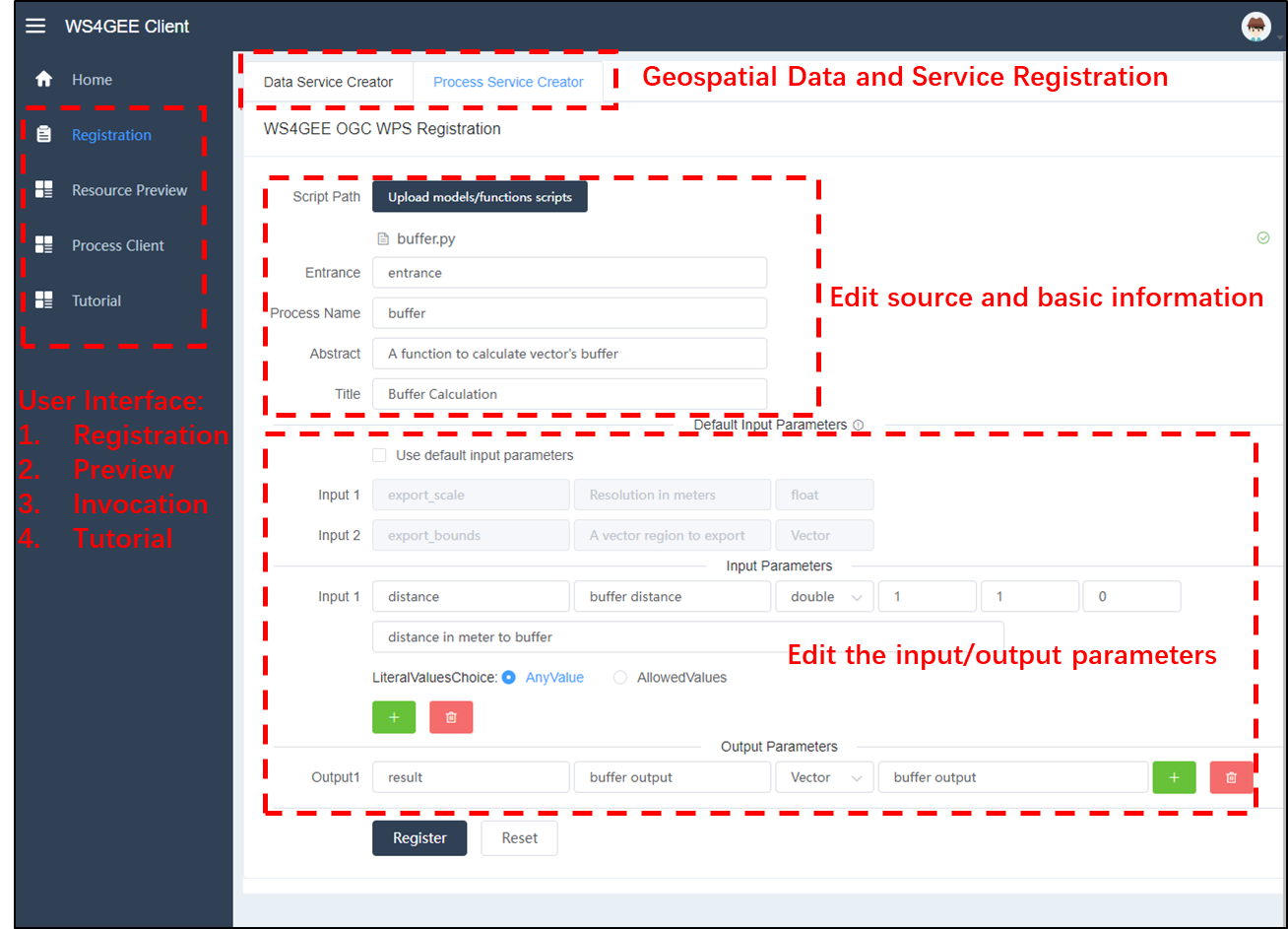
WSGEE aims at bridging the gap between Google Earth Engine (GEE) and general geospatial web service (GWS) community, and supports straight-forward access to GEE resources through OGC standards. In this prototype, we developed instances of Web Coverage Service (WCS) and Web Processing Service (WPS) to provide standardized interface for GEE-based imagery dataset, client library functions, and user-defined models.

