Qualify Developer Guide

Introduction

The Qualify API provides a programmatic mechanism for interacting with the Telestream Cloud Qualify Service. Qualify is one of many services available under the Telestream Cloud product umbrella.

The Qualify API interacts with several fundamental components which form the Qualify Service. These components are:

- **Template** – A user configurable set of test parameters
- **Project** – An organizational mechanism containing configuration information relating to a desired compute region and to storage
- **Job** – The unit of execution within the Qualify Service
- **Reports** – The results of the various QC tests

Prerequisites

This document assumes that the reader/user has successfully signed up with Telestream Cloud; and can access the Qualify Service interface.

This document assumes the user is familiar with the creation and configuration of Projects and Templates. If these concepts are unfamiliar, please see the main Qualify Service documentation.

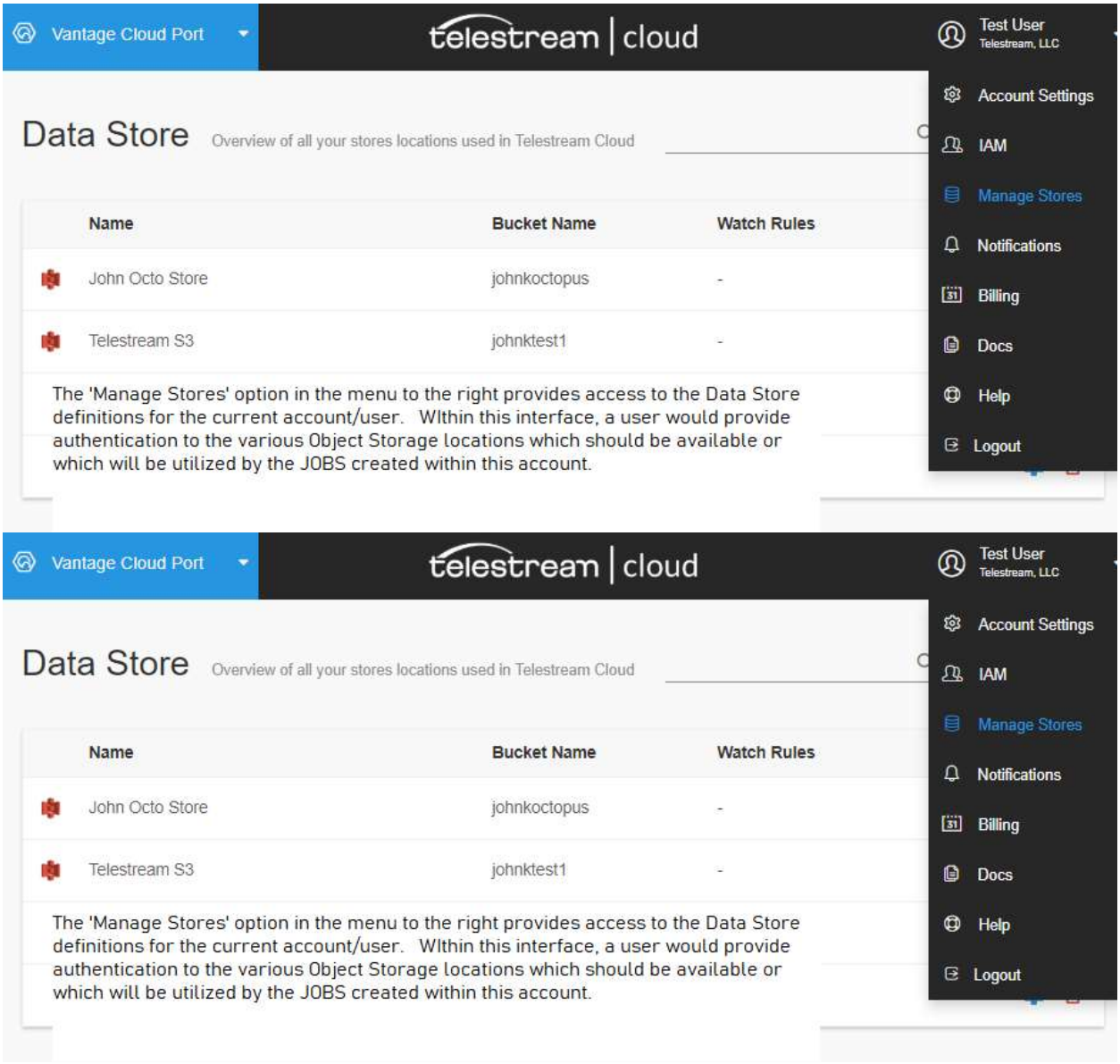
The use of the Qualify Service also assumes a level of familiarity with at least one of the following Cloud Storage providers: Amazon S3, Google Cloud Storage or Azure BLOB Storage.

