

## Cloud Vision API Product Search

# Product Search Tutorial

This tutorial demonstrates how to create a product set which contains a group of products with reference images for those products. The tutorial shows users how to create a product set via online (individual) import. After the product set has been indexed, you can query the product set using Vision Product Search.

### **Note:**

The Product Search index of products is updated approximately every 30 minutes. When images are added or deleted, the change won't be reflected in your Product Search responses until the index is next updated.

To make sure that indexing has completed successfully, check the [indexTime](https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.productSets#ProductSet) (<https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.productSets#ProductSet>) field of a product set.

In this tutorial, you will learn how to:

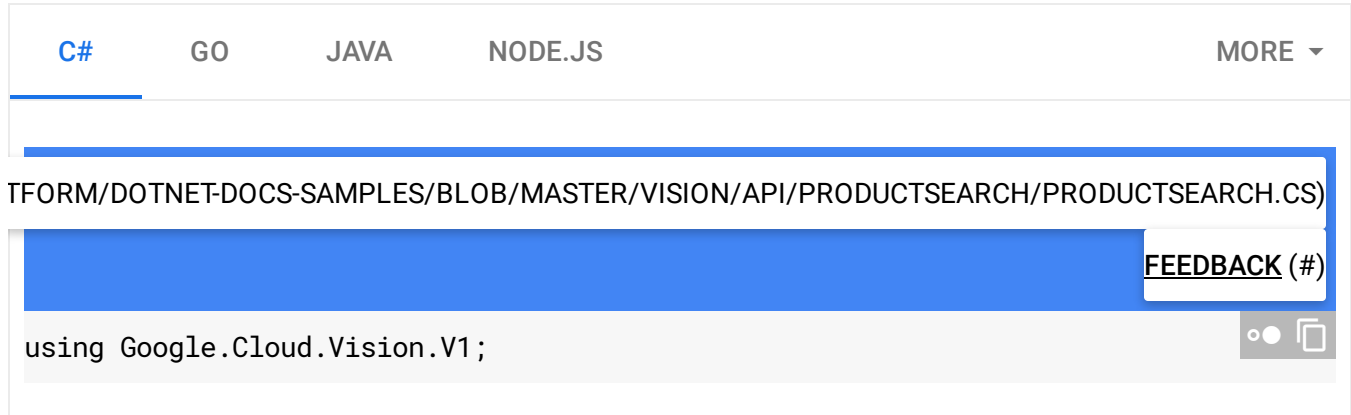
1. Create a product set through online (individual) import
2. Create an individual product
3. Add a product to a product set
4. Update a product
5. Create a reference image
6. Search for similar products

## Before you begin

Before you begin this tutorial make sure you have installed the appropriate [client libraries](https://cloud.google.com/vision/product-search/docs/libraries) (<https://cloud.google.com/vision/product-search/docs/libraries>), [enabled billing and the API](https://cloud.google.com/vision/product-search/docs/before-you-begin) (<https://cloud.google.com/vision/product-search/docs/before-you-begin>) for your project, and have properly [set up authentication](https://cloud.google.com/vision/product-search/docs/auth) (<https://cloud.google.com/vision/product-search/docs/auth>).

## Import Libraries

To use the Cloud Vision Product Search, import the following modules after downloading and installing the [client library](https://cloud.google.com/vision/product-search/docs/libraries) (<https://cloud.google.com/vision/product-search/docs/libraries>):



The screenshot shows a code editor interface with a tab labeled 'C#' selected. The code path is 'TFORM/DOTNET-DOCS-SAMPLES/BLOB/MASTER/VISION/API/PRODUCTSEARCH/PRODUCTSEARCH.CS'. The code snippet is: `using Google.Cloud.Vision.V1;`. There is a 'FEEDBACK (#)' button and a copy icon on the right side of the code editor.

## Running the Application

**Viewing Code Samples:** Most of the code samples in this tutorial are taken from larger code files located in [GitHub](https://github.com/) (<https://github.com/>). You can view and download the complete file from which a code sample is taken by clicking the "View on GitHub" button provided above a sample.

## Step 1: Create a Product Catalog

Users have two options for creating a product catalog, either via batch import using a CSV file, which allows an entire product catalog to be imported in a single API call, or via online import, which offers you control over your product sets and allows for management of one resource or relationship at a time. This primarily means individual creation of product sets, products, and reference images. Online import also allows you to incrementally update a product catalog you have already created via batch import.

