

## Introduction

Microsoft Cognitive Services is a set of APIs that use the power of Machine Learning to analyze, pictures, video, speech, and language.

Working through this set of exercises will help you understand how to call these APIs, what data is required, and what data is returned.

## Prerequisites: Azure Account and Editor

This lab requires an Azure account. If you do not yet have an Azure account, navigate to <https://azure.microsoft.com/>; click the [Start Free] button and follow the instructions to sign up for a free Azure trial.



You will need an editor for Exercise 4. I prefer Visual Studio Code, because it is free, lightweight, and will run on Windows , MacOS, or Linux. You can download Visual Studio Code at <https://code.visualstudio.com/>. But feel free to use any editor with which you are comfortable.

## Exercise 1: Get an API Key

In order to work with Cognitive Services, you will need an API key. Navigate to the Azure Portal (<https://portal.azure.com>) and log in if prompted.

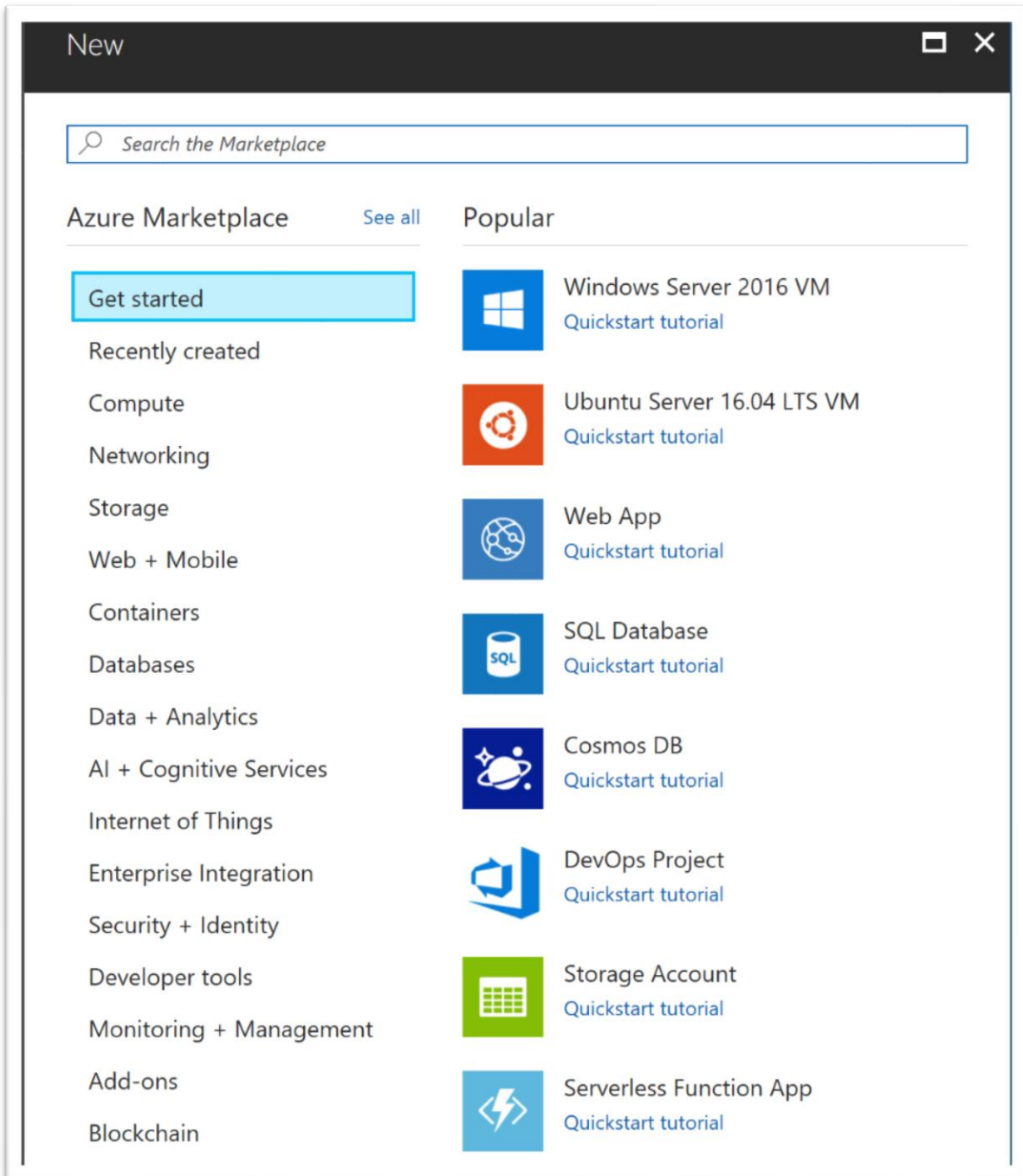
NOTE: If you do not have an Azure account, see the Prerequisites section above.

Click the [+ New] button in the left menu

NOTE: For some accounts, the word “New” is replaced by the words “Create a resource”.



This will open the New Resource blade.





In the “Search” box, enter “Emotion API” and press ENTER. The search results should list “Emotion API (preview)”.

Everything


Filter

Emotion API

Results

NAME	PUBLISHER	CATEGORY
 Emotion API (preview)	Microsoft	AI + Cognitive Services
 Cognitive Services	Microsoft	Data + Analytics

Click “Emotion API (preview)” to open the “Emotion API (preview)” blade.

 Emotion API (preview)  
Microsoft

How are you feeling? Can your app tell? With the Emotion API, you can build an app that recognizes emotions according to facial expressions-giving you the capability to provide an amazing, personalized experience.

Using facial expressions, this cloud-based API can detect happiness, neutrality, sadness, contempt, anger, disgust, fear, and surprise. The AI understands these emotions based on universal facial expressions, and it functions cross-culturally, so your app will work around the world.







Create apps that respond to moods, recognize feelings, and get personal with the Emotion API.

### Legal Notice

Microsoft will use data you send to the Cognitive Services to improve Microsoft products and services. For example, we will use content that you provide to the Cognitive Services to improve our underlying algorithms and models over time. Where you send personal data to the Cognitive Services, you are responsible for obtaining sufficient consent from the data subjects. The General Privacy and Security Terms in the [Online Services Terms](#) do not apply to the Cognitive Services.

Please refer to the Microsoft Cognitive Services section in the [Online Services Terms](#) for details.

Note: Microsoft offers policy controls that may be used to disable new Cognitive Services deployments. [Learn more](#)



---

PUBLISHER	Microsoft
	<a href="#">More about Emotion API (preview)</a>
	<a href="#">Documentation</a>
USEFUL LINKS	<a href="#">API reference for images</a>
	<a href="#">API reference for video</a>
	<a href="#">Pricing</a>
	<a href="#">Regional availability</a>

---

Create

Click the [Create] button to open the “Create Emotion API” blade.

## Create



Emotion API (preview)

\* Name

myemotion



\* Subscription

Microsoft Azure Internal Consumption



\* Location

West US



\* Pricing tier ([View full pricing details](#))

S0 (10 Calls per second)



\* Resource group



Create new



Use existing

myresourcegroup



\* I confirm I have read and understood the notice below.

Microsoft will use data you send to the Cognitive Services to improve Microsoft products and services. Where you send personal data to the Cognitive Services, you are responsible for obtaining sufficient consent from the data subjects. The General Privacy and Security Terms in the Online Services Terms do not apply to the Cognitive Services. Please refer to the Microsoft Cognitive Services section in the [Online Services Terms](#) for details. Microsoft offers policy controls that may be used to [disable new Cognitive Services deployments](#).



