SERVICE-ORIENTED SOFTWARE ENGINEERING

Programs Directory PD

Composites Service

Team

Mohammad Tomaizy

Amr Slimi

Diana Alhafi

Abstract

In our project, we design a composite service based on web service that is aimed for Tawjihi students; as we have many Universities in our country, students who have finished Tawjihi need to know about universities and their admission Requirements for the academic programs such as the minimum Tawjihi average for admission for specific academic program.

1- Introduction

In our project, we introduce a composite service to address this problem, to define a new solution for students and to reduce time and effort for admission in any academic program in any university. our composite service follow the request from student to many services to get the final result, student use his Tawjihi seat number to request this service, first, we need to connect to the Ministry of Education to get information about the student, such as Tawjihi grades and the section based on seat number of student. Second, connect to each university and based on these information's from Ministry of Education to collect data about this student such as available academic programs accepted. Third, the result about academic programs accepted for the student displayed in readable form. Forth, The students can get other information about each accepted academic programs, such as how many students pay per credit hour for specific academic program, plan hours, admission period, and how to apply for any accepted academic program.

2- Motivations

Our service provides the knowledge in all university programs suitable for the tawjihi student's grade. Rather than personally going to every university, this service helps them save time and effort, also provides them the guidance when they pick the specialty that they like, so they could look for a brief summary describing it, which gives them the confidence in the university providing this service, as it shows a great care and concern of each student. On the other hand, this service lessen the pressure on the university faculty and attracts more students to the university.

3- Implementation

Sequence Diagram

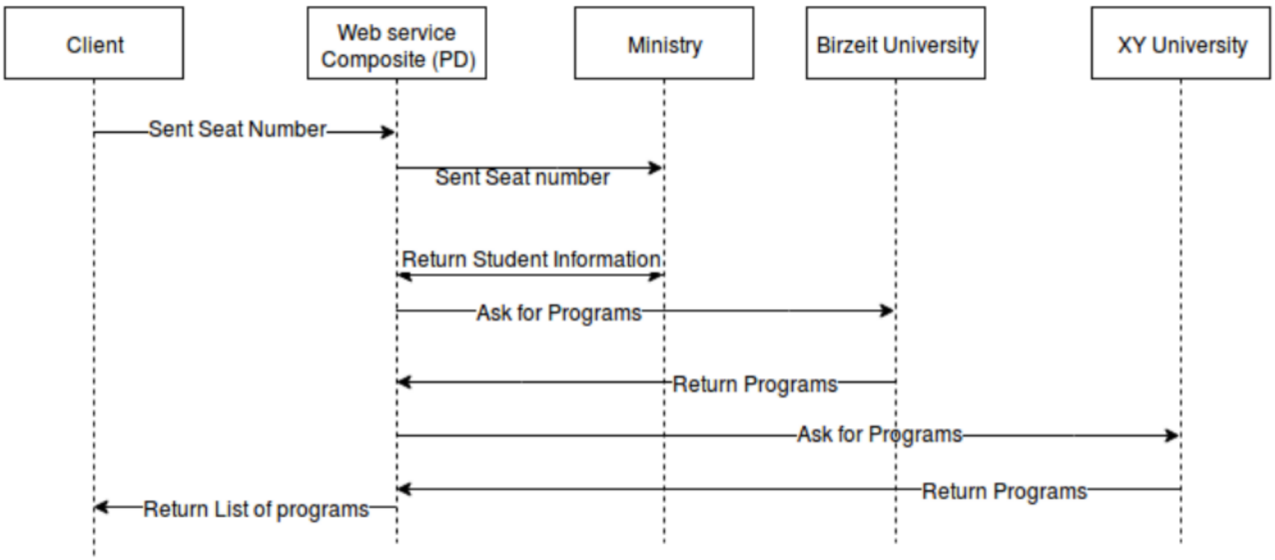

Figure 1

Components

- Client: on our service its present any user using our service
- Web Server: Which the main component working as an aggregator.
- Ministry service: we are using Ministry to request about student information's. (Name, Grade, Branch)
- Universities Service : we are using student information to search for suitable programs match students information (Grade, branch) cross all provided universities

Service architecture

Our composite service get the request from Tawjihi student with his seat

number with year of the student apply to Tawjihi exam. first our composite service send the request to high education ministry to get student Tawjihi grade, second, the composite service use this information to get academic programs the user can apply from two universities PPU and BZU, and allow the student to get all academic programs from all universities.

