Using the Watson Visual Recognition Service

This document is current as of the initial date of publication and may be changed by IBM at any time.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

Overview

In this lab, you will familiarize yourself with the Watson Visual Recognition service by learning how to classify, detect faces in, and recognize text in an image, and how to create, train and then use a custom classifier.

Prerequisites

You need the following accounts and software:

- An IBM Bluemix account
- An Internet Explorer, Firefox, or Chrome web browser
- The Postman tool . A free REST client available on Windows. Mac and Linux . Templates for all the REST calls used in the lab are made available as a Postman collection which you will import into your own Postman installation

Section 1. Initial set up

This lab has a few more pre-requisites which will take a portion of the lab time to install. However, if you are going to work further with containers on Bluemix, they are tools that are essential so it's worth the investment.

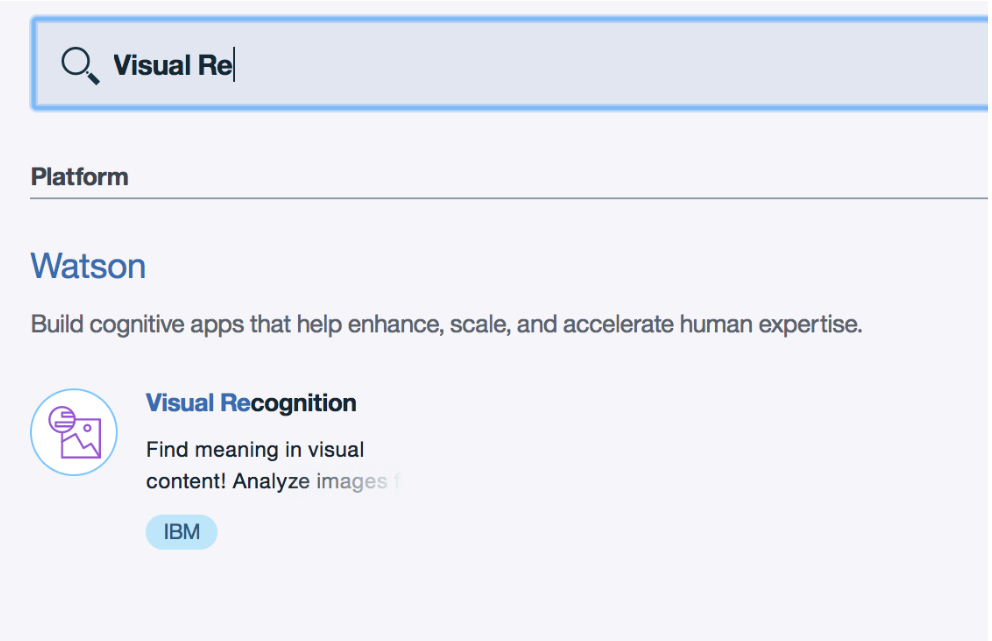
1. Download and install Postman for your platform (if you don't already have it) Instructions are available at the following website https://www.getpostman.com/

2. Create a folder on your local machine for this lab and download the following files to it. Unzip the downloaded file.

3. Create an instance of the Watson Visual Recognition Service in Bluemix..

   Open https://bluemix.net in a browser tab to load the dashboard. Once open, select **Catalog**

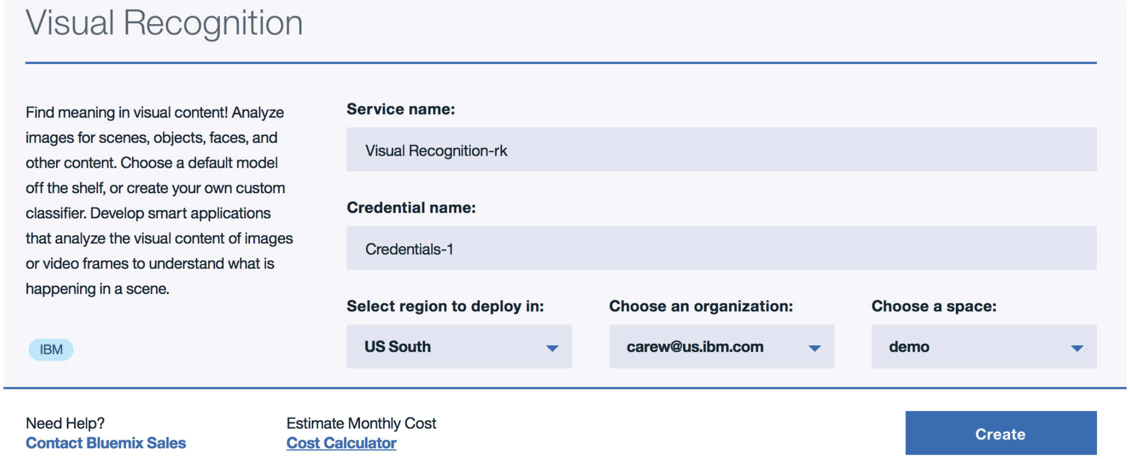
   In the search bar, type **Visual Re**.



Build cognitive apps that help enhance, scale, and accelerate human expertise.

Find meaning in visual content! Analyze images f

   Click on the service name to bring up the panel.



Visual Recognition

Find meaning in visual content! Analyze images for scenes, objects, faces, and other content. Choose a default model off the shelf, or create your own custom classifier. Develop smart applications that analyze the visual content of images or video frames to understand what is happening in a scene.

