Summer Internship Report

Company Name : DBS Bank

Roll : Full Stack Application Developer

Project Name : API Field Mapper UI

By Aniket (18CS01046)

TABLE OF CONTENT

ABSTRACT

This report is the reflection and also the journey of my internship along with the highlights of what I learned through errors, work responsibilities, and the importance of internship program at DBS Bank. The knowledge I have gained as a new developer on web application development, new frameworks, and also how to work with team is splendid. As an intern, my work was to learn and focus on web application projects specially on front end part. In this report, I have focused my work territory and explain my project which I have got through my internship at DBS Bank.

ABBRIVATION USED

- API : Application User Interface
- App : Application
- DBS : Development Bank of Singapore
- JSON : Javascript Object Notation
- POJO : Plain Old Java Object
- XML : Extensible Markup Language
- XSD : XML Schema Definition
- WSDL : Web Service Description Language

INTRODUCTION

I got the internship in Development Bank of Singapore (DBS Bank). It was a 12 weeks program from 25th May, 2021 to 13th August, 2021. Based on my skill set my manager told me to work on front end.

Basically if we are developing a web application then there are two sides of web app **front end** also called client side and **back end** also called server side.

The frontend is the part of the website users can see and interact with such as the graphical user interface (GUI) and the command line including the design, navigating menus, texts, images, videos, etc. Backend, on the contrary, is the part of the website users cannot see and interact with.

The project I was working on named as API Field Mapper UI. It was a group project of three members including me. I was given a task to work on front end part using latest tech stack – **Angular, Bootstrap** and other two member were working on backend using **Java, Spring Boot, Apache cxf**.

ABOUT THE PROJECT

There are some important points about the project API Field Mapper that I like to highlight –

❖ As discussed earlier in the introduction part it was a group project of three members and I was working on front end using angular, bootstrap and other two member were working on backend using Java, spring-boot, apache cxf.

❖The main objective of the projects are –

- **Develop an API to reduce the reductant work for DBS developer**. At DBS there were some developers that used to convert json/xml/xsd into java classes or POJO by typing menually. To avoid this my team has developed a web API for DBS developer. Now by using the API with a single click of a button developers can achieve their goal easily.
- **Stop using alternatives to avoid data loss.** As discussed in above point we have developed an API, but there are some online API available one the internet which provides a part of full solution that we as intern, trying to find. As a Bank. DBS don't want to use external online tool to solve their problem because it might lead to data loss.

IMPORTANT TERM

In the last page there was a term called **POJO.** So what exactly POJO means.

POJO stands for **Plain Old Java Object.** It is simple java class. A POJO class is not defined by what property it should have but it is defined by what property it should not have.

A POJO should not:

1) Be a child class of another class
2) Inherit properties of another class
3) Bind with annotations. Eg :-

```
@javax.persistence.Entity public class Baz{
        ......
        ......
        ......
}
```

In the next Page we will be single a simple POJO class.

POJO EXAMPLE

Below is the example showing the POJO class for Employee

```
// Employee POJO class to represent entity Employee
public class Employee
{
    String name;
    public String id;
    private double salary;

    public Employee(String name, String id, double salary)
    {
        this.name = name;
        this.id = id;
        this.salary = salary;
    }
    public String getName()
    {
        return name;
    }

    public String getId()
    {
        return id;
    }
    public Double getSalary()
    {
        return salary;
    }
}
```

In POJO class constructor is optional.

What We Have Created?

Our team has developed a tool that is used by DBS developer. We presented this tool in the form of a full responsive web application.

There were mainly two feature our web application was providing –

1) Convert X into POJO
   In this feature we are taking input X from the user with some class name, package name and other parameters depending on type of X and sending it to the server. After performing business logic on the input, the output generated in the form of zip containing POJOs is available for the user to download it.
2) Convert JSON to XML
   In this feature user has to provide JSON data in the form of text in the text editor. Now the user can beautify (with proper formatting and text coloring) the JSON data with a single click on **<Beautify>** button and if user clicks on **<JSON to XML>** button than on another text editor box the formatted XML will be printed which comes from the server after converting input JSON into XML.

3) Convert XML to JSON

In this feature user has to provide XML data in the form of text in the text editor. Now the user can beautify (with proper formatting and text coloring) the XML text with a single click on **<Beautify>** button and if user clicks on **<XML to JSON>** button than on another text editor box the formatted JSON will be printed which comes from the server after converting input XML into JSON.

