The Committee on Earth Observation Satellites (CEOS) Visualization Environment (COVE) application programming interface (API) is a suite of endpoints for executing tasks and retrieving results for a collection of COVE tools.

All tasks created using the COVE API will be available in the COVE UI, and all tasks created in the COVE UI will be available in the COVE API.

A valid COVE username and password are required for all requests. A COVE login may be created in the COVE UI by clicking on the Log In/Register link in the upper right corner.

COVE tool API sets are throttled to help reduce system overload. Throttle limits are subject to change if the need arises. The throttle limits will be detailed in each tool section below, and will be updated in this document as changes are made.

To submit a request to the service, include the appropriate JSON payload in the body of a POST request for the endpoint. Requests should include the trailing slash in the url. Examples using curl with each type of request are provided in each section below.

All responses will either be a JSON response or a binary file. All failed responses will be in JSON format, and will include a status (standard HTTP status code).

Execution start in all task status responses are in UTC time.

For CSV files, use unicode UTF-8.

Please refer to the COVE UI User's Guide for more information on the usage of each COVE tools.

The COVE tools with COVE API endpoints include:

- **Acquisition Forecaster** – A set of endpoints which provides users with the ability to execute a task in the Acquisition Forecaster tool, and retrieve results of the task (status, CSV, PNG).
• **Coverage Analyzer** – A set of endpoints which provides users with the ability to execute a task in the Coverage Analyzer tool, and retrieve results of the task (status, CSV, PNG, GeoTiff, and NetCDF).

• **Revisits Calculator** – A set of endpoints which provides users with the ability to execute a task in the Revisits Calculator tool, and retrieve results of the task (status, JSON, CSV, PNG).

• **Coincident Calculator** – A set of endpoints which provides users with the ability to execute a task in the Coincident Calculator Tool (Forecasted or Archived), retrieve results of the task (status, JSON, PNG), and includes endpoints which lists details of constellations available, and endpoints to create constellations (archived or forecasted).

• **Data Browser** – A set of endpoints which provides users with the ability to execute a task in the Data Browser tool, and retrieve results of the task (status, count, CSV, PNG).

• **Country Coverage** – A set of endpoints which provides users with the ability to execute a task in the Country Coverage tool, and retrieve results of the task (status, PDF, CSV, and PNG).

• **Utilities** – A set of endpoints for access to COVE utilities.

Other COVE API endpoints included in this document are:

• **Missions**
  - **forecasted** – An endpoint for Acquisition Forecaster, Coincident Calculator, and Revisits Calculator which lists details on forecasted missions available.
  - **archived** – An endpoint for Coverage Analyzer, Coincident Calculator, and Data Browser which lists details on archived missions available.

• **Custom Missions** – A pair of endpoints which provides users with the ability to list details on available custom missions, and create new custom missions and custom instruments.

• **Regions** – A set of endpoints which provides users with the ability to list, retrieve, and create a custom region of interest which may be used in Acquisition Forecaster, Coincident Calculator, Coverage Analyzer, Revisits Calculator, and Data Browser.
Acquisition Forecaster

Acquisition Forecaster is a tool that provides users with the ability to predict when a satellite has the ability to image a region of interest. Users may send requests for a task history, submit a new task, get a task status, retrieve a CSV with scene metadata, and retrieve a PNG map image of the task results.

Acquisition Forecaster throttle limits:
- POST: 5 requests per hour
- GET: 25 requests per hour

Retrieve task history

Lists all Acquisition Forecaster tasks in COVE for the user. The results include task ids which will allow users to retrieve results for existing tasks.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/

URL Parameters: None

Response Example:
```
[
  {
    "task_id": "<TASK ID>",
    "missions": [
      {
        "mission": "Sentinel-1A",
        "instrument": "C-SAR",
        "mode": "IWS"
      }, {
        "mission": "Sentinel-2A",
        "instrument": "MSI",
        "mode": ""
      }
    ],
    "custom_missions": [{
      "mission": "MissionA",
      "mission": "MissionB",
      "mission": "MissionC"
    }
  ]
]```
Submit a task

Task Input Parameters for Acquisition Forecaster:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to three month from the latest TLE date. Valid dates may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>end_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to three month from the latest TLE date. Valid dates may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>region_folder</td>
<td>String</td>
<td>Yes</td>
<td>Valid folder names may be found by requesting the Regions list.</td>
</tr>
<tr>
<td>region</td>
<td>String</td>
<td>Yes</td>
<td>Valid region names may be found by submitting a COVE API request for the Regions list.</td>
</tr>
<tr>
<td>custom_missions</td>
<td>CustomMissionsFilter[]</td>
<td>No</td>
<td>Available custom missions may be found by submitting a COVE API request for the Custom Missions list. Refer to the Special Data Types section for more information on CustomMissionsFilter.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>missions</td>
<td>MissionsFilter[]</td>
<td>No</td>
<td>Valid missions may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on MissionsFilter.</td>
</tr>
<tr>
<td>calculate_solar_angles</td>
<td>Boolean</td>
<td>No</td>
<td>If true, solar elevation, solar azimuth, and solar zenith will be calculated for each forecasted scene. If false, some forecasted scenes may have solar angles calculated if the scenes were cached by COVE in another task.</td>
</tr>
</tbody>
</table>

**Request Data Example:**
```
{
    "start_date": "2020-01-01",
    "end_date": "2020-01-31",
    "region_folder": "Africa",
    "region": "Kenya",
    "custom_missions": [{
        "mission": "MissionA",
        "instrument": "BeamA"
    }],
    "missions": [{
        "mission": "Sentinel-1A",
        "instrument": "C-SAR",
        "mode": "IWS"
    }]
}
```

**Request Method:** POST

**Request URL:**
https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/

**URL Parameters:** None

**Response Examples:**
Example successful response from request where the results are not ready to retrieve:
```
{"status": 202, "message": "Acquisition Forecaster task as been submitted. Please wait for the results.", "id": "<TASK ID>"}
```
Example successful response from a request where the results are ready to retrieve:
{"status": 200, "message": "Acquisition Forecaster results are ready", "id": "<TASK ID>"}

Failed Response Examples:
{"status": 400, "message": "<Error Message>"}
{"status": 417, "message": "<Error Message>"}

Retrieve task status
Once a task has been submitted, users should periodically check the task status.

Status codes:
- 200: task has completed
- 202: task is still running
- 417: task has terminated in an error

If the status code is 200 and success is true, it indicates the task has predicted scenes.
If a status code 200 is received, users may send a request for CSV or PNG results.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/<TASK ID>/

URL Parameters: None

Response Example:
{
    "execution_start": "2020-09-16 18:19:19",
    "execution_time": "00:01:34",
    "start_date": "2017-01-01",
    "end_date": "2017-01-08",
    "modes": "MSI - Sentinel-2A, IWS - C-SAR - Sentinel-1A, BeamA - MissionA",
    "region": "Ghana",
    "complete": true,
    "status": 200,
    "message": "Task successfully completed."
}
Retrieve CSV file

The CSV file (binary) will contain the predicted scene metadata generated for the task. The CSV file contains the columns: mission, acquisition_time, tle_epoch, center_latitude, center_longitude, and orbit_direction.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/csv/<TASK ID>/

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extended</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. If true, full extended CSV is returned. If false, abbreviated CSV is returned. Default value is false.</td>
</tr>
</tbody>
</table>

Failed Response Example:
{'status': 404, 'message': 'File not found.'}

Retrieve PNG results

The PNG image file is a map image which outlines scenes on a map of the requested region of interest within the start date and end date for the task.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/<TASK ID>/

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>orbit_direction</td>
<td>String</td>
<td>No</td>
<td>Values: ascending or descending. No orbit_direction will return results for both ascending and descending.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>transparent</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. If transparent is true, the map background will be omitted and the background will instead be transparent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is false, image resolution is 200dpi.</td>
</tr>
<tr>
<td>resolution</td>
<td>Integer</td>
<td>No</td>
<td>Values should be in dots per inch (dpi).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is false, resolution must be less than or equal to the default resolution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default resolution is 200dpi.</td>
</tr>
<tr>
<td>line_width</td>
<td>Float</td>
<td>No</td>
<td>Values should be greater than 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allows users to define the line width of scene footprint outlines. This parameter does not apply to region outlines.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default value is 0.5.</td>
</tr>
<tr>
<td>show_region_outline</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allows users to show or hide the region outline.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>map_coordinates</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map coordinates used to generate a PNG image. If value is true, the map coordinates will be returned instead of a PNG image. If an orbit_direction parameter is used to generate an image, it should be included in requests for map coordinates.</td>
</tr>
<tr>
<td>map_legend</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map legend used to generate a PNG image. If value is true, the map legend will be returned instead of a PNG image.</td>
</tr>
</tbody>
</table>

**Map Coordinates Response Example:**
```json
{
    "status": 200,
    "map_coordinates": [
        [-8.661654877310252, -0.16135473389262667],
        [-8.661654877310252, 14.838645266107374],
        [6.3383451226897485, 14.838645266107374],
        [6.3383451226897485, -0.16135473389262667],
        [-8.661654877310252, -0.16135473389262667]
    ]
}
```

**Map Legend Response Example:**
```json
{
    "status": 200,
    "map_legend": {
        "ALOS-2 – PALSAR-2 – Full Accessible": "#fc8d59"
    }
}
```

**Failed Response Example:**
```json
{"status": 404, 'message': 'File not found.'}
```
Example API requests using curl

Request user history:

Request for Acquisition Forecaster task:

Request task status:

Request CSV results:

Request CSV results (extended):

Request PNG map image (orbit direction: all):

Request PNG map image (orbit direction: ascending):

Request PNG map image (orbit direction: descending):

Request PNG map image (transparent):
curl -u username:password https://ceos-cove.org/en/api/acquisition_forecaster/png/<TASK ID>/?transparent=true -o filename.png

Request PNG map image (resolution):
curl -u username:password https://ceos-cove.org/en/api/acquisition_forecaster/png/<TASK ID>/?resolution=200 -o filename.png
Request PNG map image (line_width):
<TASK ID>/?line_width=0.5 -o filename.png

Request PNG map image (show_region_outline):
<TASK ID>/?show_region_outline=false -o filename.png

Request PNG map image (map_coordinates):
<TASK ID>/?map_coordinates=true -o filename.png

Request PNG map image (map_coordinates with orbit_direction):
<TASK ID>/?map_coordinates=true&orbit_direction=ascending -o filename.png

Request PNG map image (map_legend):
<TASK ID>/?map_legend=true -o filename.png

Request PNG map image (map_legend with orbit_direction):
<TASK ID>/?map_legend=true&orbit_direction=ascending -o filename.png

Request PNG map image (map_coordinates with map_legend):
<TASK ID>/?map_coordinates=true&map_legend=true -o filename.png

Request PNG map image (map_coordinates with map_legend and orbit_direction):
<TASK ID>/?map_coordinates=true&map_legend=true&orbit_direction=ascending -o filename.png
Coverage Analyzer

Coverage Analyzer is a tool that provides users with the ability to analyze the historical satellite coverage of a region of interest. Users may send requests for a task history, submit a new task, get a task status, retrieve a CSV with scene metadata, retrieve a PNG map image of the task results, retrieve a GeoTiff file, and retrieve a NetCDF file.

Coverage Analyzer throttle limits:
- POST: 5 requests per hour
- GET: 25 requests per hour

Retrieve user history

Lists all Coverage Analyzer tasks in COVE for the user. The results includes task ids which will allow users to retrieve results for existing tasks.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/coverage_analyzer/

URL Parameters: None

Response Example:
[
    {
        "task_id": "<TASK ID>",
        "missions": [
            {
                "mission": "Sentinel-2A",
                "instrument": "MSI",
                "mode": ""
            }
        ],
        "start_date": "2019-01-01",
        "end_date": "2019-01-08",
        "region": "Ghana",
        "filter_options": {}
    }
]
Submit a task

Task Input Parameters for Coverage Analyzer:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>end_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>region_folder</td>
<td>String</td>
<td>Yes</td>
<td>Valid folder names may be found by requesting the Regions list.</td>
</tr>
<tr>
<td>region</td>
<td>String</td>
<td>Yes</td>
<td>Valid region names may be found by submitting a COVE API request for the Regions list.</td>
</tr>
<tr>
<td>missions</td>
<td>MissionsFilter[]</td>
<td>Yes</td>
<td>Valid missions may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on MissionsFilter.</td>
</tr>
<tr>
<td>discretization</td>
<td>DiscretizationFilter</td>
<td>Yes</td>
<td>Refer to the Special Data Types section for more information on DiscretizationFilter.</td>
</tr>
<tr>
<td>filters</td>
<td>MetadataFilter</td>
<td>No</td>
<td>Refer to the Special Data Types section for more information on MetadataFilter.</td>
</tr>
</tbody>
</table>
Request Data Example:
{
    "start_date": "2020-01-01",
    "end_date": "2020-01-31",
    "region_folder": "South America",
    "region": "Argentina",
    "missions": [{
        "mission": "Sentinel-1A",
        "instrument": "C-SAR",
        "mode": "IWS"
    }],
    "discretization": {
        "type": "",
        "size": 1.0,
        "unit": "deg",
        "include_overlap": false
    },
    "filters": {
        "cloud_cover": null,
        "day_night": "",
        "orbit_direction": "",
        "processing_level": ""
    }
}

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/coverage_analyzer/

URL Parameters: None

Response Examples:
Example successful response from request where the results are not ready to retrieve:
{"status": 202, "message": "Coverage Analyzer task as been submitted. Please wait for the results.", "id": "<TASK ID>"}

Example successful response from a request where the results are ready to retrieve:
{"status": 200, "message": "Coverage Analyzer results are ready", "id": "<TASK ID>"}

Failed Response Examples:
{"status": 400, "message": "<Error Message>"}
{"status": 417, "message": "<Error Message>"}
Retrieve task status

Once a task has been submitted, users should periodically check the task status.

Status codes:
- 200: task has completed
- 202: task is still running
- 417: task has terminated in an error

If the status code is 200 and success is true, it indicates the task has scene data.

If a status code 200 is received, users may send a request for CSV, PNG, NetCDF, or GeoTIFF results.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/coverage_analyzer/<TASK ID>/

URL Parameters: None

Response Example:
{
   "execution_start": "2020-09-18 11:56:42",
   "execution_time": "00:01:02",
   "start_date": "2019-01-01",
   "end_date": "2019-01-08",
   "modes": "IWS - C-SAR - Sentinel-1A",
   "region": "Africa: Ghana",
   "discretization": "0.5 deg",
   "filter_options": {},
   "complete": true,
   "status": 200,
   "message": "Task successfully completed."
}

"success": true
Retrieve CSV results
The CSV file (binary) will contain the scene metadata for the task. The CSV file contains the columns: mission, name, acquisition_date, browse_url, order_url, start_time, stop_time, center_latitude, and center_longitude.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/coverage_analyzer/csv/<TASK ID>/

URL Parameters: None

Failed Response Example:
{'status': 404, 'message': 'File not found.'}

Retrieve PNG results
The PNG image file shows the discretized region of interest on a map where the color of each block in the discretized region reflects the number of scenes intersecting the block for the start date and end date and for the missions requested in the task.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/<TASK ID>/

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transparent</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is true, the map background will be omitted and the background will instead be transparent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is false, image resolution is 200dpi.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>----------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| transparency           | Float     | No       | Values should be between 0 and 1. Transparency defines the opacity of the coincidence results.透
|                        |           |          | Transparency does not apply to the region outline.                                             透
|                        |           |          | Default transparency is 0.65.                                                                 透
| resolution             | Integer   | No       | Values should be in dots per inch (dpi). If transparent is false, resolution must be less than or equal to the default resolution.透
|                        |           |          | Default resolution is 200dpi.                                                                 透
| show_region_outline    | Boolean   | No       | Values: true or false. Allows users to show or hide the region outline.                       透
| map_coordinates        | Boolean   | No       | Values: true or false. Allows users to fetch the map coordinates used to generate a PNG image.透
|                        |           |          | If value is true, the map coordinates will be returned instead of a PNG image.               透
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>map_legend</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map legend used to generate a PNG image. If value is true, the map legend will be returned instead of a PNG image. The number range on the left side of the map legend represents the number of scenes in the block.</td>
</tr>
</tbody>
</table>

Map Coordinates Response Example:
```json
{
    "status": 200,
    "map_coordinates": [
        [-8.661654877310252, -0.16135473389262667],
        [-8.661654877310252, 14.838645266107374],
        [6.3383451226897485, 14.838645266107374],
        [6.3383451226897485, -0.16135473389262667],
        [-8.661654877310252, -0.16135473389262667]
    ]
}
```

Map Legend Response Example:
```json
{
    "status": 200,
    "map_legend": {
        "0 - 0": "#d73027",
        "1 - 1": "#f46d43",
        "2 - 2": "#fdae61",
        "3 - 3": "#fee08b",
        "4 - 4": "#ffffbf",
        "5 - 5": "#d9ef8b",
        "6 - 6": "#a6d963",
        "7 - 7": "#66bd63",
        "8 - 9": "#1a9850"
    }
}
```

Failed Response Example:
```json
{'status': 404, 'message': 'File not found.'}
```
Retrieve GeoTIFF results
The file exported will be in GeoTIFF format (file extension .tif).

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/coverage_analyzer/geotiff/<TASK ID>/

URL Parameters: None

Failed Response Example:
{"status": 404, 'message': 'File not found.'}

Retrieve NetCDF results
The file exported will be in NetCDF format (file extension .nc).

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/coverage_analyzer/netcdf/<TASK ID>/

URL Parameters: None

Failed Response Example:
{"status": 404, 'message': 'File not found.'}

Example API requests using curl

Request user history:
Request for Coverage Analyzer task:

Request task status:

Request CSV File:

Request PNG map image:

Request PNG map image (transparent):
curl -u username:password https://ceos-cove.org/en/api/coverage_analyzer/png/<TASK ID>/?transparent=true -o filename.png

Request PNG map image (transparency):
curl -u username:password https://ceos-cove.org/en/api/coverage_analyzer/png/<TASK ID>/?transparency=0.65 -o filename.png

Request PNG map image (resolution):
curl -u username:password https://ceos-cove.org/en/api/coverage_analyzer/png/<TASK ID>/?resolution=200 -o filename.png

Request PNG map image (show_region_outline):

Request PNG map image (map_coordinates):
curl -u username:password https://ceos-cove.org/en/api/coverage_analyzer/png/<TASK ID>/?map_coordinates=true -o filename.png

Request PNG map image (map_coordinates): 
curl -u username:password https://ceos-cove.org/en/api/coverage_analyzer/png/<TASK ID>/?map_coordinates=true&map_legend=true -o filename.png
Request GeoTIFF File:

Request NetCDF File:
Revisits Calculator

Revisits Calculator is a tool which allows users to analyze estimated satellite coverage and revisits for a region of interest. Users may send requests for a task history, submit a new task, get a task status, retrieve JSON results, retrieve a CSV with scene metadata, and retrieve a PNG map image of the task results.

Revisits Calculator throttle limits:
- POST: 5 requests per hour
- GET: 25 requests per hour

Retrieve user history

Lists all Revisits Calculator tasks in COVE for the user. The results includes task ids which will allow users to retrieve results for existing tasks.

Request Method: GET

Request URL:

URL Parameters: None

Response Example:
[
  {
    "task_id": "<TASK ID>",
    "missions": [
      {
        "mission": "Sentinel-1A",
        "instrument": "C-SAR",
        "mode": "IWS"
      },
      {
        "mission": "Sentinel-2A",
        "instrument": "MSI",
        "mode": ""
      }
    ],
    "custom_missions": [{
      "mission": "MissionA",
      ""
Submit a task

Task Input Parameters for Revisits Calculator:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>end_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>region_folder</td>
<td>String</td>
<td>Yes</td>
<td>Valid folder names may be found by requesting the Regions list.</td>
</tr>
<tr>
<td>region</td>
<td>String</td>
<td>Yes</td>
<td>Valid region names may be found by submitting a COVE API request for the Regions list.</td>
</tr>
<tr>
<td>custom_missions</td>
<td>CustomMissionsFilter[]</td>
<td>No</td>
<td>Available custom missions may be found by submitting a COVE API request for the Custom Missions list. Refer to the Special Data Types section for more information on CustomMissionsFilter.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>missions</td>
<td>MissionsFilter[]</td>
<td>No</td>
<td>Valid missions may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on MissionsFilter.</td>
</tr>
<tr>
<td>discretization</td>
<td>DiscretizationFilter</td>
<td>Yes</td>
<td>Refer to the Special Data Types section for more information on DiscretizationFilter.</td>
</tr>
</tbody>
</table>

**Request Data Example:**

