Cloud Vision API Product Search

Method: projects.locations.productSets.import

Asynchronous API that imports a list of reference images to specified product sets based on a list of image information.

The `google.longrunning.Operation` (https://cloud.google.com/vision/product-search/docs/reference/rest/v1/locations.operations#Operation) API can be used to keep track of the progress and results of the request. `Operation.metadata` contains `BatchOperationMetadata`. (progress) `Operation.response` contains `ImportProductSetsResponse`. (results)

The input source of this method is a csv file on Google Cloud Storage. For the format of the csv file please see `ImportProductSetsGcsSource.csv_file_uri` (https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.productSets/import#ImportProductSetsGcsSource.FIELDS.csv_file_uri)

.

HTTP request

`POST`

`https://vision.googleapis.com/v1/{parent=projects/*/locations/*}/productSets:import`

The URL uses gRPC Transcoding (https://github.com/googleapis/googleapis/blob/master/google/api/http.proto) syntax.

Path parameters

Parameters	
`parent`	`string` Required. The project in which the ProductSets should be imported. Format is `projects/PROJECT_ID/locations/LOC_ID`.

Request body

The request body contains data with the following structure:

JSON representation

{
  "inputConfig": {
  object (ImportProductSetsInputConfig (https://cloud.google.com/vision/product-search/docs/r
  }
}

Fields	
inputConfig	object (ImportProductSetsInputConfig (https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.productSets/import#ImportProductSetsInputConfig)) Required. The input content for the list of requests.

Response body

If successful, the response body contains an instance of Operation (https://cloud.google.com/vision/product-search/docs/reference/rest/v1/locations.operations#Operation).

Authorization Scopes

Requires one of the following OAuth scopes:

- https://www.googleapis.com/auth/cloud-platform
- https://www.googleapis.com/auth/cloud-vision

For more information, see the Authentication Overview (https://cloud.google.com/docs/authentication/).

ImportProductSetsInputConfig

The input content for the `productSets.import` method.

JSON representation
{ "gcsSource": { object (ImportProductSetsGcsSource (https://cloud.google.com/vision/product-search/docs/ref } }

Fields	
`gcsSource`	`object (ImportProductSetsGcsSource` (https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.productSets/import#ImportProductSetsGcsSource)) The Google Cloud Storage location for a csv file which preserves a list of ImportProductSetRequests in each line.

ImportProductSetsGcsSource

The Google Cloud Storage location for a csv file which preserves a list of ImportProductSetRequests in each line.

JSON representation
{ "csvFileUri": string }

Fields	
`csvFileUri`	`string` The Google Cloud Storage URI of the input csv file.

Fields

The URI must start with **`gs://`**.

The format of the input csv file should be one image per line. In each line, there are 8 columns.

1. image-uri
2. image-id
3. product-set-id
4. product-id
5. product-category
6. product-display-name
7. labels
8. bounding-poly

The **`image-uri`**, **`product-set-id`**, **`product-id`**, and **`product-category`** columns are required. All other columns are optional.

If the **`ProductSet`** or **`Product`** specified by the **`product-set-id`** and **`product-id`** values does not exist, then the system will create a new **`ProductSet`** or **`Product`** for the image. In this case, the **`product-display-name`** column refers to **`displayName`** (https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.products#Product.FIELDS.display_name), the **`product-category`** column refers to **`productCategory`** (https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.products#Product.FIELDS.product_category), and the **`labels`** column refers to **`productLabels`** (https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.products#Product.FIELDS.product_labels).

The **`image-id`** column is optional but must be unique if provided. If it is empty, the system will automatically assign a unique id to the image.

The **`product-display-name`** column is optional. If it is empty, the system sets the **`displayName`**

Fields

(https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.products#Product.FIELDS.display_name)
field for the product to a space (" "). You can update the **displayName** later by using the API.

If a **Product** with the specified **product-id** already exists, then the system ignores the **product-display-name**, **product-category**, and **labels** columns.

The **labels** column (optional) is a line containing a list of comma-separated key-value pairs, in the following format:

```
"key_1=value_1,key_2=value_2,...,key_n=value_n"
```

The **bounding-poly** column (optional) identifies one region of interest from the image in the same manner as **referenceImages.create**. If you do not specify the **bounding-poly** column, then the system will try to detect regions of interest automatically.

At most one **bounding-poly** column is allowed per line. If the image contains multiple regions of interest, add a line to the CSV file that includes the same product information, and the **bounding-poly** values for each region of interest.

The **bounding-poly** column must contain an even number of comma-separated numbers, in the format "p1_x,p1_y,p2_x,p2_y,...,pn_x,pn_y". Use non-negative integers for absolute bounding polygons, and float values in [0, 1] for normalized bounding polygons.

The system will resize the image if the image resolution is too large to process (larger than 20MP).