Assignment No: 03

Assignment Name: Controller Rest API

Name: Binodon

ID: IT-17046

Theory:

Software-defined northbound application program interfaces (SDN northbound APIs) are usually SDN RESTful APIs used to communicate between the SDN controller and the services and applications running over the network. These APIs can be used to facilitate efficient orchestration and automation of the network to align with the needs of different applications via SDN network programibility s are the link between the applications and the SDN controller. The applications can tell the network what they need (data, storageidth, and so on) and the network can deliver those resources, or communicate what it has.

These APIs support a wide variety of applications. This is possibly why SDN northbound APIs are one of the most moldable components in an SDN environment — a variety of possible interfaces exist in different places up the stack to control different types of applications via an SDN Controller.

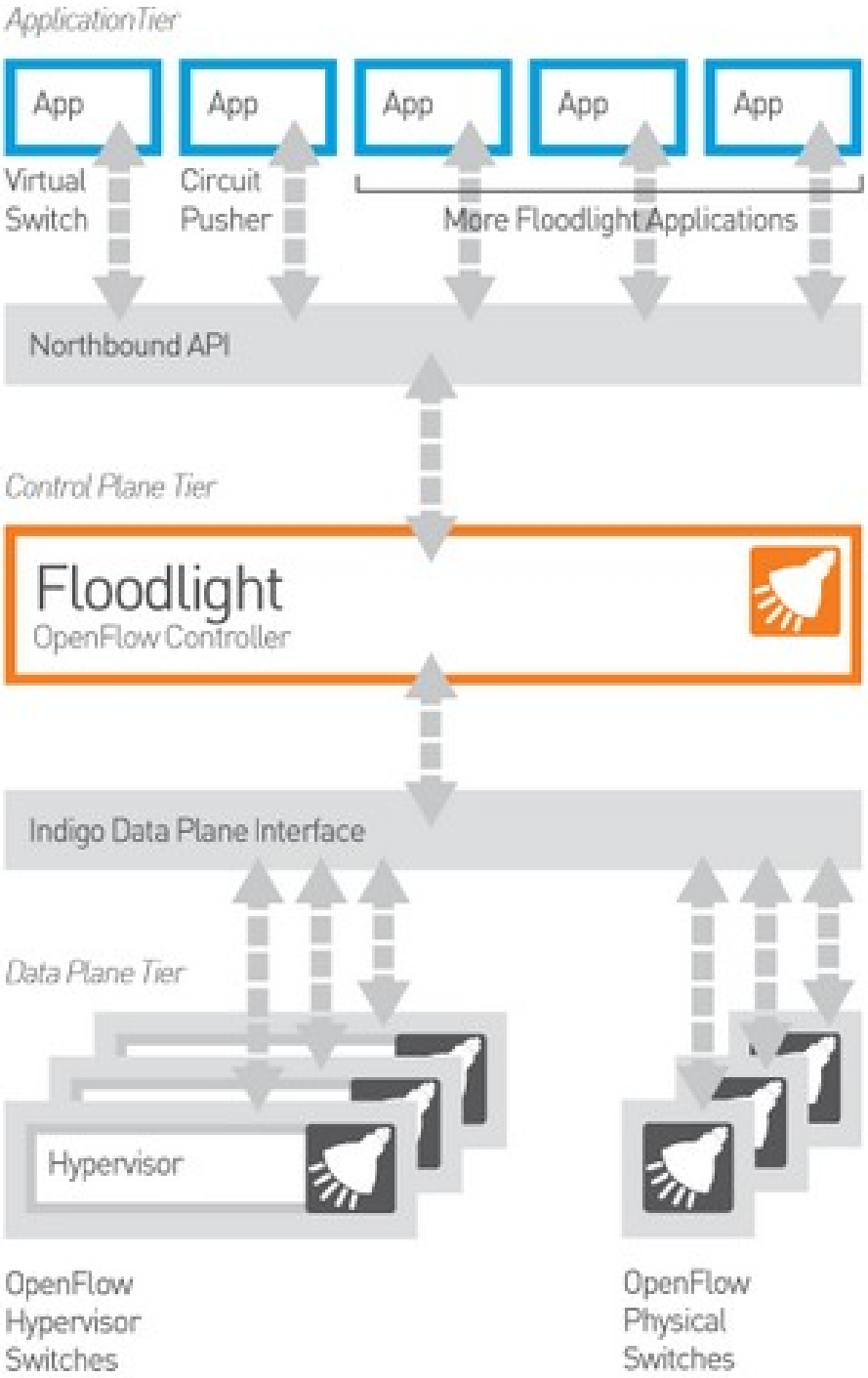
The advantages of REST for development

1. **Separation between the client and the server**: the REST protocol totally separates the user interface from the server and the data storage. This has some advantages when making developments. For example, it improves the portability of the interface to other types of platforms, it increases the scalability of the projects, and allows the different components of the developments to be evolved independently.