IBM

Service name: Visual Recognition-rk

Credential name: Credentials-1

Select region to deploy in: US South

Choose an organization: carew@us.ibm.com

Choose a space: demo

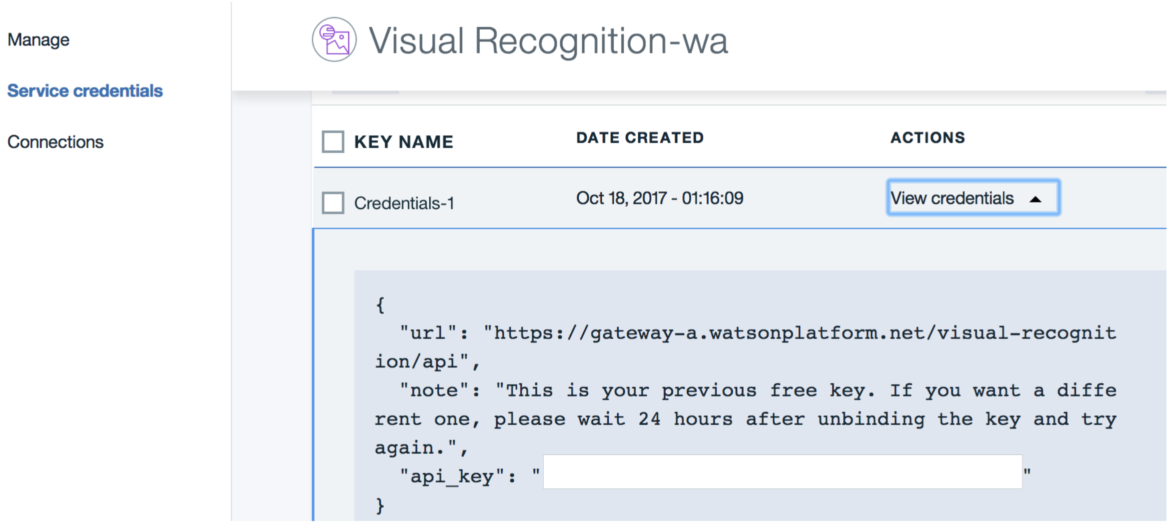
Need Help?
Contact Bluemix Sales

Estimate Monthly Cost
Cost Calculator

Keep all default values..

Click on the **Create** button to create your instance.

4. When the service is created click on **Service credentials** at the left and then click on **View credentials.** Copy the value of your api_key (without the quotes) to a text file as you'll need it several times during the lab

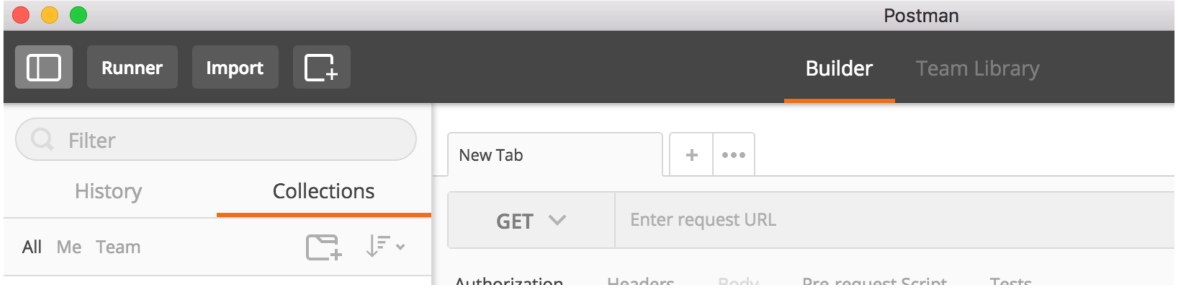


Visual Recognition-wa

KEY NAME	DATE CREATED	ACTIONS
Credentials-1	Oct 18, 2017 - 01:16:09	View credentials ▲

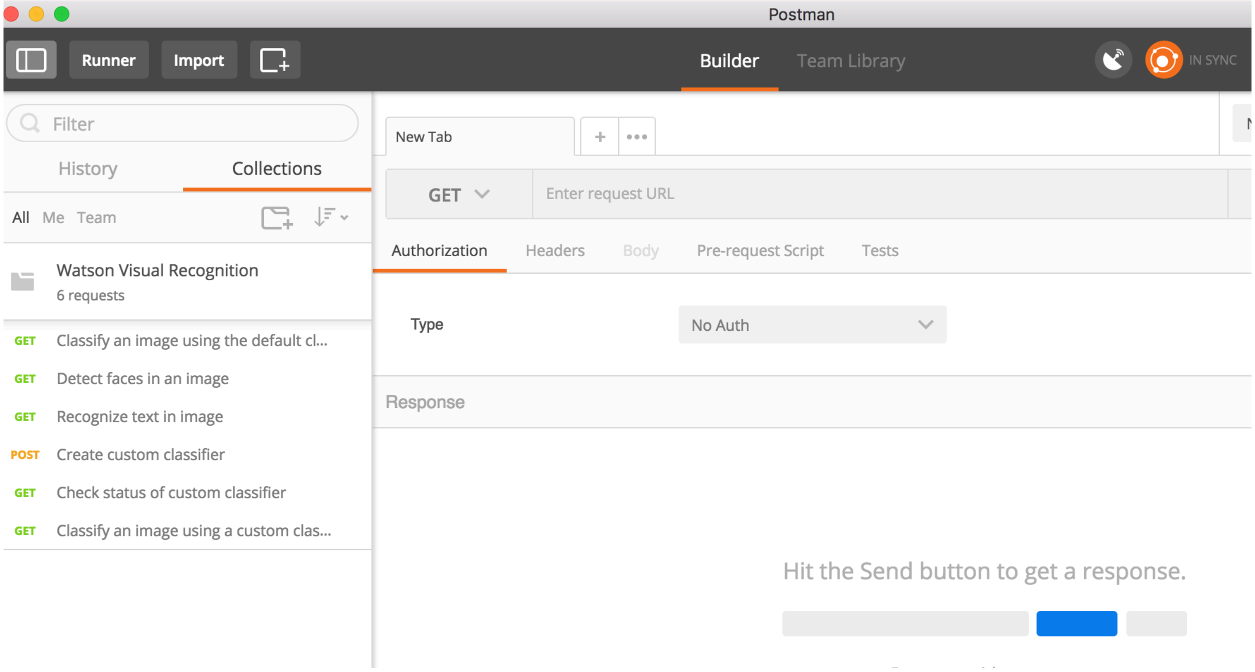
```
{
  "url": "https://gateway-a.watsonplatform.net/visual-recognit
ion/api",
  "note": "This is your previous free key. If you want a diffe
rent one, please wait 24 hours after unbinding the key and try
again.",
  "api_key": "                                    "
}
```

