

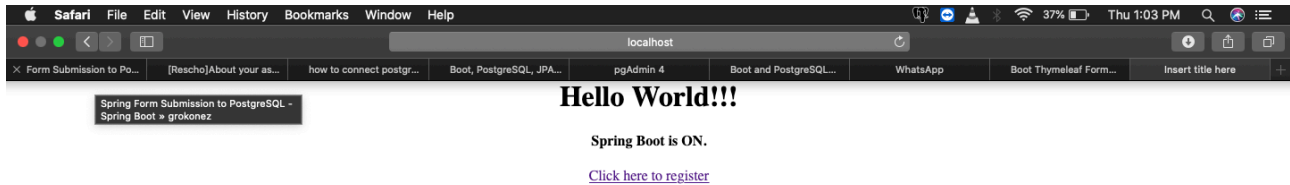
◇ Training Tasks

1. Set up environment and show up “Hello world” on the browser.

Purpose: To set up development environment.

Notice • See below item “Develop Environment” to be used.

- Showing up “Hello world” is to make sure Spring Boot is active.



This is the output of Hello world program on browser, while spring boot is active.

Along with it I had add link to a html form that will take us to a form where we can fill the details and it will be send to the server and the server will response back to the inputted html text.

url: <http://localhost:8080/index.html>

2. Input a text to HTML form, send it to a server, respond the inputted text by html from the server and show it up.

Purpose: To understand and implement the structure of receiving a request and responding by a server.

User Registration

| | |
|---|--|
| Full name: | <input type="text" value="vinay"/> |
| E-mail: | <input type="text" value="yaniv@gmail.com"/> |
| Password: | <input type="password" value="....."/> |
| Profession: | <input type="text" value="Developer"/> |
| <input type="button" value="Register"/> | |

Url: http://localhost:8080/register_form.html

In the above image, I have created a html form which take four parameters, Name, email, password and profession. As soon we click the register button , the inputted data is sent to server using POST method and then the server response back the data by displaying the data on the browser,

Registration succeeded

| | |
|-------------|-----------------|
| Full Name: | vinay |
| E-mail: | yaniv@gmail.com |
| Password: | dshgjdfig |
| Profession: | Developer |

Url: http://localhost:8080/register_success.html

The above image is the response from the server.

The screenshot shows an IDE with the following components:

- Project Explorer:** Displays the project structure, including `src/main/java` with `com.jcoreava` package, `MainController.java`, `SpringformhandlertutorialApplication`, and `User.java`. It also shows `src/main/resources` with `static` folder and `templates` folder containing `index.html`, `register_form.html`, and `register_success.html`.
- Code Editor:** Shows the `MainController.java` file with the following code:

```
1 package com.jcoreava;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 public class SpringformhandlertutorialApplication {
7
8     public static void main(String[] args) {
9         SpringApplication.run(SpringformhandlertutorialApplication.class, args);
10    }
11
12 }
13
14
```
- Console:** Displays the Spring Boot startup logs, including the message `Spring Boot :: (v2.3.0.RELEASE)` and `Tomcat started on port(s): 8080 (http)`. The logs also show the application's configuration and the successful startup of the embedded Tomcat server.
- Outline:** Shows the class hierarchy, including `com.jcoreava` and `SpringformhandlertutorialApplication`.

The above image is the screenshot of the project in Eclipse IDE. And it shows that spring boot is active.

3. Save the inputted text by HTML form to DB on a server, respond the table of saved records by JSON and show it up.
Purpose: To understand and implement DB access (i.e. create record, and read record).

Customer Form

Id:

First Name:

Last Name:

The above image is the screenshot of the html form where the input will be taken
Url: <http://localhost:/form>

SUBMIT SUCCESSFUL

[Submit another Customer Form](#)

Once we submit the data it will be send to the server using POST method request mapping and it will be store in the postgresql database. I have used Pgadmin to create the database. And the updated entered data is reflected in the table as shown below.

The screenshot shows the PgAdmin 4 interface. On the left, the 'Servers' tree is expanded to 'PostgreSQL 12' > 'Databases (1)' > 'postgres' > 'Schemas (1)' > 'public' > 'Tables (1)' > 'customer'. The main pane shows the 'Query Editor' with the following SQL query:

```
1 SELECT * FROM public.customer
2 ORDER BY id ASC
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query in a table format:

| | id [PK] bigint | firstname character varying (100) | lastname character varying (100) |
|---|-------------------|--------------------------------------|-------------------------------------|
| 1 | 122 | yaniv | jais |
| 2 | 123 | vinay | Jaiswal |
| 3 | 562 | Vivek | Jaiswal |
| 4 | 1230 | hjaflsdgf | ASHHJ |

This is what I have done till now. The rest JSON, update and delete will update and send once I finish.

Thank you.