

Triggering Jobs

A typical project will require to process raw files into spatially indexed datasets. This process takes one or more raw data entities and produces a new type of entity. In other cases, the system will use information already stored in the spatial storage to create additional entities. Some examples of this are:

- Raw point clouds used to create voxel terrains
- Heightmap raster sets used to create voxel terrains
- Block model in CSV format used to create a voxelized block model
- A report created out of multiple indexed datasets
- A high-resolution mesh exported from a mashup of datasets

Entities that require further processing will be created with its “state” property set to “PARTIAL”.

It is possible to use the REST API to trigger the processing of the entity so its state transitions from “PARTIAL” to “COMPLETE”.

These processes are time-consuming so the REST request returns immediately. The interface allows to register a callback URL. Upon completion, the system will try to deliver the notification for a configurable number of attempts. Instead of using a callback, the application using the REST interface also has the option to poll the entity state for completeness. This is doing by verifying the “state” property of the entity has switched from “PARTIAL” to either “COMPLETE” or “ERROR”.

Method

POST

URL

<server>/events.ashx

Parameters

id	Unique identifier for the entity
project	Unique identifier for the project
org	Organization ID

Post Payload

Multi-part form data containing two fields:

process	A string identifying this request to a later callback
callback	A URL that will be called when the job has completed. This is an optional parameter. The URL is opaque to the system, but must contain two placeholders: • “VF_PROCESS” – This will be substituted by the value provided in the “process” field during the original call. • “VF_ID” – This will be substituted by the ID of the entity that just completed processing • “VF_RESULT_CODE” – This will be substituted by the result code for the operation. A result of zero means the operation completed successfully. See the following table for a list of possible error codes. For example: http://myserver.mysite.com?mytoken=VF_PROCESS&id=VF_ID&myoutcome=VF_RESULT_CODE

Result Error Codes

0	Operation was successful
1	Entity not found
2	Unknown error
3	Error processing report
4	Image creation failed
5	Mesh creation failed
6	Point cloud creation failed
7	Could not index point cloud
8	Could not index mesh
9	Could not process block model
10	Could not process voxel terrain model
11	Could not ingest raw point cloud
12	Could not ingest DEM (Heightmap)
13	Could not produce Unity project
14	Could not export density data
15	Could not process density data
16	Could not index block model
17	Could not process material layer request

Returns

If completed (200 code), this call returns a JSON object that describes the result of the operation:

```
{"result" : "success"}
```

Example (PowerShell)

This POST call triggers the processing of an entity:

```
Invoke-WebRequest -Uri
```

```
"http://localhost:58697/events.ashx?project=myproject&id=4E40F10BD7C74A77AE1E4CC163F3EA98&org=2343243456  
678890" -Method "POST" -Headers @{"Origin"="http://localhost:58697"; "Accept-Encoding"="gzip, deflate, br"; "Accept-  
Language"="en-US,en;q=0.9,es;q=0.8"; "User-Agent"="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/73.0.3683.103 Safari/537.36"; "Accept"="*/*";  
"Referer"="http://localhost:58697/cloud/project.html?id=myproject"} -ContentType "multipart/form-data; boundary=----  
WebKitFormBoundaryRADf50WY1KjKrmUF" -Body ([System.Text.Encoding]::UTF8.GetBytes("-----  
WebKitFormBoundaryRADf50WY1KjKrmUF$([char]13)$([char]10)Content-Disposition: form-data;  
name=`"process`"$([char]13)$([char]10)$([char]13)$([char]10)IMPORT_VOXSURF$([char]13)$([char]10)-----  
WebKitFormBoundaryRADf50WY1KjKrmUF--$([char]13)$([char]10)"))
```

Revision #1

Created 17 March 2025 15:47:04 by admin

Updated 17 March 2025 15:48:55 by admin