Azure to Watson

Tool to convert an app from Microsoft Azure APIs to Watson APIs

Mentored by - Saif Abrar <saif.abrar@in.ibm.com>,

Hitesh Sharma <hksharma@ddn.upes.ac.in>

Team Members -

Nikunj Kumar Gupta, Ajay Kumar, Ankita Mittal, Anshul Mittal

IBM Summer Internship

INTRODUCTION

Watson is a set of Artificial Intelligence APIs provided by IBM. This project's Objective is to convert an application that is using the Microsoft Azure APIs to IBM Watson APIs, so that an application can easily migrate from Azure APIs to Watson APIs.

PROBLEM STATEMENT

Every Artificial Intelligence API has its own advantages and disadvantages, so as the Microsoft Azure API. If an application is currently using the Microsoft Azure APIs, and now wants to use the IBM Watson APIs, the conversion took huge amount of time and many changes were to be made in the code. This problem of changing the code which was time consuming, sometimes even led to rewriting the whole code.

SOLUTION

The solution is to use a Middleware Server that does the whole conversion part. The application using the Microsoft Azure APIs just need to redirect their request to the middleware server that deals with the conversion and returns the response to the application.

The middleware server gets the request from the application which was using the Microsoft Azure API and retrieves the data from the request. the data is then sent to Watson as a request and then the response from Watson is converted to the response format of Microsoft Azure. After converting the response it is sent back to the requesting application, therefore, converting it from Microsoft Azure to IBM Watson API.

TECHNOLOGIES USED

1. JAVAEE to create the server.
2. Tomcat 8 to deploy the application.
3. Watson Java SDK.
4. JSON parser for java.

RESULTS

The Solution works well for converting any Microsoft Azure APIs application to Watson APIs application, where the application is using simple HTTP request to use the APIs. Applications using the Microsoft Azure SDK will not be able to use this

solution.

Following is an example of conversion of Face Detect Azure API to Watson API -

*Only the code snippets are shown here, not the whole code

Face Detect - Azure API

```
    String uriBase

="https://westcentralus.api.cognitive.microsoft.com/face/v1.0/detect";
    public void getFace() throws Exception {
        HttpClient httpclient = new DefaultHttpClient();
        URIBuilder builder = new URIBuilder(uriBase);

        // Request parameters. All of them are optional.
        builder.setParameter("returnFaceId", "true");
        builder.setParameter("returnFaceLandmarks", "false");
        builder.setParameter("returnFaceAttributes", faceAttributes);

        // Prepare the URI for the REST API call.
        URI uri = builder.build();
        HttpPost request = new HttpPost(uri);

        // Request headers.
        request.setHeader("Content-Type", "application/json");
        request.setHeader("Ocp-Apim-Subscription-Key", subscriptionKey);

        // Request body.
        StringEntity reqEntity = new StringEntity(imageWithFaces);
        request.setEntity(reqEntity);

        // Execute the REST API call and get the response entity.
        HttpResponse response = httpclient.execute(request);
        HttpEntity entity = response.getEntity();

        // Format and display the JSON response.
        System.out.println("REST Response:\n");
        String jsonString = EntityUtils.toString(entity).trim();
        jsonArray = new JSONArray(jsonString);
    }
```

To convert the above application to Watson API application, we just need to change the value of the uriBase variable to "https://azuretowatson.herokuapp.com/detect_faces"

Output of Azure API -

Output of Watson API -

LEARNING IN THE PROJECT

1. REST API and how to use them.
2. JSON and how to read an write JSON data using java.
3. Create SSL certificate.
4. HTTP Requests and Response.

5. cURL and how to use it.

GOOD EXPERIENCE IN PROJECT

During the entire undertaking of X to watson (azure to watson) project we had encountered different things. As this was our first remote project on which we were working on, teached us a considerable measure.

Numerous things which we encountered are:

1. We experienced how to work with deadlines as we need to finish our sprints in a specific period.
2. We experienced new things like Watson APIs and Azure api's and learned a lot about these.
3. During the project we got the chance to figure out how to change Watson APIs to Azure APIs and the other way around.
4. This project likewise made us experienced how to work in group while working for a company.

FEEDBACK

The guidance of the IBM mentor was righteous and under his great supervision it was helpful for me and my team to complete our given tasks on time. Weekly conference calls were helpful enough to make the project successful.

At the beginning of the internship the conference calls were not that sufficient to provide us the necessary details of the project and to resolve our issues. There were a bit of delay in clearing our doubts which ultimately lead to delay of our initial sprints.

PROJECT IDEA WHILE WORKING WITH WATSON

If an application working on an API wants to migrate to another API or want different vendor for different AI API use. The migration is a tedious task and need huge amount of changes to be done in the code, which result in waste of time. This problem can be averted by using the following solution.

A server can be created that will allow users to use any API provider weather it be Watson, Azure, Google..etc. Although the response from these APIs are not the same, which can create a problem for the user. But, if we can create out own response format that can help user to use any kind of API.

TEAM MEMBERS

Name	Role
Nikunj Kumar Gupta	Conversion of Vision API
Ajay Kumar	Conversion of Language API
Ankita Mittal	Conversion of Speech to Text API
Anshul Mittal	Conversion of Text to Speech API