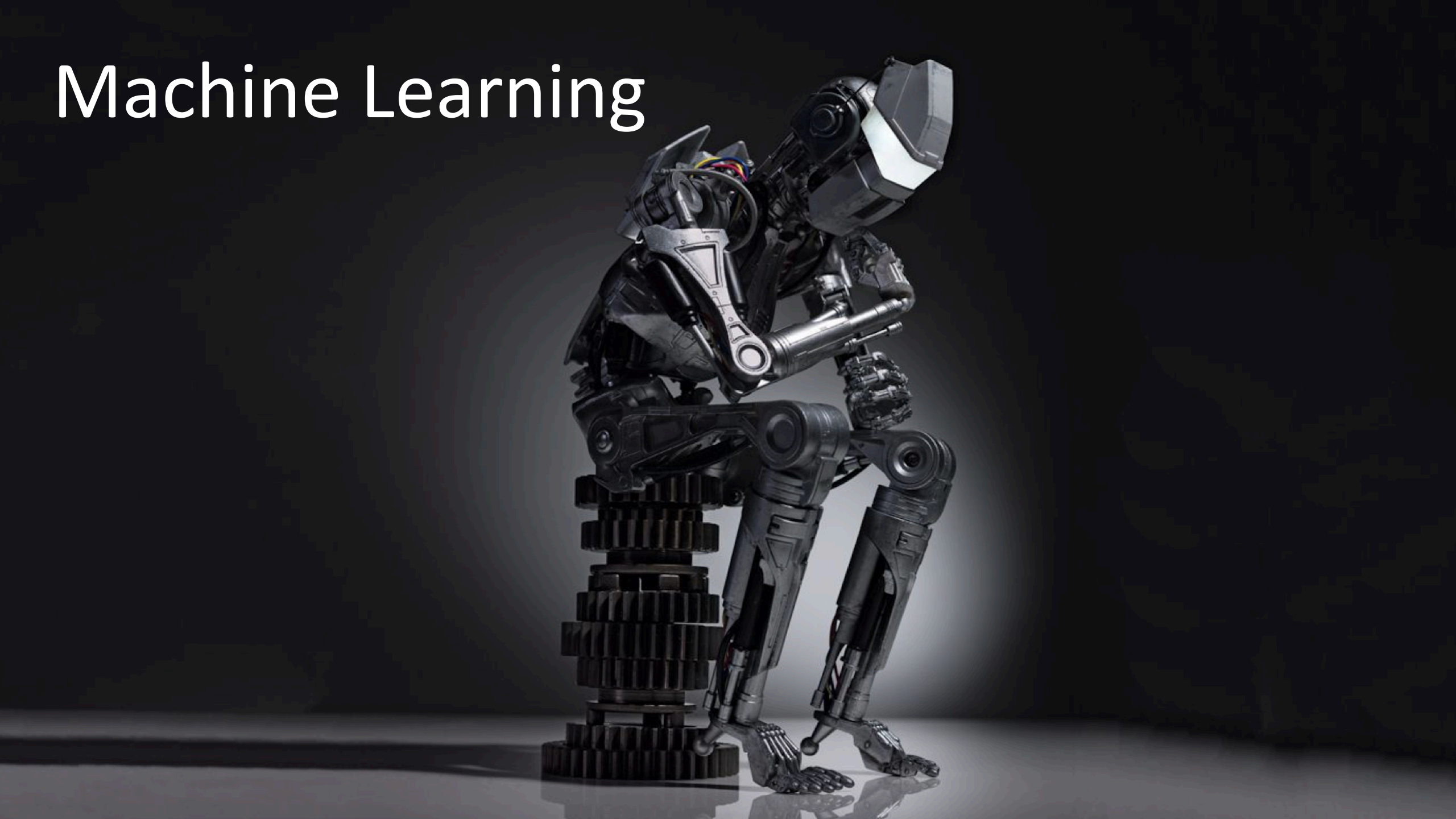
Machine Learning



Overview

- Machine Learning APIs
- Using REST APIs
  - API Hub
  - Using a REST
- Using in Code
  - First application in JS
  - Java
  - Python
- In Production
  - How to authenticate.
  - Deployment of a simple application.

