Softwareentwicklung (SW)

Pagination Pattern mit Spring Data

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

Wintersemester
2023/2024

Lernziele

- das Pattern Pagination zu verstehen
- die Klassen von Spring Data, die das Pagination unterstützen
- ein Pagination zu implementieren

Pagination - Motivation

- Clients fragen häufig das Backend ab und rufen Sammlungen von Datenelementen ab, die dem Benutzer angezeigt oder in anderen Anwendungen verarbeitet werden sollen.
- Bei vielen solchen Anfragen antwortet die Backend-Seite mit dem Senden einer großen Anzahl von Elementen.
- Der Umfang dieser Antwort kann größer sein als das, was der Kunde benötigt oder bereit ist zu konsumieren.

https://getbootstrap.com/docs/4.0/components/pagination/

Pagination - Motivation

- Es wird empfohlen, die Paginierungsfunktion zu verwenden, wenn eine große Datenmenge den Suchkriterien entspricht.
- Das gleichzeitige Abrufen und Anzeigen großer Datenmengen auf dem Bildschirm kann zu folgenden Problemen führen.
  - Speichererschöpfung auf Serverseite

    java.lang.OutOfMemoryError tritt auf, wenn mehrere Anforderungen gleichzeitig ausgeführt werden.
  - Netzwerklast

    Die Übertragung unnötiger Daten über das Netzwerk führt zu einer erhöhten Netzwerklast und kann dadurch die Reaktionszeit des gesamten Systems beeinträchtigen.
  - Verzögerung der Reaktion auf dem Bildschirm

    Der Serverprozess, der Netzwerkverkehrsprozess und der Client-Rendering-Prozess benötigen Zeit, um große Datenmengen zu verarbeiten.

https://getbootstrap.com/docs/4.0/components/pagination/

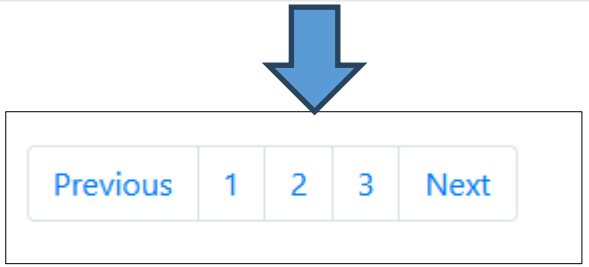
Pagination -View

https://terasolunaorg.github.io/guideline/5.2.1.RELEASE/en/ArchitectureInDetail/WebApplicationDetail/Pagination.html

Pagination – Komponent für die Views

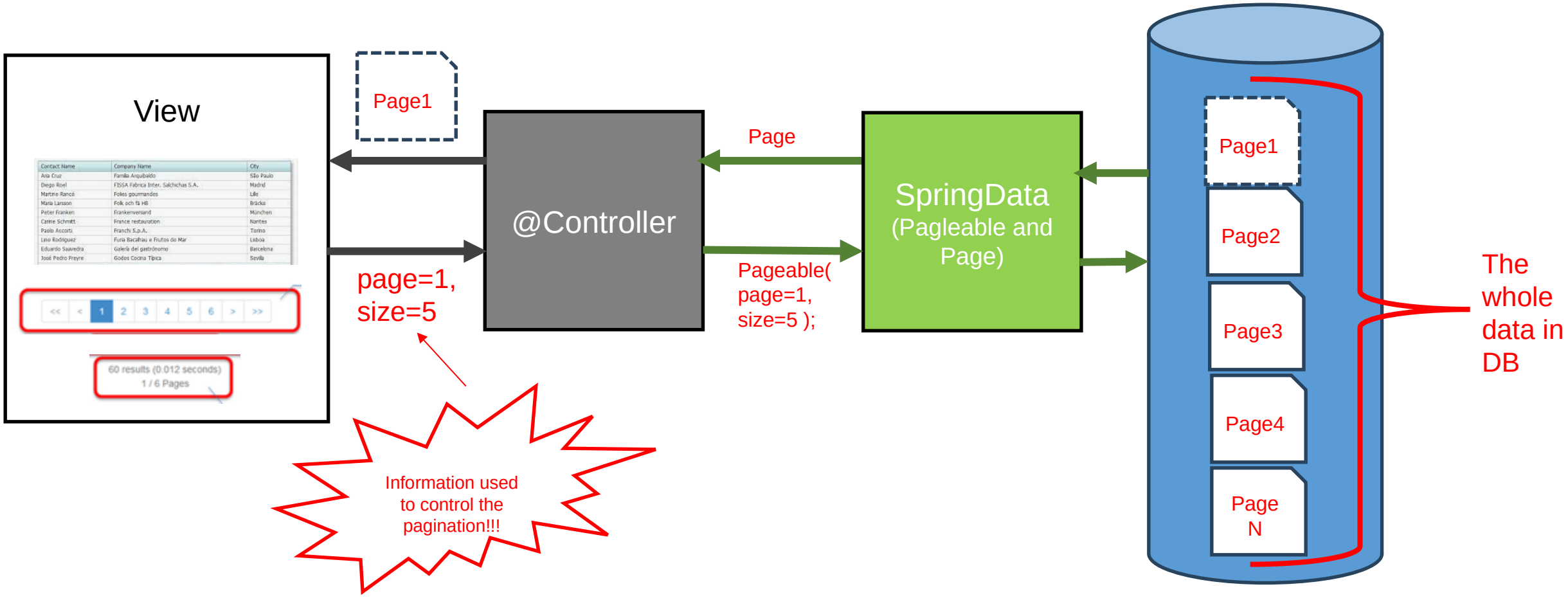
- Mithilfe von Bootstrap gibt es mehrere Möglichkeiten von Komponenten zur Darstellung der Seitendaten und der Controller für die Paginierung. **Choose Yours!**

```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Previous</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
</nav>
```



https://getbootstrap.com/docs/4.0/components/pagination/

Pagination – View- Controller und Repository Interaktion



https://terasolunaorg.github.io/guideline/5.2.1.RELEASE/en/ArchitectureInDetail/WebApplicationDetail/Pagination.html