Our composite service connect to more than one service to serve user request, and each service use the technology accepted in the organization, so high education ministry use SOAP based service to serve the request and the response for data, and the PPU university use GraphQL , but the third university use REST based service. the end user does not aware what each service request based or aware how our composite service communicate with each resource service to serve his request.

Figure 2 show the proposed composite service in our project:

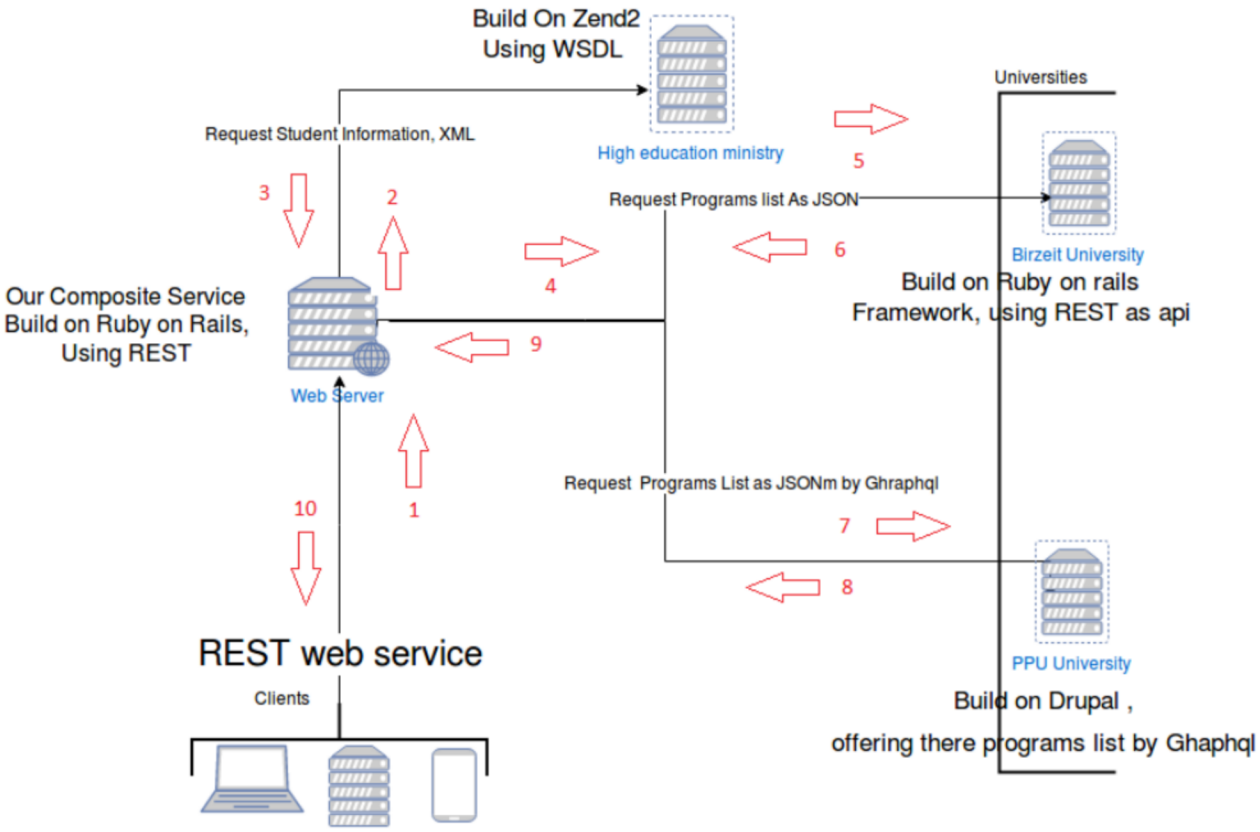

Figure 2

In the following section, we explain each service based technology on request/ response service.

Data Representation

- Programs Directory <--> Client: all request flow REST stander and JSON
- Programs Directory <--> Ministry, will flow SOP standards
- Programs Directory <--> Universitis, echo university have its own interface

High education ministry

High education ministry responsible for Tawjihi grades of students, and any

student has seat number for Tawjihi and used to get student Tawjihi grade, the second parameter is the year of the student apply for Tawjihi exam.