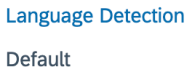
SAP API Hub

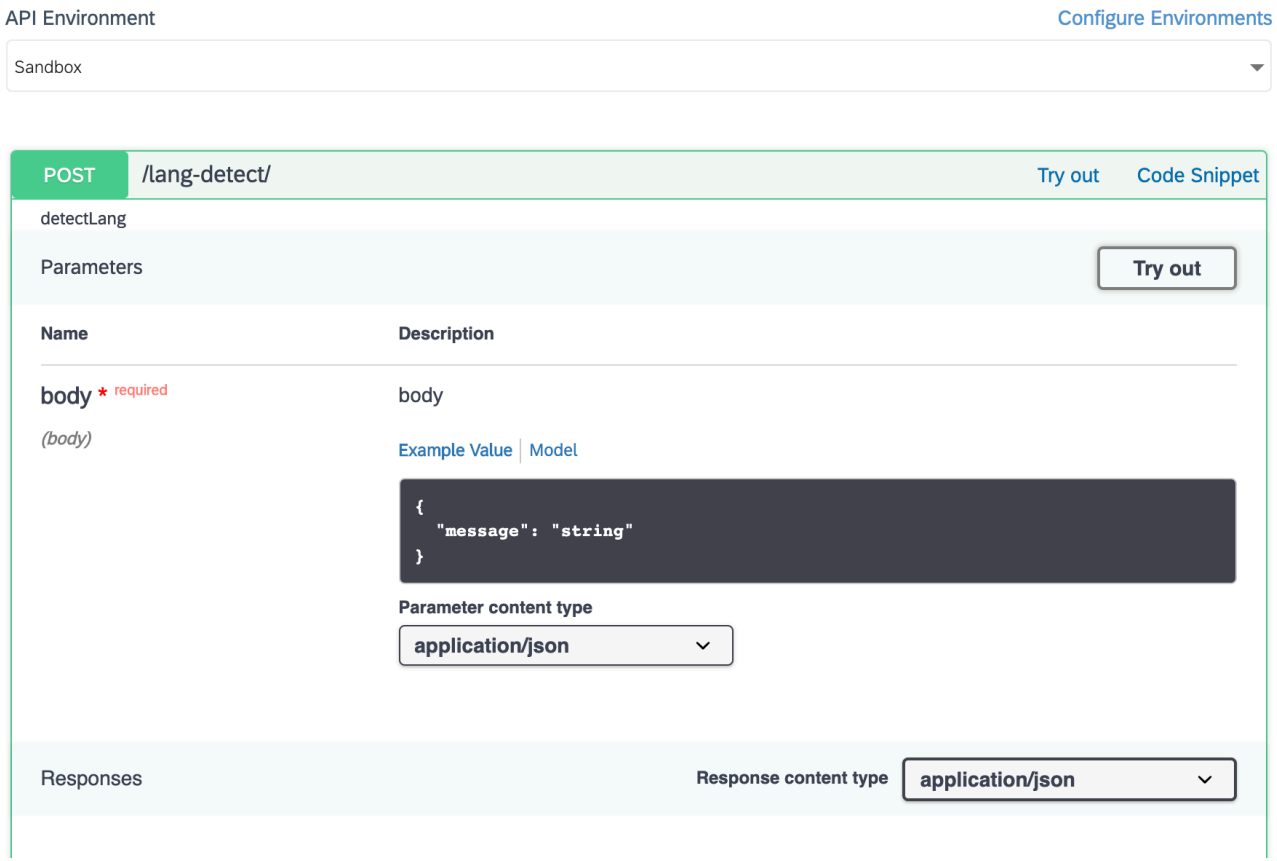
- Functional APIs are typically classification
  - Text
  - Images
  - (some exceptions)
- Business
  - Ticket
  - Finance
  - Solution
- Operation/customization
  - Customizable models or new models.