Pin to dashboard

Create

[Automation options](#)

At the “Name” field, enter a name for this key.

At the “Location” dropdown, select “West US”.

At the “Pricing Tier” dropdown, select “S0”.

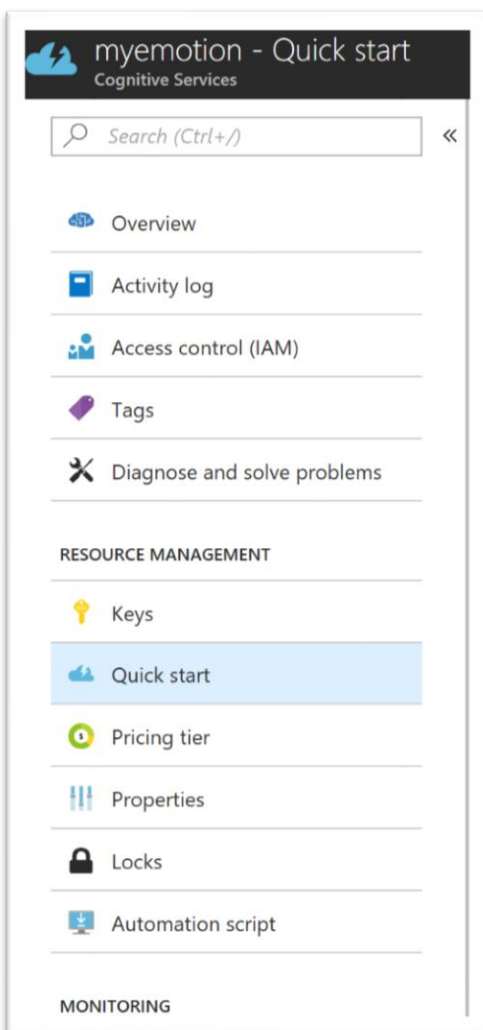
At the “Resource Group” field, select the “Create new” radio button and enter a name for a resource group. A Resource Group is a logical container for Azure resources.

Check the “Pin to dashboard” checkbox.

Check the checkbox to indicate you agree with whatever it says in the fine print below it; and click the [Create] button to create a new Emotion API key.

After a few seconds, the key will be created and a shortcut will appear on your desktop. Click the shortcut to open the newly-created Cognitive Services blade.

NOTE: You can also open this blade by “All resources” in the left menu and selecting your newly-created Cognitive Service from the list.



There are 2 key pieces of information in this blade: the endpoint and the access key. You can get to both from the “Overview” tab. Click the “Overview” link to open the “Overview” tab.

The screenshot shows the 'Overview' blade for an Emotion API resource. At the top, there is a 'Delete' button and a tab labeled 'Essentials'. Below the tab, the resource details are organized into two columns. The left column contains: 'Resource group (change)' with a link to 'myresourcegroup', 'Status' as 'Active', 'Location' as 'West US', 'Subscription name (change)' as 'Microsoft Azure Internal Consumption', and 'Subscription ID' as '442cdc0c-7236-48ee-9bcf-51fe0adb3638'. The right column contains: 'API type' as 'Emotion API (preview)', 'Pricing tier' as 'Standard', 'Endpoint' as 'https://westus.api.cognitive.microsoft.com/emotion/v1.0', and a 'Manage keys' section with a 'Show access keys ...' link. Below this is a 'Monitoring' section with three metrics: 'Total Calls and Total Errors' (100), 'Latency' (100), and 'Data In and Data Out' (100B).

Essentials	
Resource group (change) <a href="#">myresourcegroup</a>	API type Emotion API (preview)
Status Active	Pricing tier Standard
Location West US	Endpoint <a href="https://westus.api.cognitive.microsoft.com/emotion/v1.0">https://westus.api.cognitive.microsoft.com/emotion/v1.0</a>
Subscription name (change) <a href="#">Microsoft Azure Internal Consumption</a>	Manage keys <a href="#">Show access keys ...</a>
Subscription ID 442cdc0c-7236-48ee-9bcf-51fe0adb3638	

Monitoring		
Total Calls and Total Errors	Latency	Data In and Data Out
100	100	100B

Record the Endpoint (probably <https://westus.api.cognitive.microsoft.com/emotion/v1.0>). You can copy this endpoint by clicking the Copy icon to the right of the URL.

This image is a close-up of the 'Endpoint' field from the previous screenshot. It shows the text 'https://westus.api.cognitive.microsoft.com/emotion/v1.0'. To the right of the URL is a copy icon (two overlapping sheets of paper). A dark grey tooltip with the text 'Click to copy' is positioned over the copy icon.

API type  
Emotion API (preview)

Pricing tier  
Standard

Endpoint  
<https://westus.api.cognitive.microsoft.com/emotion/v1.0>

Manage keys  
[Show access keys ...](#)

Click the “Show access keys...” link to open the “Manage keys” blade.



[Home](#) > [myemotion](#) > Manage keys

# Manage keys

myemotion

1 Regenerate Key1

2 Regenerate Key2

Notice: It may take up to 10 minutes for the newly (re)generated keys to take effect.