5. Start your Postman app and click on the **Import** button



Select **Import File** and then drag and drop or select the file ***Watson Visual Recognition.postman_collection.json*** from the folder where your unzipped the lab files

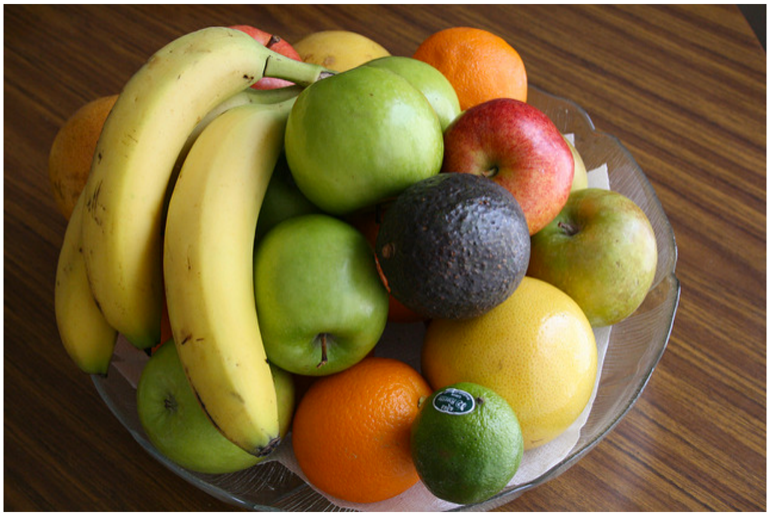
Verify that 6 members of a collection entitled *Watson Visual Recognition* were imported


Postman
Runner
Import
Builder
Team Library
IN SYNC
Filter
History
Collections
All Me Team
Watson Visual Recognition
6 requests
GET Classify an image using the default cl...
GET Detect faces in an image
GET Recognize text in image
POST Create custom classifier
GET Check status of custom classifier
GET Classify an image using a custom clas...
New Tab
GET
Enter request URL
Authorization Headers Body Pre-request Script Tests
Type
No Auth
Response
Hit the Send button to get a response.

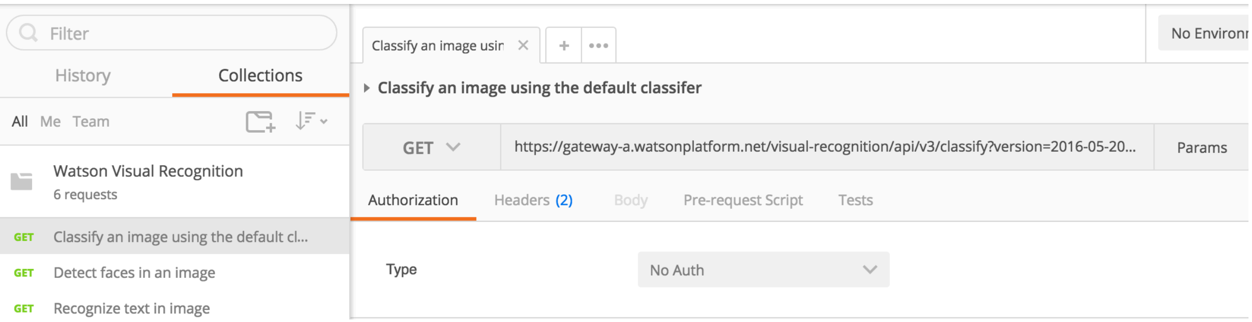
Section 2. Execute REST calls against your instance of Watson Visual Recognition

In this section, you will familiarize yourself with the Watson Visual Recognition service by making REST calls to classify, detect faces in, and recognize text in an image, and you will also learn how to create, train and then use a custom classifier.

