

Dynamic HTML

Dynamic HTML is a set of technologies that allows dynamic changes to HTML documents. An embedded script can be used to change tag attributes, contents or element style properties. These changes are not uniformly supported across the full spectrum of browsers.

Greeting.java

```
package com.example.demo;

public class Greeting {
    private long id;
    private String content;

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id=id;
    }

    public String getContent() {
        return content;
    }

    public void setContent(String content) {
        this.content=content;
    }
}
```

GreetingController.java

```
package com.example.demo;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
public class GreetingController {

    @GetMapping("/greeting")
    public String greetingForm(Model model) {
        model.addAttribute("greeting", new Greeting());
        return "greeting";
    }

    @PostMapping("/greeting")
    public String greetingSubmit(@ModelAttribute Greeting greeting, Model model)
    {
        model.addAttribute("greeting", greeting);
        return "result";
    }
}
```

Greeting.html

```
<!DOCTYPE html>

<html xmlns:th="https://www.thymeleaf.org">

<head>

<title>Getting Started: Handling From Submission</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<h1>Form</h1>

<form    action="#"    th:action="@{/greeting}"    th:object="${greeting}"
method="post">

<p>ID:<input type="text" th:field="*{id}"/></p>

<p>Message: <input type="text" th:field="*{content}"/></p>

<p><input    type="submit"    value="Submit"    /><input    type="reset"
value="Reset"/></p>

</form>

</body>

</html>
```

NewApplication.java

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class NewApplication {

    public static void main(String[] args) {

        SpringApplication.run(NewApplication.class, args);

    }

}
```

Result.html

```
<!DOCTYPE html>

<html xmlns:th="https://www.thymeleaf.org">

<head>

<title>Getting Started: Handling From Submission</title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