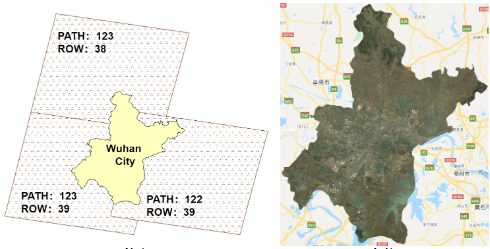
This document provides instances to generate and invoke GEE-enabled geospatial web services in the WS4GEE client side, including three parts: (1) GEE-enabled geoprocessing service generation; (2) GEE-enabled dynamic data services generation; (3) Service invocation

WS4GEE Client – Registration is the entrance to create GEE-enabled WCS and WPS services. These functions are realized in the **Data Service Creator** Tab and the **Process Services Creator** Tab. Users can create GWSs through web based UIs with simple click and setting.



**(1) GEE-enabled dynamic data services generation (Registration Window)**

This component shows how to generate GEE imagery dataset as WCS services and call relative “GetCapabilities”,”DescribeCoverage”,”GetCoverage” interfaces. We using the “Landsat 8 Surface Reflectance Tier 1” and the region – “Wuhan City” as an example. The procedure contains steps to select target dataset, stacking type, data range, operations and boundary.



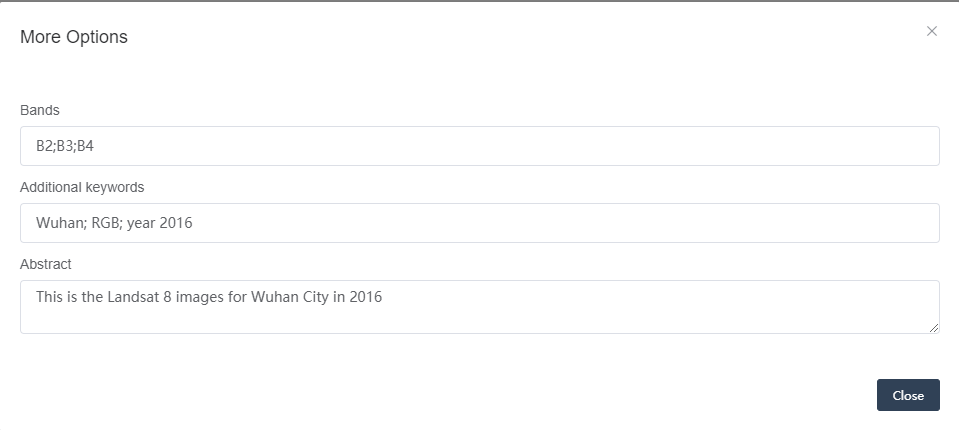
**Step 1**: Select target dataset “Landsat 8 Surface Reflectance Tier 1”, and the service type is identified automatically.

**Step 2:** The stacking type indicates the approach to generate dynamic image to a region by a group of relative images, and three methods are provided: “by mean”, “by max”, and “by min”; ( See <https://developers.google.com/earth-engine/guides/reducers_image_collection> to get more details). Here, we selected “by mean” as the stacking method.

**Step 3:** The data range is set as “2016-09-01” to “2016-12-28”.

**Step 4:** In the Other Options drop-down menu, the “no cloud” and “by month” tabs are selected. These options are chosen to initialize image generation and optimize the generation results. For instance, the “by month” option will help generate monthly images for the target region during the data range. Thus, in this example, images for Wuhan region will be generated for four months from September to December.

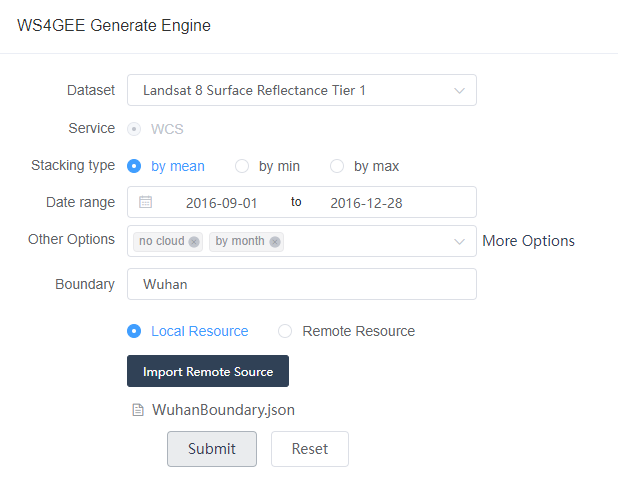
The “More Options” is used to set additional options for the services. For instance, the “Bands” is used to select bands for the images. Here, we type “B2;B3;B4”. In the input element shown below.



