

Thymeleaf

Thymeleaf

Thymeleaf è un motore di template Java per la creazione di pagine web. È un motore di template basato su XML/HTML5 che può essere utilizzato sia in ambiente web che non web (ad esempio, report, email, ecc.).

Thymeleaf è de facto il motore di template standard per Spring.

Setup

Useremo **Spring Initializr** per creare un progetto Spring Boot con Thymeleaf.

Controller

HelloWorldController.java

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

@Controller
public class HelloWorldController {

    @GetMapping("/hello")
    public String hello(Model model) {
        model.addAttribute("message", "Hello World!");
        return "helloworld";
    }
}
```

Spring Boot configura automaticamente il `view resolver` di Thymeleaf per risolvere i modelli Thymeleaf dalla cartella `resources/templates`.

Nel caso dello snippet precedente, il metodo `hello` restituirà il modello `helloworld`, ovvero andrà a cercare il file `helloworld.html` nella cartella `resources/templates`.

Template

helloworld.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <meta charset="ISO-8859-1">
    <title>Hello World App</title>
  </head>
  <body>
    <h1 th:text="'Thymeleaf ' + ${message} + ' App'"> </h1>
  </body>
</html>
```

Questo template HTML utilizza Thymeleaf per visualizzare il messaggio `Hello World!` .

`th:text` è un attributo Thymeleaf che viene utilizzato per visualizzare il valore di una variabile all'interno di un tag HTML.

`${message}` è una variabile Thymeleaf che viene passata dal controller al modello.

```
( model.addAttribute("message", "Hello World!"); )
```

Oltre a `th:text` , Thymeleaf supporta molti altri attributi per la manipolazione dei tag HTML, come `th:if` , `th:each` , `th:href` , ecc.

Risultato

Se eseguiamo l'applicazione e visitiamo `http://localhost:8080/hello`, vedremo il messaggio `Thymeleaf Hello World! App`.

th:each

loop.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <meta charset="ISO-8859-1">
    <title>Thymeleaf Loop Example</title>
  </head>
  <body>
    <h1>Thymeleaf Loop Example</h1>
    <ul>
      <li th:each="name : ${names}" th:text="${name}"></li>
    </ul>
  </body>
</html>
```

```
@GetMapping("/loop")
public String loop(Model model) {
    List<String> names = Arrays.asList("John", "Doe", "Alex");
    model.addAttribute("names", names);
    return "loop";
}
```