1. First you will classify the following image using the default classifier .



In Postman select the first request in the imported collection. A tab on the right will be opened with all the details of the request



Watson Visual Recognition
6 requests

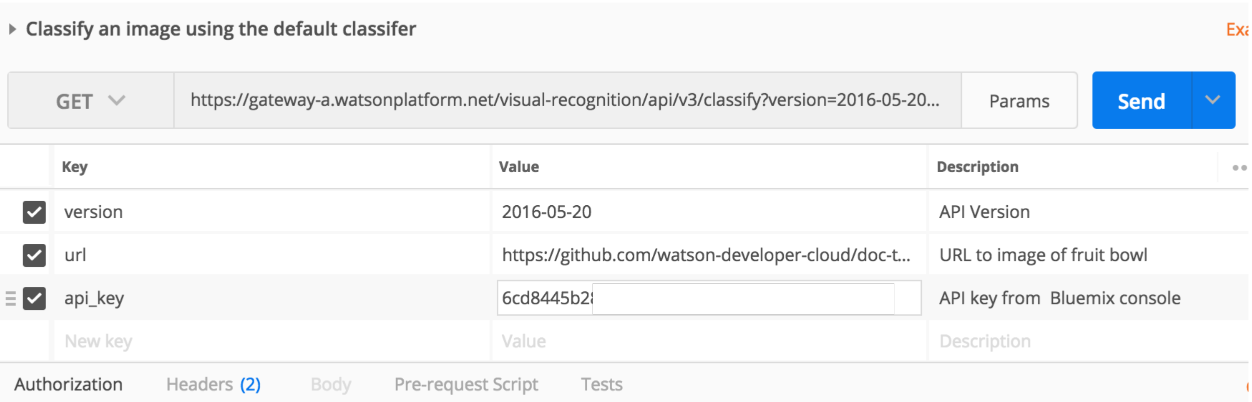
GET Classify an image using the default cl...
GET Detect faces in an image
GET Recognize text in image

Classify an image using the default classifer

GET https://gateway-a.watsonplatform.net/visual-recognition/api/v3/classify?version=2016-05-20...

Type: No Auth

Click on **Params** and then paste your API key into the **Value** column for the **api_key** parameter



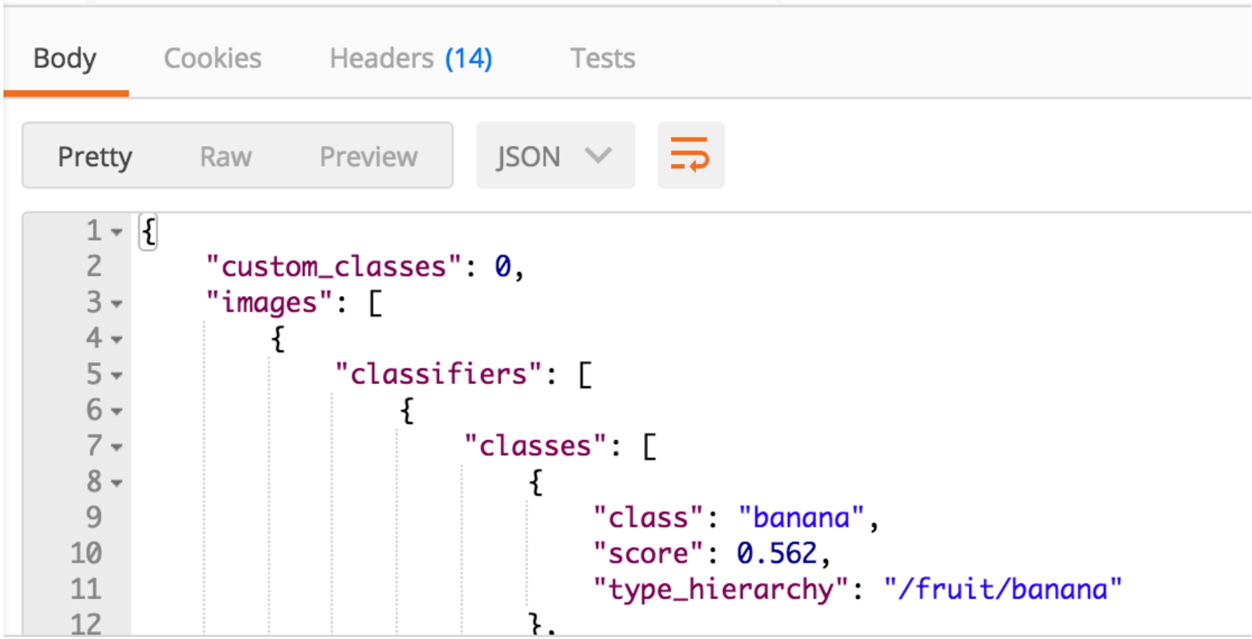
Classify an image using the default classifer

GET https://gateway-a.watsonplatform.net/visual-recognition/api/v3/classify?version=2016-05-20...

	Key	Value	Description
☑	version	2016-05-20	API Version
☑	url	https://github.com/watson-developer-cloud/doc-t...	URL to image of fruit bowl
☑	api_key	6cd8445b2	API key from Bluemix console
	New key	Value	Description

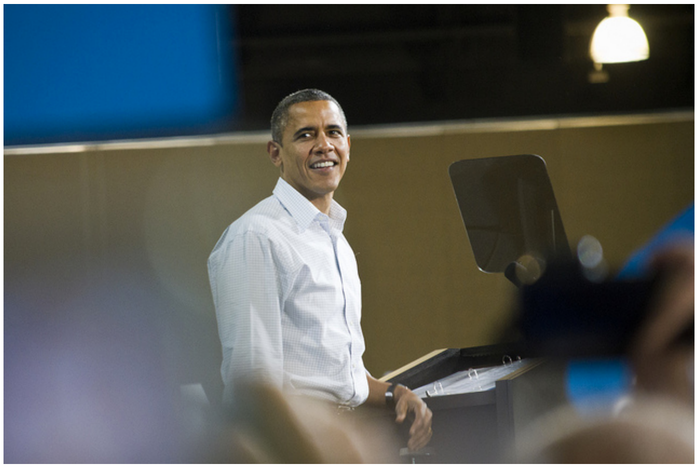
Click the **Send** button to send the request

