Softwareentwicklung (SW)

Spring Boot und Thymeleaf Layout

Prof. Dr. Alixandre Santana
alixandre.santana@oth-regensburg.de

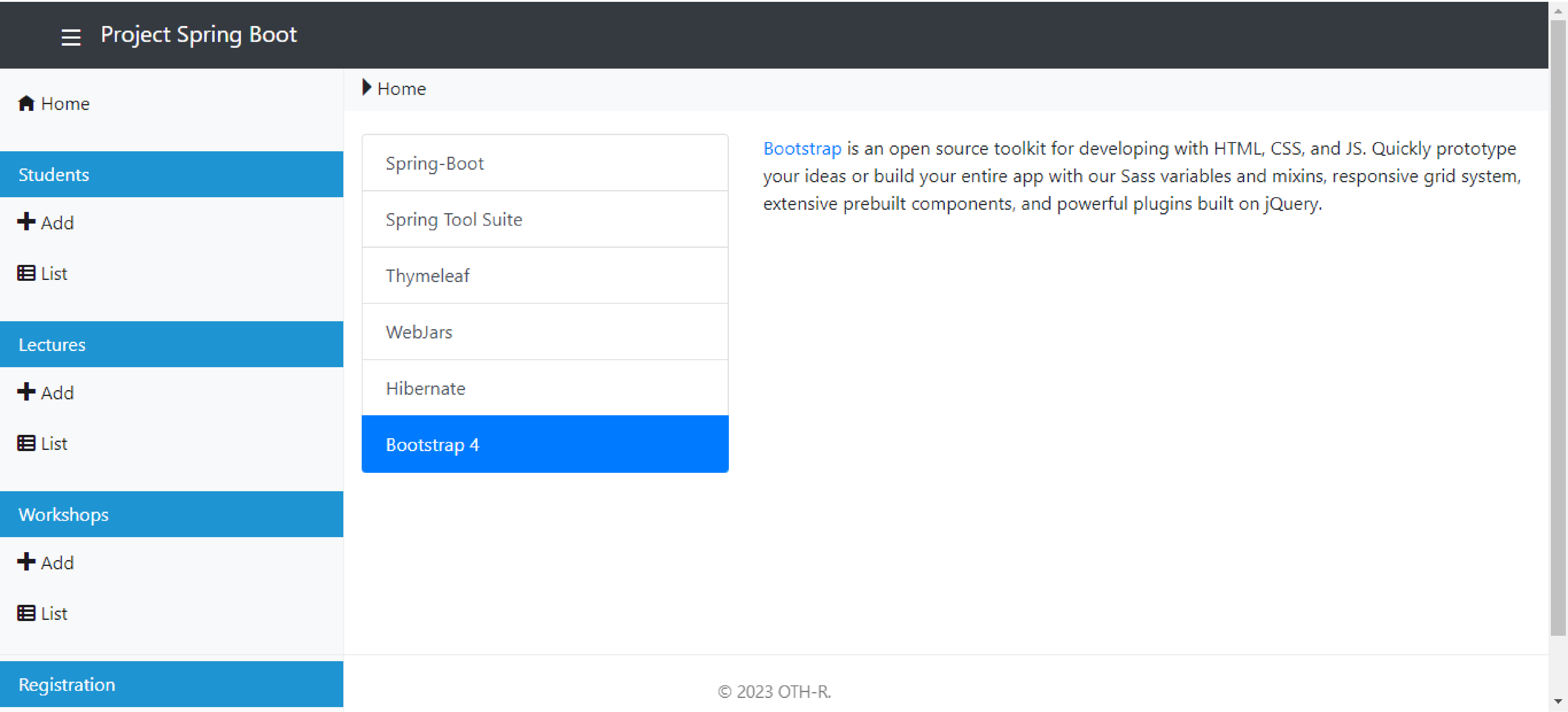
Wintersemester
2023/2024

Lernziele

- Das Konzept eines Layouts einer Ansicht zu verstehen
- Die Thymeleaf-Layout-Engine zu verstehen

