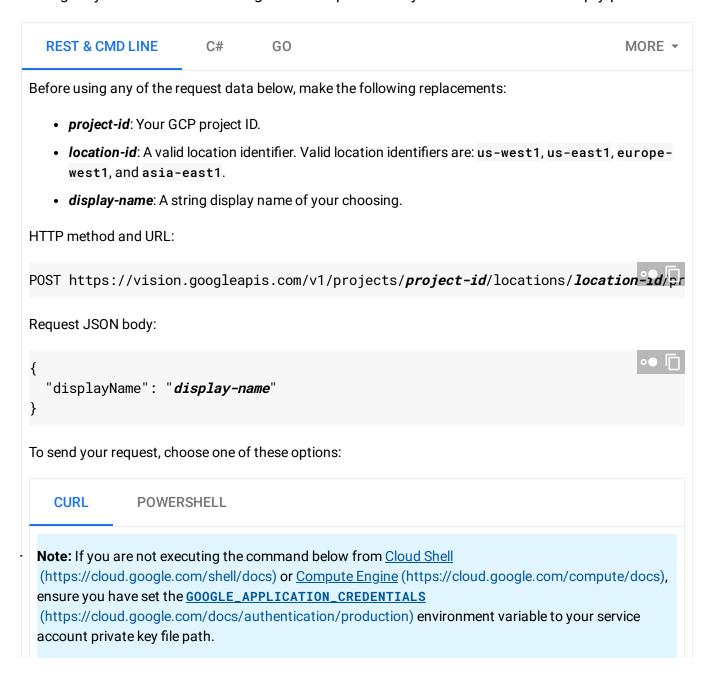Cloud Vision API Product Search

Creating a product set

A product set is a simple container for a group of products.

Creating an empty product set

It's best to use a single product set for all of your items, and create additional product sets for testing as you need. The following code samples show you how to create an empty product set.

REST & CMD LINE	C#	GO	MORE ▾

Before using any of the request data below, make the following replacements:

- **project-id**: Your GCP project ID.
- **location-id**: A valid location identifier. Valid location identifiers are: `us-west1`, `us-east1`, `europe-west1`, and `asia-east1`.
- **display-name**: A string display name of your choosing.

HTTP method and URL:

```
POST https://vision.googleapis.com/v1/projects/project-id/locations/location-id/pr
```

Request JSON body:

```
{
  "displayName": "display-name"
}
```

To send your request, choose one of these options:

CURL	POWERSHELL

Note: If you are not executing the command below from Cloud Shell (https://cloud.google.com/shell/docs) or Compute Engine (https://cloud.google.com/compute/docs), ensure you have set the `GOOGLE_APPLICATION_CREDENTIALS` (https://cloud.google.com/docs/authentication/production) environment variable to your service account private key file path.

Save the request body in a file called `request.json`, and execute the following command:

```
curl -X POST \
-H "Authorization: Bearer "$(gcloud auth application-default print-access-token)
-H "Content-Type: application/json; charset=utf-8" \
-d @request.json \
https://vision.googleapis.com/v1/projects/project-id/locations/location-id/produ
```

If the request is successful, the server returns a `200 OK` HTTP status code and the response in JSON format.

You should see output similar to the following. You can use the product set ID (`b6d809615b6dd675`, in this case) to perform other operations on the product set.

```
{
  "name": "projects/project-id/locations/location-id/productSets/b6d809615b6dd675"
  "displayName": "new-product-set"
}
```

Using bulk import to create a product set with products

You can also use the bulk import function to *simultaneously* create a product set, create multiple products and their reference images.

Various references for the product set, products, and reference images are all set in the bulk import CSV you use. For more information, see Formatting a bulk import CSV (https://cloud.google.com/vision/product-search/docs/csv-format).

Perform bulk import by running the following code:

REST & CMD LINE	C#	GO	MORE ▾

Before using any of the request data below, make the following replacements:

- **project-id**: Your GCP project ID.
- **location-id**: A valid location identifier. Valid location identifiers are: `us-west1`, `us-east1`, `europe-west1`, and `asia-east1`.

- **storage-path**: A Cloud Storage bucket/directory where your input CSV file is stored. The requesting user must have at least read permission to the bucket.

HTTP method and URL:

```
POST https://vision.googleapis.com/v1/projects/project-id/locations/location-id/pr
```

Request JSON body:

```
{
  "inputConfig": {
    "gcsSource": {
      "csvFileUri": "storage-path"
    }
  }
}
```

To send your request, choose one of these options:

CURL   POWERSHELL

> **Note:** If you are not executing the command below from Cloud Shell
> (https://cloud.google.com/shell/docs) or Compute Engine (https://cloud.google.com/compute/docs),
> ensure you have set the `GOOGLE_APPLICATION_CREDENTIALS`
> (https://cloud.google.com/docs/authentication/production) environment variable to your service
> account private key file path.

