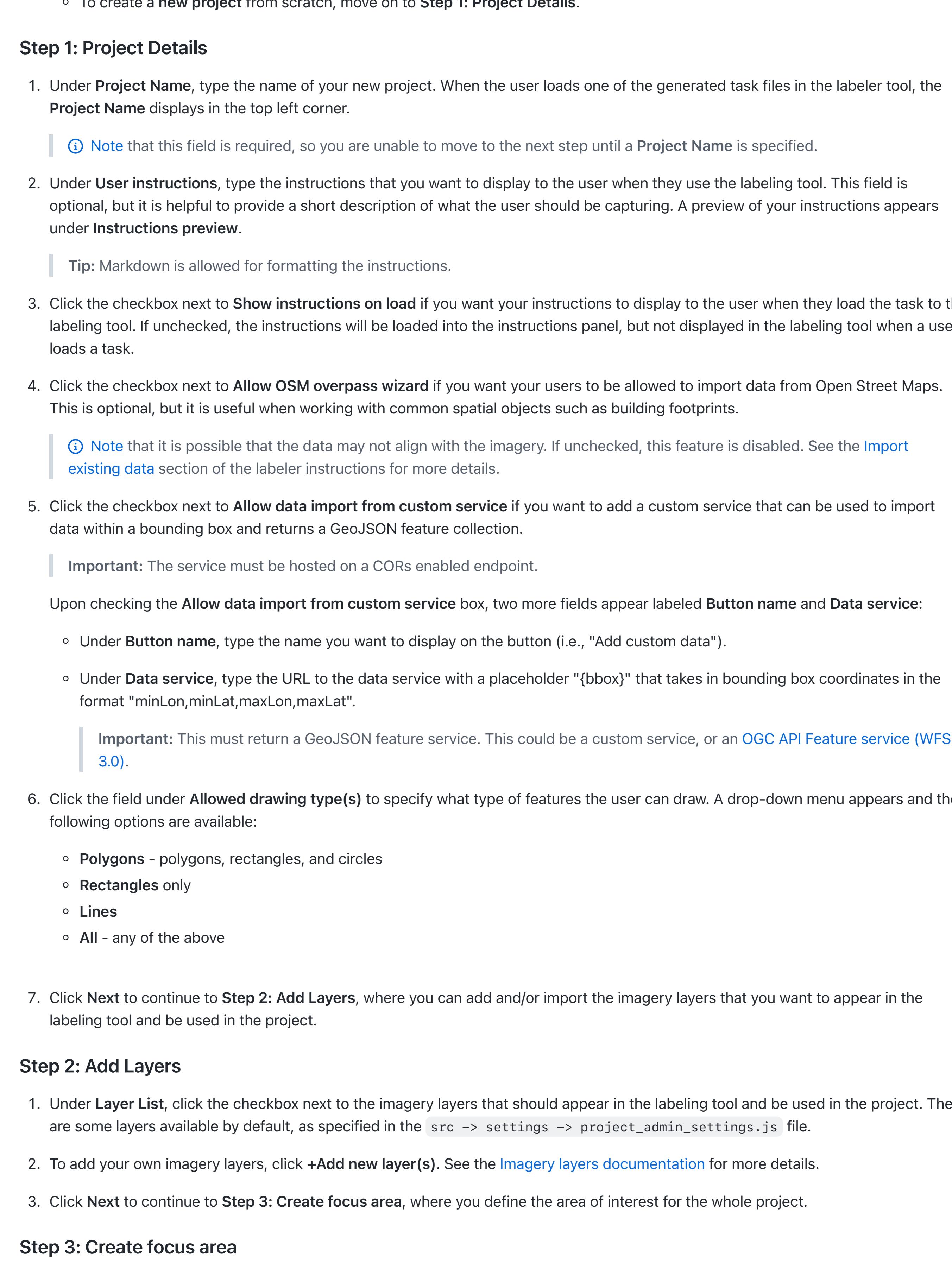


The project builder is the tool used by a project administrator to define a labeling project. In the project builder, administrators can also break the project up into a set of smaller sub-tasks that can be assigned to a group of people who will do the labeling.

Spatial imagery labeling project builder



Project Builder quickstart

- Open the [project builder tool](#). Under the heading labeled **Step 1: Project details**, there are a series of fields for you to fill in. There are five different sections like this to complete.

Note that if you have forked this project, your URL will be different.