NAME

myemotion

KEY 1

d61708-18c3-4371-b9750c01-93f74

KEY 2

f101-53-41121-7486f-533f-161

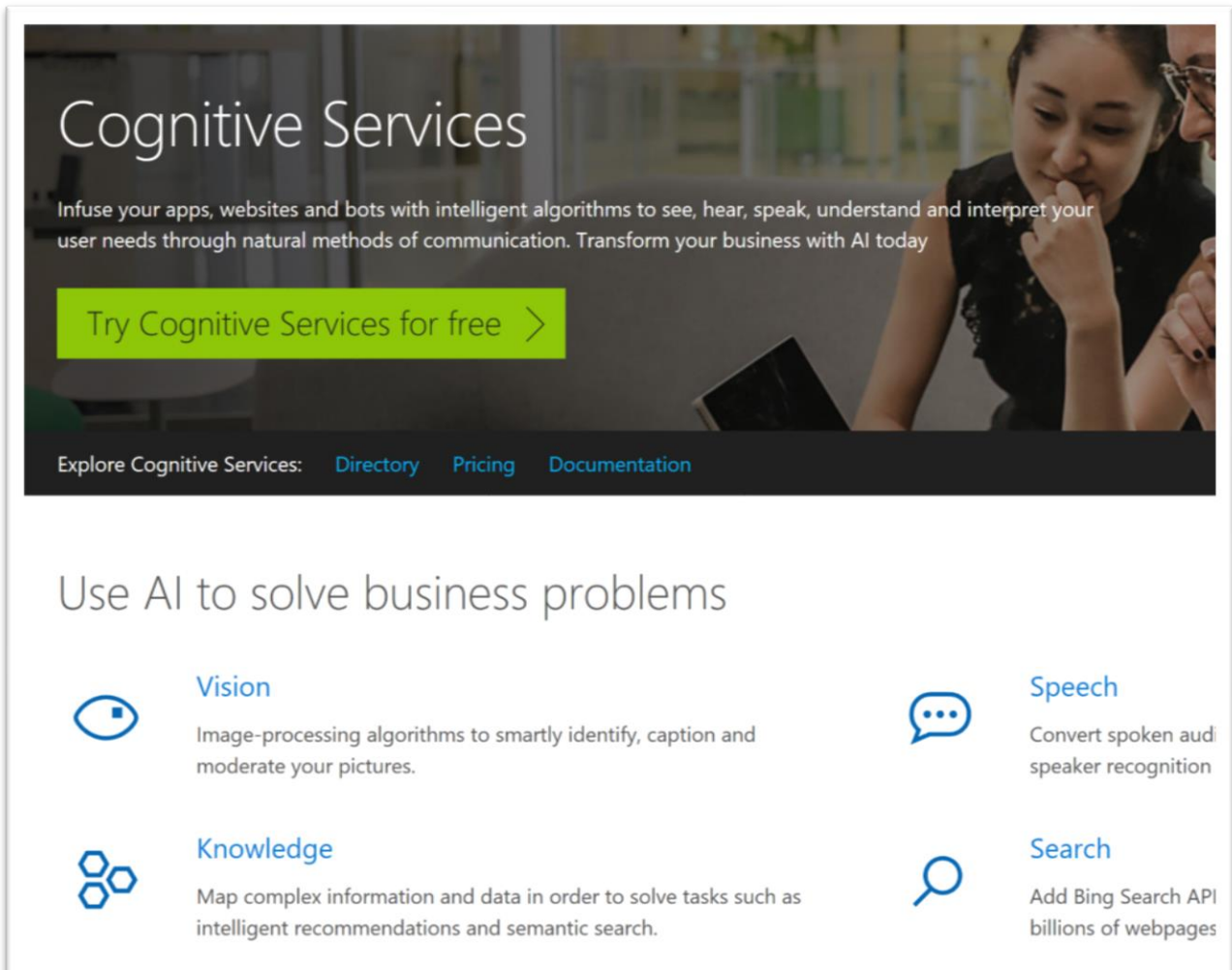
Click to copy

Record the value in KEY 1. You can copy this value by clicking the Copy icon to the right of the key.

NOTE: Cognitive Services provides 2 keys in case one of the keys is compromised and you need to re-generate it.


## Exercise 2: Explore Cognitive Services

Open a new browser tab and navigate to the Cognitive Services home page at <https://microsoft.com/cognitive-services>.



Click the "Vision" link to display the Cognitive Services Vision page.


VisionSpeechLanguageKnowledgeSearch



### Computer Vision API

Distill actionable information from images


[Try Computer Vision API](#) | [Use with an Azure subscription](#)



### Content Moderator

Automated image, text, and video moderation


[Use with an Azure subscription](#)



### Custom Vision Service PREVIEW

Easily customize your own state-of-the-art computer vision models for your unique use case


[Try Custom Vision Service](#)



### Face API

Detect, identify, analyze, organize, and tag faces in photos


[Try Face API](#) | [Use with an Azure subscription](#)



### Emotion API PREVIEW

Personalize user experiences with emotion recognition

[Try Emotion API](#) | [Use with an Azure subscription](#)



### Video Indexer PREVIEW

Unlock video insights

[Try Video Indexer](#)

Click the “Emotion API” link to navigate the Emotion API page.

Home > Products > Cognitive Services > Emotion API

# Emotion API PREVIEW

Analyze faces to detect a range of feelings and personalize your app's responses.

Try Emotion API >


Explore Cognitive Services: [Documentation](#) [API](#) [SDK](#) [Pricing](#) [Portal](#) [Try Emotion API](#) [Stack Overflow](#)

## Recognize emotions in images

The Emotion API takes a facial expression in an image as an input, and returns the confidence across a set of emotions for each face in the image, as well as bounding box for the face, using the Face API. If a user has already called the Face API, they can submit the face rectangle as an optional input.

The emotions detected are anger, contempt, disgust, fear, happiness, neutral, sadness, and surprise. These emotions are understood to be cross-culturally and universally communicated with particular facial expressions.

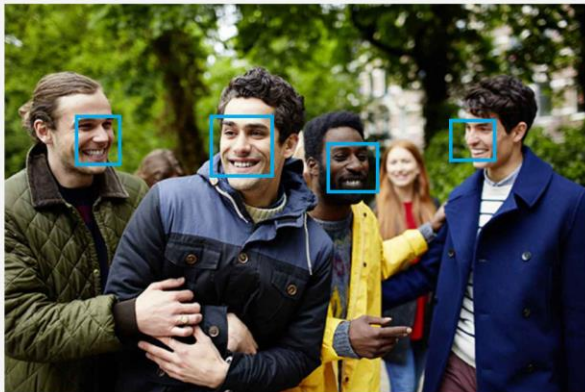
See it in action



Detection result:  
4 faces detected

Scroll down to below the “See it in action” section.

See it in action



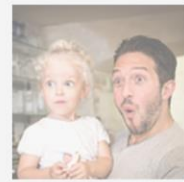
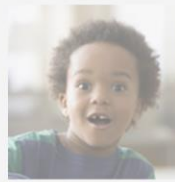
```
Detection result:
4 faces detected

JSON:
[
  {
    "faceRectangle": {
      "top": 114,
      "left": 212,
      "width": 65,
      "height": 65
    },
    "scores": {
      "anger": 1.0570484E-08,
      "contempt": 1.52679547E-09,
      "disgust": 1.60232943E-07,
      "fear": 6.00660363E-12,
      "happiness": 0.9999998,
      "neutral": 9.449728E-09,
      "sadness": 1.23025981E-08,
      "surprise": 9.91396E-10
    }
  }
]
```

Image URL

Submit

Browse



By uploading data for this demo, you agree that Microsoft may store it and use it to improve Microsoft services, including this API. To help protect your privacy, we take steps to de-identify your data and keep it secure. We won't publish your data or let other people use it.

Click each of the images; observe the JSON returned from the API call. Note that it always returns an array of faces, even though sometimes there is only one face in the array. Note that each face includes a numeric score (between 0 and 1) for 8 different emotions.

Try it with one of your own images – either from your computer or from a web search.

## Exercise 3: Use the API Testing Console

Near the top of the Emotion API page is a menu with links to documentation, the API reference, and other useful information.

Explore Cognitive Services: [Documentation](#) [API](#) [SDK](#) [Pricing](#) [Portal](#) [Try Emotion API](#) [Stack Overflow](#)

Click the “API” link in this menu to navigate to the Emotion API page.

Microsoft

Cognitive Services

APIs Documentation > API Reference

POST Emotion Recognition

POST Emotion Recognition in Video

POST Emotion Recognition with Face Rectangles

GET Get Recognition in Video Operation Result

## Emotion API

API definition

### Emotion Recognition

Recognizes the emotions expressed by one or more people in an image, as well as returns a bounding box for the face. The emotions detected are happiness, sadness, surprise, anger, fear, contempt, and disgust or neutral.

- The supported input image formats includes JPEG, PNG, GIF(the first frame), BMP. Image file size should be no larger than 4MB.
- If a user has already called the Face API, they can submit the face rectangles as an optional input. Otherwise, Emotion API will first compute the rectangles.
- The detectable face size range is 36x36 to 4096x4096 pixels. Faces out of this range will not be detected.
- For each image, the maximum number of faces detected is 64 and the faces are ranked by face rectangle size in descending order. If no face is detected, an empty array will be returned.
- Some faces may not be detected due to technical challenges, e.g. very large face angles (head-pose), large occlusion. Frontal and near-frontal faces have the best results.
- The emotions contempt and disgust are experimental.