</head>

<body>

<h1>Result</h1>

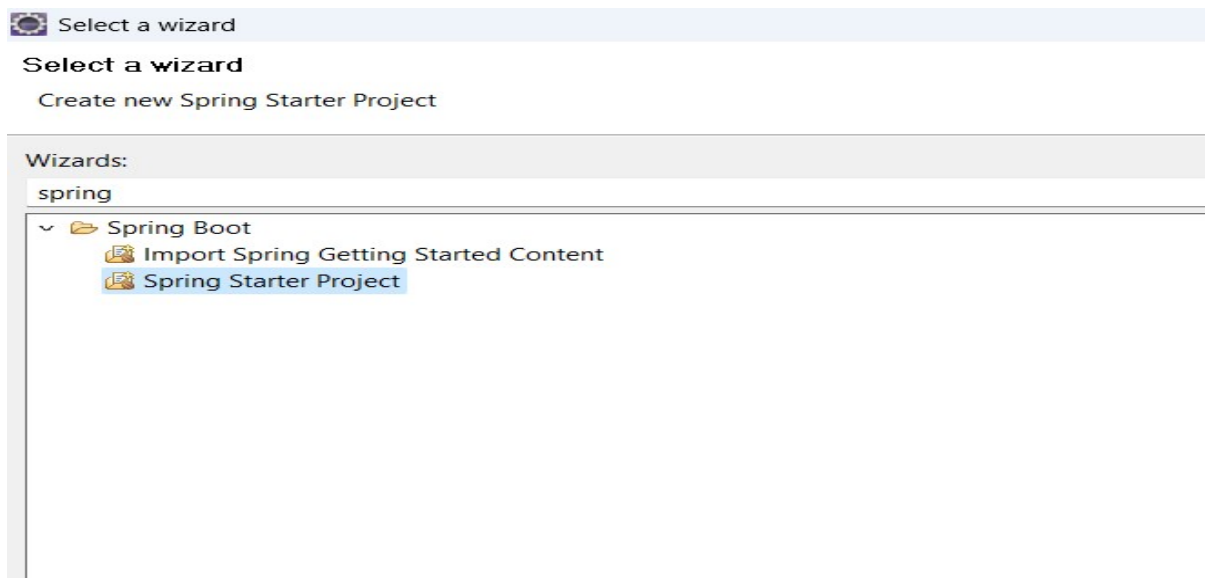
<p th:text="id: ' + ${greeting.id}"/>

<p th:text="content: ' + ${greeting.content}"/>

<a href="/greeting">Submit another message </a>

</body>

</html>
```



New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location: [Browse](#)