```json
{
    "start_date": "2020-01-01",
    "end_date": "2020-01-31",
    "region_folder": "Africa",
    "region": "Botswana",
    "region": "Botswana",
    "custom_missions": [],
    "missions": [{
        "mission": "Sentinel-2A",
        "instrument": "MSI",
        "mode": ""
    }],
    "discretization": {
        "type": "S2Tiling",
        "size": null,
        "unit": ""
    }
}
```

**Request Method:** POST

**Request URL:**

**URL Parameters:** None

**Response Examples:**

Example successful response from request where the results are not ready to retrieve:

```json
{"status": 202, "message": "Revisits Calculator task as been submitted. Please wait for the results.", "id": "<TASK ID>"}
```
Retrieve task status

Once a task has been submitted, users should periodically check the task status.

Status codes:
- 200: task has completed
- 202: task is still running
- 417: task has terminated in an error

If the status code is 200 and success is true, it indicates the task has scene data.

If a status code 200 is received, users may send a request for JSON, CSV, or PNG results.

Request Method: GET

Request URL:

URL Parameters: None

Response Example:
{
    "execution_start": "2020-09-17 22:27:20",
    "execution_time": "00:00:07",
    "start_date": "2019-01-02",
    "end_date": "2019-01-03",
    "modes": "IWS - C-SAR – Sentinel-1A",
    "region": "Africa: Ghana",
    "discretization": "1.0 deg",
    "complete": true,
    "status": 200,
    "message": "Task successfully completed."
}
Retrieve JSON results

The JSON results will contain a list of coordinates for each discretized block in the region of interest. In each discretized block, it will show a title, count of scenes in the block, mean time between revisits, median time between revisits, maximum time between revisits, and minimum time between revisits in the block for ascending orbit, descending orbit, or both.

Request Method: GET

Request URL: https://ceos-cove.org/en/api/v1_1/revisits_calculator/json/<TASK ID>/

URL Parameters: None

Response Example:

```
{
    "0.0,5.0,0.0,6.0,1.0,6.0,1.0,5.0,0.0,5.0": {
        "title": "Region Discretization Longitude 0.0, Latitude 5.0",
        "revisits": 2,
        "minimum": "12:23:00",
        "median": "12:23:00",
        "mean": "12:23:00",
        "maximum": "12:23:00",
        "desc_revisits": 1,
        "desc_minimum": "0:00:00",
        "desc_median": "0:00:00",
        "desc_mean": "0:00:00",
        "desc_maximum": "0:00:00",
        "count": 2,
        "asc_revisits": 1,
        "asc_minimum": "0:00:00",
        "asc_median": "0:00:00",
        "asc_mean": "0:00:00",
        "asc_maximum": "0:00:00"
    }
}

(JSON RESULTS TRUNCATED)
```
Retrieve CSV results

The CSV file (binary) will contain the scene metadata for the task. The CSV file contains the columns: mission, acquisition_time, tle_epoch, center_latitude, center_longitude, and orbit_direction.

Request Method: GET

Request URL:

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extended</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. If true, full extended CSV is returned. If false, abbreviated CSV is returned. Default value is false.</td>
</tr>
</tbody>
</table>

Failed Response Examples:

{"status": 400, "message": "<Error Message>"}
{"status": 417, "message": "<Error Message>"}

Retrieve PNG results

The PNG image file shows the discretized region of interest on a map where the color of each block in the discretized region reflects the number of times missions/custom missions intersect the block within the start date and end date for the task.

Request Method: GET

Request URL:
URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>orbit_direction</td>
<td>String</td>
<td>No</td>
<td>Values: ascending or descending. No orbit_direction will return results for both ascending and descending.</td>
</tr>
<tr>
<td>transparent</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is true, the map background will be omitted and the background will instead be transparent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is false, image resolution is 200dpi.</td>
</tr>
<tr>
<td>transparency</td>
<td>Float</td>
<td>No</td>
<td>Values should be between 0 and 1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Transparency defines the opacity of the coincidence results. Transparency does not apply to the region outline.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default transparency is 0.65.</td>
</tr>
<tr>
<td>resolution</td>
<td>Integer</td>
<td>No</td>
<td>Values should be in dots per inch (dpi).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is false, resolution must be less than or equal to the default resolution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default resolution is 200dpi.</td>
</tr>
<tr>
<td>show_region_outline</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allows users to show or hide the region outline.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>map_coordinates</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map coordinates used to generate a PNG image. If value is true, the map coordinates will be returned instead of a PNG image.</td>
</tr>
<tr>
<td>map_legend</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map legend used to generate a PNG image. If value is true, the map legend will be returned instead of a PNG image. The number range on the left side of the map legend represents the number of revisits in the block.</td>
</tr>
</tbody>
</table>

**Map Coordinates Response Example:**

```json
{
    "status": 200,
    "map_coordinates": [
        [-8.661654877310252, -0.16135473389262667],
        [-8.661654877310252, 14.838645266107374],
        [6.3383451226897485, 14.838645266107374],
        [6.3383451226897485, -0.16135473389262667],
        [-8.661654877310252, -0.16135473389262667]
    ]
}
```

**Map Legend Response Example:**