As a reader you might have noticed that I have mentioned X. Now I will highlight what X actually is.

X can be single file of type – JSON

XML

XSD

JSON SCHEMA

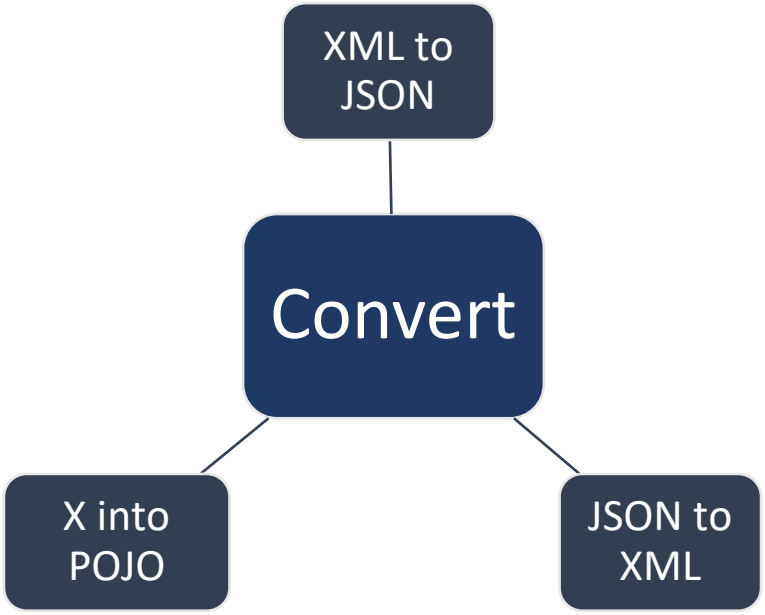
WSDL

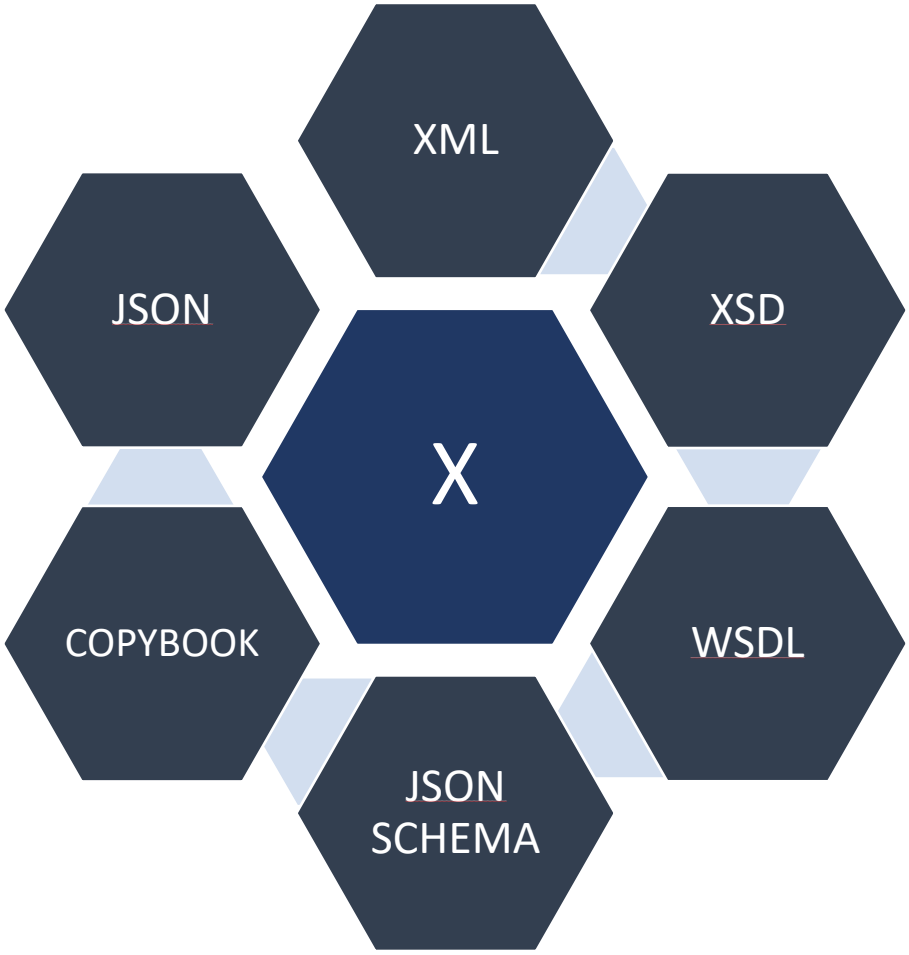
COPYBOOK

Or

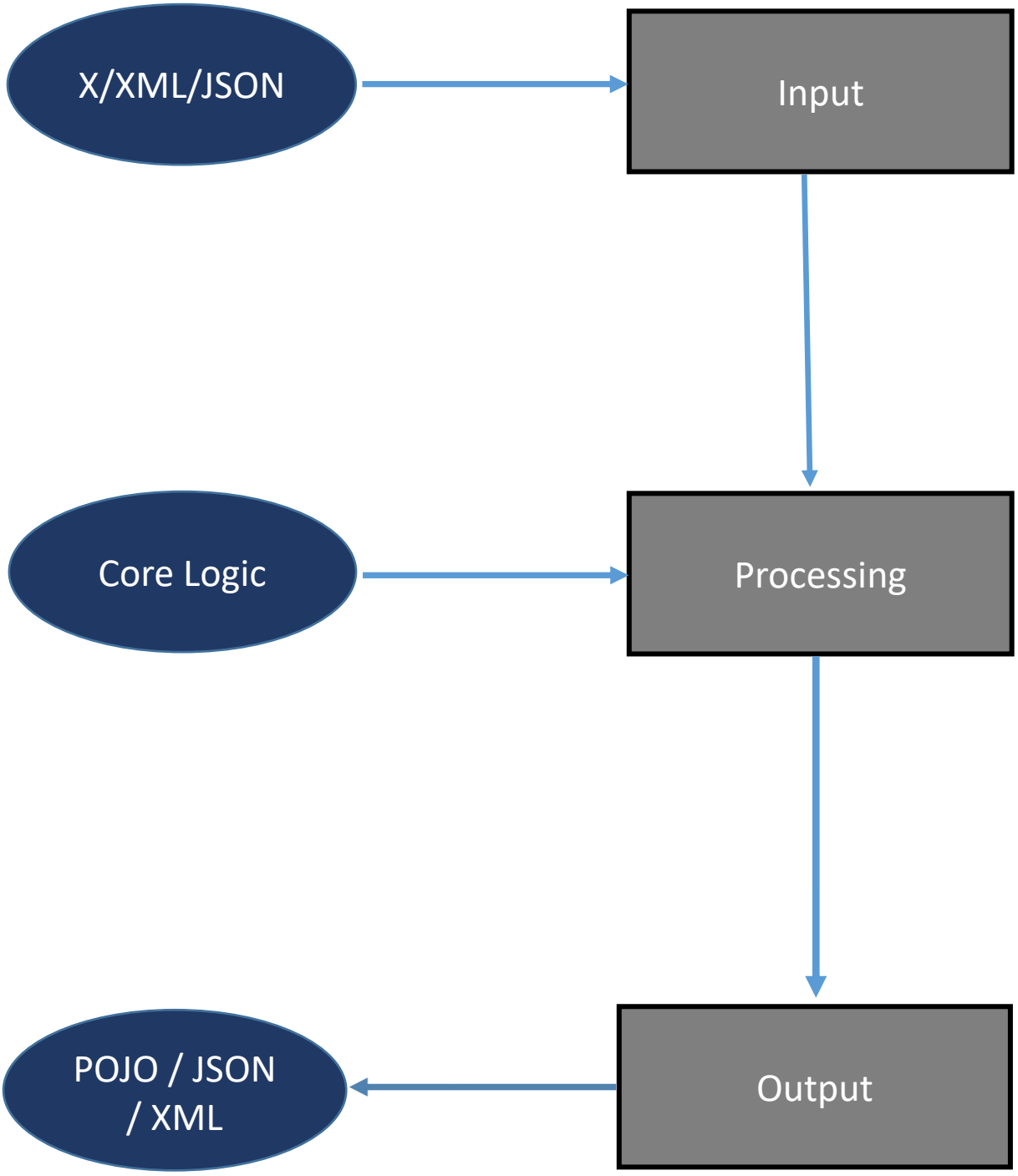
X can be zip file containing multiple files of same types

FEATURE WITH DIAGRAMS





WHATS'S HAPPINING



PROJECT DEMONSTRATION

Since DBS is a reputed international Bank, cannot make things public on the internet which by any means lead to the data lose or raise security issues.