Agenda

1. Motivation
2. Thymeleaf Layouts
3. Include-Style Layouts
4. Hierarchical Style

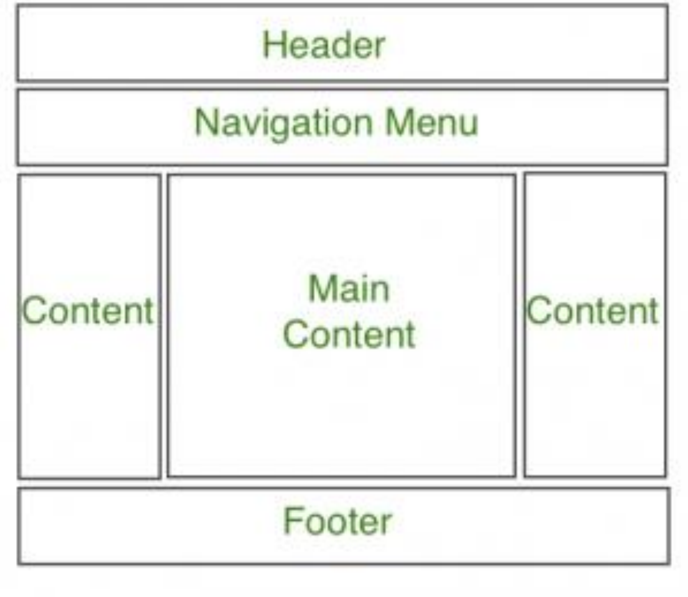


Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

© 2023 OTH-R.

1. Motivation

- Normalerweise teilen Websites gemeinsame Seitenkomponenten wie „header“, „foot“, „Menü“ und möglicherweise viele mehr.
- Diese Seitenkomponenten können von denselben oder verschiedenen Layouts verwendet werden.



Agenda

1. Motivation
2. Thymeleaf Layouts
3. Include-Style Layouts
4. Hierarchical Style

2. Thymeleaf Layouts

- Mithilfe von Fragmenten in Thymeleaf können Sie Abschnitte Ihres HTML-Codes definieren, die in mehreren Vorlagen wiederverwendet werden können.
- Anstatt Code für allgemeine Abschnitte wie head- und foot oder Navigationsleisten zu duplizieren, können Sie diese Teile als Fragmente kapseln und bei Bedarf einbinden.

- Es gibt zwei Hauptstile zum Organisieren von Layouts in Projekten : ***include* style** **und** ***hierarchical* style**.