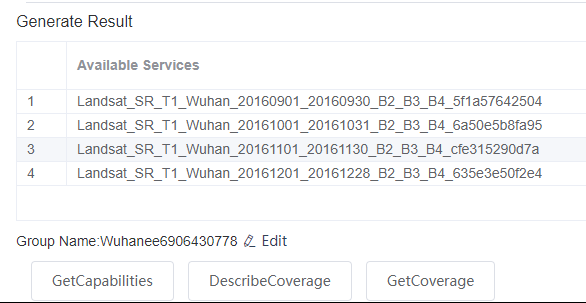
\* The system has set default values for the option setting.

**Step 5:** For the boundary, the name and the upload functions with target files (support .geojson currently) should be set. Files can be uploaded locally or remotely. We choose to upload a file called “WuhanBoundary.geojson” locally and set names.

**Step 6:** The entire setting to generate WCS is shown below. Click “Submit” to generate result.

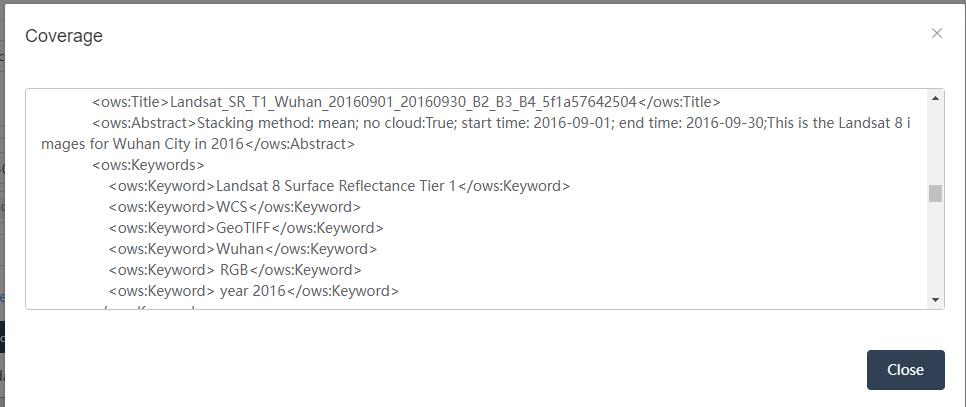


**Step 7:** A list of results are generated. The group name can be edited and re-defined.

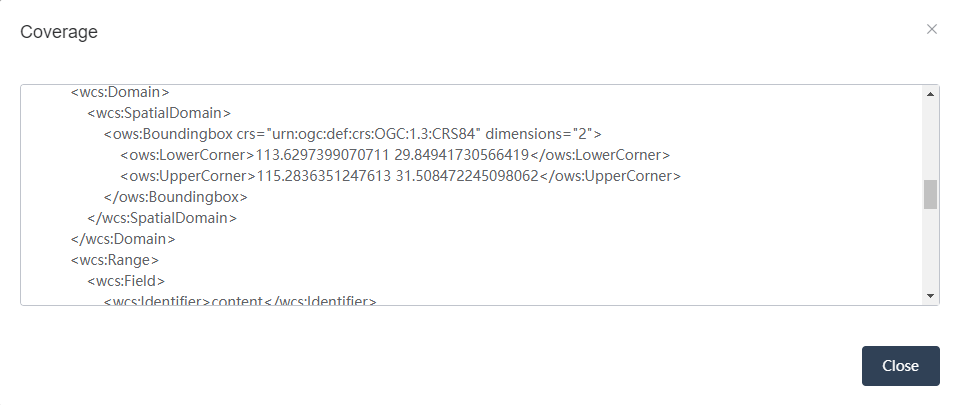


**Step 8:** Click the “GetCapabilities”,”DescribeCoverage”,”GetCoverage” to get metadata for the generated WCSs.

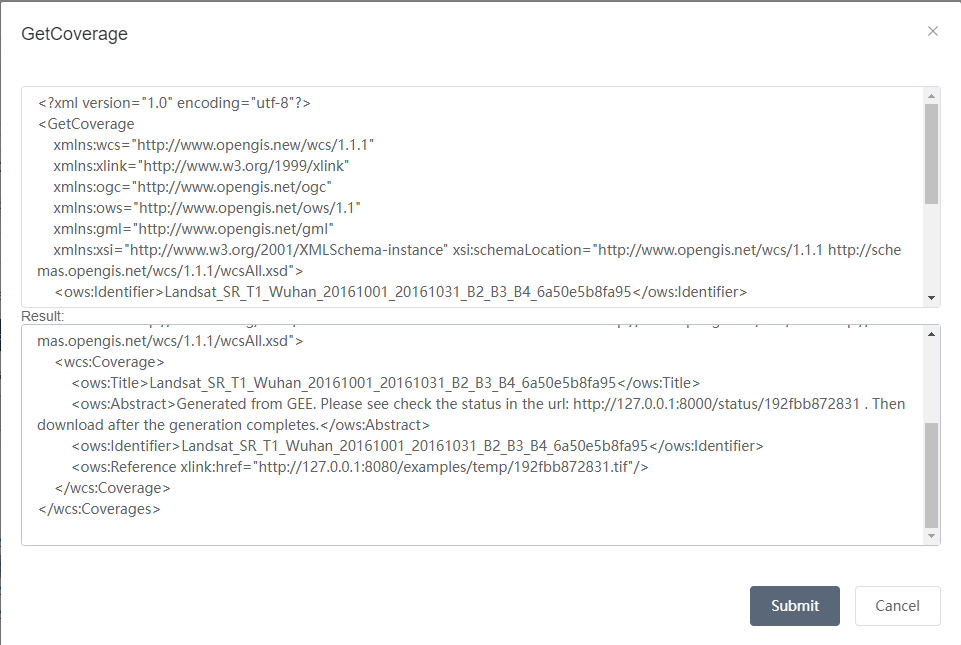
(1) GetCapabilities:



(2) DescribeCoverage:



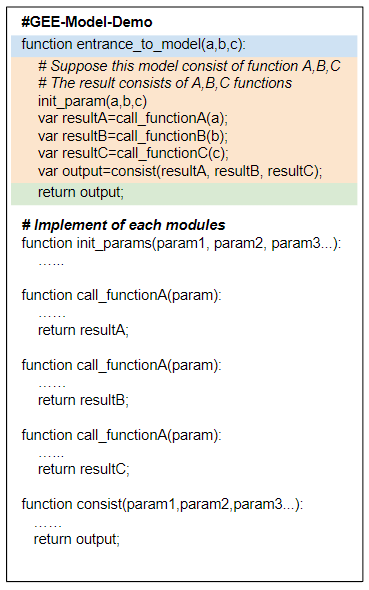
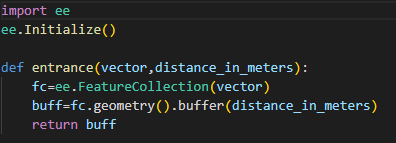
**Step 9:** The “GetCoverage” facilitates quick access the target images after the generation and downloading process through GEE is finished. This status can be checked through the link in the <abstract> (“Please see check the status in the url: http://......”).



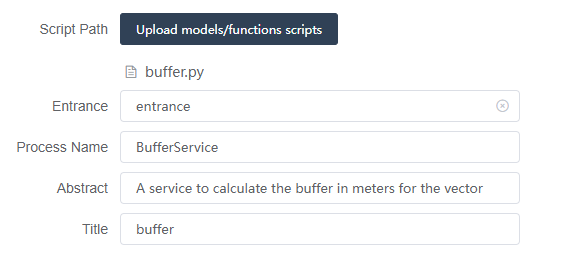
**(2) GEE-enabled geoprocessing services generation (Registration Window)**

This component shows how to generate GEE functions from Python client or user-defined GEE models as WPS services. Using a service called “buffer” as an example, which accepts two parameters – an input vector, and the distance to buffer in meters.

**Step 1:** Firstly, the **script template** to form the service calculation instance is required. Basically, this template is a section of runnable code based on GEE environment. As shown below, the function/model is called through the function called entrance, a list of functions will be called as a function chain. In summary, an “entrance” interface and a list of functions that realize the logic of the service are required. It is easy for GEE experts to adjust their models to this template. The template for the “buffer” function is shown in the right part below. Only three rows are required to realize such a function, with two rows for initialization.



