Restful Service API

For

Senior Project Website: Version 4

CIS 4911 Senior Project

Section U01

Professor Masoud Sadjadi

Prepared by

API Version 1

Nelson Capote

Michael Garcia

Antonio Vazquez

API Version 2

Jonathan Santiago

July 25th, 2014

1 COPYRIGHT

All title, including but not limited to copyrights, in and to the SENIOR PROJECT WEBSITE are owned by School of Computing and Information Sciences Faculty of Florida International University. All title and intellectual property rights in and to the content which may be accessed through use of the SENIOR PROJECT WEBSITE is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by School of Computing and Information Sciences Faculty of Florida International University.

2 ABSTRACT

In Senior Project Website version 2 there was a need to know exactly if a student is enrolled on Senior Project class in order to allow access to the site. In order to solve this problem we came up with the idea of creating an API in charge of validating a student against the list of enrolled students provided by FIU, and then, we translated this idea into creating a Restful Web Service. Now in In Senior Project Website version 4 it was necessary to modify this API to provide more data to external applications because we are getting closer to integrate all the subsystems.

This document is intended for developers who want to write applications that can interact with the Senior Project Website version 2 API. With this service you can check and obtain information about students enrolled in Senior Project class.

3 TABLE OF CONTENTS

4 GET ACCESS

The service requires that only “trusted” applications can consume it.

Your application can become a “trusted” application providing a valid TOKEN to the service.

This token will be provided temporarily by our team to all other teams that needs to make use of the service. Once the service goes into a production environment then the token will be provided by FIU SCIS network administrator (for security purposes) so then, only FIU trusted applications can have access to this information.

> Temporal token: 123FIUspw
>
> Note: Make this token configurable in your application. Do not hardcode it !!!

Fig 4.1 Token

5 API Specifications

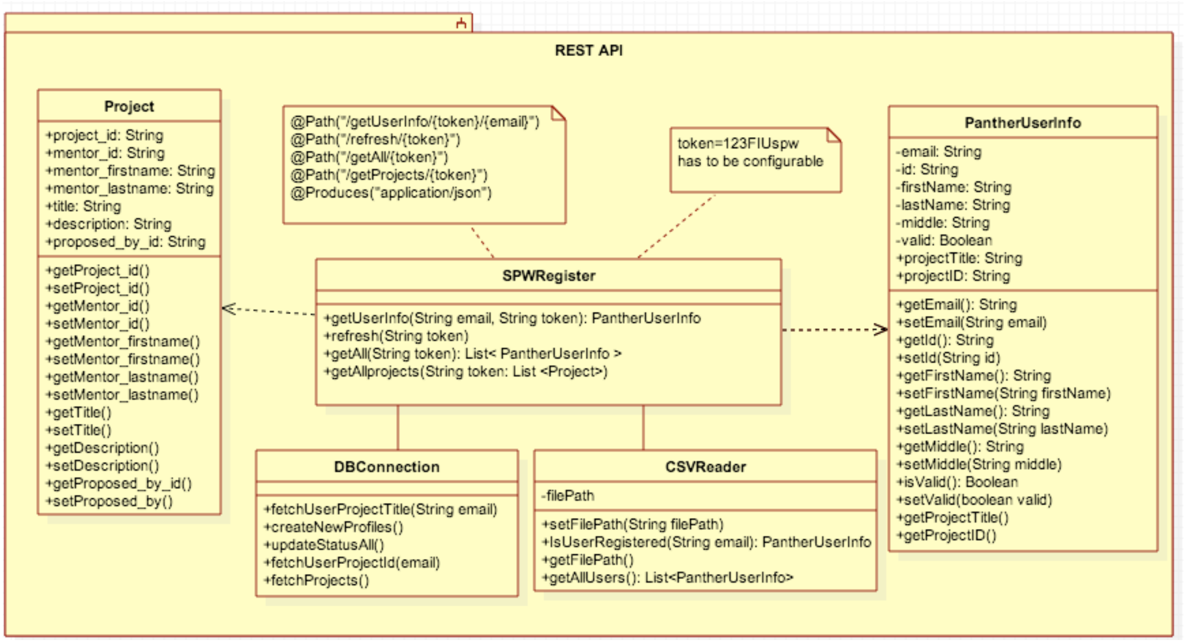
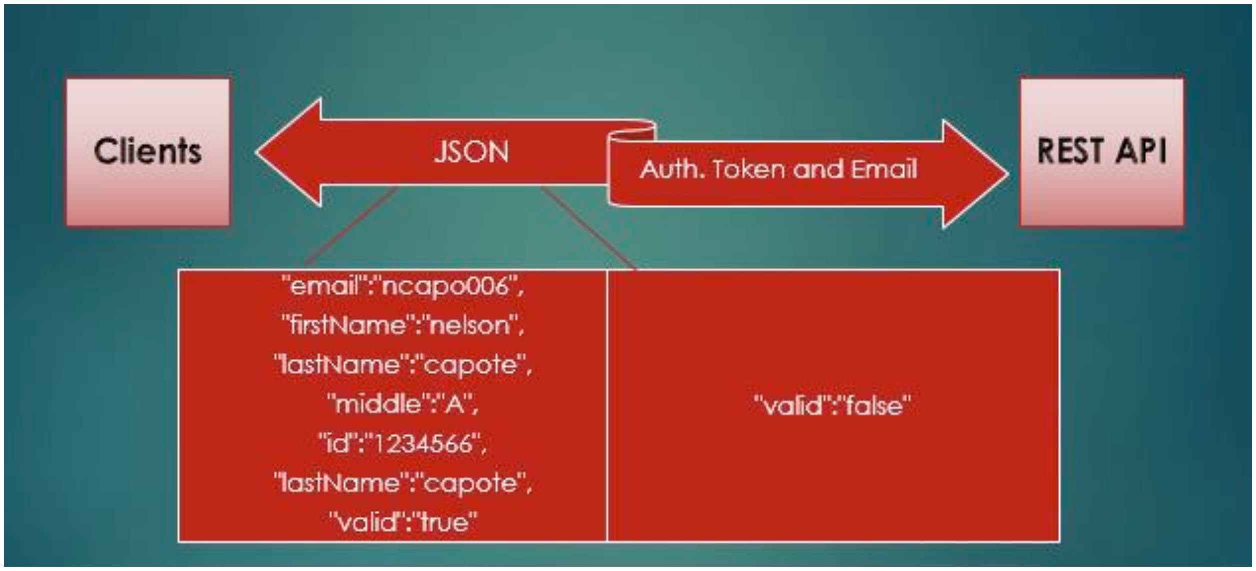