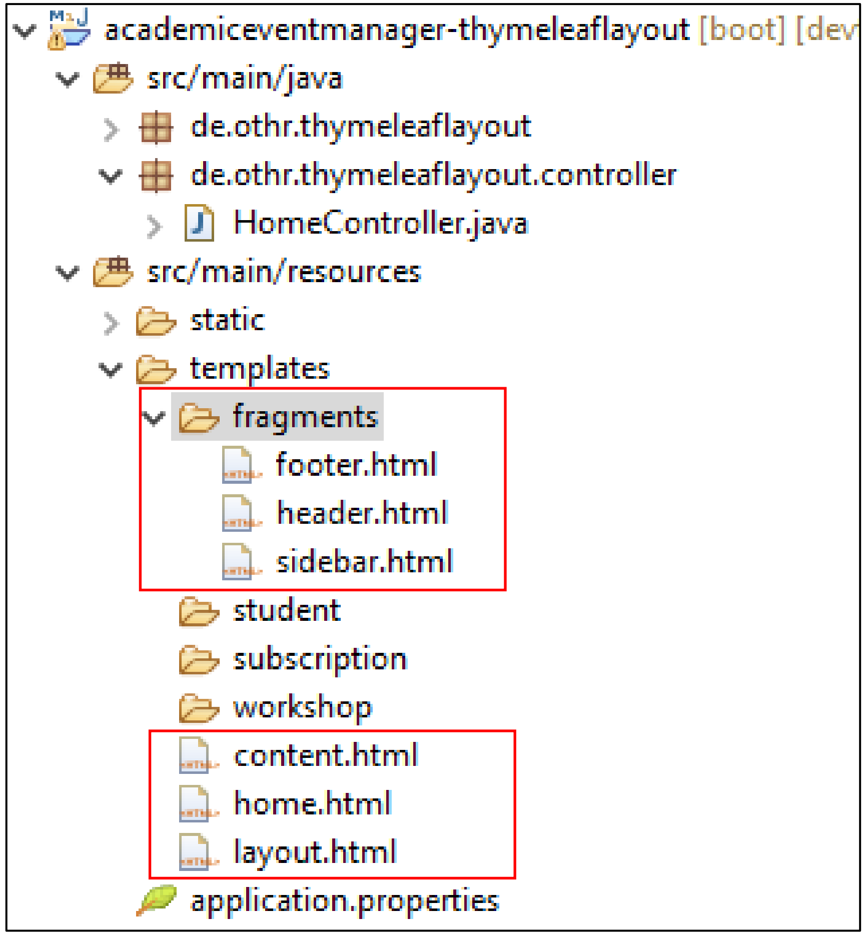
2.1 Thymeleaf Layouts

- Setup

```
<dependency>
    <groupId>nz.net.ultraq.thymeleaf</groupId>
    <artifactId>thymeleaf-layout-dialect</artifactId>
    <version>3.0.0</version>
</dependency
```

2.1 Thymeleaf Layouts

- Setup



```
academiceventmanager-thymeleaflayout [boot] [dev
  src/main/java
    de.othr.thymeleaflayout
    de.othr.thymeleaflayout.controller
      HomeController.java
  src/main/resources
    static
    templates
      fragments
        footer.html
        header.html
        sidebar.html
      student
      subscription
      workshop
      content.html
      home.html
      layout.html
    application.properties
```

2.1 Thymeleaf Layouts

- Setup:Um Thymeleaf Dialect in unseren HTML-Seiten verwenden zu können, müssen wir mithilfe des folgenden Codes Unterstützung hinzufügen:

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:th=http://www.thymeleaf.org
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
>
 ...
</html>
```

2.1 Thymeleaf Layouts

- Setup:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:th="http://www.thymeleaf.org"
xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout" >
<body>
```

1

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

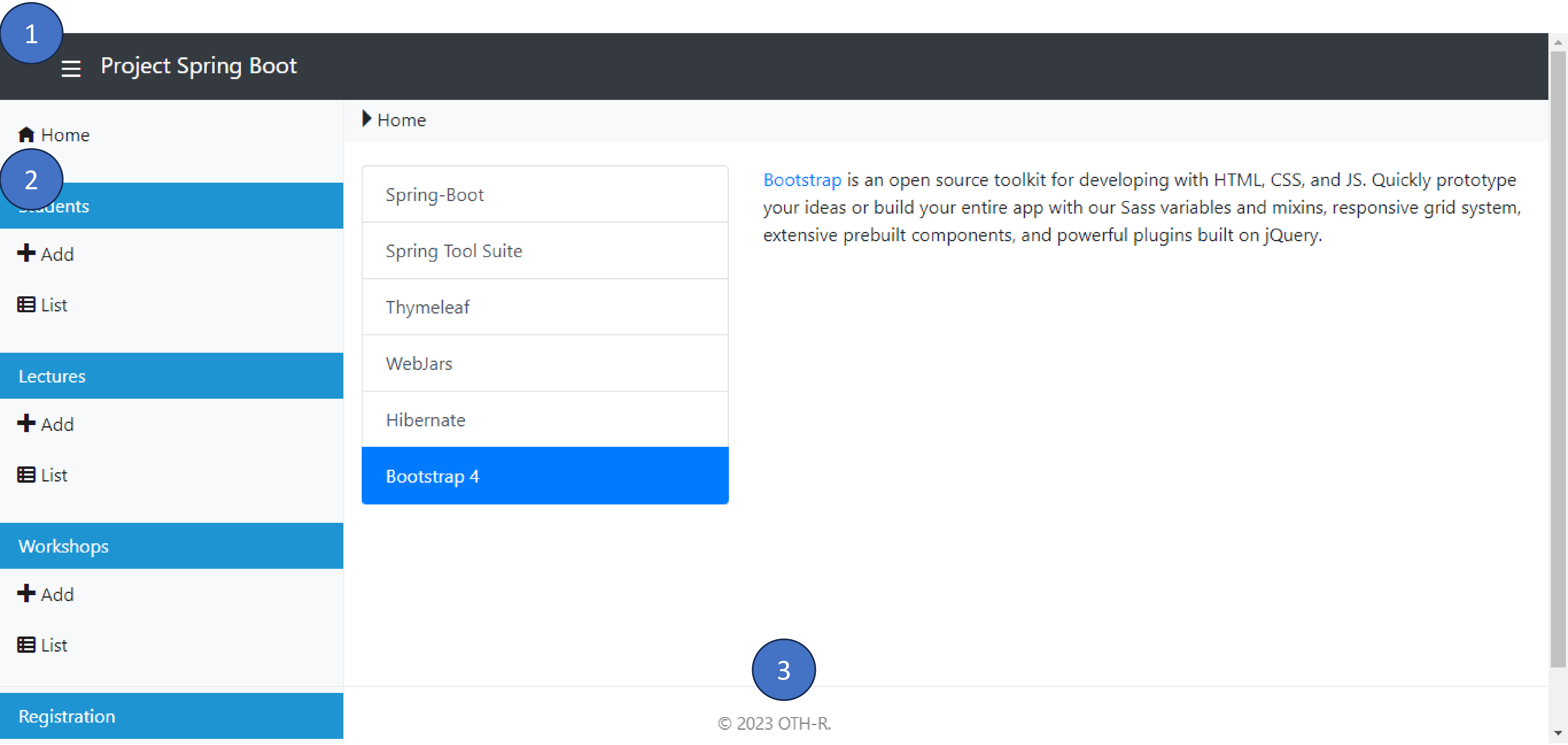
2

```
    <div class="layout-main">
        <aside>
        <nav th:replace="~{fragments/sidebar :: nav-sidebar}">
        <span>menu</span>
        </nav>
        </aside>
        <section layout:fragment="mybody">
        <div> main content </div>
        </section>
     </div> <!--end div class="layout-main" -->
```