In this tutorial you will use online import. See the [Quickstart](https://cloud.google.com/vision/product-search/docs/quickstart) (<https://cloud.google.com/vision/product-search/docs/quickstart>) for an example of batch import with a CSV.

## Online (Individual) Import

### 1. Create a Product Set

Create an empty Product Set which is a simple container for a group of products.

#### Request

Create an empty Product Set and name it "PS\_CLOTH-SHOE\_070318" by executing the following request using the method `create_product_set()`. Pass the product set id and display name as arguments.

**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`, `eu-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) (<https://cloud.google.com/shell/docs>) or [Compute Engine](https://cloud.google.com/compute/docs) (<https://cloud.google.com/compute/docs>), ensure you have set the [GOOGLE\\_APPLICATION\\_CREDENTIALS](https://cloud.google.com/docs/authentication/production) (<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"  
}
```

## Response

```
Product set name: projects/prj-prod-search-tutorials/locations/us-east1/productSets/  
Product set id: PS_CLOTH-SHOE_070318  
Product set display name: CLOTH-SHOE
```

## 2. Create a Product

After a product set has been created, the next step is to create a product. Create a product by executing the following request.

### 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`, `eu-west1`, and `asia-east1`.

- **display-name**: A string display name of your choosing.
- **product-description**: A string description of your choosing.
- **product-category**: A valid product category. The following product categories are currently available: `homegoods-v2`, `apparel-v2`, `toys-v2`, `packagedgoods-v1`, and `general-v1`.
- **productLabels**: One or more key-value pairs associated with a product. Each **key-string** must have an associated **value-string**.

HTTP method and URL:

POST [https://vision.googleapis.com/v1/projects/\*project-id\*/locations/\*location-id\*/pr](https://vision.googleapis.com/v1/projects/<i>project-id</i>/locations/<i>location-id</i>/pr)

Request JSON body:

```
{
  "displayName": "display-name",
  "description": "product-description",
  "productCategory": "product-category",
  "productLabels": [
    {
      "key": "key-string",
      "value": "value-string"
    }
  ]
}
```

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) (<https://cloud.google.com/shell/docs>) or [Compute Engine](https://cloud.google.com/compute/docs) (<https://cloud.google.com/compute/docs>), ensure you have set the [GOOGLE\\_APPLICATION\\_CREDENTIALS](https://cloud.google.com/docs/authentication/production) (<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
```

### Example request body:

```
{  
  "displayName": "sample-product-1234",  
  "description": "Athletic shorts",  
  "productCategory": "apparel-v2",  
  "productLabels": [  
    {  
      "key": "style",  
      "value": "womens"  
    },  
    {  
      "key": "color",  
      "value": "blue"  
    }  
  ]  
}
```

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 ID (**37b9811d308c4e42**, in this case) to perform other operations on the product.

```
{  
  "name": "projects/project-id/locations/location-id/products/37b9811d308c4e42",  
  "displayName": "sample-product-456",  
  "description": "Athletic shorts",  
  "productCategory": "apparel-v2",  
  "productLabels": [  
    {  
      "key": "style",  
      "value": "womens"  
    },  
    {  
      "key": "color",  
      "value": "blue"  
    }  
  ]  
}
```

## Creating a product using a specified `productId`

When creating a product resource you have the option of providing a specific product ID with your request.

This information is passed as a query to the request URL (`?productId=product-id`).

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`, `eu-west1`, and `asia-east1`.
- ***product-id***: The ID for the product that is associated with a reference image. This ID is either randomly set or specified by the user at product creation time.
- ***display-name***: A string display name of your choosing.
- ***product-description***: A string description of your choosing.
- ***product-category***: A valid product category. The following product categories are currently available: `homegoods-v2`, `apparel-v2`, `toys-v2`, `packagedgoods-v1`, and `general-v1`.
- **`productLabels`**: One or more key-value pairs associated with a product. Each ***key-string*** must have an associated ***value-string***.

HTTP method and URL:

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

Request JSON body:

```
{
  "displayName": "display-name",
  "description": "product-description",
  "productCategory": "product-category",
  "productLabels": [
    {
      "key": "key-string",
      "value": "value-string"
    }
  ]
}
```

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) (<https://cloud.google.com/shell/docs>) or [Compute Engine](https://cloud.google.com/compute/docs) (<https://cloud.google.com/compute/docs>), ensure you have set the [GOOGLE\\_APPLICATION\\_CREDENTIALS](https://cloud.google.com/docs/authentication/production) (<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
```

#### Example request URL and body:

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

```
{  
  "displayName": "sample-product-456",  
  "description": "Athletic shorts",  
  "productCategory": "apparel-v2",  
  "productLabels": [  
    {  
      "key": "style",  
      "value": "womens"  
    },  
    {  
      "key": "color",  
      "value": "blue"  
    }  
  ]  
}
```

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 ID you set (`product-id-456`, in this case) to perform other operations on the product.