Save the request body in a file called `request.json`, and execute the following command:

```
curl -X POST \
-H "Authorization: Bearer "$(gcloud auth application-default print-access-token)
-H "Content-Type: application/json; charset=utf-8" \
-d @request.json \
https://vision.googleapis.com/v1/projects/project-id/locations/location-id/produ
```

You should see output similar to the following. You can use the operation ID (`f10f34e32c40a710`, in this case) to get the status of the task. For an example, see Getting an operation's status (#get-operation):

```
{
  "name": "projects/project-id/locations/location-id/operations/f10f34e32c40a710"
}
```

After the long-running operation completes you can get the details of the import operation. The response should look similar to the following:

```json
{
  "name": "locations/location-id/operations/f10f34e32c40a710",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.vision.v1.BatchOperationMetadata",
    "state": "SUCCESSFUL",
    "submitTime": "2019-12-06T21:16:04.476466873Z",
    "endTime": "2019-12-06T21:16:40.594258084Z"
  },
  "done": true,
  "response": {
    "@type": "type.googleapis.com/google.cloud.vision.v1.ImportProductSetsResponse
    "referenceImages": [
      {
        "name": "projects/project-id/locations/location-id/products/product_id0/re
        "uri": "gs://my-storage-bucket/img_039.jpg"
      },
      {
        "name": "projects/project-id/locations/location-id/products/product_id1/re
        "uri": "gs://my-storage-bucket/img_105.jpg"
      },
      {
        "name": "projects/project-id/locations/location-id/products/product_id2/re
        "uri": "gs://my-storage-bucket/img_224.jpg"
      },
      {
        "name": "projects/project-id/locations/location-id/products/product_id3/re
        "uri": "gs://my-storage-bucket/img_385.jpg"
      }
    ],
    "statuses": [
      {},
      {},
      {},
      {}
    ]
  }
}
```

Getting the status of an operation

Several operations you request are long-running, such as product set creation via bulk import, purging a product set, and purging orphaned products. These types of requests will return a JSON with an operation ID that you can use to get the status of the operation.

For example, a batch delete (`purge`) request returns the following JSON:

```
{
"name": "projects/project-id/locations/location-id/operations/bc4e1d412863e626"
}
```

In this case, the operation ID is `bc4e1d412863e626`. The following samples show how to get the status of this operation with this ID.

REST & CMD LINE

Before using any of the request data below, make the following replacements:

- *location-id*: A valid location identifier. Valid location identifiers are: `us-west1`, `us-east1`, `europe-west1`, and `asia-east1`.
- *operation-id*: The ID of your operation. The ID is the last element of the name of your operation. For example:
  - operation name: `projects/project-id/locations/location-id/operations/bc4e1d412863e626`
  - operation id: `bc4e1d412863e626`

HTTP method and URL:

```
GET https://vision.googleapis.com/v1/locations/location-id/operations/operation-id
```

To send your request, choose one of these options:

CURL   **POWERSHELL**

Note: If you are not executing the command below from Cloud Shell (https://cloud.google.com/shell/docs) or Compute Engine (https://cloud.google.com/compute/docs), ensure you have set the **GOOGLE_APPLICATION_CREDENTIALS** (https://cloud.google.com/docs/authentication/production) environment variable to your service account private key file path.
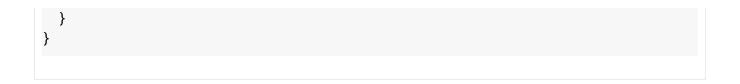
Execute the following command:

```
curl -X GET \
-H "Authorization: Bearer "$(gcloud auth application-default print-access-token)
https://vision.googleapis.com/v1/locations/location-id/operations/operation-id
```

You should see output similar to the following for a completed **product set purge operation**:

```
{
  "name": "locations/location-id/operations/operation-id",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.vision.v1.BatchOperationMetadata",
    "state": "SUCCESSFUL",
    "submitTime": "2019-09-04T15:58:39.131591882Z",
    "endTime": "2019-09-04T15:58:43.099020580Z"
  },
  "done": true,
  "response": {
    "@type": "type.googleapis.com/google.cloud.vision.v1.PurgeProductsRequest",
    "parent": "projects/project-id/locations/location-id",
    "productSetPurgeConfig": {
      "productSetId": "project-set-id"
    },
    "force": true
  }
}
```

You should see output similar to the following for a completed **purge orphaned products operation**:

```
{
  "name": "locations/location-id/operations/operation-id",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.vision.v1.BatchOperationMetadata",
    "state": "SUCCESSFUL",
    "submitTime": "2019-09-04T16:08:38.278197397Z",
    "endTime": "2019-09-04T16:08:45.075778639Z"
  },
  "done": true,
  "response": {
    "@type": "type.googleapis.com/google.cloud.vision.v1.PurgeProductsRequest",
    "parent": "projects/project-id/locations/location-id",
    "deleteOrphanProducts": true,
    "force": true
```

```
    }
}
```