Verify that a JSON response is returned. Scroll through the results to see the default classifier's analysis of this image.



```
{
    "custom_classes": 0,
    "images": [
        {
            "classifiers": [
                {
                    "classes": [
                        {
                            "class": "banana",
                            "score": 0.562,
                            "type_hierarchy": "/fruit/banana"
                        },
```

2. Next you'll use the face detection capability on this image:

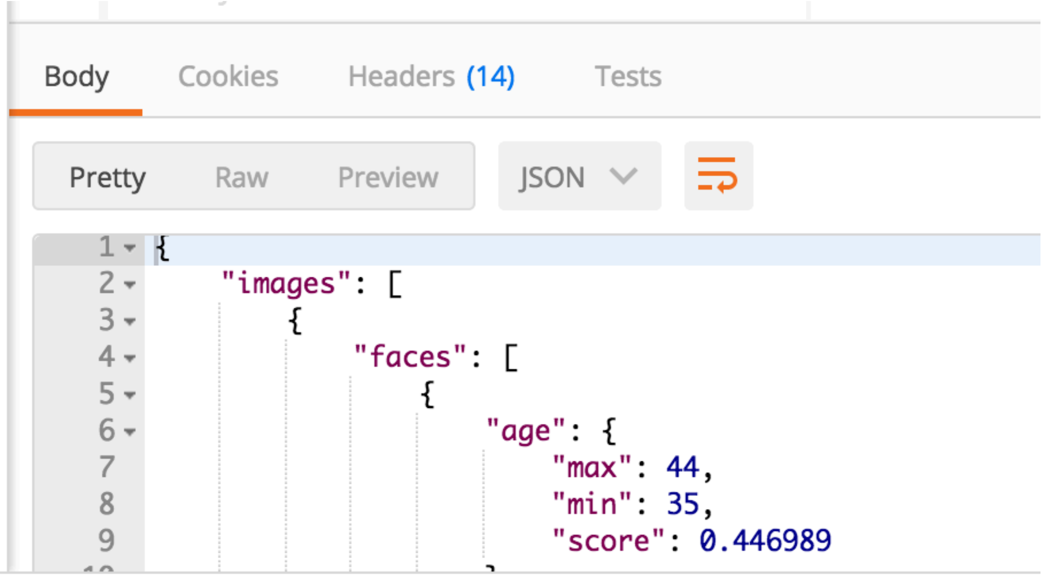


In Postman select the second request in the imported collection. A tab on the right will be opened with all the details of the request

Click on **Params** and then paste your API key into the **Value** column for the **api_key** parameter.

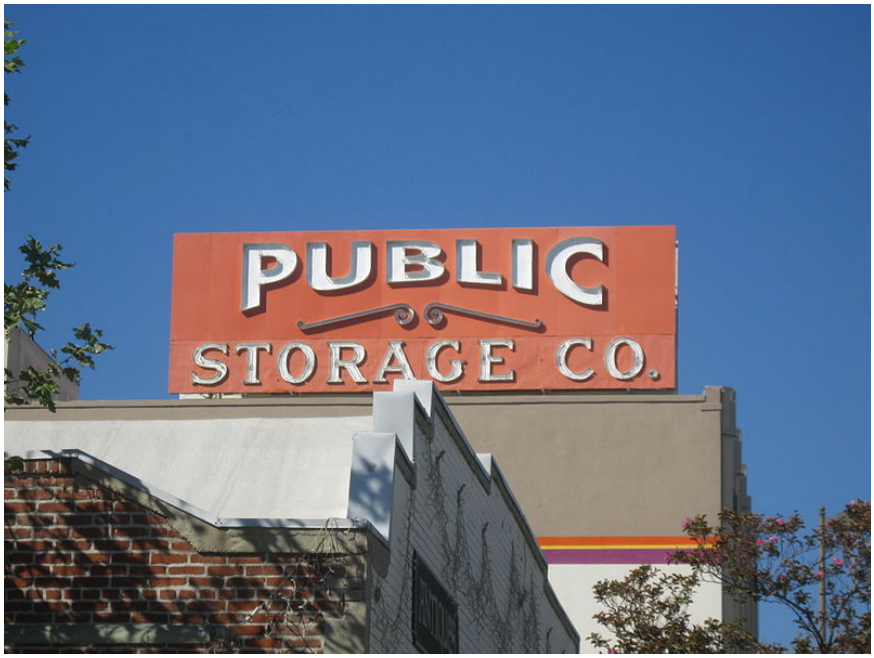
Click the **Send** button to send the request.