The High education ministry use SOAP based service to service user request, in the following example of request and response for this service:

```
<x:Envelope xmlns:x="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ec="http://ec2-35-166-183-83.us-west-2.compute.amazonaws.com/">
    <x:Header/>
    <x:Body>
        <ec:getStudentData>
            <ec:SeatNo>11111101</ec:SeatNo>
            <ec:Year>2016</ec:Year>
        </ec:getStudentData>
    </x:Body>
</x:Envelope>
```

Fig 3 : request for High education ministry service

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://ec2-35-166-183-83.us-west-2.compute.amazonaws
    .com/" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org
    /2001/XMLSchema-instance" SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Body>
        <ns1:getStudentDataResponse>
            <return SOAP-ENC:arrayType="SOAP-ENC:Struct[1]" xsi:type="SOAP-ENC:Array">
                <item xsi:type="SOAP-ENC:Struct">
                    <id xsi:type="xsd:int">1</id>
                    <Name xsi:type="xsd:string">test 1</Name>
                    <SeatNo xsi:type="xsd:int">11111101</SeatNo>
                    <Section xsi:type="xsd:string">literature</Section>
                    <Grade xsi:type="xsd:int">99</Grade>
                    <Year xsi:type="xsd:int">2016</Year>
                </item>
            </return>
        </ns1:getStudentDataResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig 4: response for High education ministry service

SOAP based service has WSDL file, used to implement this service define the data will used for this service, define the name space used to locate this service,

As in fig 5 below:

```
▼<types>
  ▼<xsd:schema targetNamespace="http://ec2-35-166-183-83.us-west-2.compute.amazonaws.com/">
    ▼<xsd:complexType name="tawjihist">
       <xsd:all/>
     </xsd:complexType>
   </xsd:schema>
  </types>
```

Figure 5

In addition, define complex data type from tawjihist type used to return data

from the service.

In addition, define the port for this service and the operation for this service serve, which is getStudentData, and input for this operation getStudentDataIn and getStudentDataOut as in Fig 6:

```
▼<portType name="Demo Web ServicePort">
  ▼<operation name="getStudentData">
      <documentation>Return tawjihist object</documentation>
      <input message="tns:getStudentDataIn"/>
      <output message="tns:getStudentDataOut"/>
    </operation>
  </portType>
```

Figure 6

After that, the file containing binding for this service, from WSDL file we can show that this service use RPC over HTTP for binding as in figure 7:

```
▼<binding name="Demo Web ServiceBinding" type="tns:Demo Web ServicePort">
   <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  ▼<operation name="getStudentData">
      <soap:operation soapAction="http://ec2-35-166-183-83.us-west-2.compute.amazonaws.com/#getStudentData"/>
     ▼<input>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://ec2-35-166-183-83.us-west-2.compute.amazonaws.com/"/
      </input>
     ▼<output>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://ec2-35-166-183-83.us-west-2.compute.amazonaws.com/"/
      </output>
    </operation>
  </binding>
```

Figure 7

Another useful information we can extract from WSDL file, the port location for this service and in Fig 8:

```
▼<service name="Demo Web ServiceService">
  ▼<port name="Demo Web ServicePort" binding="tns:Demo Web ServiceBinding">
      <soap:address location="http://ec2-35-166-183-83.us-west-2.compute.amazonaws.com/"/>
    </port>
  </service>
```

Figure 8

Final information we need from WSDL file, the messages used in this service to handle the request and the response for this service. for the operation getStudent DataIn, we can show that the service use to input parameter SeatBo and Year, which is int type used in the request for service. And the response getStudentDataOut return data from type TawjihiST which define in the earlier, as in figure 9

```
<message name="getStudentDataIn">
  <part name="SeatNo" type="xsd:int"/>
  <part name="Year" type="xsd:int"/>
</message>
<message name="getStudentDataOut">
  <part name="return" type="tns:tawjihist"/>
</message>
</definitions>
```

Figure 9

PPU University

PPU university use graphQL based service to serve the his service, and allow the user to request data he want as the fooloing in fig 7

```
{
  nodeQuery(type: "Program") {
    ... on EntityNodeProgram {
      title
      nid
      minGrade
      programType
      branch
    }
  }
}
```

Figure 10

This allow the user request academic programs based on title, id of the program, minimum grade for the program, program type and branch allow to register the program. Where GraphQL allow the service provider, define the parameter he want based on defined data of the service.