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

[< Back](#) [Next >](#) [Finish](#) [Cancel](#)

New Spring Starter Project Dependencies

Spring Boot Version:

Frequently Used:

☐ Spring Boot DevTools ☒ Spring Web ☐ Spring Web Services

☒ Thymeleaf ☐ Validation

Available:

Type to search dependencies

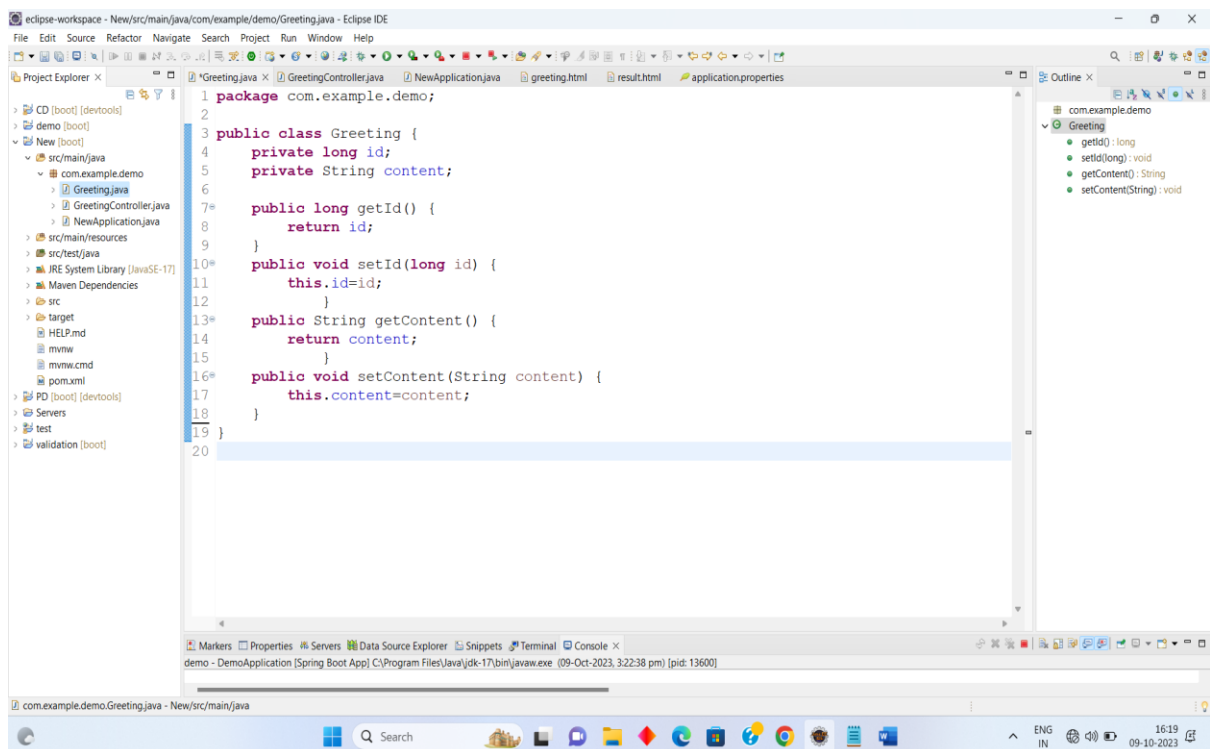
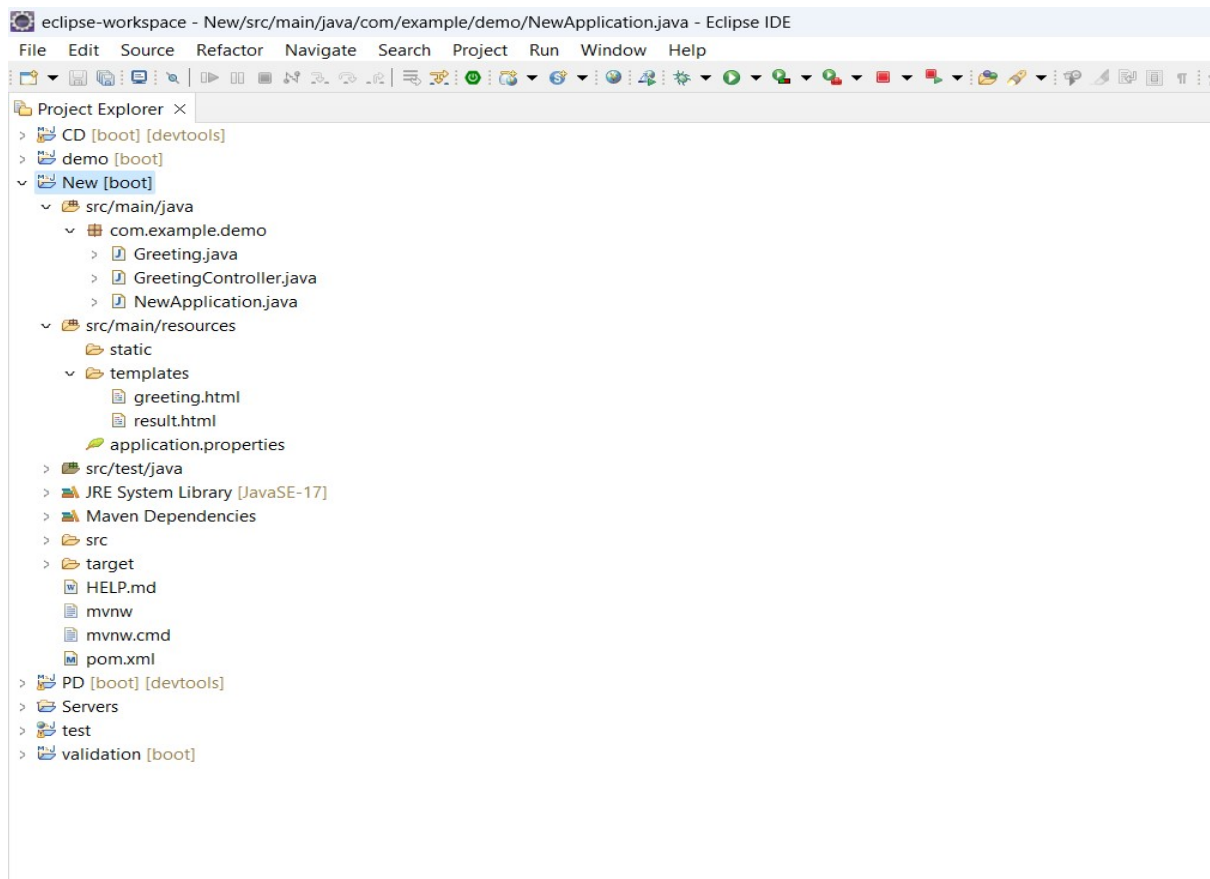
- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Observability
- Ops
- SQL
- Security
- Spring Cloud
- Spring Cloud Circuit Breaker
- Spring Cloud Config
- Spring Cloud Discovery
- Spring Cloud Messaging
- Spring Cloud Routing
- Template Engines