Excercise: API Hub

- Pick an API
- Try out
- Enter parameters
- Hit execute

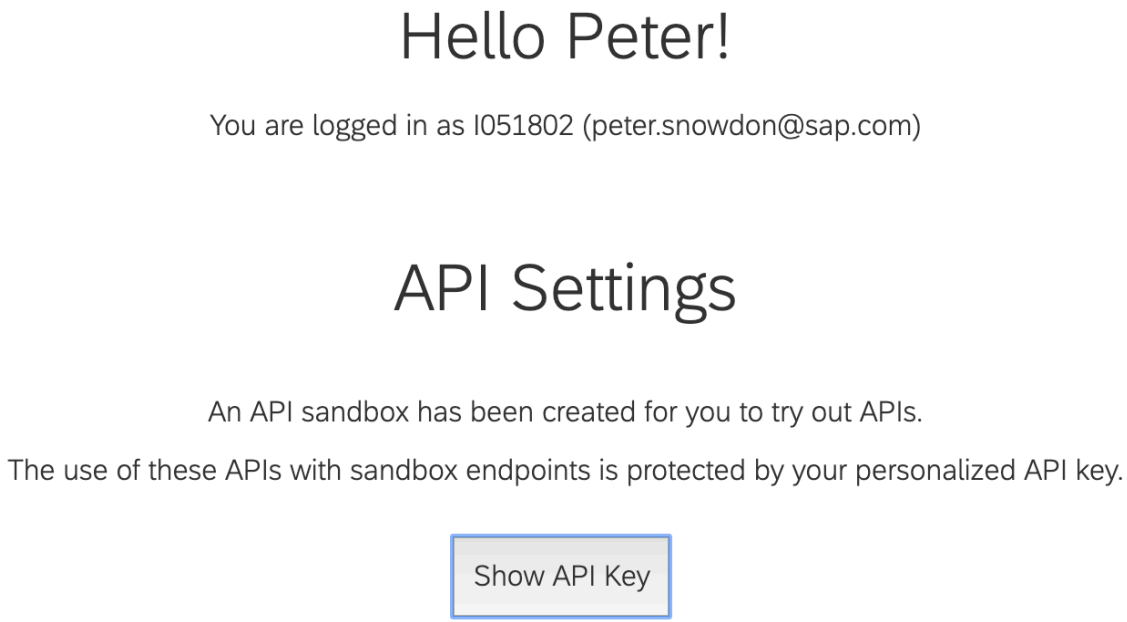
Note that parameters are typically JSON or Files (sometimes both).