Prior to the use of the Qualify Service (or any capability within Telestream Cloud); a user MUST have an account created and setup with one of these storage providers. A user will be required to setup references to a storage location from one of these providers within Telestream Cloud before any capability discussed in the document may be utilized.

If familiarity with a Cloud Storage Provider is not present or an account with a provider has not been established, the use of the Qualify Service *will not* be possible.

Cloud Storage

Before continuing with discussions about the Qualify Service API, it is important to have a clear understanding of the role that Cloud Storage (in particular the provisions made within Telestream Cloud) has upon jobs which are run and hosted on Telestream Cloud.



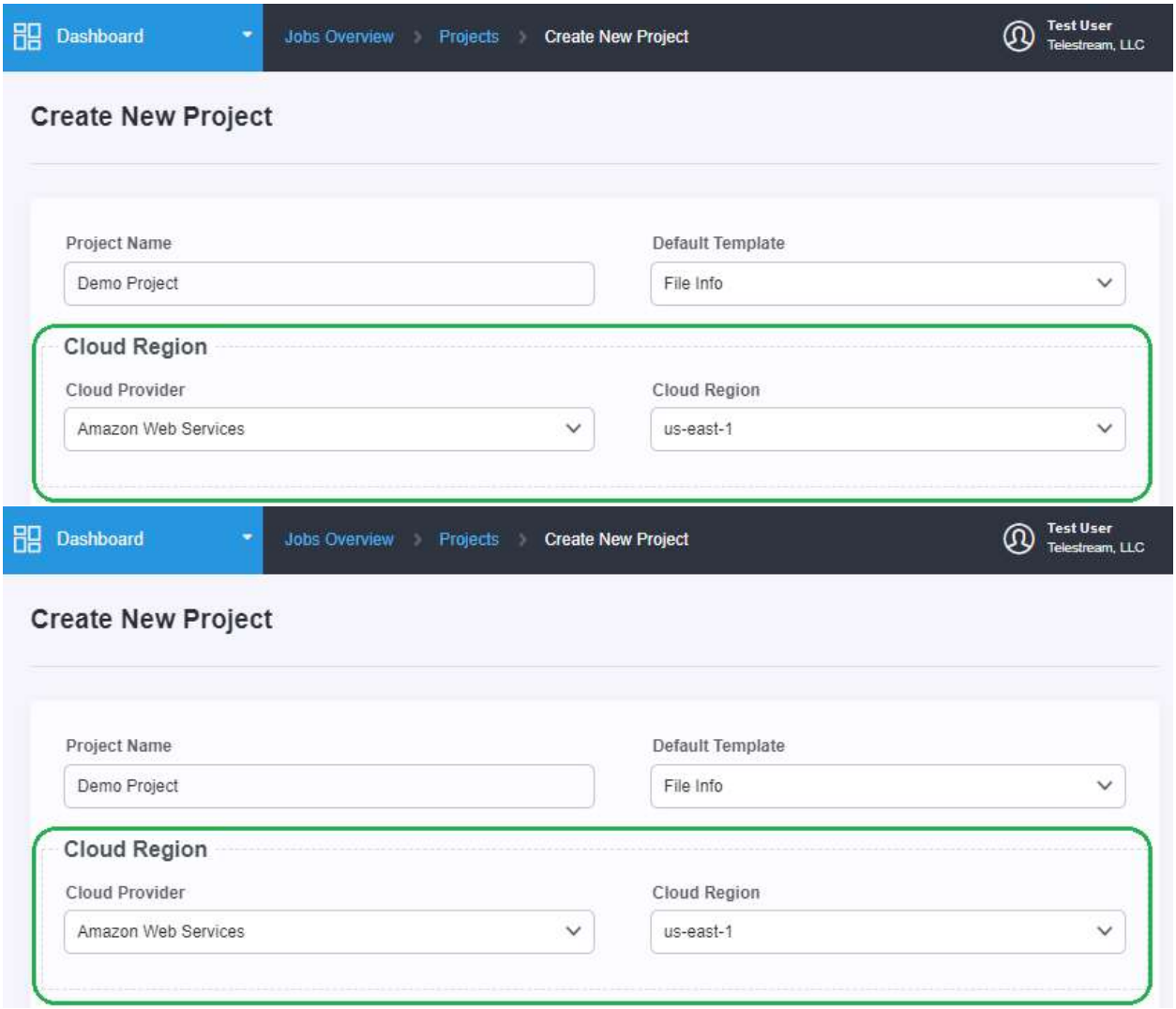
The Stores listed in the above interface are what shall be listed when Storage queries are made via the API (discussed below). Further, the storage definitions above define what capabilities that Telestream Cloud has with regards to discovering material.