Fig 5.1 API class diagram

The entry point for the service is the SPWRegister class with the method getUserInfo(String email , String token) .

Any application that consumes the service will call this method and needs to provide a valid TOKEN and a valid FIU panther email address as parameters.

Once the service validates the token it returns a JSON object representing the PantherUserInfo class.



Clients

JSON

Auth. Token and Email

REST API

"email":"ncapo006",
"firstName":"nelson",
"lastName":"capote",
"middle":"A",
"id":"1234566",
"lastName":"capote",
"valid":"true"

"valid":"false"

Fig 5.2 Interaction between clients and API

6 CONSUMING THE SERVICE

Once the service is deployed the URL to access it must have the fallowing structure:

"Server address and port "*+ "/SPW2-RegisterAPI/rest/SPWRegister/getUserInfo/ "* + ***TOKEN*** + *"/****email***"

Where:

- **Server address and port:** The server address and port where the REST API is deployed. Now it is deployed at http://spws-dev.cis.fiu.edu:8080 this could also change once the service is on production environment; developers must make this part configurable on their apps.
- **/SPW2-RegisterAPI/rest/SPWRegister/getUserInfo/**: *this portion will never change.*
- ***TOKEN:*** 123FIUspw (see section 1.1 for more detail)
- /**email :** valid FIU email address like tim@fiu.edu

This is a sample URL to consume the service.

http://spws.cis.fiu.edu:8080/SPW2-RegisterAPI/rest/SPWRegister/getAll/123FIUspw/

Please note that the service requires a POST method but now it is deployed as a GET for testing purposes.

7 JSON RETURN

The service returns a JSON object representing the class PantherUserInfo with all its properties

Valid: A Boolean representing whether the student requested it is enrolled on the Senior Project class or not. A value **true** means that the student is enrolled on the class, a value **false** means that the email provided to the service does not match any student enrolled on the class. If this property is set to **false,** then there will be no info provided for the rest of the properties. It is responsibility of the developer to validate this property first.

Email: email address of student

Id: FIU Panther ID of student

firstName: First name of student

lastName: Last name of student

middle: Middle name of student when applicable.

Sample JSON response from the service for a valid user.