Open API testing console

#### Request URL

`https://westus.api.cognitive.microsoft.com/emotion/v1.0/recognize`

#### Request headers

Header	Type	Description
<b>Content-Type</b> (optional)	string	Media type of the body sent to the API.
<b>Ocp-Apim-Subscription-Key</b>	string	Subscription key which provides access to this API. Found in your <a href="#">Cognitive Services accounts</a> .

#### Request body

application/json

application/octet-stream

```
{ "url": "http://example.com/picture.jpg" }
```

#### Response 200

A successful call returns an array of face entries and their associated emotion scores, ranked by face rectangle size in descending order. An empty response indicates that no faces were detected. An emotion entry contains the following fields:

Fields	Description
faceRectangle	Rectangle location of face in the image.
scores	Emotion scores for each face in the image.

application/json

```
{
  "faceRectangle": {
    "left": 68,
    "top": 97,
    "width": 64,
    "height": 97
  },
  "scores": {
    "anger": 0.00300731952,
    "contempt": 5.14648448E-08,
    "disgust": 9.180124E-06,
```

This page documents how to call the Emotions API web service, what parameters to pass and what response to expect. By default, the “Emotion Recognition” method is shown. Review this documentation.

Click the [Open API testing console] button to navigate to the Emotion API Testing console.

Microsoft Cognitive Services

APIs Documentation > API Reference

**POST** Emotion Recognition

**POST** Emotion Recognition in Video

**POST** Emotion Recognition with Face Rectangles

**GET** Get Recognition in Video Operation Result

## Emotion API

### Emotion Recognition

Recognizes the emotions expressed by one or more people in an image, as well as returns a bounding box for the face. The emotions detected are happiness, sadness, surprise, anger, fear, contempt, and disgust or neutral.

- The supported input image formats includes JPEG, PNG, GIF(the first frame), BMP. Image file size should be no larger than 4MB.
- If a user has already called the Face API, they can submit the face rectangles as an optional input. Otherwise, Emotion API will first compute the rectangles.
- The detectable face size range is 36x36 to 4096x4096 pixels. Faces out of this range will not be detected.
- For each image, the maximum number of faces detected is 64 and the faces are ranked by face rectangle size in descending order. If no face is detected, an empty array will be returned.
- Some faces may not be detected due to technical challenges, e.g. very large face angles (head-pose), large occlusion. Frontal and near-frontal faces have the best results.
- The emotions contempt and disgust are experimental.

Query parameters

+ Add parameter

Headers

Content-Type: application/json [Remove header](#)

Ocp-Apim-Subscription-Key: Value [Add header](#)

Request body

```
1 { "url": "http://example.com/picture.jpg" }
```

At the “Content-Type” dropdown, select “application/json”.

At the “Ocp-Apim-Subscription-Key” field, enter your Emotion API key from Exercise 1.

At the “Request body” field, enter

```
{ "url": " https://photos.smugmug.com/Tech-Community/GANGConf-2017/i-H2ZNBjt/0/c28d08df/L/IMG_7743-L.jpg" }
```

NOTE: The URL above is a photo of the 5 mostly happy people below. You can replace it with any image on the web that contains faces.



Details of the request will display at the bottom of the screen. Review these details.

### Headers

Content-Type	<input type="text" value="application/json"/>	<a href="#">✕ Remove header</a>
Ocp-Apim-Subscription-Key	<input type="text" value="b4577bbb9750c91233f74"/>	<a href="#">🔊</a>

[+ Add header](#)

### Request body

```
1 { "url": " https://photos.smugmug.com/Tech-Community/GANGConf-2017/i-H2ZNB3T/0/c28d08df/L/IMG_7743-L.jpg" } |
```

### Request URL

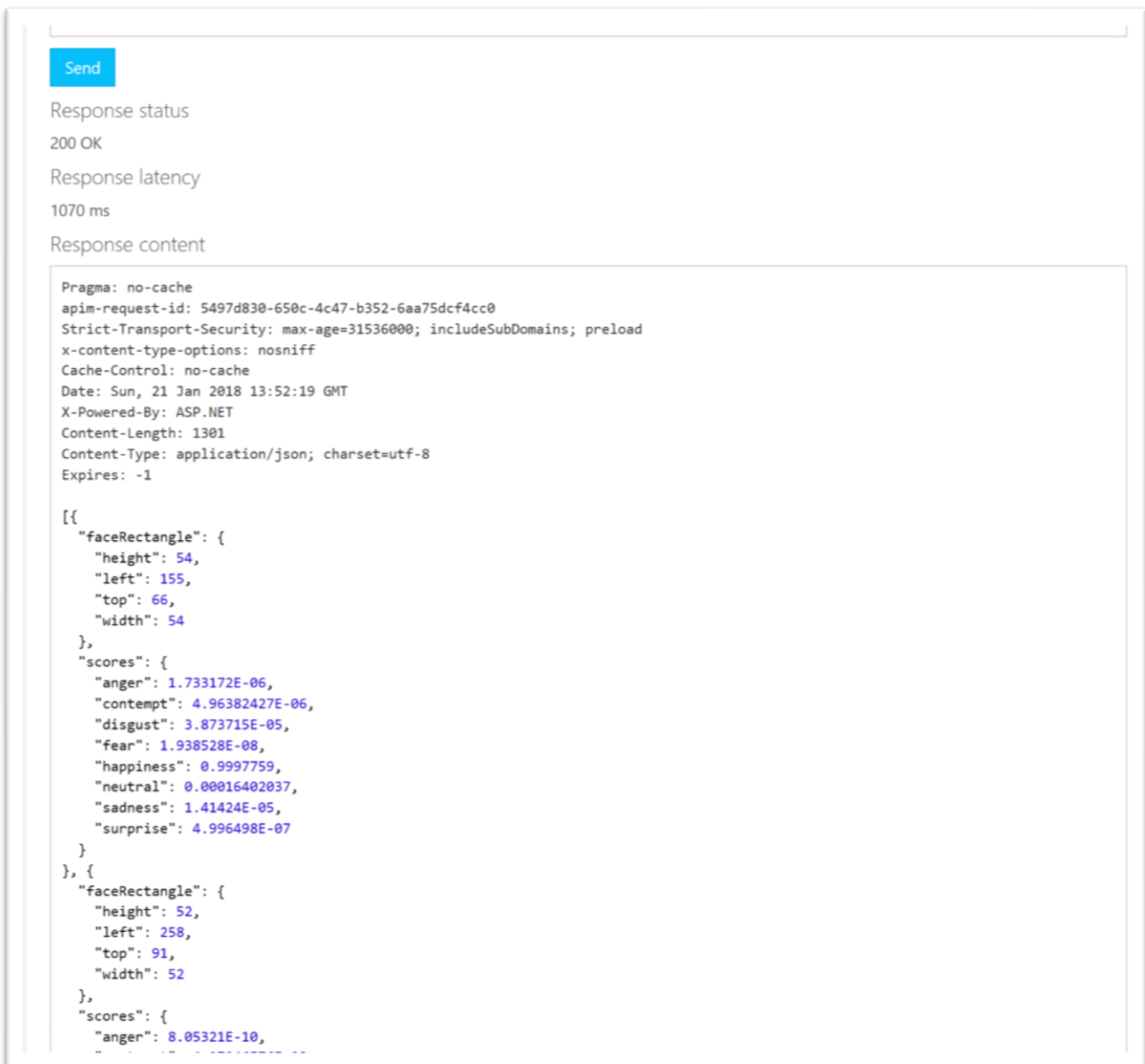
### HTTP request

```
POST https://westus.api.cognitive.microsoft.com/emotion/v1.0/recognize HTTP/1.1
Host: westus.api.cognitive.microsoft.com
Content-Type: application/json
Ocp-Apim-Subscription-Key: *****

{ "url": " https://photos.smugmug.com/Tech-Community/GANGConf-2017/i-H2ZNB3T/0/c28d08df/L/IMG_7743-L.jpg" }
```



Click the [Send] button to send an HTTP request. An HTTP request will be sent to the web service URL and you will receive an HTTP response. The details of this response will display at the bottom of the screen.