The response of the service based on parameter defined in the service we expanded before like in figure 11:


```
{
  "data": {
    "nodeQuery": [
      {
        "title": "Computer Engennering ",
        "nid": 2,
        "minGrade": 80,
        "programType": "master",
        "branch": "scientific"
      },
      {
        "title": "Biomedical Engineering ",
        "nid": 3,
        "minGrade": 77,
        "programType": "bachelor",
        "branch": "scientific"
      },
      {
        "title": "Information System",
        "nid": 4,
        "minGrade": 70,
        "programType": "bachelor",
        "branch": "literature"
      }
    ]
  }
}
```


Figure 11

BZU University

For the BZU university use REST based service, where the request and response JSON based, to request all academic programs from BZU university as this link “https://bzuprograms.herokuapp.com/programs.json” and response in JSON as in figure 12:


```
[
  ▾ {
      id: 1,
      title: "Arabic Music",
      mingrade: 70.0,
      description: "Arabic Music",
      program_type: null,
      branch: "scientific",
      created_at: "2017-05-06T16:58:27.501Z",
      updated_at: "2017-05-07T07:29:23.394Z",
      url: https://bzuprograms.herokuapp.com/programs/1.json
    },
  ▾ {
      id: 3,
      title: "Applied Statistics",
      mingrade: 70.0,
      description: "",
      program_type: null,
      branch: "scientific",
      created_at: "2017-05-19T11:04:06.051Z",
      updated_at: "2017-05-19T11:04:06.051Z",
      url: https://bzuprograms.herokuapp.com/programs/3.json
    },
  ▾ {
      id: 2,
      title: "Accounting",
      mingrade: 86.0,
      description: "Accounting",
      program_type: null,
      branch: "literature",
      created_at: "2017-05-07T07:25:33.388Z",
      updated_at: "2017-05-19T11:04:34.465Z",
      url: https://bzuprograms.herokuapp.com/programs/2.json
    }
]
```


Figure 12

4- Issues/discussion

Service hosting:

For each service used from composite service is hosted in deferent cloud host service, and can be accessed from web based on service definition. Where high education ministry service hosted in “amazon”, and PPU hosted in “ Web hosted server” and BZU hosted in “ herokuapp ”.

Caching Design Pattern

One way to reduce Internet bandwidth consumption in our composite service is to store frequently accessed objects on the local network in cache, where they can be retrieved by composite service without going out to a server on the Internet. In addition, it has the added advantage of making access for internal users faster because they are retrieving the Web objects over a fast LAN connection, typically 1 Gbps, instead of a slower Internet connection at perhaps 5-15 Mbps.

Caching accelerates the response to outbound requests when users on the internal network request a Web object from a server on the Internet. Frequently requested objects are stored on the caching server. This means they

can be retrieved via the fast local network connection instead of over a slower Internet connection.

we use this design pattern in order to speed the response on the consumer, instead of request all programs from all universities we caching program's on our server with for 15 minutes, so if a consumer as for programs we check our cache if it’s not exported server will use cached data.

Future enhancements

this work is considered proof of concept for our Idea which, We are looking forward on the feature to make this project real. Where it should including most palestinian universities also non palestinian universities, moreover we want to add more features such as scholarships directory. fees, and admission requirements.

API documentation

1. List of Programs

Returns json data about a Program's List.

- **URL** /Programs
- **Method:** GET
- **URL Params**
  - **min_grade** [Number] // Tawjihi grade
  - **type [**Bachelor | Master | Diploma **]**
  - **university** [BZU | PPU | NAU]
  - **seat_number** [number] // Tawjihi seat number
- **Success Response:**
  - **Code:** 200
  - **Content:**

```
[ {"name": "Computer science",
        "minGrade": 80,
        "university": "BZU",
        "description": "Good program",
        "feed": "40 JOD"
    }
    {   "name": "Computer Engineering",
        "minGrade": 83,
        "university": "PPU",
        "description": "Good program",
        "feed": "45 JOD"
    }]
```

- **Error Response:**

- **Code:** 404 NOT FOUND
- **Content:** { error : "Empty doesn't exist" }

2. List of Universities

Returns json data about a Program's List.

- **URL /universities**
- **Method:** GET
- **Success Response:**
  - **Code:** 200
  - **Content:**

```
[ {"name": "Birzeit University",
        "program count": 80,
        "shot name": "BZU",
        "description": "Good University",
    }
    {   "name": "Polytechnic  University",
        "program count": 50,
        "shot name": "PPU",
        "description": "Good University",
    }]
```

- **Error Response:**
  - **Code:** 404 NOT FOUND
  - **Content:** { error : "Empty doesn't exist" }

Datasheet:

Project API have four optional parameter to request, they are seat_number, year, university, branch and program_type.