- Choose one of the following:
 - To open and edit an existing project that you have saved, click [Load existing project](#), which appears above the **Step 1: Project details** heading.
 - To create a [new project](#) from scratch, move on to [Step 1: Project Details](#).

Step 1: Project Details

- Under **Project Name**, type the name of your new project. When the user loads one of the generated task files in the labeler tool, the **Project Name** displays in the top left corner.

Note that this field is required, so you are unable to move to the next step until a Project Name is specified.

- Under **User instructions**, type the instructions that you want to display to the user when they use the labeling tool. This field is optional, but it is helpful to provide a short description of what the user should be capturing. A preview of your instructions appears under **Instructions preview**.

Tip: Markdown is allowed for formatting the instructions.

- Click the checkbox next to [Show instructions on load](#) if you want your instructions to display to the user when they load the task to the labeling tool. If unchecked, the instructions will be loaded into the instructions panel, but not displayed in the labeling tool when a user loads a task.

Note that it is possible that the data may not align with the imagery. If unchecked, this feature is disabled. See the [Import existing data](#) section of the labeler instructions for more details.

- Click the checkbox next to [Allow OSM overpass wizard](#) if you want your users to be allowed to import data from Open Street Maps. This is optional, but it is useful when working with common spatial objects such as building footprints.

Note that it is possible that the data may not align with the imagery. If unchecked, this feature is disabled. See the [Import existing data](#) section of the labeler instructions for more details.

- Click the checkbox next to [Allow data import from custom service](#) if you want to add a custom service that can be used to import data within a bounding box and returns a GeoJSON feature collection.

Important: The service must be hosted on a CORS enabled endpoint.

Upon checking the [Allow data import from custom service](#) box, two more fields appear labeled **Button name** and **Data service**:

Under **Button name**, type the name you want to display on the button (i.e., "Add custom data").

Under **Data service**, type the URL to the data service with a placeholder "(bbox)" that takes in bounding box coordinates in the format "minLon,minLat,maxLon,maxLat".

Important: This must return a GeoJSON feature service. This could be a custom service, or an [OGC API Feature service \(WFS 3.0\)](#).

- Click the field under **Allowed drawing type(s)** to specify what type of features the user can draw. A drop-down menu appears and the following options are available:

- Polygons** - polygons, rectangles, and circles
- Rectangles only**
- Lines**
- All - any of the above**

- Click [Next](#) to continue to [Step 2: Add Layers](#), where you can add and/or import the imagery layers that you want to appear in the labeling tool and be used in the project.

Step 2: Add Layers

- Under **Layer List**, click the checkbox next to the imagery layers that should appear in the labeling tool and be used in the project. There are some layers available by default, as specified in the `src -> settings -> project_admin_settings.js` file.

- To add your own imagery layers, click [+Add new layer\(s\)](#). See the [Imagery layers documentation](#) for more details.

- Click [Next](#) to continue to [Step 3: Create focus area](#), where you define the area of interest for the whole project.

Step 3: Create focus area

- Zoom in on the map to the focus area of your project. You can do this a few different ways:

- Type your area of interest into the search bar at the top left of the map, next to the magnifying glass icon.

- Use your mouse or keypad to zoom in to the focus area.

- Use the plus(+) and minus(-) buttons at the bottom right corner of the map to zoom in or out of the focus area.

- Use the drawing tools to draw the area of interest by clicking on one of the icons before drawing. There are four tools to work with:

- Draw a polygon
- Draw a square
- Draw a circle
- Edit the shape

- If you want to import a spatial data file with your area polygon (or multipolygon), click [Import area](#) under **Gridding options**. Only the first shape in the data file imports.

Important: The supported spatial data file formats are GeoJSON, GeoJSONL, KML, KMZ, GeoRSS, GML, CSV (with a column with well known text of features).

When the area is drawn or loaded, a grid appears. The grid controls the amount and size of individual task areas that you are able to assign to users of the labeling tool.

- Under **Grid size units**, adjust the units if necessary.

- Under **Grid size**, use the down and up arrows to adjust the size of the grid. This helps to adjust the number of **individual task areas** that you want to create.

- Click [Next](#) to continue to [Step 4: Define classes](#), where you define the class information that you want the users to assign to the features they draw.

Step 4: Define classes

This step is particularly important for configuring a project for labeling. See examples throughout this section on how to configure a project for building segmentation and damage assessment labeling.