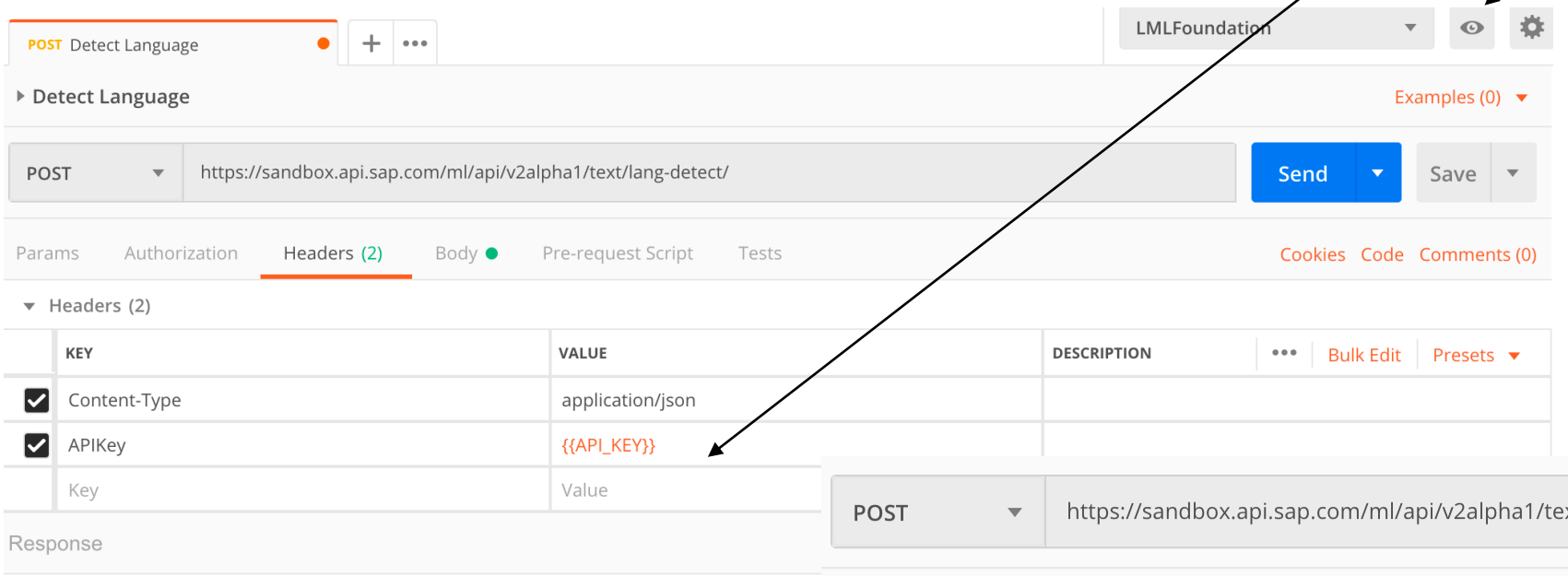
API Hub Preference

You API key can be seen in the code examples or under preferences.