If everything worked properly, you should receive a response status of “200 OK” and JSON with information about each face in the photo will display.

Review the response.

Any Response status of 400 or above indicates an error. Common errors are:

- Entering an incorrect Ocp-Apim-Subscription-Key
- Creating your key in the wrong region (This test is designed to work with West US)
- Providing an image URL that either does not exist or is not publicly accessible



## Exercise 4: Calling a Web Service from JavaScript

In this exercise, we will write some JavaScript to call the Image Analysis API and examine the data returned.

Create a folder named “CogSvcs” lab and navigate to this folder.

From this folder, launch Visual Studio Code the editor of your choice.

Create a file named “index.html”.

Add the following code to this index.html:

```
<html>
<head>
  <title>Emotions API Demo</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
  <script src="script.js"></script>
</head>
<body>
<h1>Microsoft Cognitive Services</h1>
<h2>Emotions API Lab</h2>
  <div>Image URL:</div>
    <input
      type="text"
      id="imageUrlTextbox"
      class="urlInput"
      value="https://photos.smugmug.com/Tech-Community/GANGConf-2017/i-
H2ZNBjt/0/c28d08df/L/IMG_7743-L.jpg">
    <button id="analyzeImageButton">Analyze Image </button>
    <div id="ResultsDiv"></div>
    <div id="AngerDiv"></div>
    <div id="ContemptDiv"></div>
    <div id="DisgustDiv"></div>
    <div id="FearDiv"></div>
    <div id="HappinessDiv"></div>
    <div id="NeutralDiv"></div>
    <div id="SadnessDiv"></div>
    <div id="SurpriseDiv"></div>

    <div>
      <img src="" id="imageToAnalyze">
    </div>
</body>
</html>
```

Create a file named "script.js"