```
{
  "name": "projects/project-id/locations/location-id/products/product-id-456",
  "displayName": "sample-product-456",
  "description": "Athletic shorts",
  "productCategory": "apparel-v2",
  "productLabels": [
    {
      "key": "style",
      "value": "womens"
    },
    {
      "key": "color",
      "value": "blue"
    }
  ]
}
```

Sending a request with a product ID that already exists produces the following error:

```
{
  "error": {
    "code": 409,
    "message": "Product with ID 'product-id' already exists.",
    "status": "ALREADY_EXISTS",
    "details": [
      {
        "@type": "type.googleapis.com/google.rpc.DebugInfo",
        "detail": "[ORIGINAL ERROR] generic::already_exists: Product with ID 'prod"
      }
    ]
  }
}
```

## Response

```
Product name: projects/prj-prod-search-tutorials/locations/us-east1/products/P_CLOTH
Product id: P_CLOTH-SHOE_46903668_070318
Product display name: Blue Dress
Product category: apparel
Product description: Short sleeved and 1950s style satin dress
Product labels:
  Product label 1:
```

```
    key: style
    value: women
Product label 2:
    key: category
    value: dress
Product label 3:
    key: color
    value: dark-blue
```

### 3. Add a Product to a Product Set

After a product set and a product have been created, you can add the product to the 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`, `eu-west1`, and `asia-east1`.
- **product-set-id**: The ID for the product set you want to run the operation on.
- **product-name**: The full resource name of the product. Format:
  - `projects/project-id/locations/location-id/products/product-id`

HTTP method and URL:

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

Request JSON body:

```
{
  "product": "product-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) (<https://cloud.google.com/shell/docs>) or [Compute Engine](https://cloud.google.com/compute/docs) (<https://cloud.google.com/compute/docs>), ensure you have set the [GOOGLE\\_APPLICATION\\_CREDENTIALS](https://cloud.google.com/docs/authentication/production) (<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 receive a JSON response similar to the following:

```
{}
```

## Response

```
Product added to product set.
```

## 4. Update a product

If you need to update a product or product set after it has been created, you can use our update methods. This example shows a product update where the labels are changed:

### COMMAND-LINE

[C#](#)[GO](#)[MORE](#) ▾

To update a product, send a PATCH request to the following URI.

- Replace **project-id** with the id of your Google Cloud Platform (GCP) project.
- Replace **location-id** with a valid location identifier. Valid location identifiers are: `us-west1`, `us-east1`, `eu-west1`, and `asia-east1`
- Replace **product-id** with the id for the product to update.

```
curl -X PATCH \  
-H "Authorization: Bearer $(gcloud auth application-default print-access-token)" \  
-H "Content-Type: application/json" \  
https://vision.googleapis.com/v1/projects/project-id/locations/location-id/products/  
  'updateMask': { 'paths': ['display_name'] },  
  'product': { 'display_name': 'new-display-name' }  
}"
```

## Response

```
Product name: projects/prj-prod-search-tutorials/locations/us-east1/products/P_CLOTH  
Product id: P_CLOTH-SHOE_46903668_070318  
Product display name: Blue Dress  
Updated product labels:  
  Product label 1:  
    key: style  
    value: women  
  Product label 2:  
    key: category  
    value: dress  
  Product label 3:  
    key: color  
    value: blue  
Product description: Short sleeved and 1950s style satin dress
```

## 5. Create a Product's Reference Image

Creating a reference image for an individual product allows Vision Product Search to search for the product by this image after it is indexed. You can have multiple reference images in a product, particularly if you desire a better match quality.

You can add a new reference image to a product at any time.

When creating a reference image you have the option to include bounding poly coordinates. A bounding poly identifies an area of interest in the reference image. For example, if you create a reference image for a product that is a jacket, you can provide the coordinates for the jacket in the bounding poly argument, and the system will only consider the jacket when looking for product matches. Note: You can provide multiple bounding polys at index time, though at query time the API supports only a single bounding poly.