If no parameter used in the request, service return all programs in all universities.

1- Parameter description,

1. seat_number:

Tawjihi Student Seat Number.

Must be a number with eight digits, below sample data for seat_number:

2. Year

Year Student apply Tawjihi.

If not used, use current year as default,

3. University

University join this application.

We have two Universities, BZU, and PPU.

4. Branch

Tawjihi Branch we have.

We have two Branch, literature and scientific.

5. program_type

Universities academic programs.

We have two Program Type, master and bachelor.

2- Return Data Sample:

Return Json list


```
  [{
"Id":"PPU2",
"title":"Accounting",
"program_type":"bachelor",
"university":"PPU",
"min-grade":75,
"branch":"literature"
},
{
"Id":"BZU2",
"title":"Software engeneering",
"program_type":"master",
"university":"BZU",
"min-grade":79,
"branch":"scientific"
}]
```


3- Education Ministry Database

SeatNo	Year
11111101	2016
11111102	2016
11111103	2016
11111104	2016

11111105	2016
11111106	2016
11111107	2016
11111108	2016
11111109	2016
11111110	2016
11111111	2016
11111112	2016
11111113	2016
11111114	2016
11111115	2016
11111116	2016
11111117	2016
11111118	2016
11111119	2016
11111120	2016
11111121	2016
11111122	2016
11111123	2016
11111124	2016
11111125	2016
11111126	2016
11111127	2016
11111128	2016
11111129	2016
11111130	2015
11111131	2015
11111132	2015
11111133	2015
11111134	2015
11111135	2015
11111136	2015
11111137	2015
11111138	2015
11111139	2015
11111140	2015
11111141	2015
11111142	2015
11111143	2015
11111144	2015
11111145	2015
11111146	2015
11111147	2015
11111148	2015
11111149	2015
11111150	2015

11111151	2015
11111152	2015
11111153	2015
11111154	2015
11111155	2015
11111156	2015
11111157	2015
11111158	2015
11111159	2015
11111160	2015

Client API

We implement a mobile application for the user to know from his location the weather and currency of this country, screen shot for this application bellow:

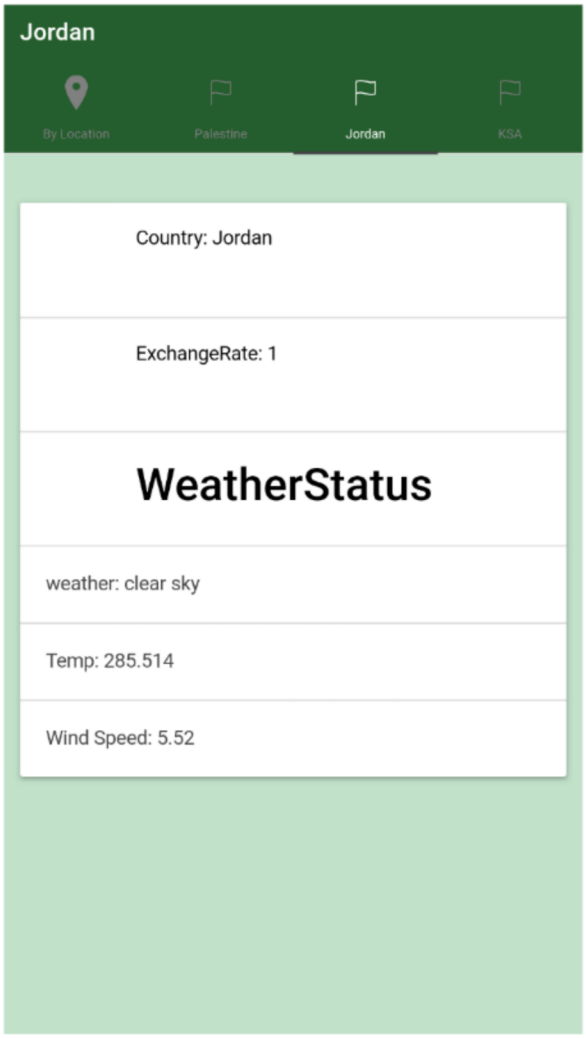

Figure 13 : API for Jordan