Add the following code to script.js:

```
$(function () {

    $("#analyzeImageButton").click(function () {
        getImageInfo();
    });

    var getImageInfo = function () {
        var subscriptionKey = "dc6d7936f86b4577bbb9750c91233f74";
        var imageUrl = $("#imageUrlTextbox").val();
        var webSvcUrl =
            "https://westus.api.cognitive.microsoft.com/emotion/v1.0/recognize";
        var resultsDiv = $("#ResultsDiv");

        if (imageUrl) {
            var body = '{ "url": "' + imageUrl + '" }';

            $.ajax({
                type: "POST",
                url: webSvcUrl,
                headers: {"Ocp-Apim-Subscription-Key": subscriptionKey},
                contentType: "application/json",
                data: body
            }).done(function (data) {
                var scores = data[0].scores;
                var angerScore = scores.anger;
                var contemptScore = scores.contempt;
                var disgustScore = scores.disgust;
                var fearScore = scores.fear;
                var happinessScore = scores.happiness;
                var neutralScore = scores.neutral;
                var sadnessScore = scores.sadness;
                var surpriseScore = scores.surprise;

                var angerText = "Anger Score: " + angerScore.toFixed(2);
                var contemptText = "Contempt Score: " + contemptScore.toFixed(2);
                var disgustText = "Disgust Score: " + disgustScore.toFixed(2);
                var fearText = "Fear Score: " + fearScore.toFixed(2);
```

```

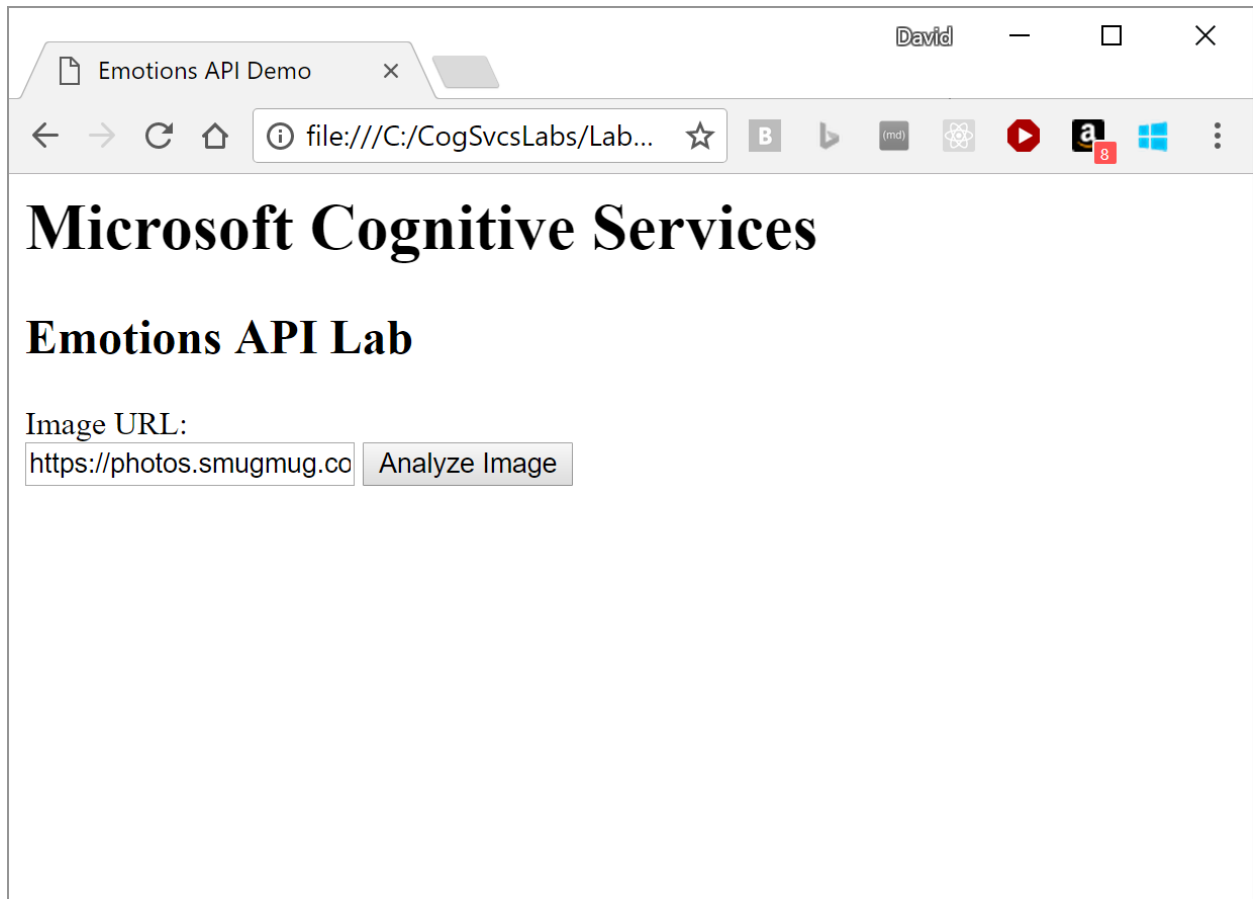
        var happinessText = "Happiness Score: " +
happinessScore.toFixed(2);
        var neutralText = "Neutral Score: " + neutralScore.toFixed(2);
        var sadnessText = "Sadness Score: " + sadnessScore.toFixed(2);
        var surpriseText = "Surprise Score: " + surpriseScore.toFixed(2);
        $("#AngerDiv").text(angerText);
        $("#ContemptDiv").text(contemptText);
        $("#DisgustDiv").text(disgustText);
        $("#FearDiv").text(fearText);
        $("#HappinessDiv").text(happinessText);
        $("#NeutralDiv").text(NeutralText);
        $("#SadnessDiv").text(sadnessText);
        $("#SurpriseDiv").text(surpriseText);

        $("#ResultsDiv").text("Success!");

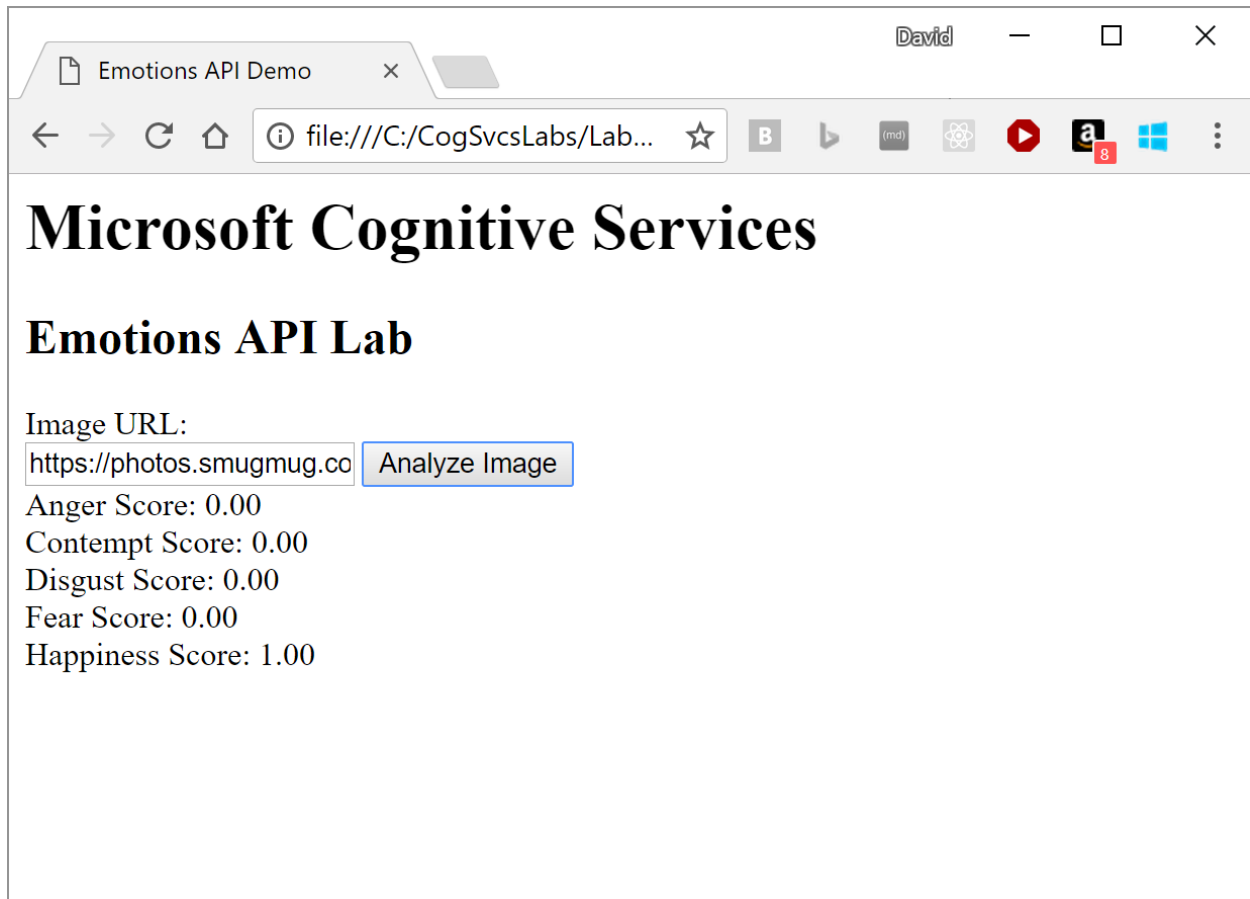
    }).fail(function (err) {
        $("#ResultsDiv").text("ERROR!" + err.responseText);
    });
}
};
});

```

Save both files. Open index.html in a web browser.



Click the [Analyze Image] button. You should see the scores displayed.



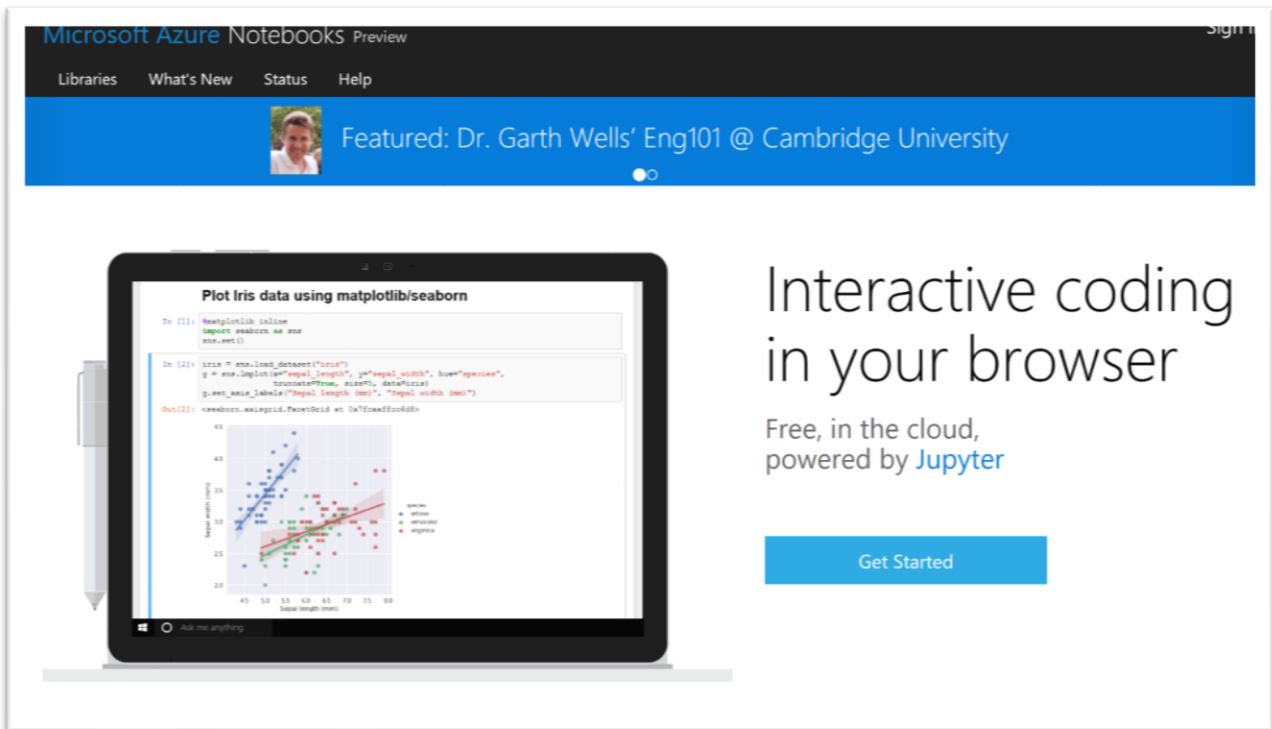
If errors are reported, use the browser's developer tools to step through and find these errors. Pressing F12 brings up these tools in most browsers.