A convenient way to get the bounding poly coordinates for an image is to use Vision API object localization. For more information on object localization, see [Detecting Multiple Objects](https://cloud.google.com/vision/docs/object-localizer) (<https://cloud.google.com/vision/docs/object-localizer>).

[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`, `eu-west1`, and `asia-east1`.
- **product-id**: The ID for the product that is associated with a reference image. This ID is either randomly set or specified by the user at product creation time.
- **cloud-storage-image-uri**: the path to a valid image file in a Cloud Storage bucket. You must at least have read privileges to the file. Example:

- `gs://storage-bucket/filename.jpg`

★ **Note:** The **boundingPolys** field is optional. If you don't specify a bounding box, Cloud Vision API will try to infer regions of interest in the image that are compatible with the **productCategory** on the parent product.

HTTP method and URL:

POST [https://vision.googleapis.com/v1/projects/\*\*project-id\*\*/locations/\*\*location-id\*\*/pr](https://vision.googleapis.com/v1/projects/<b>project-id</b>/locations/<b>location-id</b>/pr)

Request JSON body:

```
{
  "uri": "cloud-storage-image-uri",
  "boundingPolys": [
    {
      "vertices": [
        {
          "x": X_MIN,
          "y": Y_MIN
        },
        {
          "x": X_MAX,
          "y": Y_MIN
        }
      ]
    }
  ]
}
```

```

    {
      "x": X_MAX,
      "y": Y_MAX
    },
    {
      "x": X_MIN,
      "y": Y_MAX
    }
  ]
}
]
}

```

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) (<https://cloud.google.com/shell/docs>) or [Compute Engine](https://cloud.google.com/compute/docs) (<https://cloud.google.com/compute/docs>), ensure you have set the [GOOGLE\\_APPLICATION\\_CREDENTIALS](https://cloud.google.com/docs/authentication/production) (<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. The example request specified a single `boundingPoly` in the image. The vertices for the bounding box are *not* normalized; the vertex values are the actual pixel values, and *not* relative to the original image and scaled from 0 to 1. These vertices have the following values: [(33,22),(282,22),(282,278),(33,278)].

```

{
  "name": "projects/project-id/locations/location-id/products/product-id/reference

```

```
"uri": "gs://storage-bucket/filename.jpg",
"boundingPolys": [
  {
    "vertices": [
      {
        "x": 33,
        "y": 22
      },
      {
        "x": 282,
        "y": 22
      },
      {
        "x": 282,
        "y": 278
      },
      {
        "x": 33,
        "y": 278
      }
    ]
  }
]
```

## Response

```
Reference image name: projects/prj-prod-search-tutorials/locations/us-east1/products
Reference image id: I_469a896b70ba11e8be97d20059124800_070418
Reference image uri: gs://product-search-tutorial/dress-shoe-dataset/469a896b70ba11e
Reference image bounding polygons:
vertices {
  x: 80
  y: 50
}
vertices {
  x: 80
  y: 660
}
vertices {
  x: 300
  y: 50
```

```
}  
vertices {  
  x: 430  
  y: 660  
}
```

## Step 2: Search For Matching Products

This interface allows you to query the product catalog you created by taking a new image as input and searching for the best matching product.

Similar to creating a reference image, when searching for matching images you have the option to include bounding poly coordinates. A bounding poly identifies the area of interest in the source image that you want to find matches for. For example, if your source image contains both a dress and a purse, and you only want to find matches for the dress, you can identify the bounding poly coordinates for the region of the picture that contains only the dress. By default, if no bounding poly is specified the API determines the largest bounding poly and queries it automatically.

A convenient way to get the bounding poly coordinates for an image is to use Vision API object localization. For more information on object localization, see [Detecting Multiple Objects](https://cloud.google.com/vision/docs/object-localizer) (<https://cloud.google.com/vision/docs/object-localizer>). For example, you can explicitly query a full image by specifying a bounding poly of the entire image box: [(0, 0), (0, 1), (1, 1), (1, 0)].

The request returns an API response that includes the best matching product for an image with the score and the matching image. This image is returned using the highest confidence value.

**Note:** You can also search for best matching products using an image URI as an argument instead of a local filepath. See [Searching for Products](https://cloud.google.com/vision/product-search/docs/searching) (<https://cloud.google.com/vision/product-search/docs/searching>) for more information.