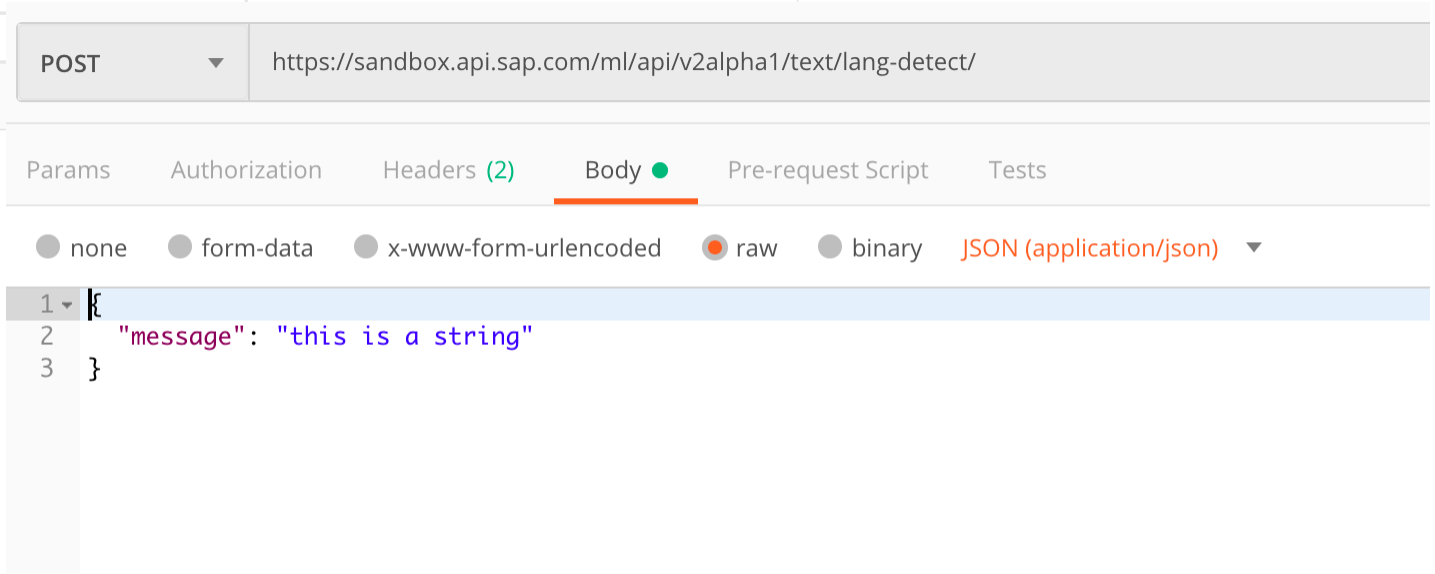


Excercise: Postman

Note the environment



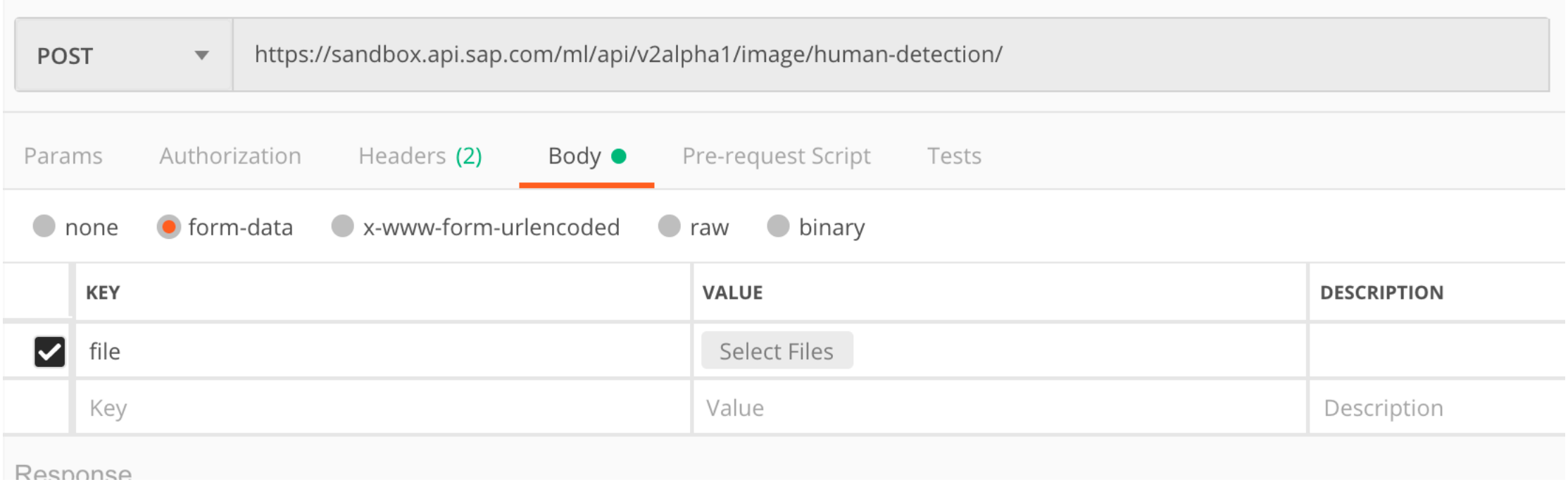
- Enter the API URL
- Enter the required headers
- Put the message as body
- Ensure you have the right method
- Hit send



Collection available in git:
https://github.com/drefter/ml-hackathon

Optional: Other Services

- You can try other services including image based services. You just need to read the doc and ensure you set the right settings:



Excercise: In Code (javascript)

The basics:

- Make a POST call
- Ensure that you have a means of authenticating the call
- Send the right message.

```
/*
url: is the end point of the call
data: is the message to send (json assumed)
callback: to get the data backout
*/
function callLML(url, oData, callback){
  request.post({
        url: url,
        body: JSON.stringify(oData),
        headers: {
            "Content-Type": "application/json",
            "Accept": "application/json",
            "APIKey": APIKey
        }
    }, function optionalCallback(err, httpResponse, body) {
        if (err) {
            return console.error('upload failed:', err);
        }
        callback(body);
  });
}
```