Search the web for more photos of faces with varying emotions and paste these into the "Image URL" text box.

## Exercise 5: Calling a Web Service from Python

We can use the free Jupyter notebooks hosted on Azure to run Python code.

Navigate to <https://notebooks.azure.com/>



The screenshot shows the Microsoft Azure Notebooks website. The header includes the logo and navigation links: Libraries, What's New, Status, and Help. A featured section highlights Dr. Garth Wells' Eng101 @ Cambridge University notebook. Below this, a large image of a tablet displays a Jupyter notebook titled "Plot Iris data using matplotlib/seaborn". The notebook code includes importing libraries, loading the Iris dataset, and creating a faceted scatter plot. The plot shows three species of Iris flowers (setosa, versicolour, virginica) with their sepal length and petal length. To the right of the tablet, the text "Interactive coding in your browser" is displayed, followed by "Free, in the cloud, powered by Jupyter". A blue "Get Started" button is located at the bottom right.

Microsoft Azure Notebooks Preview

Libraries What's New Status Help

Featured: Dr. Garth Wells' Eng101 @ Cambridge University

Plot Iris data using matplotlib/seaborn

```
In [1]: %matplotlib inline
import seaborn as sns
sns.set()

In [2]: iris = sns.load_dataset("iris")
g = sns.lmplot("sepal_length", "petal_length", hue="species",
              truncate=False, ci=95, data=iris)
g.set_axis_labels("Sepal Length (cm)", "Petal Length (cm)")

Out[2]: <seaborn.axisgrid.FacetGrid at 0x7f0eef00dd0>
```

Interactive coding  
in your browser

Free, in the cloud,  
powered by Jupyter

Get Started

Click the [Get Started] button to display a *Sample Notebooks* library.

Microsoft Azure Notebooks Preview Sign In











Libraries What's New Status Help

## Sample notebooks

Microsoft > Libraries > samples

Run Share Clone Star (36) Preview Download

Search  Show hidden items

FILE NAME ▼	FILE TYPE	MODIFIED
 Azure Notebooks - Welcome.ipynb	Notebook	Nov 9, 2017
 Creation and Deployment of an Azure ML Web Service.ipynb	Notebook	Sep 26, 2017
 Discover Sentiments in Tweets.ipynb	Notebook	Sep 26, 2017
 FSharp for Azure Notebooks.ipynb	Notebook	Sep 26, 2017
 Getting to your Data in Azure Notebooks.ipynb	Notebook	Sep 26, 2017
 Introduction to Python.ipynb	Notebook	Sep 26, 2017
 Introduction to R.ipynb	Notebook	Sep 26, 2017
 oil_price.csv	CSV	Aug 29, 2017
 README.html	HTML	Aug 24, 2017
 shell.log	LOG	Aug 29, 2017

Showing 1 to 10 search results (1 hidden)

< 1 >

Click the “Clone” link to create a copy of this library.

You may be prompted to log in with a Microsoft account (e.g., and outlook.com, live.com, or Hotmail.com login).

After you sign in, the “Clone Library” dialog displays..

# Clone Library

Library Name

Cognitive Services Labs

Library ID

https://notebooks.azure.com/anon-obet5q/libraries/

cslabs

☒ Public library

**Existing Clones:**

You've already created a clone of this library. You can open your existing clone by clicking on the clone below, or click 'Clone' to create a new clone of this library.

[Sample notebooks \(samples\)](#)

Clone

Cancel

Enter a descriptive name for the library and the url. The URL can contain only letters, numbers, and hyphens.

Click the [Clone] button to create a new library identical to the sample library.



Microsoft Azure Notebooks Preview

anon-obet5q [Configure User ID](#)

Libraries What's New Status Help

# Cognitive Services Labs

Cloned from [Microsoft/samples](#)

anon-obet5q > Libraries > cslabs

Run + New Settings Share Clone 0 Clones Star (0) Terminal Shutdown Preview Edit File Download Delete

Search [Show hidden items](#)

FILE NAME	FILE TYPE	MODIFIED
<input type="checkbox"/> Azure Notebooks - Welcome.ipynb	Notebook	Jan 25, 2018
<input type="checkbox"/> Creation and Deployment of an Azure ML Web Service.ipynb	Notebook	Jan 25, 2018
<input type="checkbox"/> Discover Sentiments in Tweets.ipynb	Notebook	Jan 25, 2018
<input type="checkbox"/> FSharp for Azure Notebooks.ipynb	Notebook	Jan 25, 2018
<input type="checkbox"/> Getting to your Data in Azure Notebooks.ipynb	Notebook	Jan 25, 2018
<input type="checkbox"/> Introduction to Python.ipynb	Notebook	Jan 25, 2018
<input type="checkbox"/> Introduction to R.ipynb	Notebook	Jan 25, 2018
<input type="checkbox"/> oil_price.csv	CSV	Jan 25, 2018
<input type="checkbox"/> README.html	HTML	Jan 25, 2018
<input type="checkbox"/> shell.log	LOG	Jan 25, 2018

Showing 1 to 10 search results (1 hidden)

< 1 >

Click the [+ New] link to open the “Add Items to Library” dialog.

## Add Items to Library

?