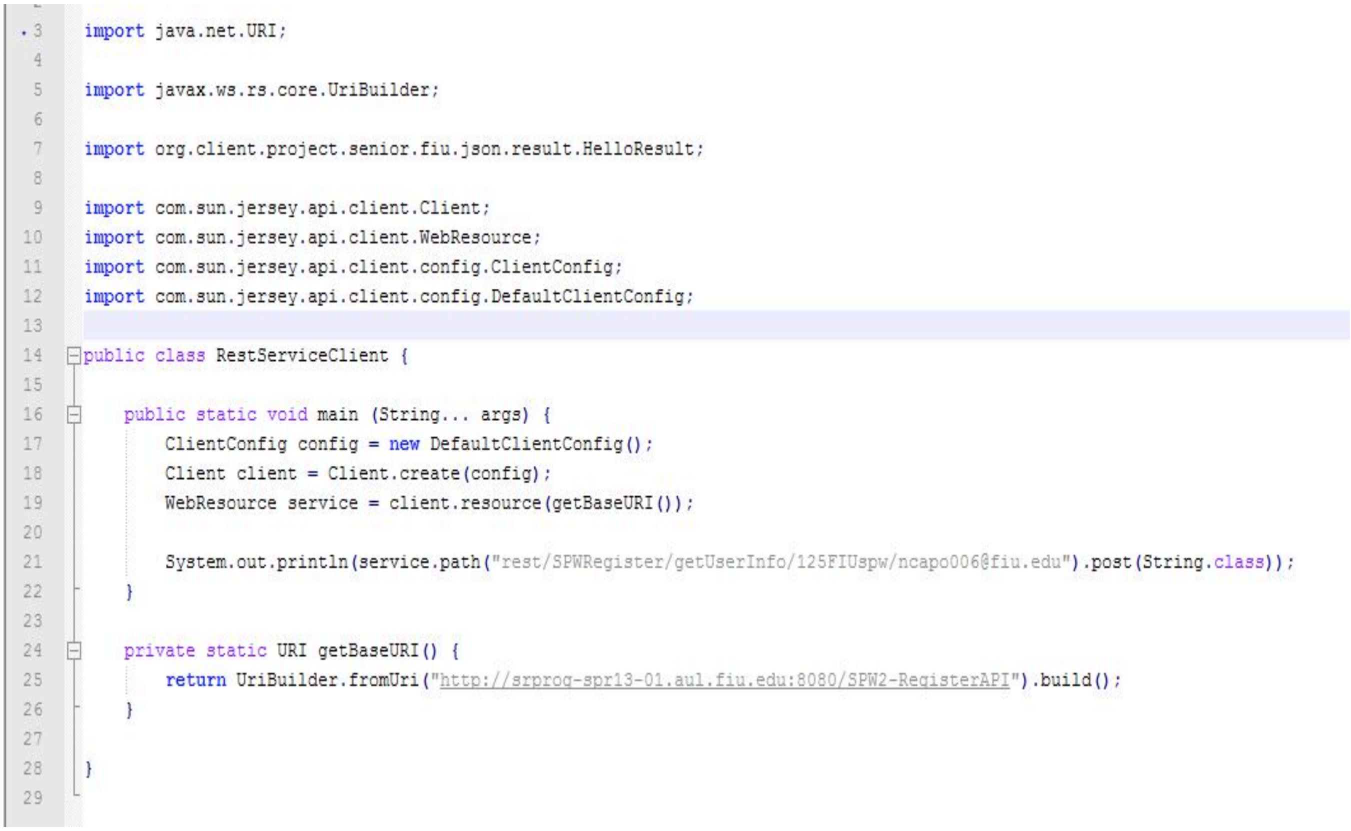
```
{
	"email":"ncapo006",
	"id":"1234566",
	"firstName":"nelson",
	"lastName":"capote",
	"middle":null,
	"valid":true
}
```

Sample JSON response from the service for a non-enrolled student.

```
{
	"email":null,
	"id":null,
	"firstName":null,
	"lastName":null,
	"middle":null,
	"valid":false
}
```

8 SAMPLE JAVA CODE TO CONSUME THE SERVICE

This is a java code for a client class that consumes the service.



```java
import java.net.URI;

import javax.ws.rs.core.UriBuilder;

import org.client.project.senior.fiu.json.result.HelloResult;

import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.WebResource;
import com.sun.jersey.api.client.config.ClientConfig;
import com.sun.jersey.api.client.config.DefaultClientConfig;

public class RestServiceClient {

    public static void main (String... args) {
        ClientConfig config = new DefaultClientConfig();
        Client client = Client.create(config);
        WebResource service = client.resource(getBaseURI());

        System.out.println(service.path("rest/SPWRegister/getUserInfo/125FIUspw/ncapo006@fiu.edu").post(String.class));
    }

    private static URI getBaseURI() {
        return UriBuilder.fromUri("http://srprog-spr13-01.aul.fiu.edu:8080/SPW2-RegisterAPI").build();
    }

}
```

Fig 8.1 Java code for a client class that consumes the service

Please refer to https://jersey.java.net/ for more detail.