EVENT MANAGEMENT APPLICATION

Requirements of the Lab: A Spring REST Application

2 clients minimum

Aserver

Implement all the CRUD operations

To run the project:

- 1. Using an IDE would be the best, I used Eclipse for the project
- 2. Download any version of Tomcat that you may like, Install the Tomcat plugin into the Eclipse IDE.
- 3. I used the Maven Build Tool for running the project, kindly follow the directory structure of all the dependencies and files.

Mine looked something like this

```
Project Explorer ≅
                                                🕒 😘 💝 🕫 🖟 EventRestController.java 🖫 ClientApp.java 🖟 ClientApp2.java 🖟 Admin.java 🖟 Event.java 🖟 exmpleClass.java
                                                                     @RequestMapping("/api")
public class EventRestController
▶ ≅ Servers
▶ ∰ sockets
 ► 🖫 Deployment Descriptor: spring-rest-demo
► 🚜 JAX-WS Web Services
  ▶ ﷺ Java Resources
▼ ➡ JavaScript Resources
    ▶ @ ECMA 3 Browser Support Library
  ▶ ⊜ Deployed Resources
                                                                                   //Delete an event
@RequestMapping(value = "events/{eventId}", method = RequestMethod.DELETE)
public String delete(@PathVariable("eventId") String eventId) {
    ▼ ⊜ main
      ▼ 🍪 java
▼ 👺 com
           ▼ @ luv2code
            ClientApp2.java
            ▼ ⊜ entity
                  Admin.java

Event.java
                   a exmpleClass.java
                   SpringSecurityConfig.java
User.java
              ▼ 🍃 rest
                                                                                    public String createAnEvent(@RequestBody exmpleClass newEvent) {
    Event newOp! = new Event(newEvent.date, newEvent.eventId, newEvent.place, newEvent.userId);
    theEvents.add(newOp!);
    return "Event Successfully created";
       @java
                                                                                    public Boolean validateUser(@RequestBody Admin ad) {
    //Admin ad = new Admin(userCred.userHame, userCred.password);
    if(ad.userName.equals("admin") 66 ad.password.contentEquals("admin"))
```

4. Add the required libraries to the configuration path

- ▼ 📂 spring-rest-aemo
 - ▶ 🛅 Deployment Descriptor: spring-rest-demo

 - - **▼** # src/main/java
 - ▶ Æ com.luv2code.Client
 - ▶ # com.luv2code.springdemo.config
 - ▶ # com.luv2code.springdemo.entity
 - ▶ Æ com.luv2code.springdemo.rest
 - ▶ **#**src/test/java
 - **▼ ≤** Libraries
 - ▶ JRE System Library [JavaSE-1.8]
 - ▶ Maven Dependencies
 - - src/main/webapp

 - ▶ **■** ECMAScript Built-In Library
 - ▶ **③** ECMA 3 Browser Support Library
- 5. Run the Tomcat server
- 6. Run the two client applications

The applications will run in two consoles. Login as an admin in one and then access the events in the other as a guest client

1. POST METHOD

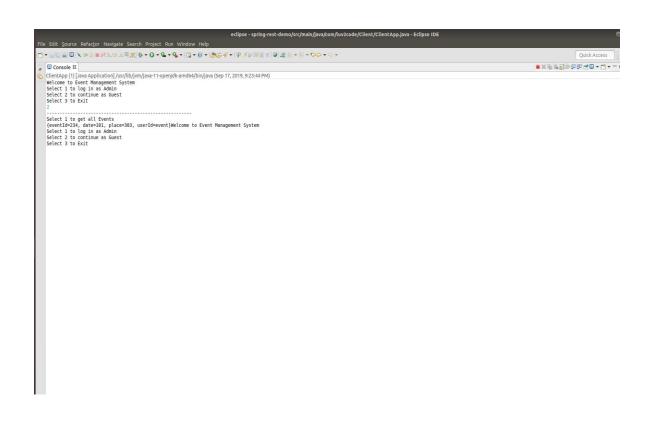
```
edipse spring-real-demofard/main/java/com/(Client/App)seva - Edipse IDE

File Edit Source Refraçor Navigate Search Project Rin Window Help

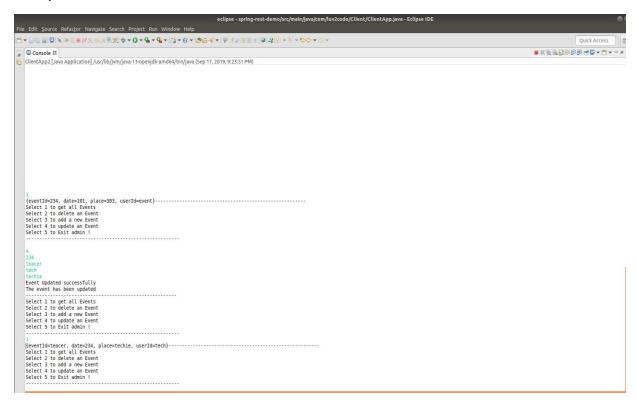
Ground III

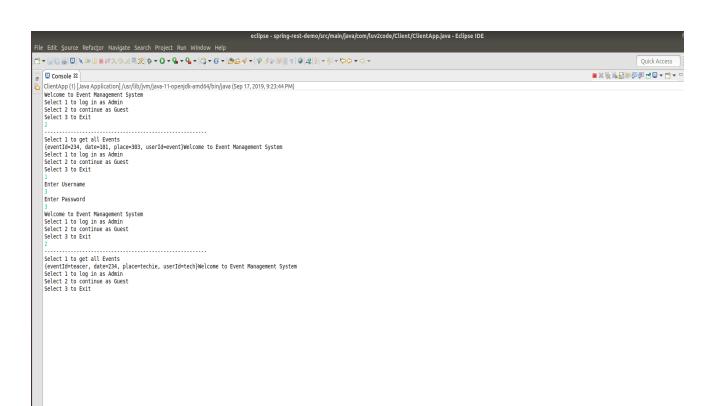
Ground III
```

2. GET METHOD

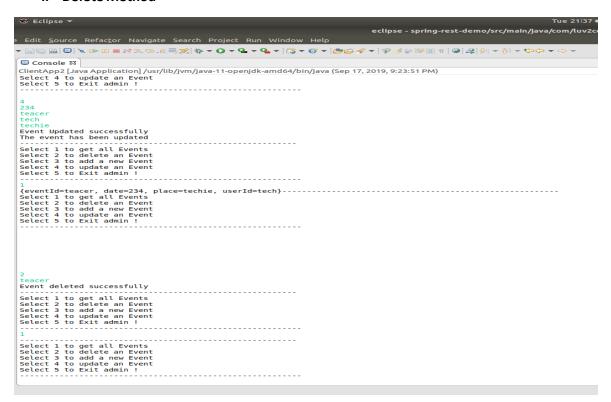


3. Update method





4. Delete Method



```
File Edit Source Refactor Navigate Search Project Run Window Help

| Commonstructure | Commonstructure
```