```json
{
    "status": 200,
    "map_legend": {
        "0 - 3":="#d73027",
        "4 - 7":="#f46d43",
        "8 - 11":="#fdbe61",
    }
}
```
Failed Response Examples:
{'status': 404, 'message': 'File not found.'}

Example API requests using curl

Request user history:

Request for Revisits Calculator task:

Request task status:

Request JSON File:

Request CSV File:

Request CSV File (extended):

Request PNG map image:
Request PNG map image (orbit direction: ascending):

Request PNG map image (orbit direction: descending):

Request PNG map image (transparent):

Request PNG map image (transparency):

Request PNG map image (resolution):

Request PNG map image (show_region_outline):

Request PNG map image (map_coordinates):

Request PNG map image (map_legend):

Request PNG map image (map_coordinates with map_legend):
Coincident Calculator

Coincident Calculator is a tool which allows users to determine the coincidence of satellite instrument modes and/or custom constellations observing a location in the same day within a region of interest using historical (archived) or predicted (forecasted) scene data. Users may send requests for a task history, submit a new task, get a task status, retrieve JSON results, and retrieve a PNG map image of the task results.

Coincident Calculator throttle limits:
- POST: 5 requests per hour
- GET: 25 requests per hour

Forecasted Task

Retrieve forecasted constellations list
Lists all custom forecasted constellations in Coincident Calculator in COVE for the user.

Request Method: GET

Request URL:

URL Parameters: None

Response Example:
```
{
  "constellations": {
    "Landsat 7/8": {
      "missions": [
        "Landsat 7 - ETM",
        "Landsat 8 - TIRS"
      ],
      "custom_missions": []
    },
    "CBERS4_SatA": {
      "missions": [
        "CBERS-4 - MUXCam"
      ]
    }
  }
}
```
Create constellation

Create a constellation from existing forecasted missions and/or custom mission.

Task Input Parameters for Coincident Calculator (Forecasted):

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constellation</td>
<td>String</td>
<td>Yes</td>
<td>Name of constellation</td>
</tr>
<tr>
<td>custom_missions</td>
<td>CustomMissionsFilter[]</td>
<td>No</td>
<td>Available custom missions may be found by submitting a COVE API request for the Custom Missions list. Refer to the Special Data Types section for more information on CustomMissionsFilter.</td>
</tr>
<tr>
<td>missions</td>
<td>MissionsFilter[]</td>
<td>No</td>
<td>Valid missions may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on MissionsFilter.</td>
</tr>
</tbody>
</table>

Request Data Example:
```json
{
    "name": "SatGroup",
    "custom_missions": [
        {
            "mission": "SatA",
            "instrument": "BeamA"
        }
    ],
    "missions": [
        {
            "mission": "Sentinel-1A",
            "instrument": "C-SAR",
            "mode": "IWS"
        }
    ]
}
```
Request Method: POST

Request URL:

URL Parameters: None

Response Examples:
Example successful response:
{"status": 200, "message": "Constellation created successfully"}

Failed Response Examples:
{"status": 400, "message": "<Error Message>"}
{"status": 417, "message": "<Error Message>"}

Retrieve user history

Lists all forecasted Coincident Calculator tasks in COVE for the user. The results includes task ids which will allow users to retrieve results for existing tasks.

Request Method: GET

Request URL:

URL Parameters: None

Response Example:
[  
  
  
  
  
    "task_id": "<TASK ID>",
    "missions": [
      
      
      
      
        "mission": "Sentinel-1A",
        "instrument": "C-SAR",
        "mode": "IWS"
    ],
  
  
  
  
  
]
"mission": "Sentinel-2A",
"instrument": "MSI",
"mode": ""
},
"custom_missions": [],
"constellations": [],
"start_date": "2019-01-01",
"end_date": "2019-01-08",
"region": "Ghana",
"coincidence_seconds": 86400
],
]

Submit a task

Task Input Parameters for Coincident Calculator (Forecasted):

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>end_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>region_folder</td>
<td>String</td>
<td>Yes</td>
<td>Valid folder names may be found by requesting the Regions list.</td>
</tr>
<tr>
<td>region</td>
<td>String</td>
<td>Yes</td>
<td>Valid region names may be found by submitting a COVE API request for the Regions list.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>constellations</td>
<td>ConstellationsFilter[]</td>
<td>No</td>
<td>Available constellations may be found by submitting a COVE API request for the constellations list. Refer to the Special Data Types section for more information on ConstellationsFilter.</td>
</tr>
<tr>
<td>custom_missions</td>
<td>CustomMissionsFilter[]</td>
<td>No</td>
<td>Available custom missions may be found by submitting a COVE API request for the Custom Missions list. Refer to the Special Data Types section for more information on CustomMissionsFilter.</td>
</tr>
<tr>
<td>missions</td>
<td>MissionsFilter[]</td>
<td>No</td>
<td>Valid missions may be found by submitting a COVE API request for the Missions forecasted list. Refer to the Special Data Types section for more information on MissionsFilter.</td>
</tr>
<tr>
<td>discretization</td>
<td>DiscretizationFilter</td>
<td>Yes</td>
<td>Refer to the Special Data Types section for more information on DiscretizationFilter.</td>
</tr>
<tr>
<td>coincidence_days</td>
<td>integer</td>
<td>No</td>
<td>Range: 0 to 5, defaults to -1. Do not use coincidence_days if coincidence_seconds is specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If neither coincidence_days or coincidence_seconds are specified, defaults to coincidence_days=0.</td>
</tr>
<tr>
<td>coincidence_seconds</td>
<td>integer</td>
<td>No</td>
<td>Range: 0 to 432000, defaults to -1. Do not use coincidence_seconds if coincidence_days is specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If neither coincidence_days or coincidence_seconds are specified, defaults to coincidence_days=0.</td>
</tr>
</tbody>
</table>
Request Data Example:
{
    "start_date": "2020-01-01",
    "end_date": "2020-01-31",
    "region_folder": "North America",
    "region": "Canada",
    "constellations": [],
    "custom_missions": [],
    "missions": [
        {
            "mission": "Sentinel-1A",
            "instrument": "C-SAR",
            "mode": "IWS"
        },
        {
            "mission": "Sentinel-2A",
            "instrument": "MSI",
            "mode": ""
        }
    ],
    "discretization": {
        "size": 1.0,
        "unit": "deg"
    },
    "coincidence_seconds": 86400
}

**Request Method:** POST

**Request URL:**

**URL Parameters:** None

**Response Examples:**
Example successful response from request where the results are not ready to retrieve:
{"status": 202, "message": "Coincident Calculator task as been submitted. Please wait for the results.", "id": "<TASK ID>"}

Example successful response from a request where the results are ready to retrieve:
{"status": 200, "message": "Coincident Calculator results are ready", "id": "<TASK ID>"}
Retrieve task status

Once a task has been submitted, users should periodically check the task status.

Status codes:
- 200: task has completed
- 202: task is still running
- 417: task has terminated in an error

If the status code is 200 and success is true, it indicates the task has scene data.

If a status code 200 is received, users may send a request for JSON or PNG results.

Request Method: GET

Request URL:

URL Parameters: None

Response Example:
{
    "execution_start": "2020-09-14 12:38:34",
    "execution_time": "00:00:04",
    "start_date": "2019-01-01",
    "end_date": "2019-01-08",
    "modes": "IWS - C-SAR - Sentinel-1A, MSI – Sentinel-2A",
    "constellations": ",",
    "region": "Ghana",
    "discretization": "1.0 deg",
    "coincidence_seconds": 86400,
    "complete": true,
    "status": 200,
    "message": "Task successfully completed.",
    "success": true
}
Retrieve JSON results

The JSON results will contain a list of coordinates for each discretized block in the region of interest. In each discretized block, it will show a title, coincidence dates, scene (acquisition) metadata, coincidence count, coincidence (dates and count) where scenes are ascending, coincidence where scenes are descending.

Request Method: GET

Request URL:

URL Parameters: None

Response Example:
{
  "0.0,8.0,0.0,9.0,1.0,9.0,1.0,8.0,0.0,8.0": {
    "title": "Region Discretization Longitude 0.0, Latitude 8.0",
    "coincidence": "2019-01-06",
    "acquisitions": [
      {
        "name": "Landsat 8 - TIRS",
        "acquisition_time": "2019-01-06 10:20:30",
        "tle_epoch": "2019-01-06 02:58:25",
        "final_prediction": false,
        "db_insert_date": "2020-09-18",
        "center_latitude": 10.27,
        "center_longitude": -0.57,
        "orbit_direction": "Descending",
        "scene_coords": "0.6660363424876055,11.962612495110209,-1.014522933011194,12.203808739278776,-1.7926181561449557,8.578126402975682,-0.13054000602536892,8.339640664025538,0.6660363424876055,11.962612495110209"
      },
      {
        "name": "CBERS-4 - MUXCam",
        "acquisition_time": "2019-01-06 22:05:30",
        "tle_epoch": "2019-01-06 22:03:25",
        "final_prediction": false,
        "db_insert_date": "2020-09-18",
        "center_latitude": 7.42,
        "center_longitude": -0.39,
        "orbit_direction": "Ascending",
        "scene_coords": "-0.5336858426123086,5.553900730294913,0.5330482808132545,5.719459991383701,-0.24584157033993395,9.280660729314569,-1.320750071368642,9.12287381737216,-0.5336858426123086,5.553900730294913"
      }
    ],
    "coincidence_asc": "",
  }
}
Retrieve PNG results

The PNG image file shows the discretized region of interest on a map where the color of each block in the discretized region reflects the number of pairs of missions/custom missions/constellations intersecting the block within the start date and end date for the task.

Request Method:  GET

Request URL:

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>orbit_direction</td>
<td>String</td>
<td>No</td>
<td>Values: ascending or descending. No orbit_direction will return results for both ascending and descending.</td>
</tr>
<tr>
<td>transparent</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. If transparent is true, the map background will be omitted and the background will instead be transparent. If transparent is false, image resolution is 200dpi.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>transparency</td>
<td>Float</td>
<td>No</td>
<td>Values should be between 0 and 1. Transparency defines the opacity of the coincidence results. Transparency does not apply to the region outline. Default transparency is 0.65.</td>
</tr>
<tr>
<td>resolution</td>
<td>Integer</td>
<td>No</td>
<td>Values should be in dots per inch (dpi). If transparent is false, resolution must be less than or equal to the default resolution. Default resolution is 200dpi.</td>
</tr>
<tr>
<td>show_region_outline</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to show or hide the region outline.</td>
</tr>
<tr>
<td>map_coordinates</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map coordinates used to generate a PNG image. If value is true, the map coordinates will be returned instead of a PNG image.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>map_legend</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map legend used to generate a PNG image. If value is true, the map legend will be returned instead of a PNG image. The number range on the left side of the map legend represents the number of coincidences in the block.</td>
</tr>
</tbody>
</table>

**Map Coordinates Response Example:**

```javascript
{
    "status": 200,
    "map_coordinates": [
        [-8.661654877310252, -0.16135473389262667],
        [-8.661654877310252, 14.838645266107374],
        [6.3383451226897485, 14.838645266107374],
        [6.3383451226897485, -0.16135473389262667],
        [-8.661654877310252, -0.16135473389262667]
    ]
}
```

**Map Legend Response Example:**

```javascript
{
    "status": 200,
    "map_legend": {
        "0 - 0": "#d73027",
        "1 - 1": "#f46d43",
        "2 - 2": "#fdae61",
        "3 - 3": "#fee08b",
        "4 - 4": "#ffffbf",
        "5 - 5": "#d9ef8b",
        "6 - 6": "#1a9850"
    }
}
```

**Failed Response Example:**

```javascript
{'status': 404, 'message': 'File not found.'}
```
Example API requests using curl

Request user history:

Request for Forecasted Coincident Calculator task:

Request forecasted custom constellations list:

Create forecasted custom constellation:

Request task status:

Request JSON File:

Request PNG map image:

Request PNG map image (orbit direction: ascending):

Request PNG map image (orbit direction: descending):
Request PNG map image (transparent):

Request PNG map image (transparency):

Request PNG map image (resolution):

Request PNG map image (show_region_outline):

Request PNG map image (map_coordinates):

Request PNG map image (map_legend):

Request PNG map image (map_coordinates with map_legend):

Archived Task

Retrieve archived constellations list
Lists all custom archived constellations in Coincident Calculator in COVE for the user.

Request Method: GET

Request URL:

URL Parameters: None
Response Example:
{
    "constellations": {
        "L7/8": [
            "Landsat 7 – ETM",
            "Landsat 8 – OLI/TIRS"
        ],
        "S1": [
            "Sentinel-1A - C-SAR – EW",
            "Sentinel-1A - C-SAR – IWS",
            "Sentinel-1A - C-SAR – SM",
            "Sentinel-1A - C-SAR – WV",
            "Sentinel-1B - C-SAR – EW",
            "Sentinel-1B - C-SAR – IWS",
            "Sentinel-1B - C-SAR – SM",
            "Sentinel-1B - C-SAR – WV"
        ]
    }
}
}

Create constellation

Create a constellation from existing archived missions.

Task Input Parameters for Coincident Calculator (Forecasted):

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constellation</td>
<td>String</td>
<td>Yes</td>
<td>Name of constellation</td>
</tr>
<tr>
<td>missions</td>
<td>MissionsFilter[]</td>
<td>No</td>
<td>Valid missions may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on MissionsFilter.</td>
</tr>
</tbody>
</table>

Request Data Example:
{
    "name": "SatGroup",
    "missions": [
        {
            "mission": "Sentinel-1A",
            "instrument": "C-SAR",
            "mode": "IWS"
        },
    ]
}
Retrieve user history
Lists all archived Coincident Calculator tasks in COVE for the user. The results includes task ids which will allow users to retrieve results for existing tasks.

Request Method: GET

Request URL:

URL Parameters: None

Response Example:
[
  {
    "task_id": "<TASK ID>",
    "missions": [}
Submit a task

Task Input Parameters for Coincident Calculator (Archived):

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>end_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>region_folder</td>
<td>String</td>
<td>Yes</td>
<td>Valid folder names may be found by requesting the Regions list.</td>
</tr>
<tr>
<td>region</td>
<td>String</td>
<td>Yes</td>
<td>Valid region names may be found by submitting a COVE API request for the Regions list.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>constellations</td>
<td>ConstellationsFilter[]</td>
<td>No</td>
<td>Valid constellations may be found by submitting a COVE API request for the constellations list. Refer to the Special Data Types section for more information on ConstellationsFilter.</td>
</tr>
<tr>
<td>missions</td>
<td>MissionsFilter[]</td>
<td>Yes</td>
<td>Valid missions may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on MissionsFilter.</td>
</tr>
<tr>
<td>discretization</td>
<td>DiscretizationFilter</td>
<td>Yes</td>
<td>Refer to the Special Data Types section for more information on DiscretizationFilter.</td>
</tr>
<tr>
<td>coincidence_days</td>
<td>integer</td>
<td>No</td>
<td>Range: 0 to 5, defaults to -1. Do not use coincidence_days if coincidence_seconds is specified. If neither coincidence_days or coincidence_seconds are specified, defaults to coincidence_days=0.</td>
</tr>
<tr>
<td>coincidence_seconds</td>
<td>integer</td>
<td>No</td>
<td>Range: 0 to 432000, defaults to -1. Do not use coincidence_seconds if coincidence_days is specified. If neither coincidence_days or coincidence_seconds are specified, defaults to coincidence_days=0.</td>
</tr>
</tbody>
</table>

Request Data Example:
```
{
  "start_date": "2020-01-01",
  "end_date": "2020-01-31",
  "region_folder": "Oceania",
  "region": "Australia",
  "constellations": [],
  "missions": []
}
```
"discretization": {
    "size": 1.0,
    "unit": "deg"
},
"coincidence_seconds": 172800
}

Request Method: GET

Request URL:

URL Parameters: None

Response examples:

Example successful response from request where the results are not ready to retrieve:
{"status": 202, "message": "Coincident Calculator task as been submitted. Please wait for the results.", "id": "<TASK ID>"}

Example successful response from a request where the results are ready to retrieve:
{"status": 200, "message": "Coincident Calculator results are ready", "id": "<TASK ID>"}

Failed Response Examples:
{"status": 400, "message": "<Error Message>"}
{"status": 417, "message": "<Error Message>"}

Retrieve task status

Once a task has been submitted, users should periodically check the task status.

Status codes:
- 200: task has completed
- 202: task is still running
- 417: task has terminated in an error

If the status code is 200 and success is true, it indicates the task has scene data.
If a status code 200 is received, users may send a request for JSON or PNG results.
Request Method: GET

Request URL:

URL Parameters: None

Response Example:
{
    "execution_start": "2020-09-14 12:31:41",
    "execution_time": "00:01:02",
    "start_date": "2019-01-01",
    "end_date": "2019-01-08",
    "modes": "IWS - C-SAR - Sentinel-1A, MSI - Sentinel-2A, OLI/TIRS - Landsat 8",
    "constellations": "",
    "region": "Africa: Ghana",
    "discretization": "5.0 deg",
    "coincidence_seconds": 0,
    "complete": true,
    "status": 200,
    "message": "Task successfully completed.",
    "success": true
}

Retrieve JSON results
The JSON results will contain a list of coordinates for each discretized block in the region of interest. In each discretized block, it will show a title, coincidence dates, scene (acquisition) metadata, coincidence count, coincidence (dates and count) where scenes are ascending, coincidence where scenes are descending.

Request Method: GET

Request URL:

URL Parameters: None

Response Example:
[
    "-4.0,11.0,-4.0,12.0,-3.0,12.0,-3.0,11.0,-4.0,11.0": {
}
"title": "Region Discretization Longitude -4.0, Latitude 11.0",
"coincidence": "2019-01-03",
"acquisitions": [
  {
    "mission": "Sentinel-2A - MSI",
    "name": "6718341",
    "acquisition_date": "2019-01-03",
    "browse_url": <BROWSE JPG URL>,
    "order_url": <ORDER URL>,
    "sun_azimuth": 150.134493948532,
    "granule_name": "L1C_T30PVT_A018452_20190103T04928",
    "sun_elevation": 39.3260813690852,
    "scene_stop_time": "2019-01-03T11:00:14.693Z",
    "cloud_cover_full": 0.0,
    "scene_start_time": "2019-01-03T10:49:28.901Z",
    "scene_center_latitude": 11.2632574,
    "scene_center_longitude": -3.4135721,
    "orbit_direction": "Descending",
    "scene_coords": 
      ",-3.9181409,11.7585667,
      ,2.910728,10.7670169,2.910728,10.7670169,-3.9181409,11.7585667"
  },
  {
    "mission": "Sentinel-1A - C-SAR – IWS",
    "name": "S1A_IW_SLC__1SDV_20190103T182746_20190103T182813_025317_02CD19_622F",
    "acquisition_date": "2019-01-03",
    "browse_url": <BROWSE JPG URL>,
    "order_url": <ORDER URL>,
    "path": 45,
    "frame": 32,
    "orbit": 25317,
    "scene_end_time": "2019-01-03T18:28:13.000000",
    "center_latitude": 11.562,
    "center_longitude": -3.8554,
    "processing_level": "SLC",
    "scene_start_time": "2019-01-03T18:27:46.000000",
    "sending_polarization": "VV",
    "receiving_polarization": "VH",
    "orbit_direction": "Ascending",
    "scene_coords": 
      ,2.560575,10.969422,-4.81989,10.522881,-5.154573,12.152341,-2.8816349999999997,12.594698,-2.560575,10.969422"
  }
],
"coincidence_asc": "",
"coincidence_desc": "",
"coincidence_count": 1,
"coincidence_asc_count": 0,
"coincidence_desc_count": 0
Retrieve PNG results

The PNG image file shows the discretized region of interest on a map where the color of each block in the discretized region reflects the number of pairs of missions/constellations intersecting the block within the start date and end date for the task.

Request Method: GET

Request URL:

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>orbit_direction</td>
<td>String</td>
<td>No</td>
<td>Values: ascending or descending. No orbit_direction will return results for both ascending and descending.</td>
</tr>
<tr>
<td>transparent</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. If transparent is true, the map background will be omitted and the background will instead be transparent. If transparent is false, image resolution is 200dpi.</td>
</tr>
<tr>
<td>transparency</td>
<td>Float</td>
<td>No</td>
<td>Values should be between 0 and 1. Transparency defines the opacity of the coincidence results. Transparency does not apply to the region outline. Default transparency is 0.65.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>resolution</td>
<td>Integer</td>
<td>No</td>
<td>Values should be in dots per inch (dpi). If transparent is false, resolution must be less than or equal to the default resolution. Default resolution is 200dpi.</td>
</tr>
<tr>
<td>show_region_outline</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to show or hide the region outline.</td>
</tr>
<tr>
<td>map_coordinates</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map coordinates used to generate a PNG image. If value is true, the map coordinates will be returned instead of a PNG image.</td>
</tr>
<tr>
<td>map_legend</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map legend used to generate a PNG image. If value is true, the map legend will be returned instead of a PNG image. The number range on the left side of the map legend represents the number of coincidences in the block.</td>
</tr>
</tbody>
</table>

Map Coordinates Response Example:
```
{
  "status": 200,
  "map_coordinates": [
    [-8.661654877310252, -0.16135473389262667],
    [-8.661654877310252, 14.838645266107374],
    [6.3383451226897485, 14.838645266107374],
  ]
}```
Map Legend Response Example:
{
    "status": 200,
    "map_legend": {
        "0 - 0": "#d73027",
        "1 - 1": "#f46d43",
        "2 - 2": "#fdae61",
        "3 - 3": "#fee08b",
        "4 - 4": "#ffffbf",
        "5 - 5": "#d9ef8b",
        "6 - 6": "#a6d96a",
        "7 - 7": "#66bd63",
        "8 - 9": "#1a9850"
    }
}