- Under **Primary class display name**, type the name that you want to display for your primary class information. This may be the primary class name itself, but if you have multiple primary classes, the display name needs to be something broader.

- Under **Primary class property name**, type the property name that captures the primary class information in the labeling tool.

- Under **Primary classes**, type the name of the primary class.

Important: When configuring a project for building segmentation and damage assessment, the primary classification might be Building.

- Click the green plus(+) button next to the primary class to officially add it. The class name is added to the list with a color picker square next to it.

Step 4: Define classes

The output of the project builder is a zip file with following structure:

Project_name

> results

> tasks

{ Project_name_0.json

{ Project_name_1.json

{ project_builder_settings.json

summary.csv

- If you choose to capture a secondary class, customize the display name, property name, and add class names in a similar manner as the primary class.

For example: when configuring a project for building segmentation and damage assessment, the secondary classification might be damage severity, with the classes for labeling being 0, 1, 2, 3, 4, and 5 as measurement for damage severity.

Spatial imagery labeling project builder

The above image shows an example project configuration that might be used for building segmentation and damage assessment.

- Click [Next](#) to continue to [Step 5: Complete](#), where you download the project files.

Step 5: Complete

- If you want to have links to the labeling tool generated with a `&taskUrl=` parameter that will load the individual labeling task from the location you have hosted the files, click the checkbox next to [Include task file links in summary](#).

Note that this requires files to be hosted on a CORS enabled endpoint.

See the [Output file format](#) section for more details on the output of the project builder tool.

- Click [Download project files](#) to download.

- Share the task files or links to the labeling tool that will load their task file with the people who will be doing the labeling. See the [Sharing task files](#) section for more details.

- Once the users have completed the labeling and sent you the results, add the output files to the **results** folder of the overall project folder.

- When you have collected all the results, zip the overall project folder and load it into the [Project viewer](#) to view and merge all the results. See the [Project viewer documentation](#) for more details.

Sharing task files

Once the task files for your project have been created you will most likely want to assign/share them with the users who will be doing the labeling.

For smaller projects

For smaller projects, the easiest method would be to email a task file to each individual user and keep track of the progress in the generated `summary.csv` file.

For larger labeling jobs

For larger labeling jobs, it will likely be much easier to host the task files on a CORS enabled endpoint, such as Azure Blob storage, and then send out a URL with the link to the labeler with the task file URL passed into the `&taskUrl=` URL parameter.

Note that the hosting location of the task files will need CORS and [https](#) enabled. See the next section for an example of one way to do this.

See the [Output file format](#) section for more details on the output of the project builder tool.

Hosting files in a CORS enabled Azure Blob storage

WARNING: The following steps will allow the files you load into this blob storage account to be publicly accessible across domains. If you have sensitive data, it is better to use local files or fork this project and modify it to work with your data in a secure manner.

Follow these steps to host files in a CORS enabled Azure blob storage:

- If you don't have one already, go into the Azure portal and create a [blob storage account resource](#).

- Go to the blob storage account resource in the Azure portal that will be hosting your files.

- Under **Settings**, click on the [Resource Sharing \(CORS\)](#) button.

- Under the **Blob services** section, add new entry with the following options:

Allowed origins	Allowed methods	Allowed headers	Exposed headers	Max age
*	GET	*	*	360000

Output file format

The output of the project builder is a zip file with following structure:

Project_name

> results

> tasks

{ Project_name_0.json

{ Project_name_1.json

{ project_builder_settings.json

summary.csv

- The **output zip file** will have the same name as the project name you specified, with spaces replaced with underscores (_).

- The **tasks** folder will contain the project name, and have a task number appended to it.

- The **project_builder_settings.json** file contains all the settings used to create the project.

- The **summary.csv** file contains all of the individual tasks that you defined using the grid in the project builder. This is a useful file for assigning those individual tasks to annotators.

See the [Result file format](#) documentation for more details on the output data schema of the labeler tool.

Task file format

The individual task files are GeoJSON files that each contain a feature collection with a single feature.

- The **geometry** of this feature defines the outline of the task area.

- The **id** of the feature is the ID of the individual task and should match the name of the task file.

- The **bbox** property is the bounding box area of the **geometry**, and is used by the labeler to set the initial viewing area when the task is loaded.

- The **properties** of the feature contain the following information:

Property name	Type	Description
---------------	------	-------------

project_name	string	The name of the project.
--------------	--------	--------------------------

name	string	The same as the id of the Feature.</