Git: https://github.com/drefter/ml-hackathon

Code | Issues 0 | Pull requests 0 | Projects 0 | Insights | Settings

Branch: master ▾ **ml-hackathon** / code-samples / node / **langauge** /

Create new file | Upload files | Find file | History

 **drefter** oops — Latest commit bed46d4 4 minutes ago

..		
.env.example	oops	4 minutes ago
index.html	updating env and turning detect to index	8 minutes ago
jquery.js	adding some files	7 days ago
language-mapping.csv	updating everything that is relevent	18 hours ago
manifest.yaml	updating everything that is relevent	18 hours ago
package.json	updating everything that is relevent	18 hours ago
server.js	oops	4 minutes ago

Setup

- If you have git you can clone the repo or just download a zip.
- Install node: https://nodejs.org/en/download/
- Check npm: https://www.npmjs.com/get-npm
- Install required packages:

```
/Users/i051802/development/git/ml-hackathon/code-samples/node/langauge
(base) C02YG2FHJHD3:langauge i051802$ npm install
```

.env.example to .env

- Note that for the environment you will need to add some properties notable the APIKey above

```
PORT=1666
APIKey="xxxxxxxxxxxxxxxxx"
clientID="xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-"
clientSecret="xxxxxxx-xxxxxxx-"
authenticationURL="https://you.domain.url/oauth/token?grant_type=client_credentials"
```

Launch the server.

- **npm install**: installs required 3rd parties

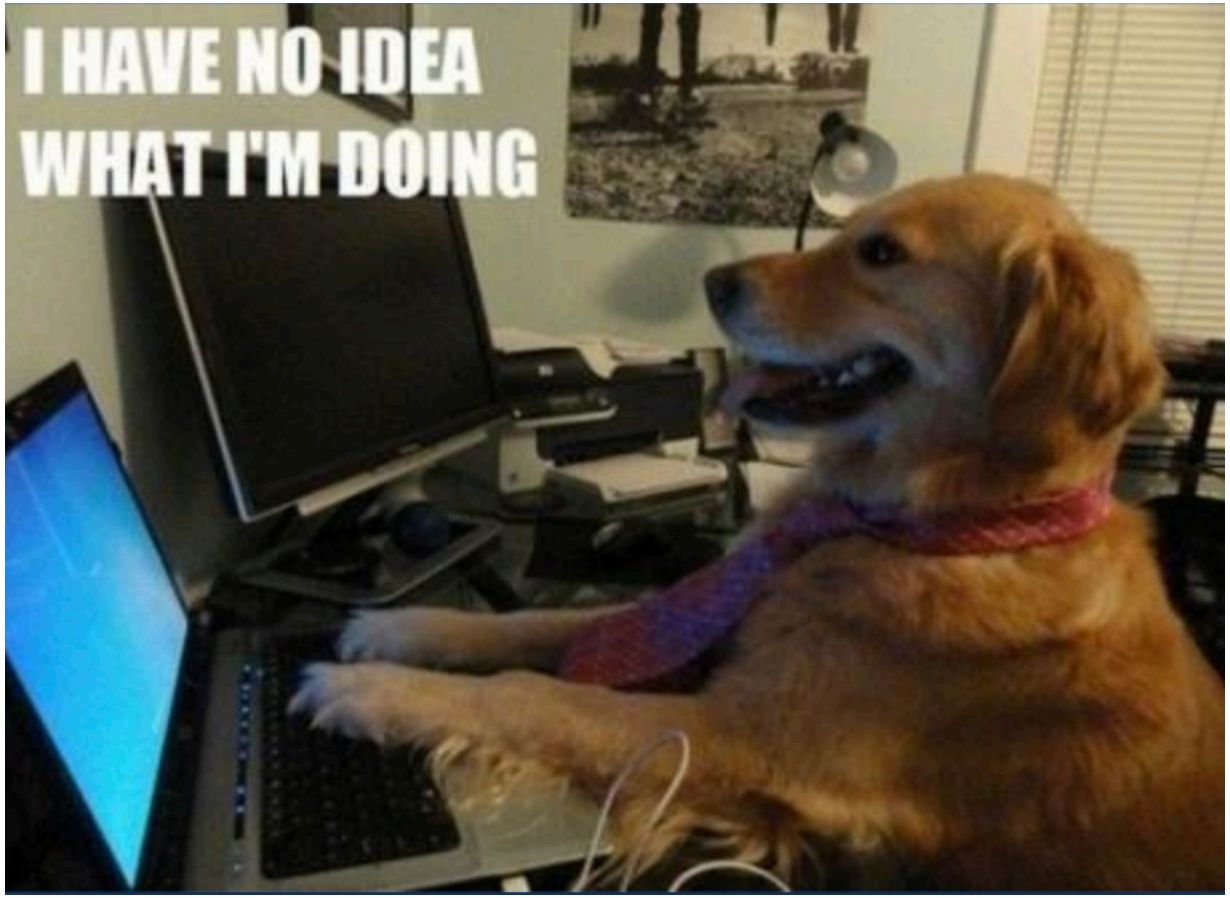
- **node server.js** (Or: **npm start**)
  - Application runs on the port that you assigned in .env

```
(base) C02YG2FHJHD3:ml-hackathon i051802$ cd code-samples/node/langauge/
(base) C02YG2FHJHD3:langauge i051802$ npm start

> language-detection@1.0.0 start /Users/i051802/development/git/ml-hackathon/code-samples/node/langauge
> node server.js

Started dashboard dev server on port 1666
```