Failed Response Example:
{"status": 404, 'message': 'File not found.'}

Example API requests using curl

Request user history:

Request for Archived Coincident Calculator task:

Request archived custom constellations list:
Create archived custom constellation:

Request task status:

Request JSON File:

Request PNG map image:

Request PNG map image (orbit direction: ascending):

Request PNG map image (orbit direction: descending):

Request PNG map image (transparent):

Request PNG map image (transparency):

Request PNG map image (resolution):

Request PNG map image (show_region_outline):

Request PNG map image (map_coordinates):
Request PNG map image (map_legend):

Request PNG map image (map_coordinates with map_legend):
Data Browser

Data Browser is a tool that allows users to view the satellite image archives from multiple CEOS missions. Users may send requests for a task history, submit a new task, get a task status, retrieve a scene count for the task, retrieve CSV results of scene metadata, and retrieve a PNG map image of the task results.

Data Browser throttle limits:
- POST: 5 requests per hour
- GET: 25 requests per hour

Retrieve user history

Lists all Data Browser tasks in COVE for the user. The results includes task ids which will allow users to retrieve results for existing tasks.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/data_browser/

URL Parameters: None

Response Example:
```
[
    {
        "task_id": "<TASK ID>",
        "missions": [
            {
                "mission": "Sentinel-1A",
                "instrument": "C-SAR",
                "mode": "IWS"
            },
            {
                "mission": "Sentinel-2A",
                "instrument": "MSI",
                "mode": ""
            }
        ],
        "start_date": "2019-01-01",
```
Submit a task

Task Input Parameters for Data Browser:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>end_date</td>
<td>DateString</td>
<td>Yes</td>
<td>Range: mission launch date to the latest acquisition date. Valid dates may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on DateString.</td>
</tr>
<tr>
<td>region_folder</td>
<td>String</td>
<td>Yes</td>
<td>Valid folder names may be found by requesting the Regions list.</td>
</tr>
<tr>
<td>region</td>
<td>String</td>
<td>Yes</td>
<td>Valid region names may be found by submitting a COVE API request for the Regions list.</td>
</tr>
<tr>
<td>missions</td>
<td>MissionsFilter[]</td>
<td>Yes</td>
<td>Valid missions may be found by submitting a COVE API request for the Missions archived list. Refer to the Special Data Types section for more information on MissionsFilter.</td>
</tr>
<tr>
<td>filters</td>
<td>MetadataFilter</td>
<td>No</td>
<td>Refer to the Special Data Types section for more information on MetadataFilter.</td>
</tr>
</tbody>
</table>
Request Data Example:
{
  "start_date": "2020-01-01",
  "end_date": "2020-01-31",
  "region_folder": "Europe",
  "region": "Spain",
  "missions": [],
  "filters": {
    "cloud_cover": null,
    "day_night": ",",
    "orbit_direction": ",",
    "processing_level": ","
  }
}

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/data_browser/

URL Parameters: None

Response Examples:
Example successful response from request where the results are not ready to retrieve:
{"status": 202, "message": "Data Browser task as been submitted. Please wait for the results.", "id": "<TASK ID>"}

Example successful response from a request where the results are ready to retrieve:
{"status": 200, "message": "Data Browser results are ready", "id": "<TASK ID>"}

Failed Response Examples:
{"status": 400, "message": "<Error Message>"}
{"status": 417, "message": "<Error Message>"}

Retrieve task status

Once a task has been submitted, users should periodically check the task status.

Status codes:
  •  200: task has completed
- 202: task is still running
- 417: task has terminated in an error

If the status code is 200 and success is true, it indicates the task has scene data.
If a status code 200 is received, users may send a request for JSON or PNG results.

**Request Method:** GET

**Request URL:**
https://ceos-cove.org/en/api/v1_1/data_browser/<TASK ID>/

**URL Parameters:** None

**Response Example:**
```
{
  "execution_start": "2020-09-14 12:31:41",
  "execution_time": "02:38:58",
  "start_date": "2019-01-01",
  "end_date": "2019-01-08",
  "modes": "OLI/TIRS - Landsat 8, IWS - C-SAR - Sentinel-1A, MSI - Sentinel-2A",
  "region": "Ghana",
  "filter_options": {},
  "complete": true,
  "status": 200,
  "message": "Task successfully completed."
}
```

**Retrieve scene count**

Returns the number of scenes in the task results.

**Request Method:** GET

**Request URL:**
https://ceos-cove.org/en/api/v1_1/data_browser/count/<TASK ID>/

**URL Parameters:** None
Retrieve CSV results

The CSV file (binary) will contain the scene metadata for the task. The CSV file contains the columns: mission, name, acquisition_date, browse_url, order_url, start_time, stop_time, center_latitude, and center_longitude.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/data_browser/csv/<TASK ID>/

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>extended</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. If true, full extended CSV is returned. If false, abbreviated CSV is returned. Default value is false.</td>
</tr>
</tbody>
</table>

Failed Response Example:
{"status": 404, 'message': 'File not found.'}

Retrieve PNG results

The PNG image file is a map image which outlines scenes on a map of the requested region of interest within the start date and end date for the task.

Request Method: GET
**Request URL:**
https://ceos-cove.org/en/api/v1_1/data_browser/png/<TASK ID>/

**URL Parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transparent</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. If transparent is true, the map background will be omitted and the background will instead be transparent. If transparent is false, image resolution is 200dpi.</td>
</tr>
<tr>
<td>resolution</td>
<td>Integer</td>
<td>No</td>
<td>Values should be in dots per inch (dpi). If transparent is false, resolution must be less than or equal to the default resolution. Default resolution is 200dpi.</td>
</tr>
<tr>
<td>line_width</td>
<td>Float</td>
<td>No</td>
<td>Values should be greater than 0. Allows users to define the line width of scene footprint outlines. This parameter does not apply to region outlines. Default value is 0.5.</td>
</tr>
<tr>
<td>show_region_outline</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to show or hide the region outline.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>map_coordinates</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false. Allows users to fetch the map coordinates used to generate a PNG image. If value is true, the map coordinates will be returned instead of a PNG image.</td>
</tr>
</tbody>
</table>

Map Coordinates Response Example:
```
{
    "status": 200,
    "map_coordinates": [
        [-8.661654877310252, -0.16135473389262667],
        [-8.661654877310252, 14.838645266107374],
        [6.3383451226897485, 14.838645266107374],
        [6.3383451226897485, -0.16135473389262667],
        [-8.661654877310252, -0.16135473389262667]
    ]
}
```

Failed Response Example:
```
{'status': 404, 'message': 'File not found.'}
```

Example API requests using curl

Request user history:

Request for Data Browser task:

Request task status:
Request scene count:

Request CSV File:

Request CSV File (extended):

Request PNG map image:

Request PNG map image (transparent):

Request PNG map image (resolution):

Request PNG map image (line_width):
curl -u username:password https://ceos-cove.org/en/api/data_browser/png/<TASK ID>/?line_width=0.5 -o filename.png

Request PNG map image (show_region_outline):

Request PNG map image (map_coordinates):
Country Coverage

Country Coverage is a tool that allows users to generate and view GFOI Country Coverage reports for over 70 countries and 3 constellations. Users may send requests for a constellations list, a countries list, a daily history, submit a new task, get a task status, retrieve JSON results, retrieve PDF results, retrieve CSV results of scene metadata, retrieve a PNG bar graph, and retrieve ZIP archive file of the reports from the task results. Users must submit a new task to retrieve monthly cached report as well as to initiate report tasks.

Country Coverage throttle limits:
- POST: 1 Country Coverage task running in COVE at a time
- GET: 25 requests per hour

Retrieve constellations list

Lists all constellations and the missions associated with each constellation. These constellations are only applicable to the Country Coverage tool and API.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/country_coverage/constellations/

URL Parameters: None

Response Example:
{
    "constellations": {
        "Landsat": [
            "Landsat 5 - TM",
            "Landsat 7 - ETM",
            "Landsat 8 - OLI/TIRS"
        ],
        "Sentinel-1": [
            "Sentinel-1A - C-SAR - EW",
            "Sentinel-1A - C-SAR - IWS",
            "Sentinel-1A - C-SAR - SM",
            "Sentinel-1A - C-SAR - WV",
            "Sentinel-1B - C-SAR - EW",
            "Sentinel-1B - C-SAR - IWS",
        ]
    }
}
Retrieve countries list

Lists over 70 countries available for generating Country Coverage reports in COVE.

**Request Method:** GET

**Request URL:**
https://ceos-cove.org/en/api/v1_1/country_coverage/countries/

**URL Parameters:** None

**Response Example:**
```
{
    "countries": [
        "Algeria",
        "Argentina",
        "Australia",
        "Bangladesh",
        "Belize",
        ...
    ]
}
```

Retrieve daily history

Lists all existing and generating Country Coverage reports in COVE for the current day.

**Request Method:** GET

**Request URL:**
https://ceos-cove.org/en/api/v1_1/country_coverage/reports/
Retrieve monthly cached history

Lists all existing and generating Country Coverage reports in COVE for the current day.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/country_coverage/reports/?daily_task=false

URL Parameters: None

Response Example:
{
  "existing": {
    "Algeria": [
      "Landsat",
      "Sentinel-1",
      "Sentinel-2"
    ],
    "Argentina": [
      "Landsat",
      "Sentinel-1"
    ],
    "Australia": [
      "Landsat"
    ]
  },
  "generating": {
    "Argentina": [
      "Sentinel-2"
    ],
    "Australia": [
      "Sentinel-1",
      "Sentinel-2"
    ]
  }
}
Submit a task

Country Coverage tasks have a size limit. The limit is the number of constellations times the number of countries requested in the task.