2.**Visibility, reliability and scalability**. The separation between client and server has one evident advantage, and that is that each development team can scale the product without too much problem. They can migrate to other servers or make all kinds of changes in the database, provided the data from each request is sent correctly. The separation makes it easier to have the front and the back on different servers, and this makes the apps more flexible to work with.

3.**The REST API is always independent of the type of platform or languages**: the REST API always adapts to the type of syntax or platforms being used, which gives considerable freedom when changing or testing new environments within the development. With a REST API you can have PHP, Java, Python or Node.js servers. The only thing is that it is indispensable that the responses to the requests should always take place in the language used for the information exchange, normally XML or JSON.

Working procedure:



Code:

```
package payroll;
import java.util.Objects;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
@Entity
class Employee {

  private @Id @GeneratedValue Long id;
  private String name;
  private String role;
  Employee() {}
  Employee(String name, String role) {

    this.name = name;
    this.role = role;
  }
  public Long getId() {
    return this.id;
  }
  public String getName() {
    return this.name;
  }
  public String getRole() {
    return this.role;
  }
  public void setId(Long id) {
    this.id = id;
```

```
  }
  public void setName(String name) {
    this.name = name;
  }
  public void setRole(String role) {
    this.role = role;
  }

  @Override
  public boolean equals(Object o) {
    if (this == o)
      return true;
    if (!(o instanceof Employee))
      return false;
    Employee employee = (Employee) o;
    return Objects.equals(this.id, employee.id) && Objects.equals(this.name, employee.name)
        && Objects.equals(this.role, employee.role);
  }

  @Override
  public int hashCode() {
    return Objects.hash(this.id, this.name, this.role);
  }

  @Override
  public String toString() {
    return "Employee{" + "id=" + this.id + ", name='" + this.name + '\'' + ", role='" + this.role
+ '\'' + '}';
  }
}
```

```
package payroll;

import org.springframework.data.jpa.repository.JpaRepository;

interface EmployeeRepository extends JpaRepository<Employee, Long> {}
```

Exercise/Output:

Test the Service:

visit http://localhost:8080/greeting.

Provide a name query string parameter by visiting http://localhost:8080/greeting?name=User. Notice how the value of the content attribute changes from Hello, World! to Hello, User!

This change demonstrates that the @RequestParam arrangement in GreetingController is working as expected. The name parameter has been given a default value of World but can be explicitly overridden through the query string.

To register a new REST route, you must specify a number of callback functions to control endpoint behavior such as how a request is fulfilled, how permissions checks are applied, and how the schema for your resource gets generated. While it is possible to declare all of these methods in an ordinary PHP file without any wrapping namespace or class, all functions declared in that manner coexist in the same global scope. If you decide to use a common function name for your endpoint logic like get_items() and another plugin (or another endpoint in your own plugin) also registers a function with that same name, PHP will fail with a fatal error because the function get_items() is being declared twice.

Notice also how the id attribute has changed from 1 to 2. This proves that you are working against the same GreetingController instance across multiple requests and that its counter field is being incremented on each call as expected.

```
$ curl -v -X PUT localhost:8080/orders/4/complete

* TCP_NODELAY set
* Connected to localhost (::1) port 8080 (#0)
> PUT /orders/4/complete HTTP/1.1
> Host: localhost:8080
> User-Agent: curl/7.54.0
> Accept: */*
>
< HTTP/1.1 405
< Content-Type: application/problem+json
< Transfer-Encoding: chunked
< Date: Mon, 27 Aug 2018 15:05:40 GMT
<
{
  "title": "Method not allowed",
  "detail": "You can't complete an order that is in the CANCELLED status"
}
```

Conclusion:

REST completely changed software engineering after 2000. This new approach to developing web projects and services was defined by Roy Fielding father of the HTTP specification and one of the leading international authorities on everything to do with Network Architecture, in his dissertation entitled “Architectural styles and the design of network based software architectures”. In the field of APIs, REST (Representational State Transfer) is today the “be all and end all” in service app development .Today there are no projects or applications that don’t have a REST API for the creation of professional services based on this software. Twitter, YouTube, Facebook identification systems… hundreds of companies generate business thanks to REST and REST APIs. Without them any horizontal growth would be practically impossible. This is because REST is the most logical, efficient and widespread standard in the creation of APIs for Internet services.