REST & CMD LINE

C#

GO

MORE ▾

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

- **base64-encoded-image:** The base64 representation (ASCII string) of your binary image data. This string should look similar to the following string:
  - /9j/4QAYRXhpZgAA...9tAVx/zDQD1Gxn//2Q==



Visit the [base64 encode](https://cloud.google.com/vision/product-search/docs/base64) (<https://cloud.google.com/vision/product-search/docs/base64>) topic for more information.

- **project-id**: Your GCP project ID.
- **location-id**: A valid location identifier. Valid location identifiers are: `us-west1`, `us-east1`, `eu-west1`, and `asia-east1`.
- **product-set-id**: The ID for the product set you want to run the operation on.

#### Field-specific considerations:

- `features.maxResults` - The maximum number of results to be returned.
- `imageContext.productCategories` - The product category to search in. Currently you can only specify one product category (homegoods, apparel, toys, packaged goods, and general).
- `imageContext.filter` - A key-value filtering expression (or multiple expressions) for product label. Format: "key=value". Filtering key-value pairs can be linked with AND or OR expressions: "color=blue AND style=mens", or "color=blue OR color=black". If using the OR expression **all keys in the expression must be the same**.

HTTP method and URL:

POST <https://vision.googleapis.com/v1/images:annotate>



Request JSON body:

```
{
  "requests": [
    {
      "image": {
        "content": base64-encoded-image
      },
      "features": [
        {
          "type": "PRODUCT_SEARCH",
          "maxResults": 5
        }
      ],
      "imageContext": {
        "productSearchParams": {
          "productSet": "projects/project-id/locations/location-id/productSets/pro
          "productCategories": [
            "apparel"
          ],
          "filter": "style = womens"
        }
      }
    }
  ]
}
```



```
}  
  }  
] }  
}
```

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) (<https://cloud.google.com/shell/docs>) or [Compute Engine](https://cloud.google.com/compute/docs) (<https://cloud.google.com/compute/docs>), ensure you have set the [GOOGLE\\_APPLICATION\\_CREDENTIALS](https://cloud.google.com/docs/authentication/production) (<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/images:annotate
```

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

The response JSON includes the two following result types:

- **productSearchResults** - Contains a list of matching products for the entire image. In the sample response the matching products are: `product_id65`, `product_id35`, `product_id34`, `product_id62`, `product_id32`.
- **productGroupedResults** - Contains bounding box coordinates and matching items for *each* product identified in the image. In the following response there is only one product identified, followed by matching products in the sample product set: `product_id65`, `product_id35`, `product_id34`, `product_id93`, `product_id62`.

Note that while there is overlap in the two result types, there may also be differences (for example, `product_id32` and `product_id93` in the response).

**Note: Zero coordinate values omitted.** When the API detects a coordinate ("x" or "y") value of 0, *that coordinate is omitted in the JSON response*. Thus, a response with a bounding poly around the entire

image would be

`[{"x": 1}, {"x": 1, "y": 1}, {"y": 1}]`. For more information, see the [API Reference documentation](https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.products.referenceImages#boundingpoly) (<https://cloud.google.com/vision/product-search/docs/reference/rest/v1/projects.locations.products.referenceImages#boundingpoly>)

## Response

```
{
  "responses": [
    {
      "productSearchResults": {
        "indexTime": "2019-09-04T21:03:35.662099907Z",
        "results": [
          {
            "product": {
              "name": "projects/project-id/locations/location-id/products/product_",
              "displayName": " ",
              "productCategory": "apparel",
              "productLabels": [
                {
                  "key": "style",
                  "value": "women"
                },
                {
                  "key": "category",
                  "value": "dress"
                }
              ]
            }
          },
          {
            "score": 0.38946953,
            "image": "projects/project-id/locations/location-id/products/product_i"
          },
          {
            "product": {
              "name": "projects/project-id/locations/location-id/products/product_",
              "displayName": " ",
              "productCategory": "apparel",
              "productLabels": [
                {
                  "key": "style",
                  "value": "women"
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

```
    {
      "key": "category",
      "value": "dress"
    }
  ]
},
"score": 0.3847863,
"image": "projects/project-id/locations/location-id/products/product_i
},
{
  "product": {
    "name": "projects/project-id/locations/location-id/products/product_
    "displayName": " ",
    "productCategory": "apparel",
    "productLabels": [
      {
        "key": "style",
        "value": "women"
      },
      {
        "key": "category",
        "value": "dress"
      }
    ]
  },
  "score": 0.33896044,
  "image": "projects/project-id/locations/location-id/products/product_i
},
{
  "product": {
    "name": "projects/project-id/locations/location-id/products/product_
    "displayName": " ",
    "productCategory": "apparel",
    "productLabels": [
      {
        "key": "style",
        "value": "women"
      },
      {
        "key": "category",
        "value": "dress"
      }
    ]
  },
  "score": 0.32509044,
```

```

    "image": "projects/project-id/locations/location-id/products/product_i
  },
  {
    "product": {
      "name": "projects/project-id/locations/location-id/products/product_
      "displayName": " ",
      "productCategory": "apparel",
      "productLabels": [
        {
          "key": "style",
          "value": "women"
        },
        {
          "key": "category",
          "value": "dress"
        }
      ]
    },
    "score": 0.3237155,
    "image": "projects/project-id/locations/location-id/products/product_i
  }
],
"productGroupedResults": [
  {
    "boundingPoly": {
      "normalizedVertices": [
        {
          "x": 0.00458825,
          "y": 0.11000001
        },
        {
          "x": 0.988353,
          "y": 0.11000001
        },
        {
          "x": 0.988353,
          "y": 0.9290588
        },
        {
          "x": 0.00458825,
          "y": 0.9290588
        }
      ]
    },
    "results": [