Task Input Parameters for Country Coverage:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constellations</td>
<td>ConstellationsFilter[]</td>
<td>Yes</td>
<td>Refer to section Retrieve constellations list for a list of constellations. Refer to the Special Data Types section for more information on ConstellationsFilter.</td>
</tr>
<tr>
<td>countries</td>
<td>CountriesFilter[]</td>
<td>Yes</td>
<td>Refer to section Retrieve countries list for a list of valid countries. Refer to the Special Data Types section for more information on CountriesFilter.</td>
</tr>
<tr>
<td>daily_task</td>
<td>Boolean</td>
<td>No</td>
<td>Default value is true. If value is true, reports will be generated based on current data. If value is false, resulting task id may be used to retrieve monthly cached reports.</td>
</tr>
</tbody>
</table>
Request Data Example:
{
    "constellations": ["Landsat", "Sentinel-1", "Sentinel-2"],
    "countries": ["Algeria", "Argentina", "Australia", "Bangladesh", "Belize"],
    "daily_task": true
}

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/country_coverage/reports/

URL Parameters: None

Response Examples:
Example successful response from request where the results are not ready to retrieve:
{"status": 202, "message": "Country Coverage reports task as been submitted. Please wait for the results.", "id": "<TASK ID>"}

Example successful response from a request where the results are ready to retrieve:
{"status": 200, "message": "Country Coverage reports results are ready", "id": "<TASK ID>"}

Failed Response Examples:
{"status": 429,
 "message": "Another country coverage task is currently running. Please try again later."}
{"status": 400, "message": "<Error Message>"}
{"status": 417, "message": "<Error Message>"}
Retrieve task status

Once a task has been submitted, users should periodically check the task status and progress percentage.

Status codes:
- 200: task has completed
- 202: task is still running
- 417: task has terminated in an error

If the status code is 200 and success is true, it indicates the task has scene data.
If a status code 200 is received, users may send a request for JSON or PNG results.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/country_coverage/reports/<TASK ID>/

URL Parameters: None

Response Example:
```json
{
  "execution_start": "2020-09-18 14:11:21",
  "execution_time": "03:05:07",
  "constellations": "Landsat",
  "countries": "Benin",
  "complete": true,
  "status": 200,
  "message": "Your task has completed successfully.",
  "progress": 85,
  "completed": {
    "Benin": [
      "Landsat"
    ]
  }
}
```

Retrieve JSON results

The JSON results will contain scene counts for every year since mission launch for all missions in all constellations and all countries in a task request. If a URL parameter is
included in the request URL, the results will be filtered by the parameter(s) provided.

Request Method: GET

Request URL: https://ceos-cove.org/en/api/v1_1/country_coverage/json/<TASK ID>/

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constellation</td>
<td>String</td>
<td>No</td>
<td>Refer to section Retrieve constellations list.</td>
</tr>
<tr>
<td>country</td>
<td>String</td>
<td>No</td>
<td>Refer to section Retrieve countries list.</td>
</tr>
</tbody>
</table>

Response Example:

```
{
    "Sentinel-1": {
        "Benin": [
            {
                "year": 2014,
                "Sentinel-1A": 0,
                "Sentinel-1B": 0
            },
            {
                "year": 2015,
                "Sentinel-1A": 442,
                "Sentinel-1B": 0
            },
            {
                "year": 2016,
                "Sentinel-1A": 575,
                "Sentinel-1B": 0
            }
        ],
        (RESULTS TRUNCATED)
    }
}
```

Retrieve PDF results

The PDF file will contain a chart with the totals of scenes for the country and mission in the constellation requested for each year since launch. It will also contain a bar graph of the totals for each year. Reports for Sentinel-1 include an additional bar graph where values are
separated by processing level. If the task associated with the task id only requests one constellation and one country, no URL parameters are needed.

**Request Method:** GET

**Request URL:**
https://ceos-cove.org/en/api/v1_1/country_coverage/pdf/<TASK ID>/

**URL Parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constellation</td>
<td>String</td>
<td>No</td>
<td>The constellation parameter is only required if multiple constellations are associated with the task. Refer to section Retrieve constellations list.</td>
</tr>
<tr>
<td>country</td>
<td>String</td>
<td>No</td>
<td>The country parameter is only required if multiple countries are associated with the task. Refer to section Retrieve countries list.</td>
</tr>
</tbody>
</table>

**Failed Response Example:**
{'status': 404, 'message': 'File not found.'}

**Retrieve CSV results**
The CSV file (binary) will contain the scene metadata. The CSV file contains the columns:

- **Landsat:** Date Acquired, Year, Mission, Path, Row, Scene Cloud Cover, Processing Level, Tier, Product ID, Scene ID
- **Sentinel-1:** Date Acquired, Year, Mission, Path, Frame, Processing Level, Orbit Direction, Granule Name
- **Sentinel-2:** Date Acquired, Year, Mission, Scene Center Latitude, Scene Center Longitude, Scene Cloud Cover, Scene ID
Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/country_coverage/csv/<TASK ID>/

URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constellation</td>
<td>String</td>
<td>No</td>
<td>The constellation parameter is only required if multiple constellations are associated with the task. Refer to section Retrieve constellations list.</td>
</tr>
<tr>
<td>country</td>
<td>String</td>
<td>No</td>
<td>The country parameter is only required if multiple countries are associated with the task. Refer to section Retrieve countries list.</td>
</tr>
</tbody>
</table>

Failed Response Example:
{'status': 404, 'message': 'File not found.'}

Retrieve PNG results

The PNG image file is a bar graph which shows the number of scenes for each year for each mission for the country and constellation parameter specified in the request. If the task associated with the task id only requests one constellation and one country, no URL parameters are needed.

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/country_coverage/png/<TASK ID>/

URL Parameters:
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>constellation</td>
<td>String</td>
<td>No</td>
<td>The constellation parameter is only required if multiple constellations are associated with the task. Refer to section Retrieve constellations list.</td>
</tr>
<tr>
<td>country</td>
<td>String</td>
<td>No</td>
<td>The country parameter is only required if multiple countries are associated with the task. Refer to section Retrieve countries list.</td>
</tr>
</tbody>
</table>

**Failed Response Example:**

```json
{'status': 404, 'message': 'File not found.'}
```

**Retrieve ZIP results**

The ZIP archive file will contain all PDF and CSV files for all constellations and countries in the task request.

**Request Method:** GET

**Request URL:**

https://ceos-cove.org/en/api/v1_1/country_coverage/zip/<TASK ID>/

**URL Parameters:** None

**Failed Response Example:**

```json
{'status': 404, 'message': 'File not found.'}
```

**Example API requests using curl**

**Request Constellations List:**

```bash
```

**Request Countries List:**

```bash
```
Request Country Coverage daily history:

Request Country Coverage monthly cached history:

Request for Country Coverage daily reports task:
curl -H "Content-Type: application/json" -X POST -u username:password -d '{"constellations": ["<CONSTELLATION>"], "countries": ["<COUNTRY>"]}, daily_task=true'}
https://ceos-cove.org/en/api/country_coverage/reports/

Request for Country Coverage monthly cached reports task:
curl -H "Content-Type: application/json" -X POST -u username:password -d '{"constellations": ["<CONSTELLATION>"], "countries": ["<COUNTRY>"]}, daily_task=false'}
https://ceos-cove.org/en/api/country_coverage/reports/

Request task status:

Request JSON results for all countries and constellations in task:

Request JSON results for all constellations for a country in task:

Request JSON results for all countries for a constellation in task:

Request JSON results for a country and a constellation in task:

Request PDF report:

Request CSV File:
Request PNG Bar Chart:
curl -u username:password https://ceos-cove.org/en/api/country_coverage/reports/png/<TASK ID>/?country=<COUNTRY>&constellation=<CONSTELLATION> -o filename.png

Request ZIP File:
Utilities

Utilities is a collection of calculators to provide users with a means of estimating various parameters of interest about CEOS satellite missions. There are two collections of calculators in utilities repeating orbit and swath calculators. Repeating orbit calculators available are ground track interval, period/velocity, and sun synchronous orbit. Swath calculators available are incidence to pointing, fov from swath width, swath width from fov, swath from incidence, and swath/off-nadir.

Utilities throttle limits:
- POST: 25 requests per hour

Repeating Orbit

Ground Track Interval

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 200 to 1500</td>
</tr>
<tr>
<td>days_to_repeat</td>
<td>Float</td>
<td>Yes</td>
<td>Greater than 0</td>
</tr>
<tr>
<td>latitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 0 to 90</td>
</tr>
<tr>
<td>revs_to_repeat</td>
<td>Float</td>
<td>No</td>
<td>Greater than 0; Calculated if not provided.</td>
</tr>
</tbody>
</table>

Request Data Example:

```json
{
    "altitude": 800,
    "days_to_repeat": 16,
    "latitude": 0,
    "revs_to_repeat": 228
}
```

Request Method: POST
**Request URL:**
https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/ground_track_interval/

**URL Parameters:** None

**Response Example:**
```json
{
    "altitude": 800.0,
    "days_to_repeat": 16.0,
    "latitude": 0.0,
    "revs_to_repeat": 228.0,
    "fundamental_interval_deg_long": 25.2632,
    "fundamental_interval_km": 2812.3,
    "subinterval_deg_long": 1.5789,
    "subinterval_km": 175.8
}
```

---

**Period/Velocity**

**Input parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 200 to 1500</td>
</tr>
</tbody>
</table>

**Request Data Example:**

```json
{
    "altitude": 800
}
```

**Request Method:** POST

**Request URL:**
https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/period_velocity/

**URL Parameters:** None

**Response Example:**

```json
{
    "altitude": 800.0,
    "period": 100.87,
    "velocity": 7.45
}
```
Sun Synchronous Orbit

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 200 to 1500</td>
</tr>
<tr>
<td>revs_to_repeat</td>
<td>Float</td>
<td>Yes</td>
<td>Greater than 0; Calculated if not provided.</td>
</tr>
<tr>
<td>days_to_repeat</td>
<td>Float</td>
<td>No</td>
<td>Greater than 0</td>
</tr>
</tbody>
</table>

Request Data Example:

```json
{
    "altitude": 800,
    "revs_to_repeat": 16,
    "days_to_repeat": 1
}
```

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/sun_synchronous_orbit/

URL Parameters: None

Response Example:

```json
{
    "altitude": 800.0,
    "revs_to_repeat": 16.0,
    "days_to_repeat": 1.0,
    "estimated_altitude": 268.1315,
    "estimated_inclination": 96.5607
}
```
Swath Calculator

Incidence To Pointing

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 200 to 1500</td>
</tr>
<tr>
<td>incidence_angle</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 0 to 90</td>
</tr>
</tbody>
</table>

Request Data Example:

```json
{
    "altitude": 800,
    "incidence_angle": 15
}
```

Request Method: POST

Request URL:

URL Parameters: None

Response Example:

```json
{
    "altitude": 800.0,
    "incidence_angle": 15.0,
    "off_nadir_angle": 13.3
}
```

FOV from Swath Width

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 200 to 1500</td>
</tr>
<tr>
<td>swath_width</td>
<td>Float</td>
<td>Yes</td>
<td>Greater than 0</td>
</tr>
</tbody>
</table>
Request Data Example:
{
  "altitude": 800,
  "swath_width": 15
}

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/utilities/swath_calculator/fov_from_swath_width/

URL Parameters: None

Response Example:
{
  "altitude": 800.0,
  "swath_width": 210.9,
  "fov": 15.0,
  "half_fov": 7.5
}

### Swath Width from FOV

**Input parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 200 to 1500</td>
</tr>
<tr>
<td>fov</td>
<td>Float</td>
<td>Yes</td>
<td>Greater than 0</td>
</tr>
</tbody>
</table>

Request Data Example:
{
  "altitude": 800,
  "fov": 15
}

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/utilities/swath_calculator/swath_width_from_fov/
Swath From Incidence

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 200 to 1500</td>
</tr>
<tr>
<td>min_incidence_angle</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 0 to 90</td>
</tr>
<tr>
<td>max_incidence_angle</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 0 to 90</td>
</tr>
</tbody>
</table>

Request Data Example:

```
{
    "altitude": 800,
    "min_incidence_angle": 0,
    "max_incidence_angle": 16
}
```

Request Method: POST

Request URL:

URL Parameters: None

Response Example:

```
{
    "altitude": 800.0,
    "min_incidence_angle": 0.0,
    "max_incidence_angle": 16.0,
    "off_nadir_angle": 7.09,
}
```
Swath/Off-Nadir

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>altitude</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 200 to 1500</td>
</tr>
<tr>
<td>off_nadir_angle</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 0 to 90</td>
</tr>
<tr>
<td>fov</td>
<td>Float</td>
<td>Yes</td>
<td>Range: 0 to 90</td>
</tr>
</tbody>
</table>

Request Data Example:

```json
{
   "altitude": 800,
   "off_nadir_angle": 25,
   "fov": 5
}
```

Request Method: POST

Request URL:

URL Parameters: None

Response Example:

```json
{
   "altitude": 800.0,
   "off_nadir_angle": 25.0,
   "fov": 5.0,
   "swath_width": 88.93
}
```

Example API requests using curl

Calculator: Repeating Orbit: Ground Track Interval:
curl -H "Content-Type: application/json" -X POST -u username:password -d "{"altitude": <ALTITUDE>, "days_to_repeat": <DAYS TO REPEAT>, "latitude": <LATITUDE>, "revs_to_repeat": <REVOLUTIONS TO REPEAT}>" https://ceos-
cove.org/en/api/utilities/repeating_orbit/ground_track_interval/

**Calculator: Repeating Orbit: Period/Velocity:**

**Calculator: Repeating Orbit: Sun Synchronous Orbit:**
curl -H "Content-Type: application/json" -X POST -u username:password -d '{"altitude": <ALTITUDE>, "revs_to_repeat": <REVOLUTIONS TO REPEAT>, "days_to_repeat": <DAYS TO REPEAT>}' https://ceos-cove.org/en/api/utilities/repeating_orbit/sun_synchronous_orbit/

**Calculator: Swath Calculator: Incidence To Pointing:**

**Calculator: Swath Calculator: FOV From Swath Width:**

**Calculator: Swath Calculator: Swath Width From FOV:**

**Calculator: Swath Calculator: Swath From Incidence:**

**Calculator: Swath Calculator: Swath/Off-Nadir:**
Missions

Missions is a list of forecasted or archived missions. Forecasted missions are used in Acquisition Forecaster, Revisits Calculator, and Coincident Calculator. Archived missions are used in Data Browser, Coverage Analyzer, and Coincident Calculator. The missions results will help users determine correct mission, instrument, and mode names as well as determine valid start and end dates when creating tasks.

