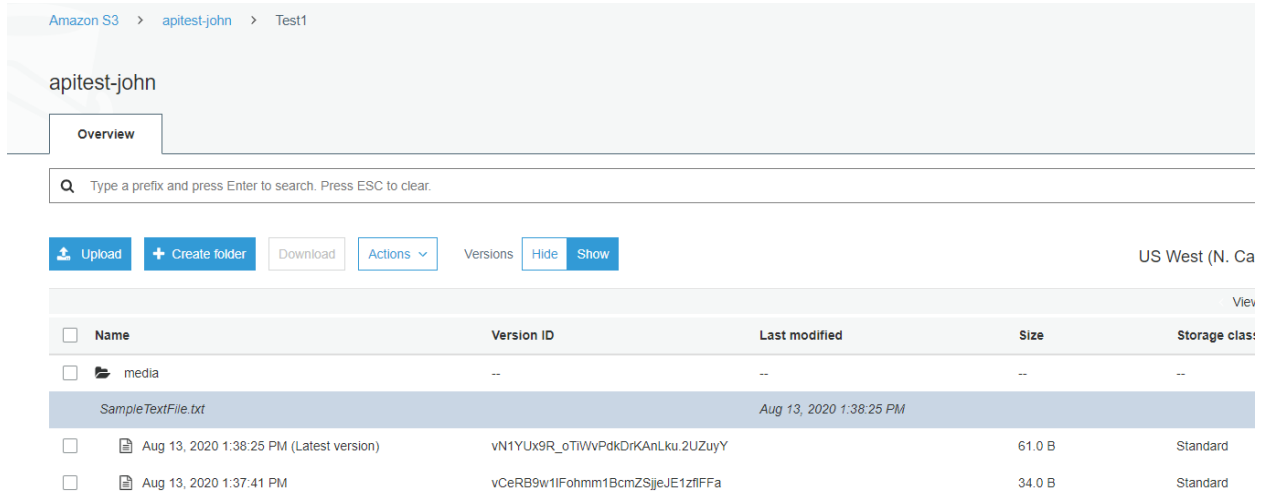
Once versioning is enabled for a BUCKET, any NEW material placed into that BUCKET will have a unique VERSION tag associated with it.

Vantage will support requesting a specific VERSION of the material by supplying this TAG in the S3 URL that is constructed and supplied to Vantage.

For example, consider the following BUCKET with the file: SampleTextFile.text present. If VERSIONING is enabled, and the option to Show versions is turned on, the following is displayed:



848 Gold Flat Road  
 Nevada City, CA 95959  
 +1 530 470 1300  
[www.telestream.net](http://www.telestream.net)



The screenshot above shows that the file: `SampleTextFile.txt` has two (2) distinct versions:

<i>SampleTextFile.txt</i>		
<input type="checkbox"/>	Aug 13, 2020 1:38:25 PM (Latest version)	vN1YUx9R_oTiWvPdkDrKAnLku.2UZuyY
<input type="checkbox"/>	Aug 13, 2020 1:37:41 PM	vCeRB9w1lFohmm1BcmZSjjeJE1zf1FFa

The ID after the upload date for each entry is the Version TAG.

For example, the file: `SampleTextFile.txt` was originally uploaded on Aug 13 at 1:37:41PM. The ID of this version is: `vCeRB9w1lFohmm1BcmZSjjeJE1zf1FFa`

When a URL to this file is constructed, Vantage supports the use of the query parameter: `versionId`

This allows the API user the ability to specify a distinct version of the file.

If a user wishes to submit the original version uploaded to S3 (ie: the file uploaded at 1:37:41PM); the following URL would be supplied as part of the SDK submission to Vantage:

`s3://AKIXXXXXXXXXXXXXXXXX:xtfRZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ@s3.amazonaws.com/apitest-john/Test1/SampleTextFile.txt?versionId=vCeRB9w1lFohmm1BcmZSjjeJE1zf1FFa`

NOTE: The ID and Secret Key for the account in question have been obscured above.

NOTE: If the most recent (Latest Version) is desired, this is what is retrieved if NO version information is supplied, ie:



848 Gold Flat Road  
Nevada City, CA 95959  
+1 530 470 1300  
[www.telestream.net](http://www.telestream.net)

```
s3://AKIXXXXXXXXXXXXXX:XtFRZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ@s3.amazonaws.com/apitest-  
john/Test1/SampleTextFile.txt
```

Would refer to the LATEST version (ie: no version designator was specified).

A complete JSON submit message would be similar to:

```
{  
  "Attachments": [],  
  "JobName": "Populate with your job name",  
  "Labels": [],  
  "Medias": [  
    {  
      "Identifier": "abb3f7e5-8d7e-49ba-84c5-674a55edb700",  
      "Data": "If sending in-  
band data (eg: CML); place a UTF8 BASE64 encoded version of the data in this field and  
do not send a file path.",  
      "Description": "The original version of content encountered or created by  
Vantage.",  
      "Files": [  
        "s3://AKIXXXXXXXXXXXXXX:XtFRZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ  
@s3.amazonaws.com/apitest-  
john/Test1/SampleTextFile.txt?versionId=vCeRB9w1lFohmm1BcmZSjjeJE1zf1FFa"  
      ],  
      "Name": "Original"  
    }  
  ],  
  "Priority": 0,  
  "Variables": []  
}
```

Where the **highlighted** text above illustrates the use of the versionId query parameter.