

## Lab 5

### REST AND Jersey

#### 1. โหลด Library ของ Jersey และทำการทดลองใช้งานดังนี้

##### a. ใช้ Dependency ดังนี้

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.13.1</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-core</artifactId>
  <version>2.13.1</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.13.1</version>
</dependency>

<dependency>
  <groupId>org.glassfish.jersey.core</groupId>
  <artifactId>jersey-server</artifactId>
  <version>2.27</version>
</dependency>

<dependency>
  <groupId>org.glassfish.jersey.containers</groupId>
  <artifactId>jersey-container-servlet</artifactId>
  <version>2.27</version>
</dependency>
<dependency>
  <groupId>org.glassfish.jersey.inject</groupId>
  <artifactId>jersey-hk2</artifactId>
  <version>2.27</version>
</dependency>

<dependency>
  <groupId>com.sun.xml.bind</groupId>
  <artifactId>jaxb-core</artifactId>
  <version>2.3.0.1</version>
</dependency>
```

```

<dependency>
  <groupId>javax.xml.bind</groupId>
  <artifactId>jaxb-api</artifactId>
  <version>2.3.1</version>
</dependency>

<dependency>
  <groupId>com.sun.xml.bind</groupId>
  <artifactId>jaxb-impl</artifactId>
  <version>2.3.1</version>
</dependency>

```

- b. จาก Lab 4 ไปที่ package Entity เดิมที่มีคลาส Customer และ PhoneNumber โดย Entity นั้นสนับสนุนข้อมูลในรูปแบบของ JSON
- c. ให้ คลาส Customer และ PhoneNumber ทำการ implement Serializable เพื่อใช้สำหรับการบันทึกข้อมูลทั้ง Object

```
public class Customer implements Serializable{
```

- d. สร้าง คลาส CustomerDAO หรือถ้ามีอยู่แล้วให้ใช้ของเดิม โดยทำการแก้ไข CustomerDAO เดิมโดยการสร้างเมธอด
  - i. getAllCustomers
  - ii. saveCustomerList

```

public ArrayList<Customer> getAllCustomers() {

    ArrayList<Customer> listOfCustomers = new ArrayList<>();

    try {

        File file = new File("Customer.dat");

        if (!file.exists()) {
            Customer c1 = new Customer("Mr.Smith");
            Customer c2 = new Customer("Mrs.Smith");

            listOfCustomers.add(c1);
            c1.getPhoneNumbers().add(new PhoneNumber("145-1654-
1495"));

            listOfCustomers.add(c2);
            saveCustomerList(listOfCustomers);
        } else {
            FileInputStream fis = new FileInputStream(file);
            ObjectInputStream ois = new ObjectInputStream(fis);
            listOfCustomers = (ArrayList<Customer>)
ois.readObject();
            ois.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

```

        return listOfCustomers;
    }

    private void saveCustomerList(List<Customer> cusList) {
        try {
            File file = new File("Users.dat");
            FileOutputStream fos;
            fos = new FileOutputStream(file);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(cusList);
            oos.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

e. ที่ package controller ให้สร้างคลาส CustomerService ดังนี้

```

@Path("/services")
public class CustomerService {

    CustomerDAO cusDao = new CustomerDAO();
    @GET
    @Path("/customers")
    @Produces(MediaType.APPLICATION_JSON)

    public List<Customer> getUsers(){

        return cusDao.getAllCustomers()    }

}

```

f. ที่ web.xml ทำการ config ไฟล์ดังนี้

```

<?xml version = "1.0" encoding = "UTF-8"?>
<web-app xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xmlns = "http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
    id = "WebApp_ID" version = "3.0">
    <display-name>Customer Management</display-name>
    <servlet>
        <servlet-name>jersey-servlet</servlet-name>
        <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
        <init-param>
            <param-name>jersey.config.server.provider.packages</param-name>
            <param-value>th.ac.ku.kps.eng.cpe.controller</param-value>
        </init-param>
    </servlet>

```

```

<init-param>
  <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
  <param-value>>true</param-value>
</init-param>

</servlet>

<servlet-mapping>
  <servlet-name>jersey-servlet</servlet-name>
  <url-pattern>/rest/*</url-pattern>
</servlet-mapping>
</web-app>

```

g. เรียก RESTful WS โดยไปที่ <http://localhost:8080/web/rest/services/customers>

2. ทดสอบการดักค้นหา เพิ่ม ลบ แก้ไข ข้อมูลของ Customer และ PhoneNumbers ดังนี้

a. ที่ CustomerDAO ทำการเพิ่มเมธอด

i. getCustomerByName

```

public Customer getCustomerByName(String name) {
    List<Customer> cusList = getAllCustomers();

    for (Customer cus : cusList) {
        if (cus.getName().equalsIgnoreCase(name)) {
            return cus;
        }
    }
    return null;
}

```

b. ที่ CustomerService ทำการเพิ่มเมธอด

i. getCustomerByName

```

@GET
@Path("/customers/{param}")
public Customer getCustomerByname(@PathParam("param") String name){
    return cusDao.getCustomerByName(name);
}

```

c. ทดสอบ RESTful Jersey ดังนี้

```
import com.sun.jersey.api.client.Client;
import com.sun.jersey.api.client.ClientResponse;
import com.sun.jersey.api.client.WebResource;

public class Main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {

            Client client = Client.create();

            WebResource webResource =
client.resource("http://localhost:8080/jersey/rest/services/customers/A"
);

            ClientResponse response =
webResource.accept("application/json").get(ClientResponse.class);

            if (response.getStatus() != 200) {
                throw new RuntimeException("Failed : HTTP error
code : " + response.getStatus());
            }

            String output = response.getEntity(String.class);

            System.out.println("Output from Server .... \n");
            System.out.println(output);

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

d. ที่ CustomerDAO ทำการเพิ่มเมธอด addCustomer

```
public int addCustomer(Customer cus) {
    List<Customer> cusList = getAllCustomers();
    boolean cusExists = false;
    for (Customer c : cusList) {
        if (c.name.equalsIgnoreCase(cus.name)) {
            cusExists = true;
            break;
        }
    }
}
```

```

    }
    if (!cusExists) {
        cusList.add(cus);
        saveCustomerList(cusList);
        return 1;
    }
    return 0;
}

```

e. ที่ CustomerService ทำการเพิ่มเมธอด

i. createCustomer

```

@POST
@Path("/customers/create")
@Consumes(MediaType.APPLICATION_JSON)
public Response createCustomer(Customer cus) throws IOException {

    int i = cusDao.addCustomer(cus);
    if (i == 1)
        return Response.status(201).entity(" create
successfully").build();
    else
        return Response.status(201).entity(" create fail").build();

}

```