- Testing

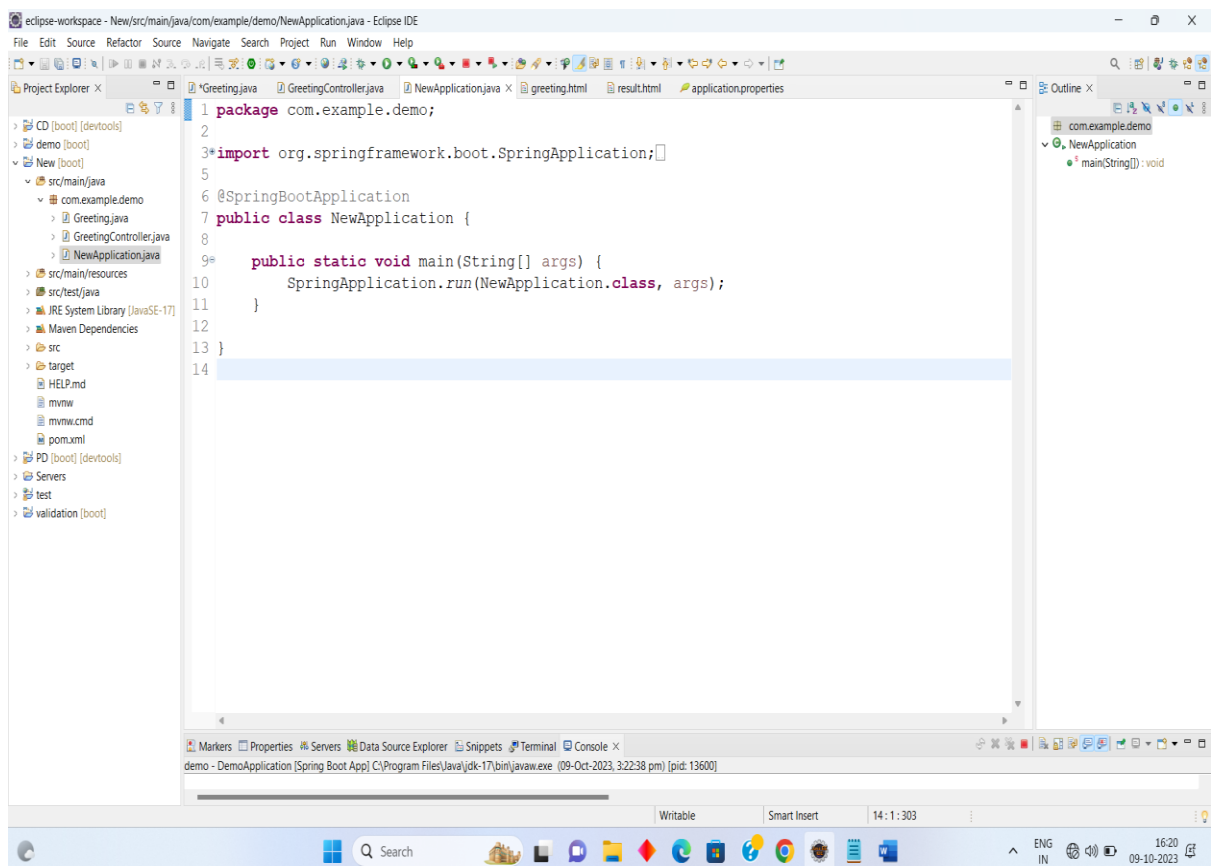
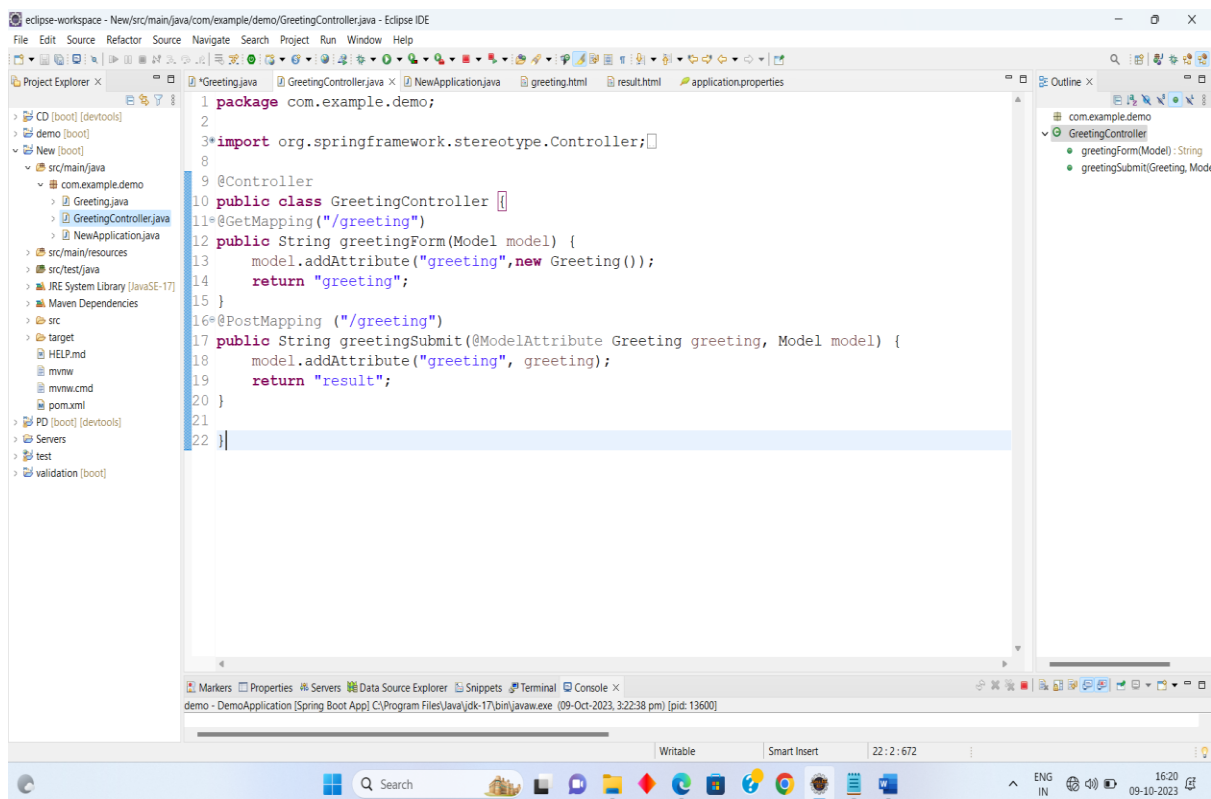
Selected:

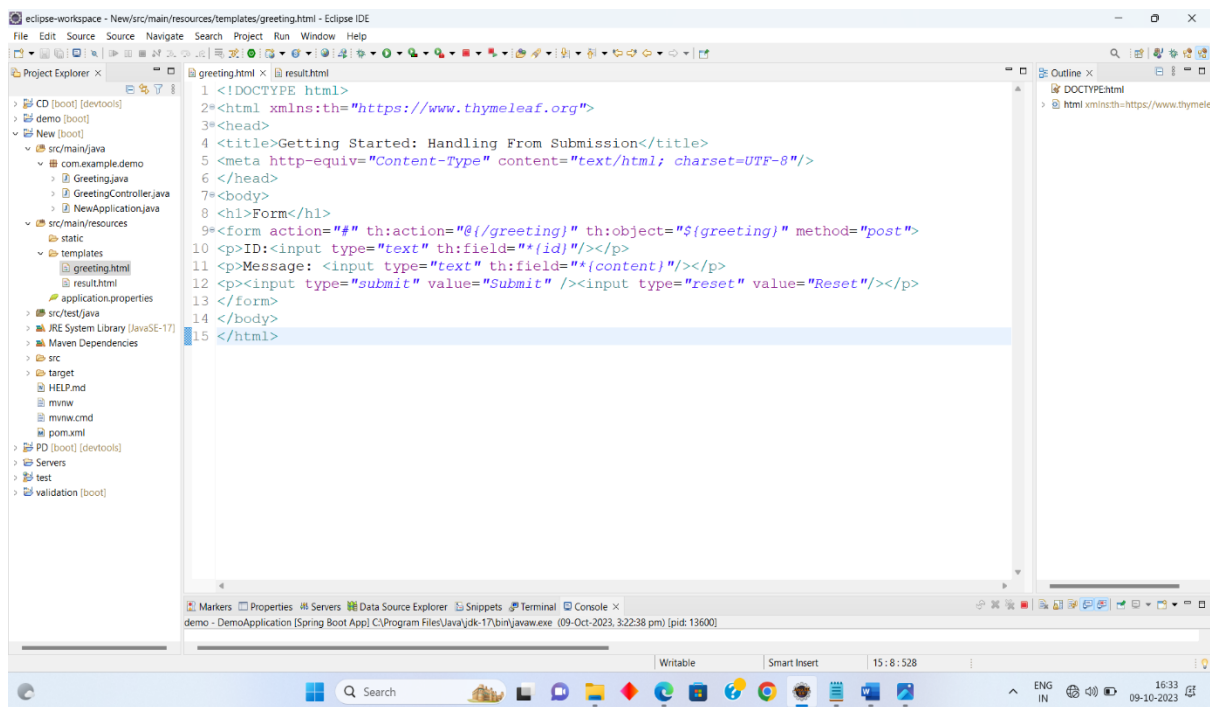
- X Thymeleaf
- X Spring Web

[Make Default](#) [Clear Selection](#)

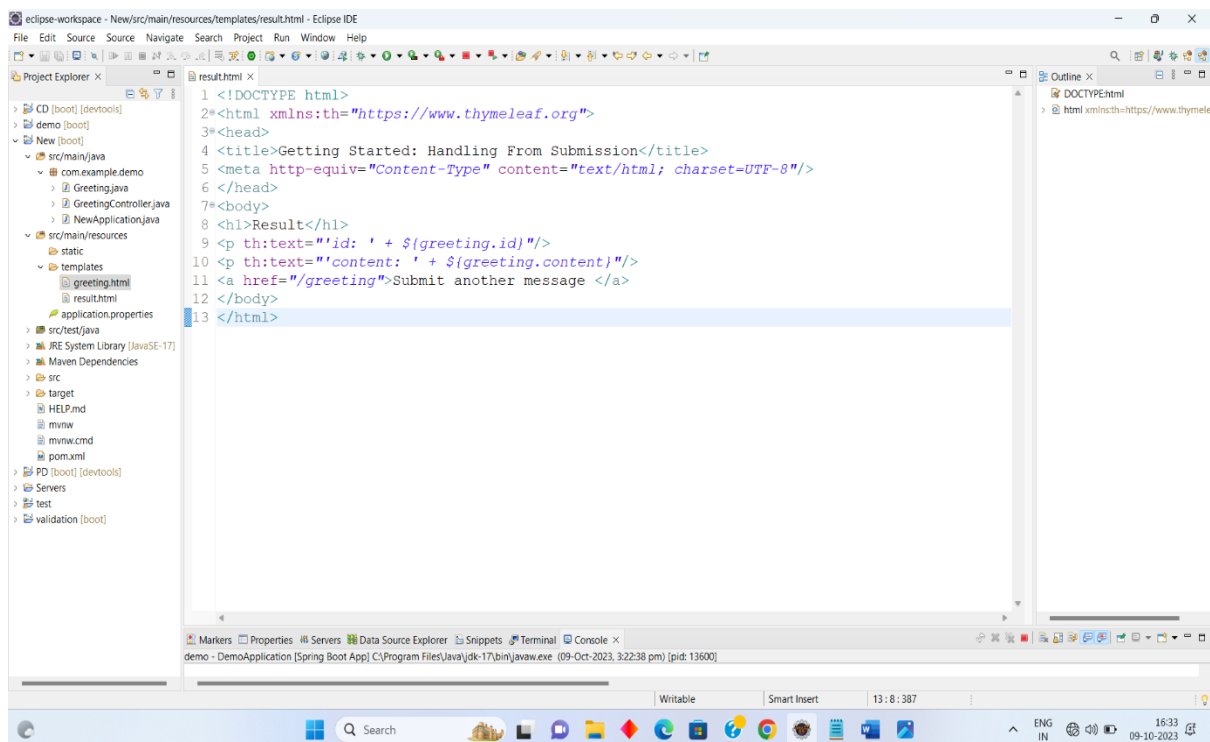
[< Back](#) [Next >](#) [Finish](#) [Cancel](#)







