

Use the IBM Image Recognition API 🕒

In this article, we will show how to use the Image Recognition API from IBM to identify an image.

This API takes an image (JPG, PNG, GIF, etc) and returns a JSON file explaining the result of the machine learning algorithm trying to recognize it.

Prerequisites 🕒

Before we begin, make sure you have done the following:

Download and unzip the photo from the link below, and place it at the root of your project folder.

Clownfish photo

Read the IBM tutorial and obtain your own API Key from the link below. (The CURL section is not relevant for this case.)

IBM APIs: Visual Recognition Tutorial

Install the AIMMS HTTP Client Library according to AIMMS Documentation: Adding the HTTP Client Library.

Basics of IBM Cloud APIs 🕒

You can find information about this API in IBM Cloud API Docs: Visual Recognition.

The Authentication system we'll be using is a simple API key. You can use a GET request using the URL of a photo, or a POST request to send the document to analyze to the server. (Here, we'll be using the POST request, explained in IBM docs in section Methods > Classify images.)

There are 3 kinds of parameters we can set for this method.

Headers:

`../../_images/customparam.png`

We'll set the language for the answer file to en for English.

Query parameters:

`../../_images/queryparam.png`

We'll specify the version of the API we want to use, in this case 2018-03-19.

Form parameters:

`../../_images/formparam.png`

We'll be using the `images_file` parameters that is used to set the photo we want to analyze. We'll also set the parameters `threshold` and `Classifiers_ids` as an example.

The IBM documentation gives examples using CURL, a client for formulating HTTP requests. Here we'll find out how to translate the CURL requests into AIMMS and set the parameters.

Example project🕒

You can download the example project here: AIMMS project

We'll first show how to get the JSON file.

Then we will talk about how to specify the following:

Query parameters

Form parameters

Request headers

Getting the JSON file🕒

The final code will be:

```
SP_requestFileName:="clownfish.JPG";
```

```
SP_responseFileName := "Answer.json";
```

```
SP_apikey:="YOUR_API_KEY";
```

```
SP_Requestparam:={
```

```
    'version' : "2018-03-19",
```

```
    'threshold' : "0.6",
```

```
    'classifier_ids' : "default"
```

```
};
```

```
web::query_format(SP_Requestparam,SP_formattedRequestParam);
```

!endpoint given on the IBMCloud documentation

```
SP_requestURI := "https://gateway.watsonplatform.net/visual-  
recognition/api/v3/classify?"+SP_formattedRequestParam;
```

!initialize request

```
web::request_create(SP_requestId);
```

!Set Headers and add a new one.

```
web::HttpHeaders:= DATA{ Accept, Accept-Encoding, Authorization, Cache-Control, Content-Length,  
Content-Type, Transfer-Encoding, Location ,Accept-Language};
```

```
web::request_getHeaders(SP_requestId, SP_myHttpHeaders);
```

```
SP_myHttpHeaders[ 'Accept-Language' ] := "en";
```

!Authentication

```
web::base64_encode( "apikey" + ":" + SP_apikey, SP_authorization);
```

```
SP_myHttpHeaders[ 'Authorization' ] := "Basic " + SP_authorization;
```

```
web::request_setHeaders(SP_requestId, SP_myHttpHeaders);
```

!Set request

```
web::request_setURL(SP_requestId, SP_requestURI);
```

```
web::request_setMethod(SP_requestId, "POST");
```

```
web::request_setRequestBody(SP_requestId, 'File', SP_requestFileName);
```

```
web::request_setResponseBody(SP_requestId, 'File', SP_responseFileName);
```

```
web::request_getOptions(SP_requestId,SP_requestOption);
```

```
SP_requestOption['requestTimeout'] := "30";
```

```
web::request_setOptions(SP_requestId, SP_requestOption);
```

```
web::request_invoke(SP_requestId, P_responseCode);
```

Additionally, you'll need these identifiers:

```
StringParameter SP_requestFileName;

StringParameter SP_responseFileName;

StringParameter SP_requestURI;

Parameter P_responseCode;

StringParameter SP_requestId;

StringParameter SP_requestOption {
    IndexDomain: web::co;
}

StringParameter SP_apikey;

StringParameter SP_myHttpHeaders {
    IndexDomain: web::httpHeader;
}

StringParameter SP_authorization;

Set S_Param {
    Index: i_p;

    InitialData: DATA{version,threshold,classifier_ids};
}

StringParameter SP_Requestparam {
    IndexDomain: i_p;
}

StringParameter SP_formattedRequestParam;
```