Finally, click the “Upload models/functions scripts” to set up the script templates, and type the “entrance” to the entrance input box, because the function “entrance” is our entrance to this service. Then, the process name, abstract and titles of this functions should be set as follows for readability.

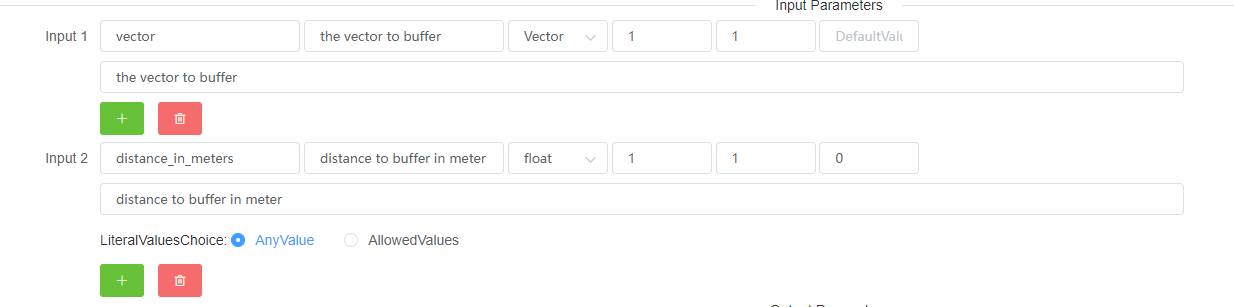


**Step 2:** Set up the input parameters and relative configuration.

For function with a raster output, two default parameters: “export\_scale”, and “export\_bounds” are require. The system has already defined these setting. They are ready to use. For the “buffer” service, the output is a vector, so this default input parameters setting is not required. Click the checkbox to unselect this setting.



**Step 3:** As mention previously, the “buffer” service accepts two parameters – an input vector, and the distance to buffer in meters. According to the python code, these input parameters are called “vector” and “distance\_in\_meters”. We recommend to set the input name in keeping with the parameters in the python code (the script template) so that other users can match and edit the templates easily. Thereafter, the setting is configured as follow.



It is noted that for the minOccurs/maxOccurs parameter, the default setting is 1, which means this service accept input this parameter once. For literal data type (int, float, double, string), the “LiteralValuesChoice” can be set (Default: AnyValue). If the “AllowedValues” is set, a selection of the value to use should be given.

**Step 4:** Similarly, set up the output parameter. Finally click the submit button to register this service as WPS.



**(3) Service invocation (Resource Preview and Process Client Window)**

Since WS4GEE integrates the service as OGC data and processing services. These services can be called through standardized OGC interfaces:

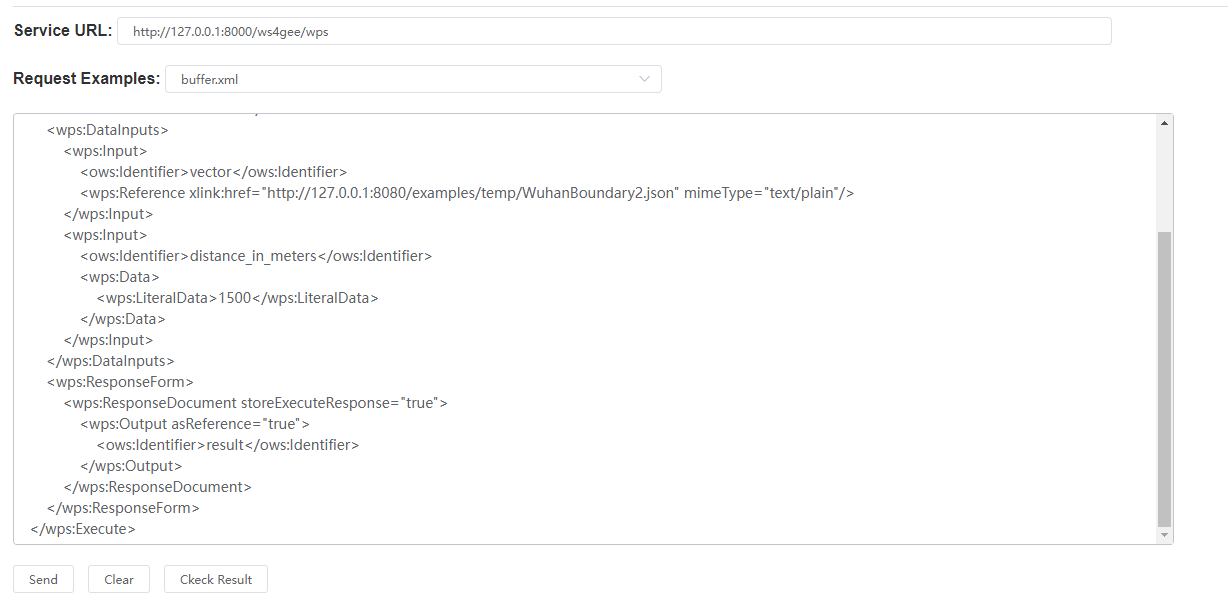
|  |  |  |
| --- | --- | --- |
| Service | URL | Http Methods |
| WCS  GetCapabilities | http://{ServerAddress:port}/ows/{GroupName}/wcs?service=WCS&version=1.1.0&request=GetCapabilities | Get |
| WCS  DescribeCoverage | http://{ServerAddress:port}/ows/{GroupName}/wcs?service=WCS&version=1.1.0&request=describecoverage&identifiers={Image Name} | Get |
| WCS  GetCoverage | http://{ServerAddress:port}/ows/{GroupName}/wcs  *\* The WCS GetCoverage Request XML template is required. See the attachment to get the example* | Post |
| WPS  GetCapabilities | http://{ServerAddress:port}/ws4gee/wps?request=GetCapabilities&service=WPS&version=1.1.0 | Get |
| WPS  DescribeProcess | http://{ServerAddress:port}/ws4gee/wps?service=WPS&version=1.1.0&request=DescribeProcess&identifiers={Process Name} | Get |
| WPS  Execute | http://{ServerAddress:port}/ws4gee/wps | Post |

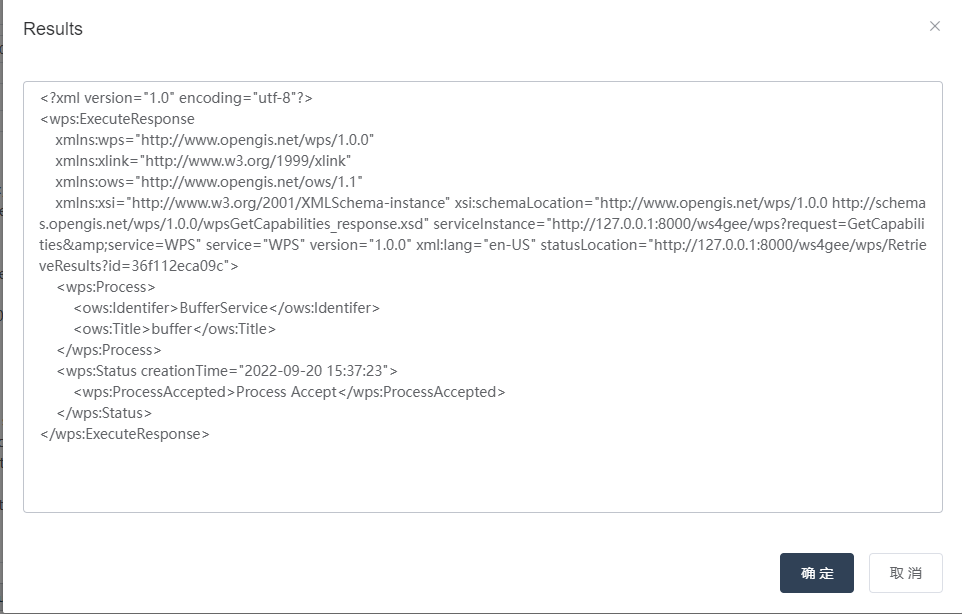
We also provide web-based UIs to invoke these services to facilitate UI-only users:

For WCS service in the WS4GEE system, use the **Resource Preview** window to check and obtain the raster that is pre-registered in the previous procedure.

For WPS service in the in the WS4GEE system, user can either use the **Resource Preview** window to check the metadata through WPS OGC DescribeProcess interface, or call these WPSs in the Process Client window

Using the “BufferServices” registered in (2) as an instance:





The result then can be obtained through the url from “statusLocation”