Missions throttle limits:
  - GET: 25 requests per hour

Retrieve forecasted list

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/missions/forecasted/

URL Parameters: None

Response Example:

```json
[
  {
    "mission": "ALOS",
    "norad_id": "28931",
    "mim_id": 242,
    "launch_date": "2006-01-24",
    "decommissioned_date": "2011-04-22",
    "earliest_tle_date": "2006-01-25",
    "latest_tle_date": "2011-07-20",
    "status": "deactivated",
    "instrument_modes": [
      {
        "instrument": "PRISM",
        "mode": "Nadir"
      },
      {
        "instrument": "AVNIR-2",
        "mode": "Standard"
      },
      {
        "instrument": "AVNIR-2",
        "mode": "Full Accessible"
      },
      {
        "instrument": "PALSAR",
        "mode": ""
      }
    ]
  }
]```
Retrieve archived list

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/missions/archived/

URL Parameters: None

Response Example:
[
  {
    "mission": "Landsat 5",
    "norad_id": "14780",
    "mim_id": 226,
    "launch_date": "1984-03-01",
    "decommissioned_date": "2013-06-05",
    "latest_acquisition_date": "2012-05-05",
    "status": "deactivated",
    "instrument_modes": [
      {
        "instrument": "TM",
        "mode": ""
      }
    ],
  },
  (MISSIONS LIST TRUNCATED)
]

Example API requests using curl

Forecasted missions list:

Archived missions list:
The Custom Missions feature allows users to create notional or proposed missions, and request a list of custom missions a user has already created. Custom Missions may be used in Acquisition Forecaster, Revisits Calculator, and Coincident Calculator to submit tasks.

Custom Missions throttle limits:
- GET: 25 requests per hour

**Retrieve custom missions list**

**Request Method**: GET

**Request URL**: https://ceos-cove.org/en/api/v1_1/custom_missions/

**URL Parameters**: None

**Response Example**:
```
{
  "custom_missions": [{
    "custom_mission_name": "Mission_A",
    "orbit_type": "Sun Synchronous",
    "revolutions_per_day_revs": 14.56,
    "altitude_km": 705.0,
    "revolutions_to_repeat_revs": 233.0,
    "local_time": "11:10",
    "direction": "ascending",
    "instruments": [{
      "custom_instrument_name": "Beam_A",
      "field_of_view_deg": 5.0,
      "pointing_angle_deg": 5.0
    }]
  }
}
```

**Create a custom mission**

The endpoint will allow users to create one custom mission with multiple custom instruments. Custom missions may be created from existing missions or with new mission details. Custom instruments may be created from an existing mission's instrument-mode or using new instrument details. Custom instruments may be added to an existing custom instrument.
Refer to the COVE UI User's Guide for additional details on creating custom missions.

Input parameters to create a Custom Mission:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>custom_mission</td>
<td>ExistingMissionFilter, CircularOrbitFilter,</td>
<td>Yes</td>
<td>Refer to the Special Data Types section for more information on ExistingMissionFilter, CircularOrbitFilter, SunSynchronousOrbitFilter, or AdvancedOrbitFilter.</td>
</tr>
<tr>
<td></td>
<td>SunSynchronousOrbitFilter, or AdvancedOrbitFilter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>custom_instruments</td>
<td>CustomInstrumentsFilter[]</td>
<td>Yes</td>
<td>Refer to the Special Data Types section for more information on CustomInstrumentsFilter.</td>
</tr>
</tbody>
</table>

Example request to add a new instrument to an existing custom mission CustomInstrumentFilter:

```json
{
    "custom_mission": {
        "name": "Mission1"
    },
    "custom_instruments": [
        {
            "name": "beam",
            "existing": true,
            "mission": {
                "mission": "Sentinel-2A",
                "instrument": "MSI",
                "mode": ""
            }
        }
    ]
}
```

Example request to create a custom mission from an existing mission using ExistingMissionFilter and CustomInstrumentFilter:

```json
{
    "custom_mission": {
        "name": "Euclid",
        "existing": true,
        "mission": "Sentinel-1A"
    },
    "custom_instruments": [
        {
            "name": "beam",
            "existing": true,
```
Example request to create a custom mission with a circular orbit using CircularOrbitFilter:
{
  "custom_mission": {
    "name": "Pythagoras",
    "existing": false,
    "orbit_type": "circular",
    "inclination": 0,
    "altitude": 600,
    "longitude": 180
  },
  "custom_instruments": [
    {
      "name": "beam",
      "existing": false,
      "field_of_view": 5.0,
      "pointing_angle": 5.0
    }
  ]
}

Example request to create a custom mission with a sun synchronous orbit using SunSynchronousOrbitFilter:
{
  "custom_mission": {
    "name": "Archimedes",
    "existing": false,
    "orbit_type": "sun_synchronous",
    "altitude": 705.0,
    "revolutions_to_repeat": 233.0,
    "local_time": "11:10",
    "direction": "ascending"
  },
  "custom_instruments": [
    {
      "name": "beam",
      "existing": false,
      "field_of_view": 1.2,
      "pointing_angle": 5.0
    }
  ]
}
Example request to create a custom mission with an advanced orbit using `AdvancedOrbitFilter`:

```json
{
  "custom_mission": {
    "name": "Newton",
    "existing": false,
    "orbit_type": "advanced",
    "semimajor_axis": 6800.0,
    "eccentricity": 0,
    "inclination": 35,
    "argument_of_perigee": 90,
    "raan": 130,
    "true_anomaly": 250
  },
  "custom_instruments": [
    {
      "name": "beam",
      "existing": false,
      "field_of_view": 9.0,
      "pointing_angle": 0
    }
  ]
}
```

**Request Method:** POST

**Request URL:**
https://ceos-cove.org/en/api/v1_1/custom_missions/

**URL Parameters:** None

**Response Examples:**

**Example successful response:**
```json
{"status": 200, "message": "Custom mission created successfully"}
```

**Failed Response Examples:**
```json
{"status": 417, "message": "An error occurred while creating custom mission"}
```
Example API requests using curl

Custom missions list:

Add instrument to existing custom mission:

Create custom mission from existing mission:

Create custom mission (circular orbit):

Create custom mission (sun synchronous orbit):

Create custom mission (advanced orbit):
Regions

Regions allows users to create custom regions of interest. Regions are used in Acquisition Forecaster, Coverage Analyzer, Revisits Calculator, Coincident Calculator, and Data Browser. Users may send requests for a regions list, details of a region, import a region by KML string, KML binary file, or Shapefile, export a region by KML file or Shapefile, and retrieve a PNG map image of the region.

Regions throttle limits:
- POST: 5 requests per hour
- GET: 25 requests per hour

Retrieve regions list

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/v1_1/regions/

URL Parameters: None

Response Example:
[
  {
    "folder": "Africa",
    "region": "Algeria"
  },
  {
    "folder": "Africa",
    "region": "Angola"
  },
  {
    "folder": "Africa",
    "region": "Benin"
  },
  {
    "folder": "Africa",
    "region": "Botswana"
  },
  (REGIONS LIST TRUNCATED)
]
Retrieve region details

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>region_folder</td>
<td>String</td>
<td>No</td>
<td>Default folder is User Folder.</td>
</tr>
<tr>
<td>region</td>
<td>String</td>
<td>Yes</td>
<td>Region Name</td>
</tr>
</tbody>
</table>

Request Data Example:

```json
{
    "region_folder": "Africa",
    "region": "Ghana"
}
```

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/regions/

URL Parameters: None

Response Example:

```json
{
    "status": 200,
    "region_folder": "Africa",
    "region": "Ghana",
    "id": 171
}
```

Import region KML string

The KML (Keyhole Markup Language) is an XML formatted file, and is was developed for use with GoogleEarth. It contains polygon coordinates of the region of interest. KML files must contain only one region. If the KML file contains multiple regions, it must be imported from the COVE UI.
Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>region</td>
<td>String</td>
<td>Yes</td>
<td>Region name</td>
</tr>
<tr>
<td>data</td>
<td>String</td>
<td>Yes</td>
<td>KML file passed as a string.</td>
</tr>
</tbody>
</table>

Request Data Example:
```
{
    "region": "Barekese Dam",
    "data": "<?xml version="1.0" encoding="UTF-8"?><kml xmlns="http://www.opengis.net/kml/2.2"><Placemark><Polygon><outerBoundaryIs><LinearRing><coordinates>-1.722,6.844,0 -1.721,6.844,0 -1.722,6.843,0 -1.721,6.839,0
-1.723,6.837,0 -1.721,6.836,0 -1.723,6.832,0 -1.721,6.832,0 -1.72,6.831,0
-1.718,6.83,0 -1.716,6.827,0 -1.716,6.83,0 -1.715,6.832,0 -1.712,6.83,0
-1.707,6.825,0 -1.707,6.827,0 -1.71,6.831,0 -1.713,6.834,0 -1.713,6.837,0
-1.71,6.838,0 -1.707,6.837,0 -1.707,6.837,0 -1.708,6.839,0 -1.707,6.84,0
-1.704,6.839,0 -1.703,6.841,0 -1.703,6.842,0 -1.706,6.845,0 -1.705,6.846,0
-1.706,6.849,0 -1.7,6.85,0 -1.697,6.852,0 -1.695,6.85,0 -1.695,6.85,0
-1.695,6.852,0 -1.694,6.855,0 -1.693,6.855,0 -1.69,6.855,0 -1.688,6.856,0
-1.686,6.854,0 -1.684,6.854,0 -1.684,6.856,0 -1.687,6.858,0 -1.686,6.861,0
-1.687,6.863,0 -1.692,6.86,0 -1.697,6.858,0 -1.701,6.861,0 -1.707,6.86,0
-1.701,6.861,0 -1.697,6.858,0 -1.701,6.855,0 -1.705,6.855,0 -1.707,6.854,0
-1.71,6.855,0 -1.712,6.854,0 -1.712,6.852,0 -1.713,6.85,0 -1.712,6.847,0
-1.715,6.843,0 -1.716,6.844,0 -1.716,6.842,0 -1.716,6.842,0 -1.718,6.841,0
-1.721,6.841,0
-1.722,6.844,0</coordinates></LinearRing></outerBoundaryIs></Polygon></Placemark></kml>"
}
```

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/regions/

URL Parameters: None

Failed Response Example:
```
{'status': 415, 'message': 'Unable to parse KML string.'}
```
Import region KML file

KML files must contain only one region. If the KML file contains multiple regions, it must be imported from the COVE UI.

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>KML (binary)</td>
<td>Yes</td>
<td>Region name will be the name of the file received. If the file name is unable to be determined, the default name is User Region.</td>
</tr>
</tbody>
</table>

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/regions/upload/

URL Parameters: None

Failed Response Example:
{'status': 415, 'message': '<Error Message>'}

Import region shapefile

Shapefiles must contain only one region. If the shapefile contains multiple regions, it must be imported from the COVE UI.

Input parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>Shapefile</td>
<td>Yes</td>
<td>Region name will be the name of the file received. If the file name is unable to be determined, the default name is User Region.</td>
</tr>
</tbody>
</table>

Request Method: POST

Request URL:
https://ceos-cove.org/en/api/v1_1/regions/upload/
Export region KML file

**Request Method:** GET

**Request URL:**
https://ceos-cove.org/en/api/v1_1/regions/kml/

**URL Parameters:** None

---

Export region shapefile

The shapefile file exported (ZIP) is used in geographic information systems (GIS) and contains polygon coordinates of the region of interest.

**Request Method:** GET

**Request URL:**
https://ceos-cove.org/en/api/v1_1/regions/shapefile/

**URL Parameters:** None

---

Retrieve region PNG

The PNG image file shows a map of the region of interest, and allows users to view existing and imported regions.

**Request Method:** GET

**Request URL:**
https://ceos-cove.org/en/api/v1_1/regions/png/<REGION ID>/
### URL Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transparent</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is true, the map background will be omitted and the background will instead be transparent.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is false, image resolution is 200dpi.</td>
</tr>
<tr>
<td>resolution</td>
<td>Integer</td>
<td>No</td>
<td>Values should be in dots per inch (dpi).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If transparent is false, resolution must be less than or equal to the default resolution.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default resolution is 200dpi.</td>
</tr>
<tr>
<td>line_width</td>
<td>Float</td>
<td>No</td>
<td>Values should be greater than 0.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allows users to define the line width for the region outline.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default value is 2.0.</td>
</tr>
<tr>
<td>line_color</td>
<td>String</td>
<td>No</td>
<td>Values should be a standard six character hex color string. The leading # character is optional.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allows users to define the line color for the region outline.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Default value is #FFFFFF (white).</td>
</tr>
<tr>
<td>map_coordinates</td>
<td>Boolean</td>
<td>No</td>
<td>Values: true or false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allows users to fetch the map coordinates used to generate a PNG image.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If value is true, the map coordinates will be returned <strong>instead of</strong> a PNG image.</td>
</tr>
</tbody>
</table>
Map Coordinates Response Example:
{
  "status": 200,
  "map_coordinates": [
    [-8.661654877310252, -0.16135473389262667],
    [-8.661654877310252, 14.838645266107374],
    [6.3383451226897485, 14.838645266107374],
    [6.3383451226897485, -0.16135473389262667],
    [-8.661654877310252, -0.16135473389262667]
  ]
}

Failed Response Example:
{"status": 404, 'message': 'File not found.'}

Example API requests using curl

Request region list:
curl -H "Accept: application/json; indent=4" -u username:password
https://ceos-cove.org/en/api/regions/

Request region information:
curl -H "Content-Type: application/json" -X POST -u username:password -d
'{"region_folder": "Africa", "region": "Ghana"}' https://ceos-
cove.org/en/api/regions/

Import region KML string:
curl -H "Content-Type: application/json" -X POST -u username:password -d
'{"region": "RegionA", "data": "<KML STRING>"}' https://ceos-
cove.org/en/api/regions/

Import region KML file:
curl -H "Content-Type: multipart/form-data" -X POST -u username:password -F
file=@region_name.kml https://ceos-cove.org/en/api/regions/upload/

Import region shapefile:
curl -H "Content-Type: multipart/form-data" -X POST -u username:password -F
file=@region_name.zip https://ceos-cove.org/en/api/regions/upload/

Export region PNG map image:
-o region.png

Request PNG map image (transparent):
?transparent=true -o filename.png
Request PNG map image (resolution):

Request PNG map image (line_width):
curl -u username:password https://ceos-cove.org/en/api/regions/png/<REGION ID>/?line_width=0.5 -o filename.png

Request PNG map image (line_color):

Request PNG map image (map_coordinates):
curl -u username:password https://ceos-cove.org/en/api/regions/png/<REGION ID>/?map_coordinates=true -o filename.png

Export region KML file:

Export region shapefile:
Version Information

In future versions of the COVE API, the default base URL will always point to the most recent version, and older versions may be depreciated. If COVE API endpoints are integrated into custom applications, please use the versioned URL. Users may refer to this document for version information or by sending a request to the request URL below.

COVE API details:

- Current version: v1_1
- Supported versions: v1_0, v1_1
- Deprecated versions: None
- Revision history:
  v1_1:
  - Added calculate_solar_angles option to acquisition forecaster task.
  - Added endpoints to create constellations (archived and forecasted).
  - Added options to PNG requests:
    - line_width: in Acquisition Forecaster, Data Browser, and Regions
    - line_color: in Regions
    - transparent: in Acquisition Forecaster, Coverage Analyzer, Coincident Calculator, Revisits Calculator, Data Browser, and Regions
    - transparency: in Coverage Analyzer, Coincident Calculator, Revisits Calculator
    - resolution: in Acquisition Forecaster, Coverage Analyzer, Coincident Calculator, Revisits Calculator, Data Browser, and Regions
    - show_region_outline: in Acquisition Forecaster, Coverage Analyzer, Coincident Calculator, Revisits Calculator, and Data Browser
    - map_coordinates: in Acquisition Forecaster, Coverage Analyzer, Coincident Calculator, Revisits Calculator, Data Browser, and Regions
  - Added extended option to CSV requests in Acquisition Forecaster, Revisits Calculator, and Data Browser
- Default base url: https://ceos-cove.org/en/api/
- Versioned base url: https://ceos-cove.org/en/api/v1_1/
Retrieve version details

Request Method: GET

Request URL:
https://ceos-cove.org/en/api/

URL Parameters: None

Response Example:
{
    "current_version": "<CURRENT VERSION>",
    "default_version": "<CURRENT VERSION>",
    "supported_versions": [
        "<CURRENT VERSION>",
        "<OLDER VERSION>"
    ],
    "deprecated_versions": [
        {
            "<DEPRECATED VERSION>": "<DEPRECATION DATE>
        }
    ],
    "revision_history": {
        "<VERSION>": [ 
        ...
    }
}

Example API requests using curl

Version information:
curl -H "Accept: application/json; indent=4" -u username:password
https://ceos-cove.org/en/api/
AdvancedOrbitFilter

A dictionary containing values needed to create a custom mission using parameters required for an advanced orbit type. Used in Custom Missions.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>Yes</td>
<td>Name of custom mission</td>
</tr>
<tr>
<td>existing</td>
<td>boolean</td>
<td>Yes</td>
<td>Value: false</td>
</tr>
<tr>
<td>Create custom mission using a circular orbit.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>orbit_type</td>
<td>string</td>
<td>Yes</td>
<td>Value: advanced</td>
</tr>
<tr>
<td>semimajor_axis</td>
<td>float</td>
<td>Yes</td>
<td>Range: 6778.127 to 7878.137</td>
</tr>
<tr>
<td>eccentricity</td>
<td>float</td>
<td>Yes</td>
<td>Range: 0 to 1</td>
</tr>
<tr>
<td>inclination</td>
<td>float</td>
<td>Yes</td>
<td>Range: 0 to 180</td>
</tr>
<tr>
<td>argument_of_perigee</td>
<td>float</td>
<td>Yes</td>
<td>Range: 0 to 360</td>
</tr>
<tr>
<td>raan</td>
<td>float</td>
<td>Yes</td>
<td>Range: 0 to 360</td>
</tr>
<tr>
<td>true_anomaly</td>
<td>float</td>
<td>Yes</td>
<td>Range: 0 to 360</td>
</tr>
</tbody>
</table>

Filter example for request:

