

Thymeleaf

1

Sommaire

- Thymeleaf.
- Spring Expression Language.
- Attributs du modèle de vues.
- Paramètres de requête.
- Attributs de session.

Thymeleaf

- Thymeleaf est un moteur de templates Java destiné principalement aux développement web. Il offre une solution élégante et puissante pour intégrer des templates HTML avec des données provenant d'application Java.
- Thymeleaf est un logiciel Open Source créé à l'origine par un ingénieur logiciel espagnol appelé Daniel Fernández.

Spring Expression Language

Spring EL est un langage qui prend en charge l'interrogation et la manipulation d'un graphe d'objets lors de l'exécution.

```
<html xmlns:th="http://www.thymeleaf.org"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout">
```

```
<div th:fragment="header">
    <h1>Welcome to the website</h1>
</div>
```

```
<div th:replace="fragments/header"></div>
```

```
<div th:include="fragments/header"></div>
```

```
<p>Welcome, <span th:text="${user.name}"></span>!</p>
```

```
<div th:if="${user.isAdmin}">  
  <p>Welcome, Admin!</p>  
</div>  
<div th:unless="${user.isAdmin}">  
  <p>Welcome, User!</p>  
</div>
```

```
<p th:if="${user.age > 18}">Majeur</p>
```

```
<p th:text="${user.isAdmin ? 'Admin' : 'Utilisateur'}"></p>
```

```
<ul>  
  <li th:each="product : ${products}" th:text="${product.name}"></li>  
</ul>
```

```
<p th:text="${subtotal + (subtotal * taxRate)}"></p>
```

Attributs du modèle de vues

Dans le langage Thymeleafv, une variables de contexte.


```
@RequestMapping(value = "message", method = RequestMethod.GET)
public String messages(Model model) {
    model.addAttribute("messages", messageRepository.findAll());
    return "message/list";
}
```

```
<tr th:each="message : ${messages}">
    <td th:text="${message.id}">1</td>
    <td><a href="#" th:text="${message.title}">Title ...</a></td>
    <td th:text="${message.text}">Text ...</td>
</tr>
```

```
@RequestMapping(value = "message", method = RequestMethod.GET)
public ModelAndView messages() {
    ModelAndView mav = new ModelAndView("message/list");
    mav.addObject("messages", messageRepository.findAll());
    return mav;
}
```

```
<tr th:each="message : ${messages}">
    <td th:text="${message.id}">1</td>
    <td><a href="#" th:text="${message.title}">Title ...</a></td>
    <td th:text="${message.text}">Text ...</td>
</tr>
```

```
@ModelAttribute("messages")  
public List<Message> messages() {  
    return messageRepository.findAll();  
}
```

```
<tr th:each="message : ${messages}">  
    <td th:text="${message.id}">1</td>  
    <td><a href="#" th:text="${message.title}">Title ...</a></td>  
    <td th:text="${message.text}">Text ...</td>  
</tr>
```

Paramètres de requête

`https://example.com/query?q=Thymeleaf+Is+Great!`

```
@Controller
public class SomeController {
    @RequestMapping("/")
    public String redirect() {
        return "redirect:/query?q=Thymeleaf+Is+Great!";
    }
}
```

```
<p th:text="${param.q}">Test</p>
```

Attributs de session

```
@RequestMapping("/")
String index(HttpSession session) {
    session.setAttribute("mySessionAttribute", "someValue");
    return "index";
}
```

```
<p th:text="${session.mySessionAttribute}" th:unless="${session == null}">[...]</p>
```

Ce code définit un attribut de session dans une méthode de contrôleur, puis l'affiche dans une vue Thymeleaf en utilisant la syntaxe `${session.mySessionAttribute}`.

Cela permet de stocker des données spécifiques à une session utilisateur et de les afficher dans différentes vues de l'application.

Sources

- <https://www.thymeleaf.org/>
- <https://www.thymeleaf.org/doc/articles/springmvcaccessdata.html>
- <https://www.baeldung.com/>