3

```
    <footer class="layout-footer">
    <div th:replace="~{fragments/footer::footer}">
    </div>
    </footer>
</body>
```

Layout.html



Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

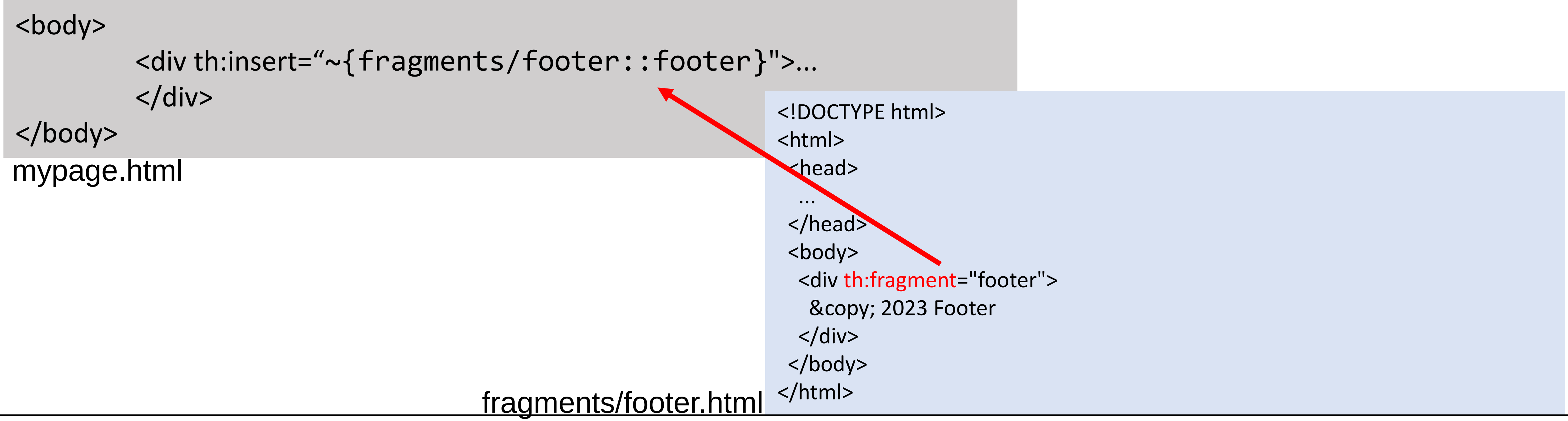
© 2023 OTH-R.