```json
{
    "name": "Newton",
    "existing": false,
    "orbit_type": "advanced",
    "semimajor_axis": 6800.0,
    "eccentricity": 0,
    "inclination": 35,
    "argument_of_perigee": 90,
    "raan": 130,
    "true_anomaly": 250
}
```
**CircularOrbitFilter**

A dictionary containing values needed to create a custom mission using parameters required for a circular orbit type. Used in Custom Missions.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>Yes</td>
<td>Name of custom mission</td>
</tr>
<tr>
<td>existing</td>
<td>boolean</td>
<td>Yes</td>
<td>Value: false</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Create custom mission using a circular orbit.</td>
</tr>
<tr>
<td>orbit_type</td>
<td>string</td>
<td>Yes</td>
<td>Value: circular</td>
</tr>
<tr>
<td>inclination</td>
<td>integer or float</td>
<td>Yes</td>
<td>Range: 0 to 180</td>
</tr>
<tr>
<td>altitude</td>
<td>integer or float</td>
<td>Yes</td>
<td>Range: 400 to 1500</td>
</tr>
<tr>
<td>longitude</td>
<td>integer or float</td>
<td>Yes</td>
<td>Range: 0 to 360</td>
</tr>
</tbody>
</table>

Filter example for request:

```json
{
    "name": "Pythagorus",
    "existing": false,
    "orbit_type": "circular",
    "inclination": 0,
    "altitude": 600,
    "longitude": 180
}
```
**ConstellationsFilter**
A list of constellation name strings. Used in Coincident Calculator and Country Coverage.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>No:</td>
<td>Coincident Calculator: Filter optional.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes:</td>
<td>Country Coverage: At least one constellation required.</td>
</tr>
</tbody>
</table>

**Filter example for request:**

`["Landsat", "Sentinel-1", "Sentinel-2"]`

**CustomInstrumentFilter**
A dictionary of containing new custom instrument parameters. Used in Custom Missions.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>Yes</td>
<td>Name of custom instrument</td>
</tr>
<tr>
<td>existing</td>
<td>boolean</td>
<td>Yes</td>
<td>Value: false</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------</td>
<td>----------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>field_of_view</td>
<td>Integer or float</td>
<td>Yes</td>
<td>Range: 0 to 60</td>
</tr>
<tr>
<td>pointing_angle</td>
<td>Integer or float</td>
<td>Yes</td>
<td>Range: -60 to 60</td>
</tr>
</tbody>
</table>

Filter example for request:
```
{
  "name": "beam",
  "existing": False,
  "field_of_view": 5.0,
  "pointing_angle": -5.0
}
```

**CustomInstrumentsFilter**
A list of ExistingInstrumentFilter and/or CustomInstrumentFilter. Used in Custom Missions.

**CustomMissionFilter**
A dictionary of containing custom mission name, and custom instrument name. Used in Acquisition Forecaster, Revisits Calculator, Coincident Calculator, and Custom Missions.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mission</td>
<td>string</td>
<td>Yes</td>
<td>Valid existing custom missions (mission and instrument) may be found by submitting a COVE API request for the Custom Missions list.</td>
</tr>
<tr>
<td>instrument</td>
<td>string</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

Filter example for request:
```
{
  "mission": "sputnik",
  "instrument": "beam"
}
```
**CustomMissionsFilter**
A list of CustomMissionFilter containing custom mission names, and custom instrument names. Used in Acquisition Forecaster, Revisits Calculator, and Coincident Calculator.

**CountriesFilter**
A list of country names. Used in Country Coverage.

**Input Parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>Yes</td>
<td>Refer to section Retrieve countries list to submit a request for a valid list of countries.</td>
</tr>
</tbody>
</table>

Filter example for request:
["Algeria", "Argentina", "Australia", "Bangladesh", "Belize"]

**DateString**
Date string format YYYY-MM-DD. Example: 2020-01-31

**DiscretizationFilter**
A dictionary values used to divide a region of interest into blocks. Used in Coverage Analyzer, Revisits Calculator, and Coincident Calculator.

**Input Parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
</table>
| type          | string    | No       | Landsat WRS: wrsdata  
S2 Tiling Scheme: s2tiling  
Coverage Analyzer:  
Value: s2tiling or wrsdata.  
Revisits Calculator:  
Value: s2tiling or wrsdata.  
Coincident Calculator:  
Invalid parameter. |
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unit</td>
<td>string</td>
<td>No</td>
<td>Unit parameter should be an empty string or omitted if type parameter is specified. Unit parameter required if size parameter is specified. Value: deg, km.</td>
</tr>
<tr>
<td>size</td>
<td>float</td>
<td>No</td>
<td>Size parameter should be null or omitted if type parameter is specified. Size parameter required if unit parameter is specified. Coverage Analyzer: Value (deg): between 5.0 to 0.1 Value (km): between 12 to 555 Revisits Calculator: Value (deg): between 5.0 to 0.1 Value (km): between 12 to 555 Coincident Calculator: Value (deg): between 1.0 to 0.1 Value (km): between 12 to 555</td>
</tr>
<tr>
<td>include_overlap</td>
<td>boolean</td>
<td>No</td>
<td>Coverage Analyzer: Value should only be true if type is s2tiling and missions is Sentinel-2A or Sentinel-2B. Value: true, false. Revisits Calculator: Invalid parameter. Coincident Calculator: Invalid parameter.</td>
</tr>
</tbody>
</table>

Filter example for request for Coverage Analyzer:
```json
{
    "type": "s2tiling",
    "size": null,
    "unit": "",
    "include_overlap": true
}
```
Filter example for request for Revisits Calculator:
{
    "type": "",
    "size": 5.0,
    "unit": "deg"
}

Filter example for request for Coincident Calculator:
{
    "size": 0.25,
    "unit": "deg"
}

**ExistingInstrumentFilter**

A dictionary containing values needed to create a custom mission from an existing mission. Used in Custom Missions.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Yes</td>
<td>Name of custom mission</td>
</tr>
<tr>
<td>existing</td>
<td>Boolean</td>
<td>Yes</td>
<td>Value: true</td>
</tr>
<tr>
<td>mission</td>
<td>MissionsFilter</td>
<td>Yes</td>
<td>Create custom instrument using existing mission/instrument/mode.</td>
</tr>
</tbody>
</table>

Filter example for request:
{
    "name": "beam",
    "existing": True,
    "mission": {
        "mission": "Sentinel-1A",
        "instrument": "C-SAR",
        "mode": "IWS"
    }
}

**ExistingMissionFilter**
A dictionary containing values needed to create a custom mission from an existing mission. Used in Custom Missions.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>Yes</td>
<td>Name of custom mission</td>
</tr>
<tr>
<td>existing</td>
<td>boolean</td>
<td>Yes</td>
<td>Value should be true. Create custom mission using existing mission.</td>
</tr>
<tr>
<td>mission</td>
<td>string</td>
<td>Yes</td>
<td>Valid mission name may be found by submitting a COVE API request for the Missions forecasted list.</td>
</tr>
</tbody>
</table>

Filter example for request:
```
{
    "name": "Euchlid",
    "existing": True,
    "mission": "ALOS-2"
}
```

**MetadataFilter**
Used in Coverage Analyzer, and Data Browser.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloud_cover</td>
<td>float</td>
<td>No</td>
<td>Cloud cover filter only applies to Landsat 5, Landsat 7, Landsat 8, Sentinel-2A, and Sentinel-2B. Values: between 0 and 100</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>day_night</td>
<td>string</td>
<td>No</td>
<td>Day/Night filter only applies to Landsat 5, Landsat 7, and Landsat 8. Values: day, night</td>
</tr>
<tr>
<td>processing_level</td>
<td>string</td>
<td>No</td>
<td>Processing level filter only applies to Sentinel-1A, and Sentinel-1B. Values: GRD, SLC</td>
</tr>
<tr>
<td>orbit_direction</td>
<td>string</td>
<td>No</td>
<td>Orbit direction filter only applies to Sentinel-1A, Sentinel-1B, Sentinel-2A, and Sentinel-2B. Ascending: asc Descending: desc Values: asc, desc</td>
</tr>
</tbody>
</table>

Filter example for request:
```json
{
    "cloud_cover": null,
    "day_night": "",
    "orbit_direction": "ascending",
    "processing_level": ""
}
```

MissionsFilter

Used in Acquisition Forecaster, Coverage Analyzer, Revisits Calculator, Coincident Calculator, and Data Browser.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mission</td>
<td>string</td>
<td>Yes</td>
<td>Acquisition Forecaster: Valid mission, instrument, and mode may be found by submitting a COVE API request for the Missions forecasted list.</td>
</tr>
<tr>
<td>instrument</td>
<td>string</td>
<td>Yes</td>
<td>Coverage Analyzer: Valid mission, instrument, and mode may be found by submitting a COVE API request for the Missions archived list.</td>
</tr>
<tr>
<td>mode</td>
<td>string</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
Filter example for request:
```
{
    "mission": "Sentinel-2A",
    "instrument": "MSI",
    "mode": ""
}
```

**SunSynchronousOrbitFilter**

A dictionary containing values needed to create a custom mission using parameters required for a sun synchronous orbit type. Used in Custom Missions.

**Input Parameters:**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>string</td>
<td>Yes</td>
<td>Name of custom mission</td>
</tr>
<tr>
<td>existing</td>
<td>boolean</td>
<td>Yes</td>
<td>Value: false</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Create custom mission using a circular orbit.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Data Type</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>orbit_type</td>
<td>string</td>
<td>Yes</td>
<td>Value: sun_synchronous</td>
</tr>
<tr>
<td>revolutions_per_day</td>
<td>integer or float</td>
<td>No</td>
<td>Range: 12.4 to 15.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Must be included if altitude is omitted.</td>
</tr>
<tr>
<td>altitude</td>
<td>integer or float</td>
<td>No</td>
<td>Range: 400 to 1500</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Must be included if revolutions_per_day is omitted.</td>
</tr>
<tr>
<td>revolutions_to_repeat</td>
<td>Integer or float</td>
<td>Yes</td>
<td>Range: 1 to 1500</td>
</tr>
<tr>
<td>local_time</td>
<td>string</td>
<td>Yes</td>
<td>Format: HH:MM</td>
</tr>
<tr>
<td>orbit_direction</td>
<td>integer or float</td>
<td>Yes</td>
<td>Values: ascending or descending</td>
</tr>
</tbody>
</table>

**Filter example for request using revolutions_per_day:**
```json
{
    "name": "Archimedes",
    "existing": False,
    "orbit_type": "sun_synchronous",
    "revolutions_per_day": 14.56,
    "revolutions_to_repeat": 233.0,
    "local_time": "11:10",
    "direction": "ascending"
}
```

**Filter example for request using altitude:**
```json
{
    "name": "Archimedes",
    "existing": False,
    "orbit_type": "sun_synchronous",
    "altitude": 705.0,
    "revolutions_to_repeat": 233.0,
    "local_time": "11:10",
    "direction": "ascending"
}
```
<table>
<thead>
<tr>
<th>Request Method</th>
<th>Request Type</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>User History</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/</a></td>
</tr>
<tr>
<td>POST</td>
<td>Task</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/</a></td>
</tr>
<tr>
<td>GET</td>
<td>Status</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>CSV</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/csv/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/csv/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>CSV</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/csv/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/csv/</a>&lt;TASK ID&gt;/?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>extended=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>orbit_direction=ascending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>orbit_direction=descending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transparent=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>resolution=200</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>line_width=0.5</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>show_region_outline=false</td>
</tr>
<tr>
<td>Request Method</td>
<td>Request Type</td>
<td>Request URL</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_coordinates=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_coordinates=true&amp;orbit_direction=ascending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_coordinates=true&amp;orbit_direction=descending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_legend=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_legend=true&amp;orbit_direction=ascending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_legend=true&amp;orbit_direction=descending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_coordinates=true&amp;map_legend=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_coordinates=true&amp;map_legend=true&amp;orbit_direction=ascending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/">https://ceos-cove.org/en/api/v1_1/acquisition_forecaster/png/</a>&lt;TASK ID&gt;/?map_coordinates=true&amp;map_legend=true&amp;orbit_direction=descending</td>
</tr>
</tbody>
</table>

**Coverage Analyzer**

<table>
<thead>
<tr>
<th>Request Method</th>
<th>Request Type</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>User History</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/</a></td>
</tr>
<tr>
<td>POST</td>
<td>Task</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/</a></td>
</tr>
<tr>
<td>GET</td>
<td>Status</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/</a>&lt;TASK_ID&gt;/</td>
</tr>
<tr>
<td>Request Method</td>