Verify that a JSON response is returned. Scroll through the results to see the what is returned for the face detection function



```
{
    "images": [
        {
            "faces": [
                {
                    "age": {
                        "max": 44,
                        "min": 35,
                        "score": 0.446989
```

3. Next you'll use the text recognition capability on this image:



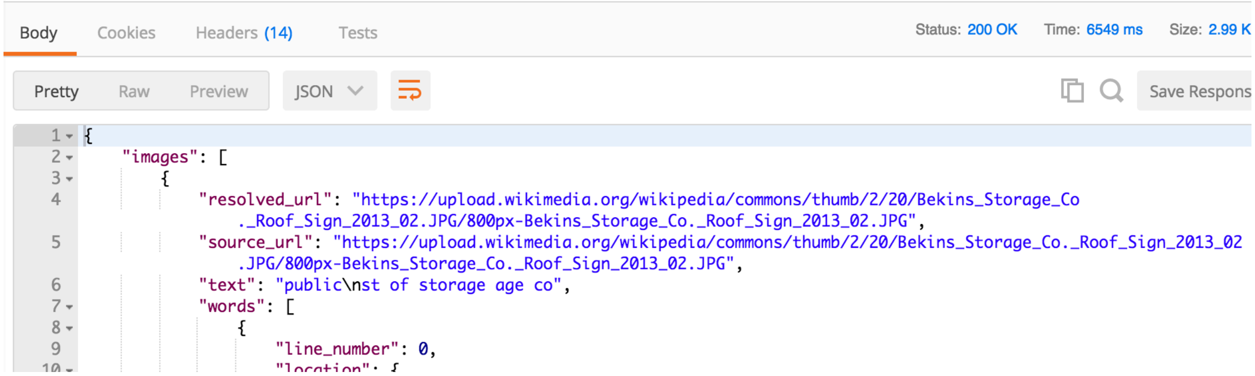
”

In Postman select the third request in the imported collection. A tab on the right will be opened with all the details of the request

Click on **Params** and then paste your API key into the **Value** column for the **api_key** parameter.

Click the **Send** button to send the request

Verify that a JSON response is returned. Scroll through the results to see the what is returned for the text recognition function



Status: 200 OK   Time: 6549 ms   Size: 2.99 K

```
{
    "images": [
        {
            "resolved_url": "https://upload.wikimedia.org/wikipedia/commons/thumb/2/20/Bekins_Storage_Co
                ._Roof_Sign_2013_02.JPG/800px-Bekins_Storage_Co._Roof_Sign_2013_02.JPG",
            "source_url": "https://upload.wikimedia.org/wikipedia/commons/thumb/2/20/Bekins_Storage_Co._Roof_Sign_2013_02
                .JPG/800px-Bekins_Storage_Co._Roof_Sign_2013_02.JPG",
            "text": "public\nst of storage age co",
            "words": [
                {
                    "line_number": 0,
                    "location": {
```

4. Next you'll learn how to create a custom classifier to recognize various breeds of dogs

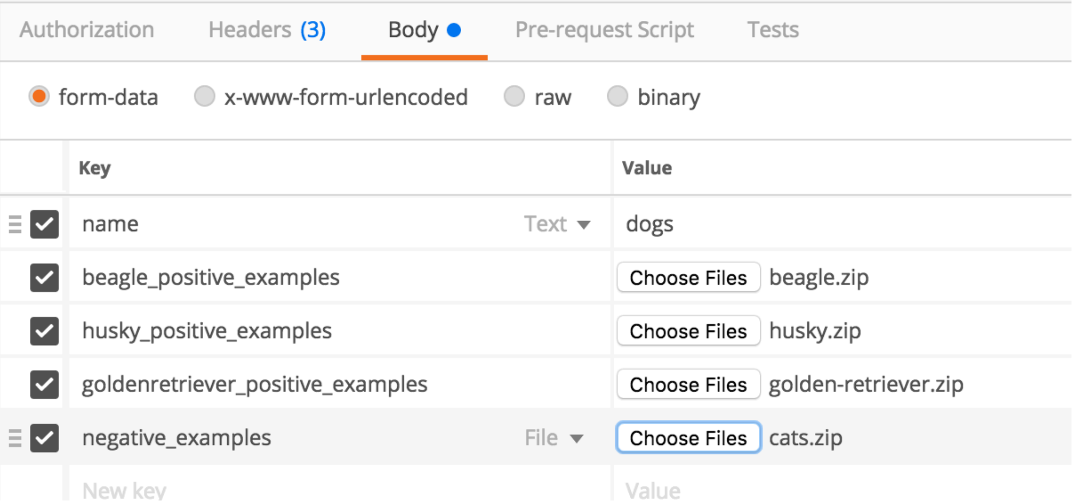
    In Postman select the fourth request in the imported collection. A tab on the right will be opened with all the details of the request

    Click on **Params** and then paste your API key into the **Value** column for the **api_key** parameter.

    Next click on Body to see the Form Data that is being passed to the request. Add the following files from the files you unzipped for this lab by clicking on Choose Files next to the corresponding parameter

    i. Select ***beagle.zip*** for **beagle_positive_examples**
    ii. Select ***husky.zip*** for **husky_positive_eaxamples**
    iii. Select ***golden_retriever.zip*** for **goldenretriver_positive_examples**
    iv. Select ***cats.zip*** for **negative_examples**

    When you're done the **Body** should look like this:

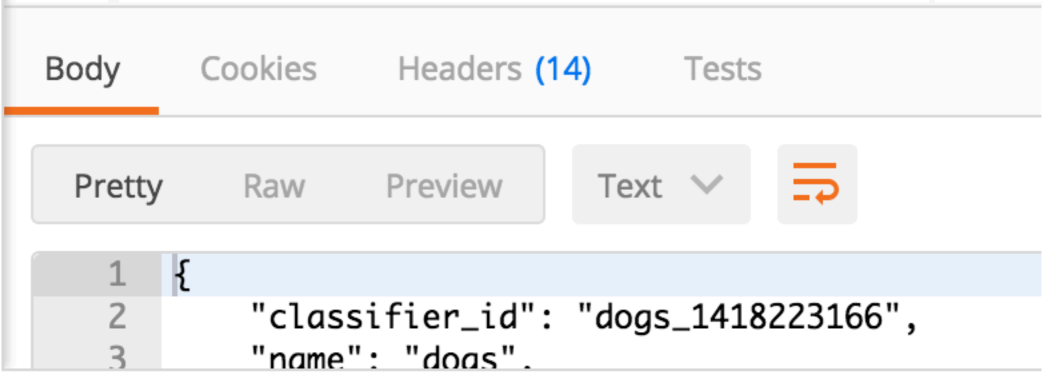


● form-data ○ x-www-form-urlencoded ○ raw ○ binary

	Key		Value
☑	name	Text	dogs
☑	beagle_positive_examples		Choose Files beagle.zip
☑	husky_positive_examples		Choose Files husky.zip
☑	goldenretriever_positive_examples		Choose Files golden-retriever.zip
☑	negative_examples	File	Choose Files cats.zip
	New key		Value

Click the **Send** button to send the request The request will take awhile to complete as all the files need to be uploaded to the service .

Copy and paste the returned **classifier_id** to the same file as your api_key. You'll need this is the next steps:



```
{
    "classifier_id": "dogs_1418223166",
    "name": "dogs",
```

5. Custom classifiers are trained asynchronously so in this step you'll check the status of the custom classifier you just created.

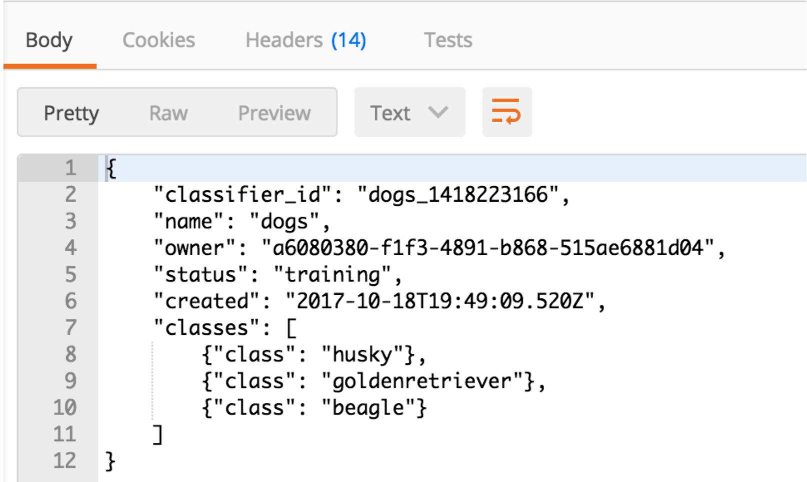
   In Postman select the fifth request in the imported collection. A tab on the right will be opened with all the details of the request

   Click on **Params** and then paste your API key into the **Value** column for the **api_key** parameter.

   Edit the URL of the request and change *dogs_XXXXXXXX* to the id of the classifier you saved in the previous step

Click the **Send** button to send the request

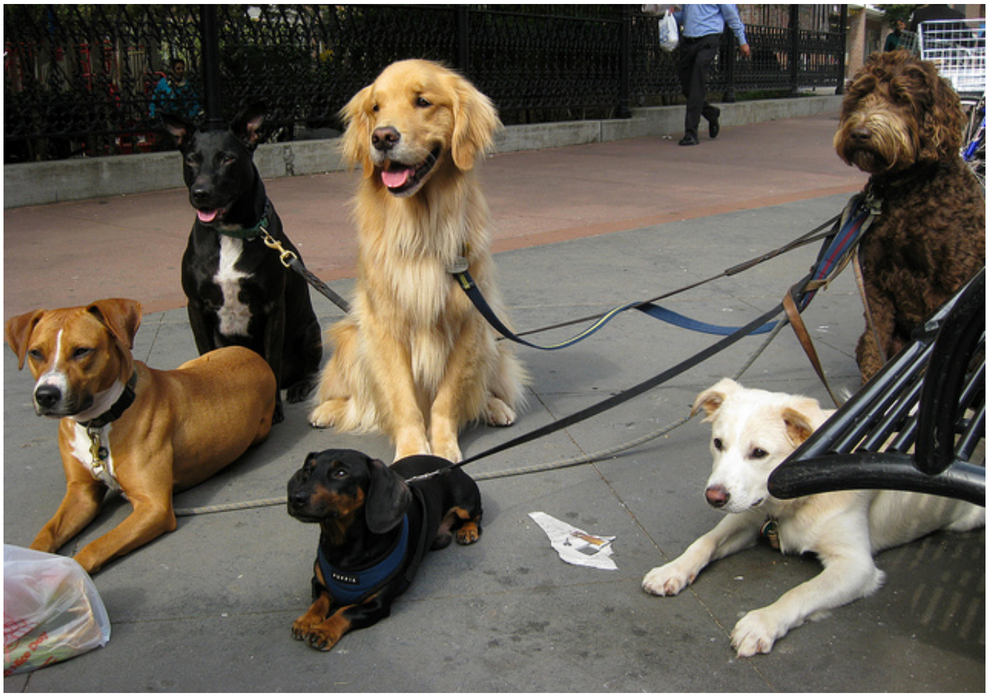
Note the returned JSON. The status field shows the state of the custom classifier. If the status is set to *training* it means that the classifier has not been completely trained yet



```
{
    "classifier_id": "dogs_1418223166",
    "name": "dogs",
    "owner": "a6080380-f1f3-4891-b868-515ae6881d04",
    "status": "training",
    "created": "2017-10-18T19:49:09.520Z",
    "classes": [
        {"class": "husky"},
        {"class": "goldenretriever"},
        {"class": "beagle"}
    ]
}
```

Keep sending this request until the status changes to *ready*

6. In this final part of the exercise you'll run the following image against both your custom classifier and the default classifier.