```
1 <!DOCTYPE html>
2 <html xmlns:th="https://www.thymeleaf.org">
3 <head>
4 <title>Getting Started: Handling From Submission</title>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
6 </head>
7 <body>
8 <h1>Form</h1>
9 <form action="#" th:action="@{/greeting}" th:object="${greeting}" method="post">
10 <p>ID:<input type="text" th:field="**{id}"/></p>
11 <p>Message: <input type="text" th:field="**{content}"/></p>
12 <p><input type="submit" value="Submit" /><input type="reset" value="Reset"/></p>
13 </form>
14 </body>
15 </html>
```



```
1 <!DOCTYPE html>
2 <html xmlns:th="https://www.thymeleaf.org">
3 <head>
4 <title>Getting Started: Handling From Submission</title>
5 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
6 </head>
7 <body>
8 <h1>Result</h1>
9 <p th:text="id: ' + ${greeting.id}"/>
10 <p th:text="content: ' + ${greeting.content}"/>
11 <a href="/greeting">Submit another message </a>
12 </body>
13 </html>
```

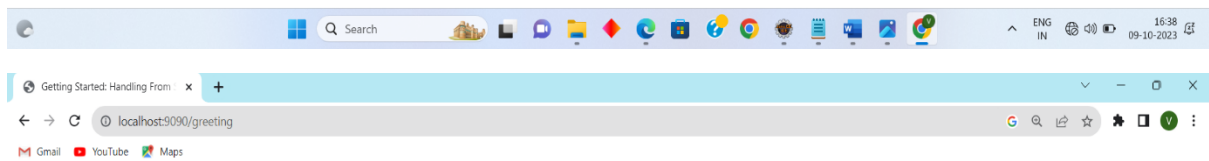
Output:-



Form

ID:

Message:



Result

id: 12

content: hello

[Submit another message](#)