```

```
{
  "product": {
    "name": "projects/project-id/locations/location-id/products/prod",
    "displayName": " ",
    "productCategory": "apparel",
    "productLabels": [
      {
        "key": "style",
        "value": "women"
      },
      {
        "key": "category",
        "value": "dress"
      }
    ]
  },
  "score": 0.41785678,
  "image": "projects/project-id/locations/location-id/products/produ
},
{
  "product": {
    "name": "projects/project-id/locations/location-id/products/prod",
    "displayName": " ",
    "productCategory": "apparel",
    "productLabels": [
      {
        "key": "style",
        "value": "women"
      },
      {
        "key": "category",
        "value": "dress"
      }
    ]
  },
  "score": 0.3803885,
  "image": "projects/project-id/locations/location-id/products/produ
},
{
  "product": {
    "name": "projects/project-id/locations/location-id/products/prod",
    "displayName": " ",
    "productCategory": "apparel",
    "productLabels": [
      {
```

```
        "key": "style",
        "value": "women"
      },
      {
        "key": "category",
        "value": "dress"
      }
    ]
  },
  "score": 0.36055994,
  "image": "projects/project-id/locations/location-id/products/produ
},
{
  "product": {
    "name": "projects/project-id/locations/location-id/products/prod
    "displayName": " ",
    "productCategory": "apparel",
    "productLabels": [
      {
        "key": "style",
        "value": "women"
      },
      {
        "key": "category",
        "value": "shoe"
      },
      {
        "key": "kids",
        "value": "true"
      }
    ]
  },
  "score": 0.33286288,
  "image": "projects/project-id/locations/location-id/products/produ
},
{
  "product": {
    "name": "projects/project-id/locations/location-id/products/prod
    "displayName": " ",
    "productCategory": "apparel",
    "productLabels": [
      {
        "key": "style",
        "value": "women"
      },
    ],
```

```
        {
          "key": "category",
          "value": "dress"
        }
      ]
    },
    "score": 0.32263064,
    "image": "projects/project-id/locations/location-id/products/produ
  }
],
"objectAnnotations": [
  {
    "mid": "/m/01d40f",
    "name": "Dress",
    "score": 0.95488
  }
]
}
]
}
]
}
]
}
```

## Apparel Response Example

Search Image:



D:/product/final/images-20180618T073733Z-01/images/469355b570ba11e88ff2d2005912480





#### Similar product information:

Product id: 46930b6b

Product display name: Evening gown

Product description: Blue evening gown in 1940s style

Product category: apparel

style: women

category: dress

color: blue



### Searching with a Label

The following search example includes a filter based on color.

**Note:** The filtering system only works on user-defined labels, so the label of interest (color, in this case) has to be provided when creating the products.

## Request

Make a search request by executing the following request with method `get_similar_products_file()` or `get_similar_products_uri()`. Product set id, local image file path and filter are passed as arguments. This input image is also present in "resources/input/".

### PYTHON

```
python product_search.py get_similar_products_file "12000002" "D:/product/final/ima
```

## Response

Search Image:

D:/product/final/images-20180618T073733Z-001/images/469355b570ba11e88ff2d200591248



Similar product information:

Product id: p569d4e7a1

Product display name: Wedding Dress

Product description: Elegant Wedding Dress for women

Product category: apparel

style: women

category: dress

color: white

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated January 23, 2020.*