Agenda

1. Motivation
2. Thymeleaf Layouts
3. Include-Style Layouts
4. Hierarchical Style

3. Thymeleaf “Include”- style layouts

- In this style pages are built by embedding common page component code directly within each view to generate the final result. In Thymeleaf this can be done using Thymeleaf Standard Layout System:



```
<body>
        <div th:insert=“~{fragments/footer::footer}">...
        </div>
</body>
```

mypage.html

```
<!DOCTYPE html>
<html>
  <head>
   ...
  </head>
  <body>
    <div th:fragment="footer">
      &copy; 2023 Footer
    </div>
  </body>
</html>
```

fragments/footer.html

3. Thymeleaf “Include”- style layouts

- Note how that the referenced fragments are specified with th:fragment attributes. This way we can define multiple fragments in one template file.’

```
<!DOCTYPE html>
<html>
  <head>
    ...
  </head>
  <body>
    <div th:fragment="footer">
      &copy; 2023 Footer
    </div>
  </body>
</html>
```

fragments/footer.html

3. Thymeleaf “Include”- style layouts

- In templatename :: selector, both “templatename” and “selector” can be dynamic expressions.
- In the below example we want to include different fragments depending on a condition. If the authenticated user is an Admin, we will show a different footer than for a regular user:

```
<div th:replace="fragments/footer :: ${#authentication.principal.isAdmin()} ? 'footer-admin' : 'footer'">
  &copy; 2023 The Static Templates
</div>
```

anypage.html

3. Thymeleaf “Include”- style layouts

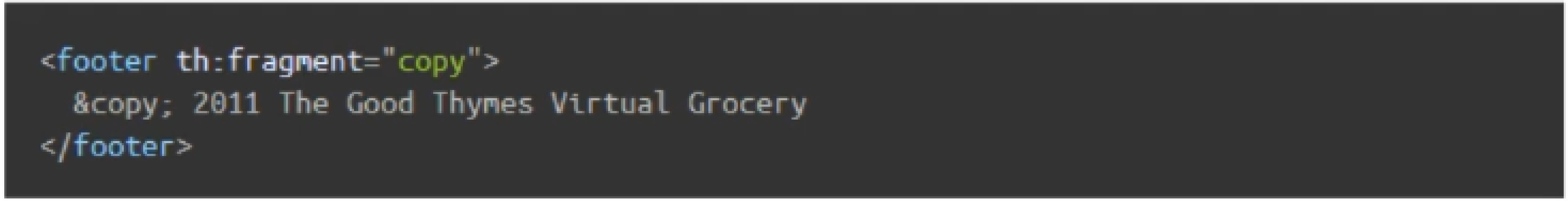
- fragments/footer.html has slightly changed, as we need to have two footers defined:

```
<!DOCTYPE html>
<html>
  <head>
    ...
  </head>
  <body>
    <!-- /*  Multiple fragments may be defined in one file */-->
    <div th:fragment="footer“>
      &copy; 2023 Footer
    </div>
    <div th:fragment="footer-admin">
      &copy; 2023 Admin Footer
    </div>
  </body>
</html>
```