The sample contains your challenge

If this isn’t you then there are other code examples in the repo:

- Image comparison
- Text detection.



Example: java

```
String url = "https://sandbox.api.sap.com/mlfs/api/v2/text/translation";

CloseableHttpClient httpClient = HttpClients.createDefault();
HttpPost uploadFile = new HttpPost(url);
uploadFile.addHeader("Accept", "application/json");
uploadFile.addHeader("APIKey", apiKey);

String data = createTranslationJSON("Can you translate this please?").toString();

StringEntity requestEntity = new StringEntity(data, ContentType.APPLICATION_JSON);
uploadFile.setEntity(requestEntity);


CloseableHttpResponse response = httpClient.execute(uploadFile);
HttpEntity responseEntity = response.getEntity();

JSONTokener tokener = new JSONTokener(responseEntity.getContent());
JSONObject results = new JSONObject(tokener);
```

Example: python

```python
import requests
import sys

if len(sys.argv) < 2:
    print("please pass in your apikey")
    exit();

# defining the api-endpoint
API_ENDPOINT = "https://sandbox.api.sap.com/ml/api/v2alpha1/text/lang-detect/"

# your API key here
APIKey = sys.argv[1];

# data to be sent to api
data = '{"message": "this is a string"}'

headers = {'APIKey': APIKey, "Content-Type" : "application/json"}

# sending post request and saving response as response object
r = requests.post(url = API_ENDPOINT, data = data, headers=headers)

# extracting response text
response = r.text
print("The response is:%s"%response)
```

Using Production Accounts

- API hub gives you trial access to a subset of the LML APIs.
- To access everything you need a production account.
- And also API authentication.