If the file that is submitted to the API as a source is some type of reference file (for example: a CML file that may reference other pieces of Media, an IMF sequence or a QuickTime reference file or an IMX OP1B collection); then Telestream Cloud must have

access to the entire Storage repository where these files reside. If this is not provided, then Telestream Cloud (and the underlying job which was being processed) will not be able to locate/access all the required material needed to process the job.

Project Settings

Further, and as equally as important, the specification of a Cloud Provider and Region within the creation/setup of a Project (detailed below) defines the Provider and Region where the Qualify Service Job shall be run. This is *EXTREMELY IMPORTANT*.



For example, the new Project created in the screenshot above indicates that the **US EAST 1** region of **Amazon S3** as part of its definition.

When any JOB is created which targets this project, if the input for a job resides in some other region (for example: **US WEST**), then the processing of this job shall take place in

a REGION different from where the input is not located. In order to move the input to the region where the processing shall take place, EGRESS charges shall be incurred by the user's account.

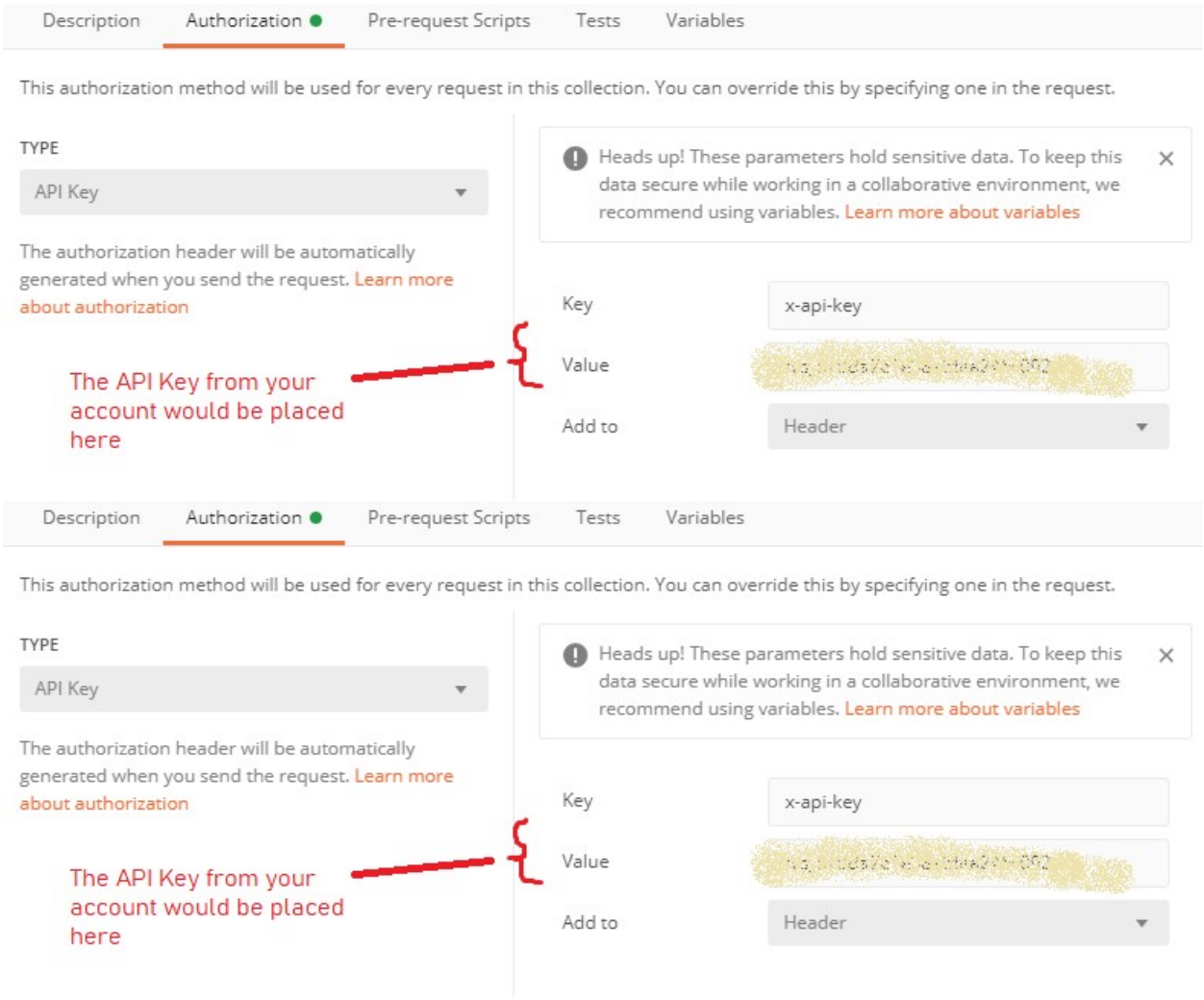
To avoid un-necessary EGRESS charges, it is highly recommended that the PROJECT which is referenced during a new job submission always MATCH the Provider and Region where the INPUT for that job resides.