fragments/footer.html

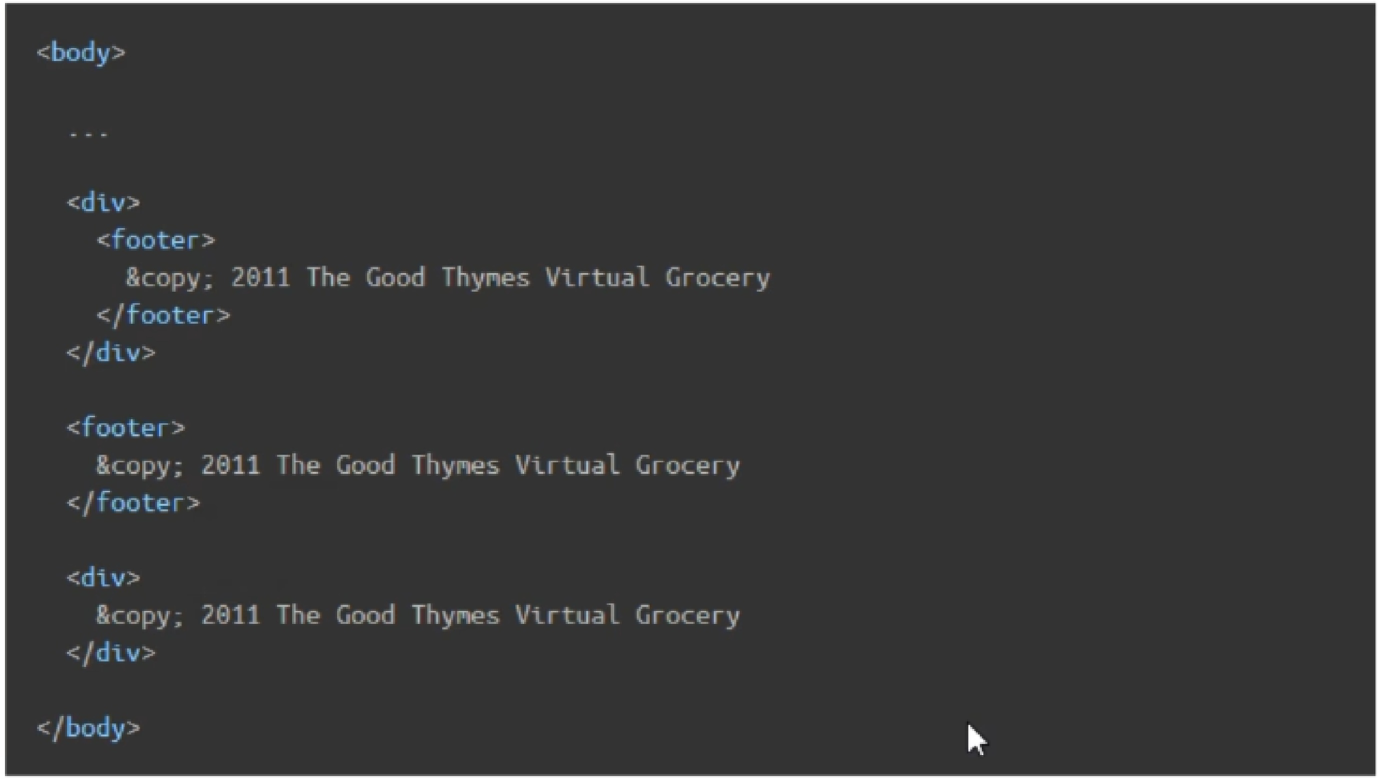
3. Unterschiede zwischen th:insert, replace, include

fragments/footer.html



```
<footer th:fragment="copy">
  &copy; 2011 The Good Thymes Virtual Grocery
</footer>
```

```
<body>

 ...

 <div th:insert="~{footer :: copy}"></div>

 <div th:replace="~{footer :: copy}"></div>

<div th:include="~{footer :: copy}"></div>

</body>
```



```
<body>

  ...

  <div>
    <footer>
      &copy; 2011 The Good Thymes Virtual Grocery
    </footer>
  </div>

  <footer>
    &copy; 2011 The Good Thymes Virtual Grocery
  </footer>

  <div>
    &copy; 2011 The Good Thymes Virtual Grocery
  </div>

</body>
```