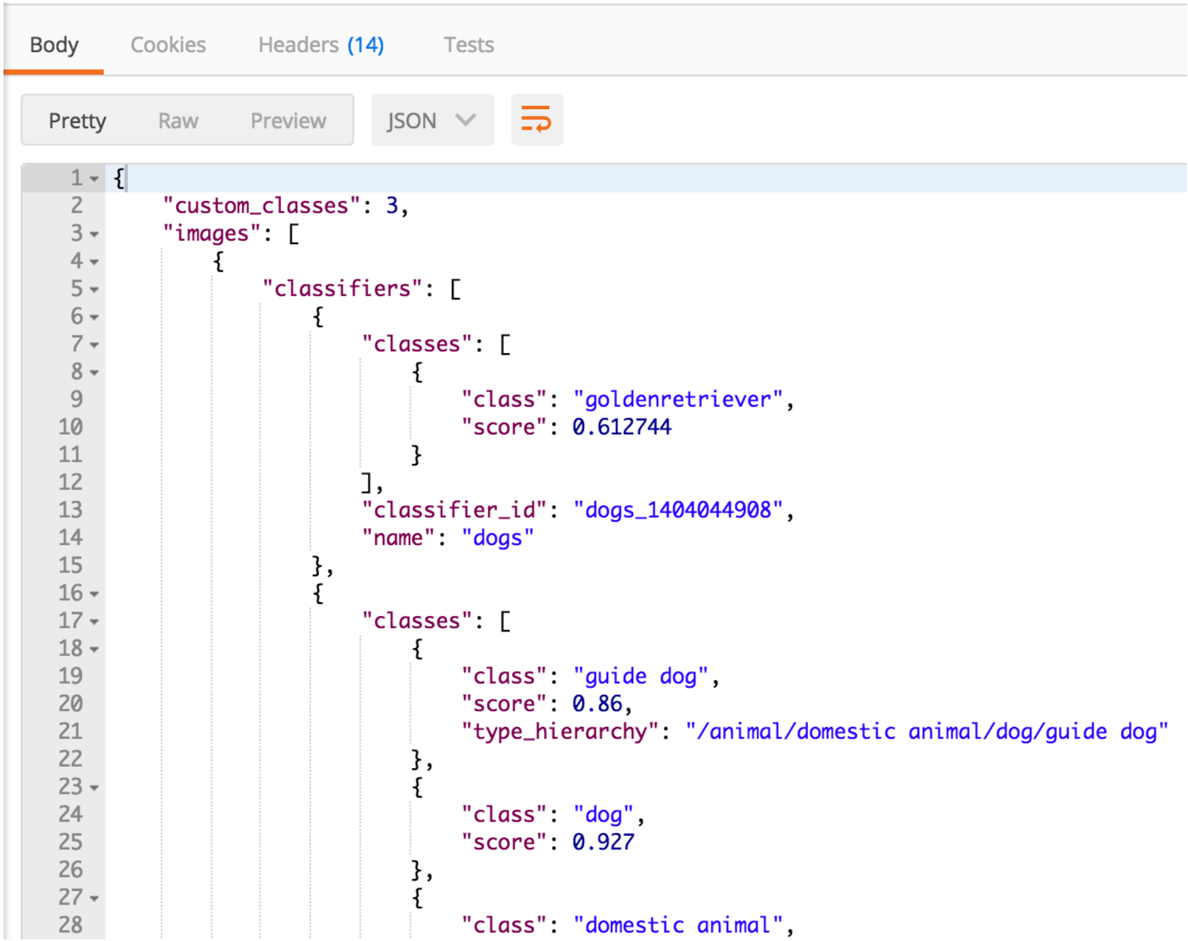
In Postman select the sixth request in the imported collection. A tab on the right will be opened with all the details of the request

Click on **Params** and then paste your API key into the **Value** column for the **api_key** parameter.

Change the *dogs_XXXXX* part of the **classifer_ids** parameter to the classifier id of your custom classifier

Click the **Send** button to send the request

Note the returned JSON. Scroll through the results to see the output organized by classifier . The results from your custom classifier should appear first followed by the results from the default classifier.



```
{
    "custom_classes": 3,
    "images": [
        {
            "classifiers": [
                {
                    "classes": [
                        {
                            "class": "goldenretriever",
                            "score": 0.612744
                        }
                    ],
                    "classifier_id": "dogs_1404044908",
                    "name": "dogs"
                },
                {
                    "classes": [
                        {
                            "class": "guide dog",
                            "score": 0.86,
                            "type_hierarchy": "/animal/domestic animal/dog/guide dog"
                        },
                        {
                            "class": "dog",
                            "score": 0.927
                        },
                        {
                            "class": "domestic animal",
```

Summary

In this lab, You created an instance of the Watson Visual Recognition Service and completed the following tasks

1. Classifying an image
2. Detecting faces in an image

3. Recognizing text in an image
4. Creating a custom classifier
5. Checking the status of a custom classifier
6. Classifying an image with a custom classifier