



Thymeleaf

ThymeLeaf

01.

Qué es

02.

Características

03.

Integración con Spring

04.

Etiquetas, expresiones
e interpolación



Thymeleaf es un motor de plantillas (template engine) Java. Se utiliza para generar contenido HTML de manera dinámica y se integra a la perfección con el framework Spring.

Permite mezclar HTML con código Java para crear aplicaciones web dinámicas, robustas, personalizables y escalables

La sintaxis de Thymeleaf es intuitiva y fácil de aprender. Utilizando atributos especiales en tus etiquetas HTML, como `th:text` y `th:if`, permite manipular dinámicamente el contenido.

Permite incrustar datos dinámicos en las plantillas HTML, para que las páginas web muestren información que puede cambiar según la lógica de la aplicación.

Características y usos de Thymeleaf

Generación dinámica de contenido HTML:

Permite incrustar datos dinámicos en las plantillas HTML, muestran información que puede cambiar según la lógica de la aplicación.

Sintaxis legible:

Utiliza una sintaxis fácil de leer y comprender, lo que lo hace accesible para desarrolladores y diseñadores web por igual. Los marcadores Thymeleaf se integran en el código HTML utilizando atributos y elementos específicos, lo que hace que las plantillas sean más legibles.

Características y usos de Thymeleaf

Integración con frameworks web:

Se puede integrar fácilmente con varios frameworks y tecnologías web en el ecosistema Java, como Spring Framework, Spring Boot y otros.

Soporte para internacionalización y localización:

Incluye características para manejar de manera efectiva la internacionalización y localización en aplicaciones web, lo que facilita la creación de sitios web multilingües.

Se ejecuta en el servidor:

Los datos se procesan y se generan las vistas en el servidor antes de enviarlas al navegador del usuario. Esto lo diferencia de algunas tecnologías JavaScript que generan contenido en el lado del cliente.

Integración con Spring

Thymeleaf es una herramienta muy útil para desarrolladores Java que desean crear vistas web dinámicas y flexibles. Facilita la incorporación de datos dinámicos en plantillas HTML y es especialmente popular en el contexto de aplicaciones web basadas en Java y Spring Framework.

Thymeleaf se integra perfectamente con el framework Spring, lo que te permite aprovechar al máximo su potencial. Puedes usar Thymeleaf en combinación con Spring MVC para crear



Etiquetas

Sintaxis de las etiquetas: Las etiquetas Thymeleaf se integran en el código HTML utilizando atributos y elementos específicos. Aquí tienes algunos ejemplos de etiquetas Thymeleaf:

Th:if y Th:unless: Se utilizan para crear estructuras condicionales en la plantilla HTML. Por ejemplo, puedes mostrar o ocultar contenido basado en una condición:

```
<p th:if="${loggedIn}">Bienvenido,  
usuario</p>
```

```
<p th:unless="${loggedIn}">Por  
favor, inicia sesión</p>
```

Th:each: Se utiliza para iterar sobre una lista de elementos y generar contenido repetitivo:

```
<ul>  
  <li th:each="item : ${items}"  
    th:text="${item}">Item</li>  
</ul>
```

Th:attr: Permite modificar atributos HTML. Por ejemplo, puedes cambiar dinámicamente el valor de un atributo "src" de una imagen:

```

```

Th:text: Se utiliza para establecer el texto de un elemento HTML basado en una expresión:

```
<p th:text="${message}">  
Mensaje  
predeterminado</p>
```


Expresiones Las expresiones Thymeleaf se utilizan dentro de las etiquetas para acceder a datos y realizar operaciones.

Por ejemplo, `${loggedIn}` es una expresión que accede a la variable `loggedIn`.

Interpolación Thymeleaf también permite la interpolación de texto dentro de atributos, lo que facilita la inclusión de valores dinámicos en cualquier atributo HTML. Por ejemplo:

```
<a th:href="@{/profile/' + ${userId}'}">Perfil del  
usuario</a>
```


Conclusión

Thymeleaf es una herramienta creativa y poderosa que permite crear plantillas enriquecidas para proyectos web. Su integración con Spring, su sintaxis sencilla y sus numerosas características lo convierten en una opción ideal para desarrolladores.