Agenda

1. Motivation
2. Thymeleaf Layouts
3. Include-Style Layouts
4. Hierarchical Style

4. Thymeleaf Layout – Hierarchical Style

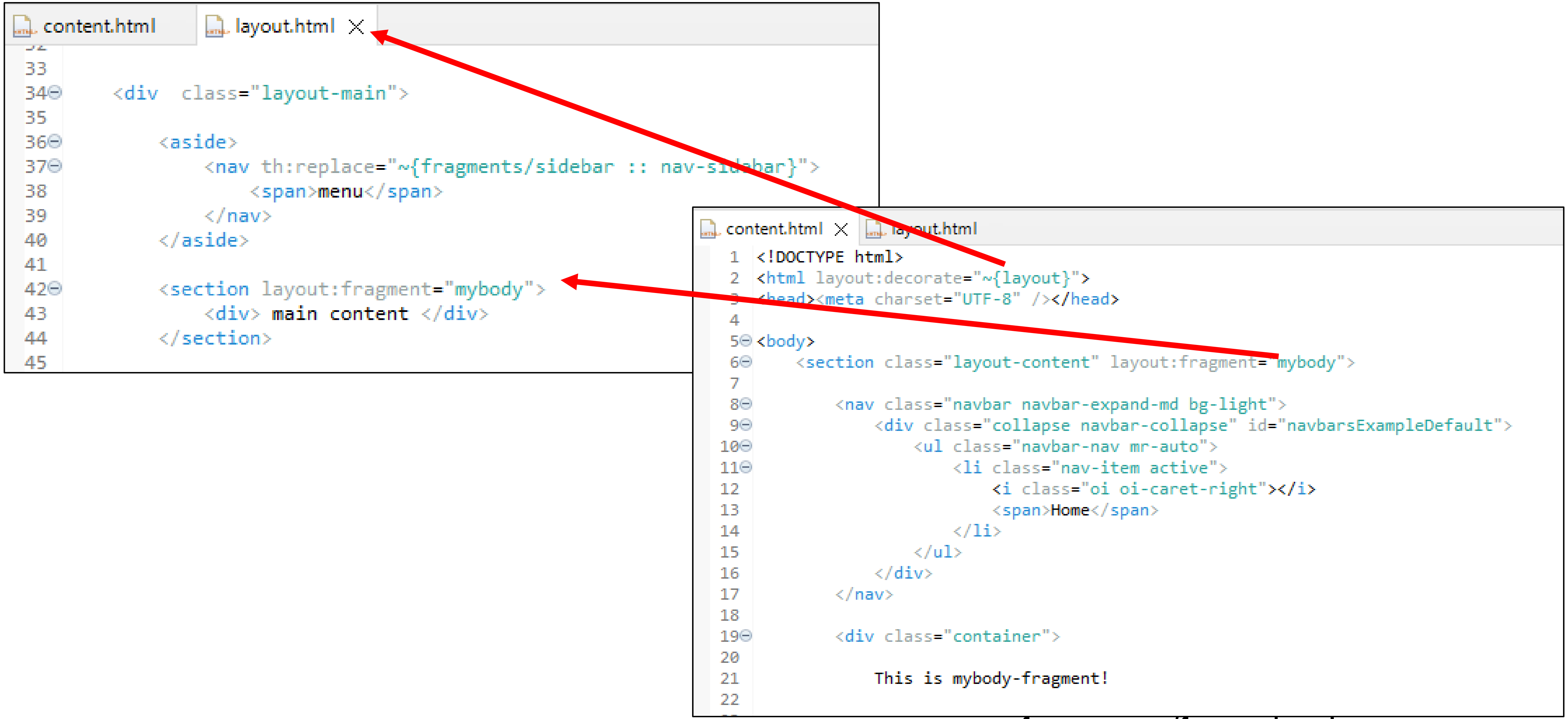
- **In hierarchical style**, werden normalerweise mit einer Eltern-Kind-Beziehung erstellt, vom allgemeineren Teil (z.B. „layout.html") bis zu den spezifischsten (z. B. Seite „content.html").

- Jede Komponente der Vorlage kann dynamisch eingebunden werden, basierend auf der Einbeziehung und Ersetzung von Vorlagenfragmenten.

- Die Hauptvorteile dieser Lösung sind die Wiederverwendung atomarer Teile der Ansicht und der modulare Aufbau

fragments/footer.html

4. Thymeleaf Layout – Hierarchical Style



```
    <div  class="layout-main">

        <aside>
            <nav th:replace="~{fragments/sidebar :: nav-sidebar}">
                <span>menu</span>
            </nav>
        </aside>

        <section layout:fragment="mybody">
            <div> main content </div>
        </section>
```

```
<!DOCTYPE html>
<html layout:decorate="~{layout}">
<head><meta charset="UTF-8" /></head>

<body>
    <section class="layout-content" layout:fragment="mybody">

        <nav class="navbar navbar-expand-md bg-light">
            <div class="collapse navbar-collapse" id="navbarsExampleDefault">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item active">
                        <i class="oi oi-caret-right"></i>
                        <span>Home</span>
                    </li>
                </ul>
            </div>
        </nav>

        <div class="container">

            This is mybody-fragment!

```

fragments/footer.html

Referenzen

- https://techblogstation.com/spring-boot/thymeleaf-with-spring-boot/
- https://ultraq.github.io/thymeleaf-layout-dialect/getting-started/
- https://www.thymeleaf.org/doc/articles/layouts.html
- https://www.youtube.com/watch?v=LRZb02GCcrM
- https://bootstrapious.com/p/bootstrap-multilevel-dropdown

fragments/footer.html