<td>Request Type</td>
<td>Request URL</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GET</td>
<td>CSV</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/csv/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/csv/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/</a>&lt;TASK ID&gt;/?transparent=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/</a>&lt;TASK ID&gt;/?transparency=0.65</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/</a>&lt;TASK ID&gt;/?resolution=200</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/</a>&lt;TASK ID&gt;/?show_region_outline=false</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/</a>&lt;TASK ID&gt;/?map_coordinates=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/png/</a>&lt;TASK ID&gt;/?map_legend=true</td>
</tr>
<tr>
<td>GET</td>
<td>NetCDF</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/netcdf/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/netcdf/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>GeoTiff</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coverage_analyzer/geotiff/">https://ceos-cove.org/en/api/v1_1/coverage_analyzer/geotiff/</a>&lt;TASK ID&gt;/</td>
</tr>
</tbody>
</table>

### Revisits Calculator

<table>
<thead>
<tr>
<th>Request Method</th>
<th>User History</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Method</td>
<td>Request Type</td>
<td>Request URL</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/">https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/</a>&lt;TASK ID&gt;/?orbit_direction=descending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/">https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/</a>&lt;TASK ID&gt;/?transparent=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/">https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/</a>&lt;TASK ID&gt;/?transparency=0.65</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/">https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/</a>&lt;TASK ID&gt;/?resolution=200</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/">https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/</a>&lt;TASK ID&gt;/?show_region_outline=false</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/">https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/</a>&lt;TASK ID&gt;/?map_coordinates=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/">https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/</a>&lt;TASK ID&gt;/?map_coordinates=true&amp;orbit_direction=descending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/">https://ceos-cove.org/en/api/v1_1/revisits_calculator/png/</a>&lt;TASK ID&gt;/?map_legend=true</td>
</tr>
<tr>
<td>Request Method</td>
<td>Request Type</td>
<td>Request URL</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>map_legend=true&amp;orbit_direction=ascending</td>
</tr>
<tr>
<td></td>
<td></td>
<td>map_legend=true&amp;orbit_direction=descending</td>
</tr>
<tr>
<td></td>
<td></td>
<td>map_coordinates=true&amp;map_legend=true</td>
</tr>
<tr>
<td></td>
<td></td>
<td>map_coordinates=true&amp;map_legend=true&amp;orbit_direction=ascending</td>
</tr>
<tr>
<td></td>
<td></td>
<td>map_coordinates=true&amp;map_legend=true&amp;orbit_direction=descending</td>
</tr>
</tbody>
</table>

**Coincident Calculator**

<table>
<thead>
<tr>
<th>Request Method</th>
<th>Request Type</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/</a>&lt;TASK ID&gt;/? transparent=true</td>
</tr>
<tr>
<td>Request Method</td>
<td>Request Type</td>
<td>Request URL</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/</a>&lt;TASK ID&gt;/?transparency=0.65</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/</a>&lt;TASK ID&gt;/?resolution=200</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/</a>&lt;TASK ID&gt;/?show_region_outline=false</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/</a>&lt;TASK ID&gt;/?map_coordinates=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/</a>&lt;TASK ID&gt;/?map_legend=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/forecasted/png/</a>&lt;TASK ID&gt;/?map_coordinates=true&amp;map_legend=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/</a>&lt;TASK ID&gt;/?transparent=true</td>
</tr>
<tr>
<td>Request Method</td>
<td>Request Type</td>
<td>Request URL</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/</a>&lt;TASK ID&gt;/?transparency=0.65</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/</a>&lt;TASK ID&gt;/?resolution=200</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/</a>&lt;TASK ID&gt;/?map_coordinates=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/</a>&lt;TASK ID&gt;/?map_legend=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/">https://ceos-cove.org/en/api/v1_1/coincident_calculator/archived/png/</a>&lt;TASK ID&gt;/?map_coordinates=true&amp;map_legend=true</td>
</tr>
</tbody>
</table>

### Data Browser

<table>
<thead>
<tr>
<th>Request Method</th>
<th>Request Type</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>User History</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/">https://ceos-cove.org/en/api/v1_1/data_browser/</a></td>
</tr>
<tr>
<td>POST</td>
<td>Task</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/">https://ceos-cove.org/en/api/v1_1/data_browser/</a></td>
</tr>
<tr>
<td>GET</td>
<td>Status</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/">https://ceos-cove.org/en/api/v1_1/data_browser/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>Scene Count</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/count/">https://ceos-cove.org/en/api/v1_1/data_browser/count/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>CSV</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/csv/">https://ceos-cove.org/en/api/v1_1/data_browser/csv/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>CSV</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/csv/">https://ceos-cove.org/en/api/v1_1/data_browser/csv/</a>&lt;TASK ID&gt;/?extended=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/png/">https://ceos-cove.org/en/api/v1_1/data_browser/png/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/png/">https://ceos-cove.org/en/api/v1_1/data_browser/png/</a>&lt;TASK ID&gt;/?orbit_direction=ascending</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/png/">https://ceos-cove.org/en/api/v1_1/data_browser/png/</a>&lt;TASK ID&gt;/?orbit_direction=descending</td>
</tr>
<tr>
<td>Request Method</td>
<td>Request Type</td>
<td>Request URL</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/png/">https://ceos-cove.org/en/api/v1_1/data_browser/png/</a>&lt;TASK ID&gt;?transparent=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/png/">https://ceos-cove.org/en/api/v1_1/data_browser/png/</a>&lt;TASK ID&gt;?line_width=0.5</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/png/">https://ceos-cove.org/en/api/v1_1/data_browser/png/</a>&lt;TASK ID&gt;?show_region_outline=false</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/data_browser/png/">https://ceos-cove.org/en/api/v1_1/data_browser/png/</a>&lt;TASK ID&gt;?map_coordinates=true</td>
</tr>
</tbody>
</table>

### Country Coverage

<table>
<thead>
<tr>
<th>Request Method</th>
<th>Request Type</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Countries List</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/countries/">https://ceos-cove.org/en/api/v1_1/country_coverage/countries/</a></td>
</tr>
<tr>
<td>GET</td>
<td>Constellations List</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/constellations/">https://ceos-cove.org/en/api/v1_1/country_coverage/constellations/</a></td>
</tr>
<tr>
<td>GET</td>
<td>Daily History</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/</a></td>
</tr>
<tr>
<td>GET</td>
<td>Monthly Cached History</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/?daily_task=false">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/?daily_task=false</a></td>
</tr>
<tr>
<td>POST</td>
<td>Task</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/</a></td>
</tr>
<tr>
<td>GET</td>
<td>Status</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>JSON</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/json/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/json/</a>&lt;TASK ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>JSON</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/json/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/json/</a>&lt;TASK ID&gt;/?country=&lt;COUNTRY&gt;&amp;constellation=&lt;CONSTELLATION&gt;</td>
</tr>
<tr>
<td>GET</td>
<td>JSON</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/json/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/json/</a>&lt;TASK ID&gt;/?country=&lt;COUNTRY&gt;</td>
</tr>
<tr>
<td>GET</td>
<td>JSON</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/json/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/json/</a>&lt;TASK ID&gt;/?constellation=&lt;CONSTELLATION&gt;</td>
</tr>
<tr>
<td>Request Method</td>
<td>Request Type</td>
<td>Request URL</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GET</td>
<td>PDF</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/pdf/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/pdf/</a>&lt;TASK ID&gt;/?country=&lt;COUNTRY&gt;&amp;constellation=&lt;CONSTELLATION&gt;</td>
</tr>
<tr>
<td>GET</td>
<td>CSV</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/csv/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/csv/</a>&lt;TASK ID&gt;/?country=&lt;COUNTRY&gt;&amp;constellation=&lt;CONSTELLATION&gt;</td>
</tr>
<tr>
<td>GET</td>
<td>ZIP</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/country_coverage/reports/zip/">https://ceos-cove.org/en/api/v1_1/country_coverage/reports/zip/</a>&lt;TASK ID&gt;/?country=&lt;COUNTRY&gt;&amp;constellation=&lt;CONSTELLATION&gt;</td>
</tr>
</tbody>
</table>

### Utilities

**POST** Calculator [https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/ground_track_interval/](https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/ground_track_interval/)

**POST** Calculator [https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/period_velocity/](https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/period_velocity/)

**POST** Calculator [https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/sun_synchronous_orbit/](https://ceos-cove.org/en/api/v1_1/utilities/repeating_orbit/sun_synchronous_orbit/)


**POST** Calculator [https://ceos-cove.org/en/api/v1_1/utilities/swath_calculator/fov_from_swath_width/](https://ceos-cove.org/en/api/v1_1/utilities/swath_calculator/fov_from_swath_width/)


### Missions


**GET** List [https://ceos-cove.org/en/api/v1_1/missions/archived/](https://ceos-cove.org/en/api/v1_1/missions/archived/)
<table>
<thead>
<tr>
<th>Request Method</th>
<th>Request Type</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>List</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/custom_missions/">https://ceos-cove.org/en/api/v1_1/custom_missions/</a></td>
</tr>
<tr>
<td>POST</td>
<td>Create</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/custom_missions/">https://ceos-cove.org/en/api/v1_1/custom_missions/</a></td>
</tr>
</tbody>
</table>

**Custom Missions**

**Regions**

<table>
<thead>
<tr>
<th>Request Method</th>
<th>Request Type</th>
<th>Request URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>List</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/">https://ceos-cove.org/en/api/v1_1/regions/</a></td>
</tr>
<tr>
<td>POST</td>
<td>Region Details</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/">https://ceos-cove.org/en/api/v1_1/regions/</a></td>
</tr>
<tr>
<td>POST</td>
<td>Import KML String</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/">https://ceos-cove.org/en/api/v1_1/regions/</a></td>
</tr>
<tr>
<td>POST</td>
<td>Import KML File</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/upload/">https://ceos-cove.org/en/api/v1_1/regions/upload/</a></td>
</tr>
<tr>
<td>POST</td>
<td>Import Shapefile</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/upload/">https://ceos-cove.org/en/api/v1_1/regions/upload/</a></td>
</tr>
<tr>
<td>GET</td>
<td>Export Shapefile</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/shapefile/">https://ceos-cove.org/en/api/v1_1/regions/shapefile/</a>&lt;REGION ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/png/">https://ceos-cove.org/en/api/v1_1/regions/png/</a>&lt;REGION ID&gt;/</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/png/">https://ceos-cove.org/en/api/v1_1/regions/png/</a>&lt;REGION ID&gt;?transparent=true</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/png/">https://ceos-cove.org/en/api/v1_1/regions/png/</a>&lt;REGION ID&gt;?line_width=0.5</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/png/">https://ceos-cove.org/en/api/v1_1/regions/png/</a>&lt;REGION ID&gt;?line_color=%23FFFFFF</td>
</tr>
<tr>
<td>GET</td>
<td>PNG</td>
<td><a href="https://ceos-cove.org/en/api/v1_1/regions/png/">https://ceos-cove.org/en/api/v1_1/regions/png/</a>&lt;REGION ID&gt;?map_coordinates=true</td>
</tr>
</tbody>
</table>
Contact Us

SEO Sponsor
Brian D. Killough, Ph.D.
CEOS Systems Engineering Office (SEO)
Email: Brian.D.Killough@nasa.gov
Phone: 757-864-7047

Technical Support
CEOS-COVE Admin
AMA, Inc.
Email: ceoscove@gmail.com
Phone: 757-864-3176

Send a message via the Contact Us page with details of any issues and the COVE team will attempt to rectify the situation in a timely manner.

Refer to the COVE UI User’s Guide for more information on the usage of each COVE tools.