API Keys/Authorization

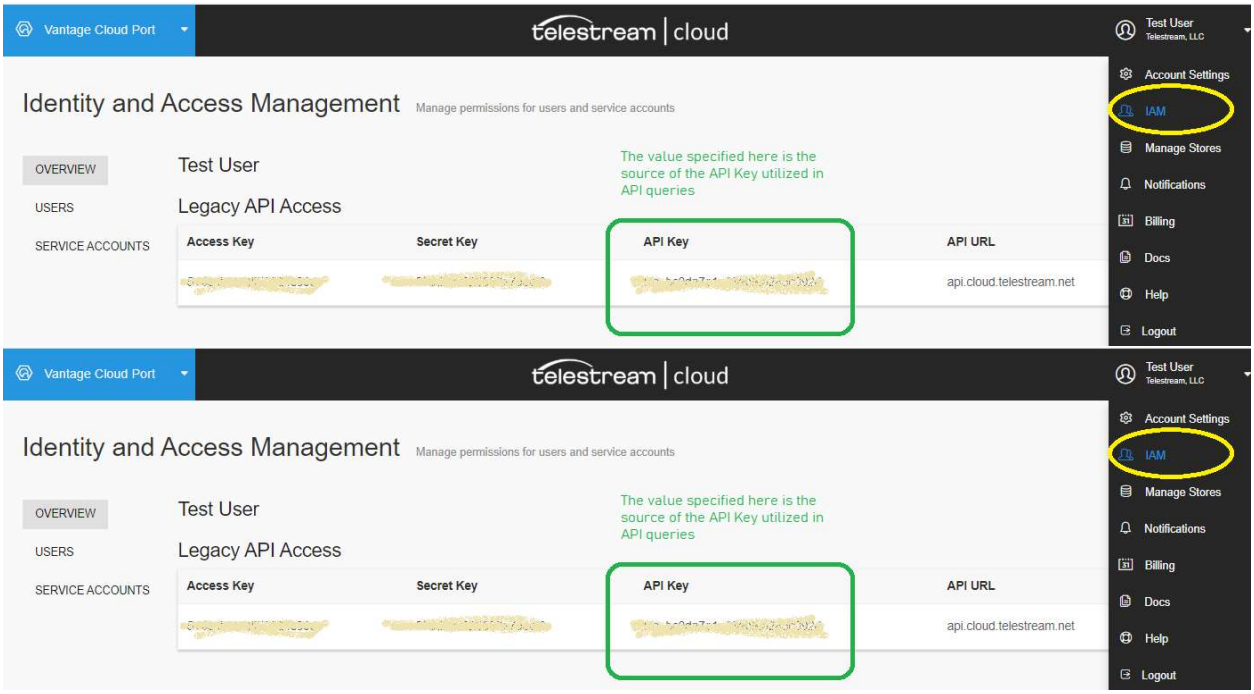
All Qualify Service API capabilities require the specification of an API Key. This key is available in the IAM section of a given account. This key is passed within the header of any HTTP requests made to the Qualify Service API.

All examples which follow shall be made using the Postman application.

The authentication is specified as follows within Postman:



The API can be located within your Telestream Cloud account within the IAM action:



Job Results

The JSON that is generated when job status is queried contains several fields. It is important to understand the distinction between two fields in particular: Job Status and Job State.

Job Status refers to the disposition of the overall job. For example, whether the job was successfully run, or the job encountered an error. Only jobs which successfully run will produce QC results.

An example of a scenario which results in a Job Status of Failure is submitting a reference to a file which does not exist. In this case, the job shall fail; and the results of the QC test are immaterial (i.e.: non-existent).

Job State refers to the outcome of all QC tests for a job. The Job State may be “passed” or “reject” or “warning”.

Only jobs which were Successfully run will produce a State. If a job is not successfully run, then the State field shall not be produced.

Interpreting the results of a job should adhere to the following algorithm/pseudo-code:

Text

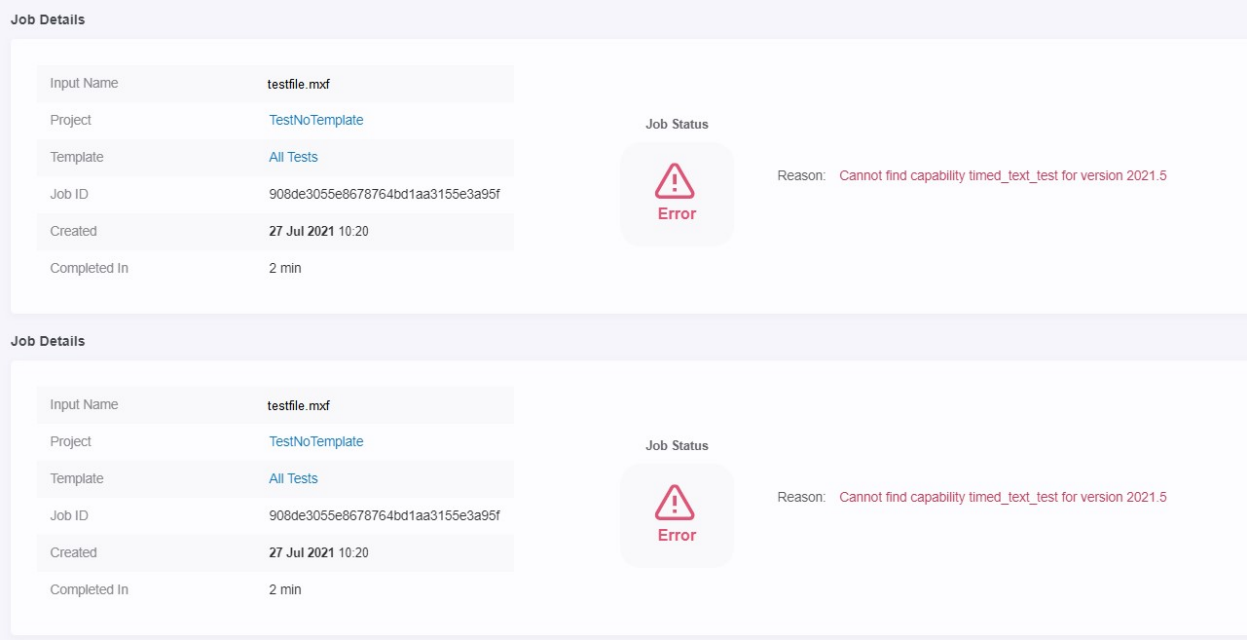
```
If the final job Status is Successful
{
    Examine the job.state field
```

```
    If ( job.state == 'passed') this implies all tests in the job passed
successfully

    If ( job.state == 'warning') this implies that certain tests raised warnings

    If (job.state == 'reject' ) this implies that certain tests resulted in
failures/the file should be rejected based upon the structure of the tests
}
```

If the job status is not successful; this implies something prohibited the job from successfully completing (e.g.: a file was unavailable or the file was not recognized/understood).

Examples of Job Status for completed jobs are as follows:

From the Qualify Service UI:

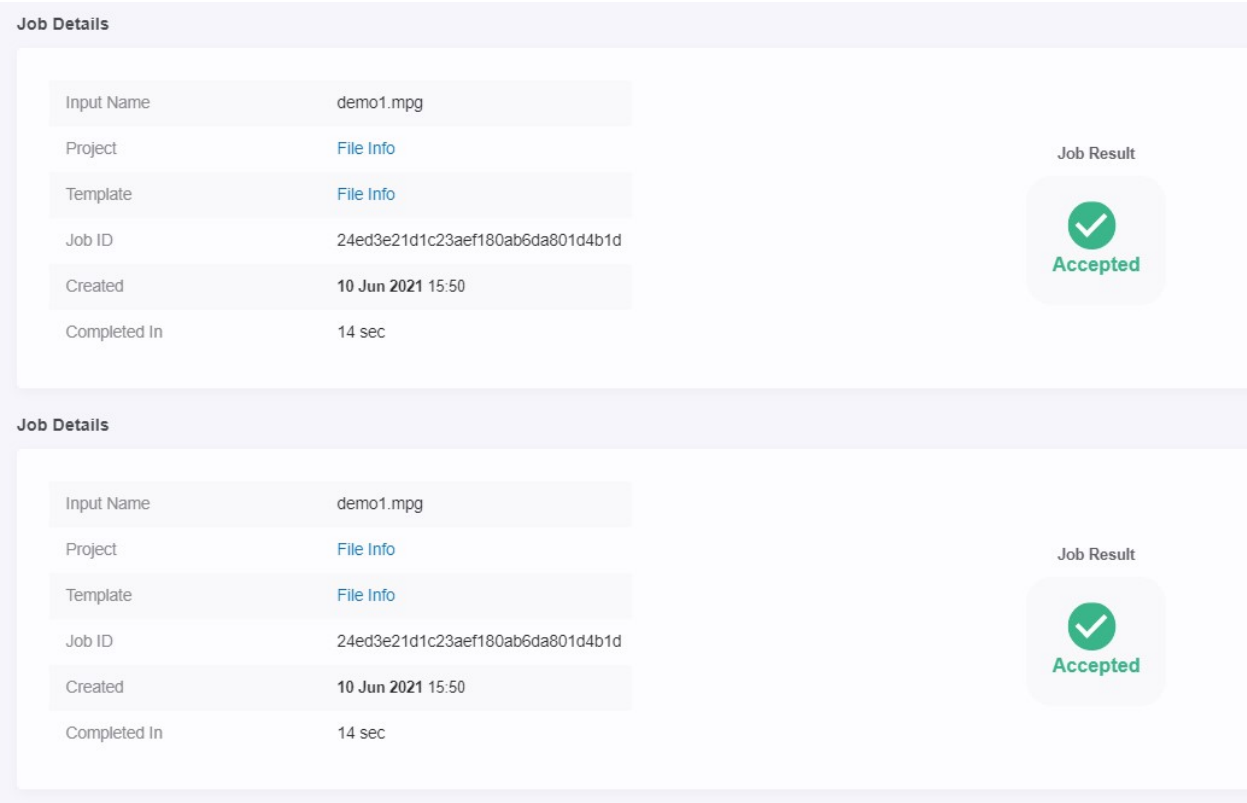


Corresponding Job result JSON:

JSON


```
{
"id": "908de3055e8678764bd1aa3155e3a95f",
"created_at": "2021-07-27T14:20:53Z",
"updated_at": "2021-07-27T14:22:29Z",
"status": "error",
"progress": 0,
"message": "Cannot find capability timed_text_test for version 2021.5",
"url": "\<URL to the Source\>",
"project_id": "de3544bec669985d7293fb92d4895d4f",
"project_name": "TestNoTemplate",
"template_id": "2c5ec555fb789719b890f5e0e72d6f34",
"template_name": "All Tests",
"file_name": "testfile.mxf"
```

}



Corresponding Job result JSON:

JSON


```
{
"id": "24ed3e21d1c23aef180ab6da801d4b1d",
"created_at": "2021-06-10T19:50:40Z",
"updated_at": "2021-06-10T19:50:54Z",
"status": "success",
"state": "passed",
"progress": 100,
"url": "\<URL to the Source\>",
"file_info": { \<omitted for brevity\> },
"project_id": "9a4616eb47ef0cc47b923b6f235eefdb",
"project_name": "File Info",
"template_id": "ce05da9b5f1a2bf726e9e26e5e5b5b8c",
"template_name": "File Info",
"file_name": "demo1.mpg"
}
```


Status and State

A complete list of all possible values for the Status and State fields within the Job result JSON are as follows:

Job.status {queued, downloading, processing, success, error, cancelled}

- **Queued** implies that the job has been accepted and is awaiting processing.
- **Downloading** implies that the source file(s) are currently being localized to the instance used for this job.
- **Processing** implies that the current job is ‘in-process’ or being worked upon.
- **Success** implies that the current job has completed successfully.
- **Error** implies that the current job was unable to complete due to a problem processing the job.
- **Cancelled** implies that the job was previously in-process; but was cancelled/stopped by a user request.

Job.state {passed, warning, reject}

- **Passed** implies all of the tests defined in the job template were completed successfully
- **Warning** implies that one or more of the tests associated with the job template encountered a condition they were specifically designed to alert about.
- **Reject** implies that one or more tests defined in the job template encountered a characteristic in the media which should not be allowed. (for example, if a template were configured to reject any material with a video width < 720; submitting a 640x480 MPEG1 video would cause the material to receive a 'reject' state).