Authentication with Production

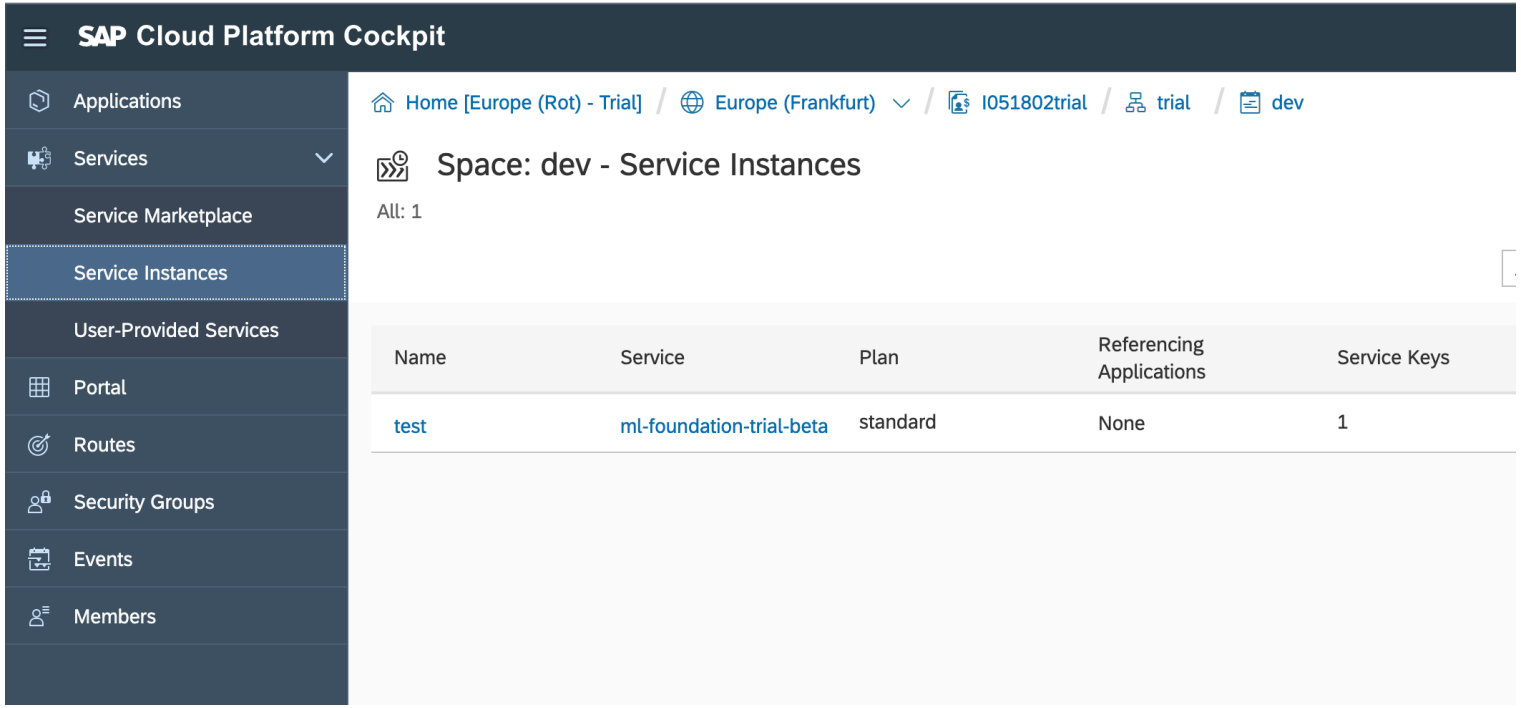
- With a production account you will need to authenticate using Oauth.
- You will need 3 values that can be found in your account.
  - clientID
  - clientSecret
  - authenticationURL

```
PORT=1666
APIKey="xxxxxxxxxxxxxxxxx"
clientID="xxxxxxx-xxxxxxx-xxxxxxx-xxxxxxx-"
clientSecret="xxxxxxx-xxxxxxx-"
authenticationURL="https://you.domain.url/oauth/token?grant_type=client_credentials"
```

CF Account

In your CF account you will either have or need to create a service instance.

Under that instance you will find your service variables like authentication and endpoints.



Cloud Foundry Deployment.

Returning to our Node example its possible to deploy to CF for example by creating a yaml file.

```yaml
---
applications:

- name: language-detection
  host: language-detection
  buildpack: nodejs_buildpack
  memory: 512M
  command: node server.js
  instances: 1
  random-route: true

```

- cf login account
  - User/pass
- Select your space
- cf push

Hackathon Guidelines

- Theme: **Intelligent Enterprise**
- Scoring Guide:
  - Staying on topic.
  - Technical Merit.
  - Conceptual Brilliance.

- Note:
  - No presentations
  - ‘Working’ application
  - Using an ML API
  - More than one!!

Strictly: 6 minutes per team.