The project or the web app that we have created is totally meant for the usage of DBS developers and it is not hosted publically on the internet. Therefor I don't have the codes and link for the web application.

FEATURE 1

Now I will try to explain more about the first feature of web app that is mentioned on the page number 8.

First thing is – user has to provide input X.

As I mentioned earlier X can be JSON, XML, XSD, JSON SCHEMA, WSDL, COPYBOOK or a zip file containing multiple files of the same type.

Let say X is JSON file and data inside file is shown below

```
{
    "employee": {
        "name":         "sonoo",
        "salary":       56000,
        "married":      true
    }
}
```

Now after clicking on upload button above JSON will be sent to server if no error found in the input JSON, if error found then an error message will be displayed on the screen saying provide JSON. In the server side above JSON will be converted into POJO and user will be able to see a download button. After clicking on download button a zip file will be automatically downloaded. POJO will be inside zip file.

The generated POJO for above JSON example will be –

```
public class Employee{
    public String name;
    public int salary;
    public boolean married;
}
```

With this if the user wants to convert multiple JSON files at once than he can give a zip file containing all the JSON files. Generated zip file will be containing POJOs for all the JSON files provided by the user.

Now let say X is XML file and data inside file is shown below –

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <mainheading>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </mainheading>
</note>
```

Generated POJO for above XML will be –

```java
public class note {
    public class mainheading {
    public String to;
    public String from;
    public String heading;
    public String body;
    }
}
```

Similarly, In the same manner for other types of input file their POJOs will be generated.

FEATURE 2

Now let's discuss the JSON to XML conversion

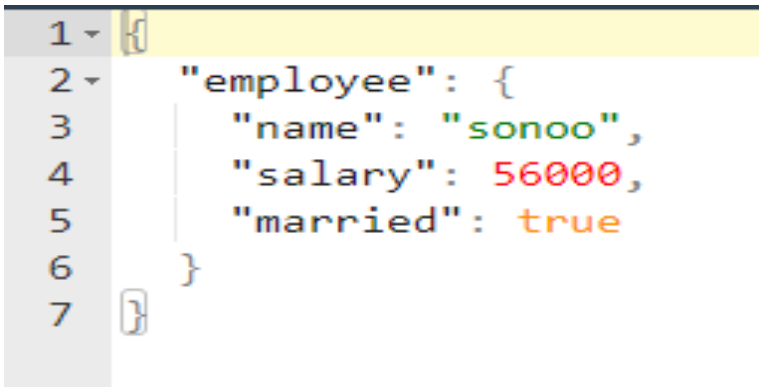
In this feature user has to provide JSON data in the form of text in the text editor.

Suppose he provides the below JSON text –

{ "employee": { "name": "sonoo", "salary": 56000, "married": true } }

Since the above JSON is not formatted and user wants to clearly see what are the key value pair then he can click on **<beautify>** button.

If user clicks on **<beautify>** button then above JSON will looks like below formatted JSON –



And after clicking on **<JSON to XML>** button the generated XML for above JSON example will be –

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<employee>
    <name>sonoo</name>
    <salary>56000</salary>
    <married>true</married>
</employee>
```

FEATURE 3

Feature three is the reverse of feature two.

Here we are converting input XML into JSON. By following the all steps mentioned in feature 2 we can convert XML into JSON.

INSIGHTS GATHERED FROM INTERNSHIP

As an intern and part of 12 weeks journey with DBS, was really amazing and insightful. Being a beginner it was difficult to cope up with such a big reputed firms. But DBS people and other interns supported me throughout the journey. When any of our teammates was facing any issues and not able to get the solution than as a group we all look into the problem and collectively try to find the solution.

Before this internship I was not familiar with the Angular, But at DBS and with support of my mentor I got the opportunity to learn Angular and implement it on the live project during whole time in internship.

One big thing I got to learn in this internship is how the people works on group in a single project, i.e., how to use GitHub or Bit bucket for big projects.

Really it was great experience of my life and I am very thankful to DBS that they gave me this opportunity.

REFERENCES

- Angular
- JSON to POJO
- XML to POJO
- JSON to XML
- XML to JSON
- Bootstrap 5
- POJO
- Fontawesome to insert icon in web application.
- Icons8 to insert icon in web application.