

Project Structure

The screenshot displays an IDE interface with three main panels:

- Project Explorer:** Shows the project structure for 'EmployeeService (in Employee-Service-Rest-Security)'. The 'src/main/java' directory contains:
 - `com.greatLearning.employeeService` (containing `EmployeeService.java`)
 - `com.greatLearning.employeeService.controller` (containing `EmployeeRestController.java`)
 - `com.greatLearning.employeeService.dao` (containing `EmployeeRepository.java`, `RoleRepository.java`, and `UserRepository.java`)
 - `com.greatLearning.employeeService.entity` (containing `Employee.java`, `Role.java`, and `User.java`)
 - `com.greatLearning.employeeService.security` (containing `MyUserDetails.java` and `WebSecurityConfig.java`)
 - `com.greatLearning.employeeService.service` (containing `EmployeeService.java`, `EmployeeServiceImpl.java`, and `UserDetailsServiceImpl.java`)The 'src/main/resources' directory contains:
 - `static`
 - `templates`
 - `application.properties`
 - `data.sql` (selected)
- SQL Editor:** Displays the content of `data.sql`, which includes SQL scripts for creating and inserting data into tables: `USERS`, `employee`, `roles`, and `users_roles`. The script is as follows:

```
1 --DROP TABLE IF EXISTS TEST;
2 --CREATE TABLE USERS(user_id INT PRIMARY KEY, USERNAME VARCHAR(255),PASSWORD
3 INSERT INTO USERS VALUES(1, '$2a$12$yZI4gjdk6vxb/TfM.oRv0xsDNeR02ffJGk9nG7L
4 INSERT INTO USERS VALUES(2, '$2a$12$8k06jIAxi4unZY0hTG7.VuJcBofA3yw51QFM4M9E
5 --CREATE TABLE employee(id INT PRIMARY KEY, first_name VARCHAR(255),last_nam
6 INSERT INTO employee VALUES(1, 'John','Smith','johnsmith@gmail.com');
7 --CREATE TABLE roles(role_id INT PRIMARY KEY, name VARCHAR(255));
8 INSERT INTO roles VALUES(1, 'admin');
9 INSERT INTO roles VALUES(2, 'user');
10 --CREATE TABLE users_roles(user_id INT,role_id INT);
11 INSERT INTO users_roles VALUES(1,1);
12 INSERT INTO users_roles VALUES(2,2);
13
```
- Console:** Shows the output of the application. The logs indicate that the JPA EntityManagerFactory is initialized, the LiveReload server is running on port 35729, the filter chain is created, and Tomcat started on port 8081. The application also logs that Spring Data repositories are initialized and the EmployeeService is started in 2.233 seconds.