New From computer From URL

Item Name

Emotion API demo.ipynb

Item type Python 3.6 Notebook

New Cancel

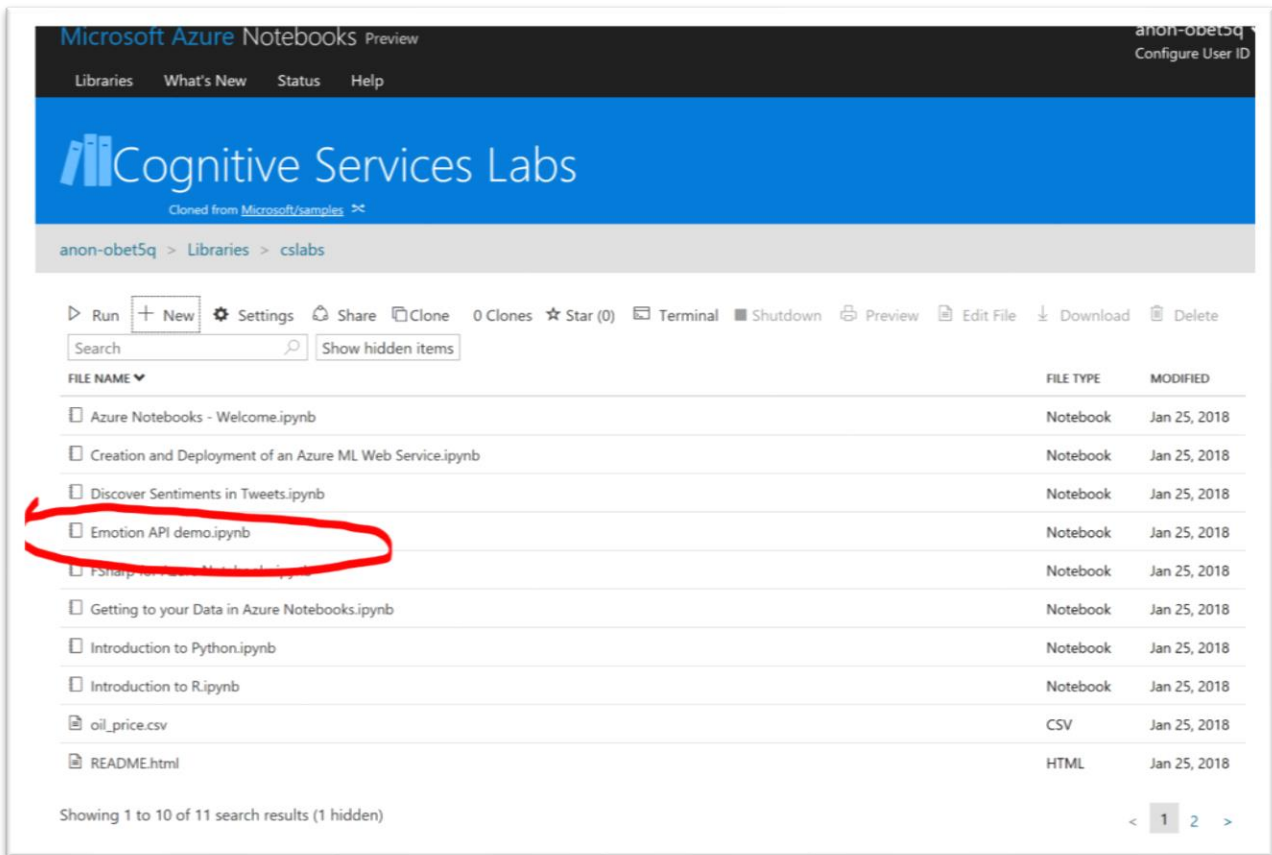
At the “Item Name” textbox, enter “Emotion API demo”.

From the “Item type” dropdown, select “Python 3.6 Notebook”.

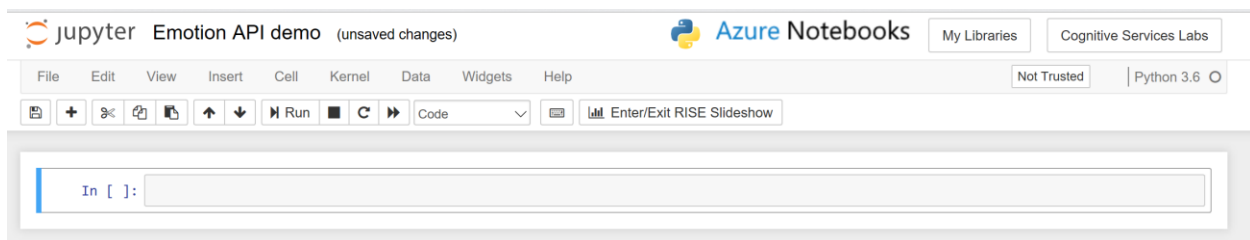
Click the [New] button to create a new Jupyter notebook and add it to this library.

The newly-created notebook will now be listed in the library.

nb



Click the “Emotion API demo.ipynb” link to open this notebook. The notebook will be empty except for an empty code box.



Paste the following code into the code box:

```
import http.client, urllib.request, urllib.parse, urllib.error, base64
headers = {
    # Request headers
    'Content-Type': 'application/json',
```

```

'Ocp-Apim-Subscription-Key': 'EmotionAPIKey',
}

params = urllib.parse.urlencode({

try:
    conn = http.client.HTTPSConnection('westus.api.cognitive.microsoft.com')
    conn.request("POST", "/emotion/v1.0/recognize?%s" % params, "{url':
'https://photos.smugmug.com/Tech-Community/GANGConf-2017/i-H2ZNBjt/0/c28d08df/L/IMG_7743-
L.jpg'}", headers)
    response = conn.getresponse()
    data = response.read()
    print(data)
    conn.close()
except Exception as e:
    print("[Errno {0}] {1}".format(e.errno, e.strerror))

```

Replace *EmotionAPIKey* with your Emotion API key created in Exercise 1.

Click the [Run] button on the toolbar.

If everything is correct, you should see JSON returned from the Emotion API describing the image in the URL.

Jupyter Emotion API demo (unsaved changes) Azure Notebooks My Libraries Cognitive Services Labs

File Edit View Insert Cell Kernel Data Widgets Help Not Trusted Python 3.6

Enter/Exit RISE Slideshow

```
In [1]: import http.client, urllib.request, urllib.parse, urllib.error, base64

headers = {
    # Request headers
    'Content-Type': 'application/json',
    'Ocp-Apim-Subscription-Key': 'e0a98793af8c41aebeca20f9a0384a0c',
}

params = urllib.parse.urlencode({})

try:
    conn = http.client.HTTPSConnection('westus.api.cognitive.microsoft.com')
    conn.request("POST", "/emotion/v1.0/recognize?%s" % params, '{"url": "https://photos.smugmug.com/Tech-Community/GANGConf-2017'}
    response = conn.getresponse()
    data = response.read()
    print(data)
    conn.close()
except Exception as e:
    print("[Errno {0}] {1}".format(e.errno, e.strerror))

b'[{"faceRectangle":{"height":54,"left":155,"top":66,"width":54},"scores":{"anger":1.733172E-06,"contempt":4.96382427E-06,"disgust":3.873715E-05,"fear":1.938528E-08,"happiness":0.9997759,"neutral":0.00016402037,"sadness":1.41424E-05,"surprise":4.996498E-07}},"faceRectangle":{"height":52,"left":258,"top":91,"width":52},"scores":{"anger":8.05321E-10,"contempt":4.27246576E-08,"disgust":9.711659E-09,"fear":1.852482E-12,"happiness":0.9999972,"neutral":2.74413537E-06,"sadness":2.55643479E-10,"surprise":2.38442444E-09}},"faceRectangle":{"height":50,"left":407,"top":148,"width":50},"scores":{"anger":0.00115220936,"contempt":0.00464260252,"disgust":0.0006162417,"fear":1.78585869E-05,"happiness":0.007853591,"neutral":0.9761079,"sadness":0.009153373,"surprise":0.0004562266}},"faceRectangle":{"height":48,"left":333,"top":120,"width":48},"scores":{"anger":8.192019E-07,"contempt":8.954231E-08,"disgust":1.56902024E-05,"fear":1.81517468E-10,"happiness":0.999976337,"neutral":6.12861731E-06,"sadness":9.080684E-07,"surprise":2.46937226E-08}},"faceRectangle":{"height":45,"left":211,"top":188,"width":45},"scores":{"anger":7.4214006E-07,"contempt":2.12904965E-08,"disgust":2.58160435E-05,"fear":1.809541E-10,"happiness":0.999973,"neutral":2.53957154E-07,"sadness":7.843519E-09,"surprise":1.346782E-07}}]'
```

In [ ]:

If an error is reported, review your code and correct the error.