Pagination mit Spring Data

- org.springframework.data.domain.Pageable

Wir extrahieren die für die Seitensuche erforderlichen Informationen (Speicherort der zu durchsuchenden Seite, Anzahl der abzurufenden Datensätze und Sortierbedingung) aus dem Anforderungsparameter und übergeben die extrahierten Informationen als Objekte von org.springframework.data.domain.Pageable an die Repository-Layer.

- org.springframework.data.domain.Page

Es speichert die Seiteninformationen (Gesamtzahl der Datensätze, Daten der entsprechenden Seite, Speicherort der zu durchsuchenden Seite, Anzahl der abzurufenden Datensätze und Sortierbedingung).

https://getbootstrap.com/docs/4.0/components/pagination/

Pagination – Spring Data „Page“

Schauen wir uns das Page-Objekt an.

Page ist eine Unterschnittstelle von Slice mit einigen zusätzlichen Methoden. Es enthält die Gesamtzahl der Elemente und die Gesamtseitenzahl der gesamten Liste.

```
public interface Page<T> extends Slice<T> {
  static <T> Page<T> empty();
  static <T> Page<T> empty(Pageable pageable);
  long getTotalElements();
  int getTotalPages();
  <U> Page<U> map(Function<? super T,? extends U> converter);
}
```

https://www.bezkoder.com/thymeleaf-pagination/

Pagination -Spring Data “Pageable”

Jetzt sehen wir den Pageable-Parameter für das „Repository“. Die Spring Data-Infrastruktur erkennt diesen Parameter automatisch, um Paginierung und Sortierung auf die Datenbank anzuwenden.

Die Pageable-Schnittstelle enthält die Informationen über die angeforderte Seite, wie z. B. die Größe und die Nummer der Seite.

```
public interface Pageable {
  int getPageNumber();
  int getPageSize();
  long getOffset();
  Sort getSort();
  Pageable next();
  Pageable previousOrFirst();
  Pageable first();
  boolean hasPrevious();
  ...
}
```

https://getbootstrap.com/docs/4.0/components/pagination/

Pagination – Changes in the @Repository

Wir müssen unserem Repository die Methoden hinzufügen, um die Abfrage mit Paginierung durchzuführen.

```
public interface CourseRepositoryI extends MyBaseRepository<Course, Long>
{


List<Course> findByDescriptionContainingIgnoreCase (String description);
Page <Course> findAll(Pageable pageable);
Page <Course> findByDescriptionContainingIgnoreCase (String description,
Pageable pageable);


}
```

https://getbootstrap.com/docs/4.0/components/pagination/

Pagination – Changes in the @Controller

CourseController.java

```
@GetMapping(value = {"", "/"})
public String showCourseList(Model model, @RequestParam(required = false) String keyword,
@RequestParam(required = false, defaultValue = "1") int page, @RequestParam(required = false, defaultValue =
"5") int size) {
try {
     List<Course> courses = new ArrayList<Course>();
     Pageable paging = PageRequest.of(page - 1, size);

     Page<Course> pageCourses;
     pageCourses = courseService.getAllCourses(keyword, paging);
     model.addAttribute("keyword", keyword);

     courses = pageCourses.getContent();
     //information used by the pagination component in the view:
     model.addAttribute("courses", courses);
     model.addAttribute("currentPage", pageCourses.getNumber() + 1);
     model.addAttribute("totalItems", pageCourses.getTotalElements());
     model.addAttribute("totalPages", pageCourses.getTotalPages());
     model.addAttribute("pageSize", size);
} catch (Exception e) {
     model.addAttribute("message", e.getMessage());
}
  return "/courses/course-all";
}
```

https://getbootstrap.com/docs/4.0/components/pagination/

Pagination – Changes in the View (1/2)

/courses/course-all.html

```
<!--pagination code -->
<nav aria-label="Pagination" th:if="${totalPages > 0}">
     <ul class="pagination justify-content-center">
     <li class="page-item" th:classappend="${currentPage == 1} ? 'disabled'">
     <a th:replace="~{fragments/paging :: paging(1, '<<', 'First Page')}"></a>
     </li>
     <li class="page-item font-weight-bold" th:classappend="${currentPage == 1} ? 'disabled'">
     <a th:replace="~{fragments/paging :: paging(${currentPage - 1}, 'Prev', 'Previous Page')}"></a>
     </li>
     <li class="page-item disabled" th:if="${currentPage - 2 > 1}">
     <a class="page-link" href="#">...</a>
     </li>
     <li class="page-item" th:classappend="${page == currentPage} ? 'active'"
     th:each="page : ${#numbers.sequence(currentPage > 2 ? currentPage - 2 : 1, currentPage + 2 < totalPages ?
     currentPage + 2 : totalPages)}">
     <a th:replace="~{fragments/paging :: paging(${page}, ${page}, 'Page ' + ${page})}"></a>
     </li>
     <li class="page-item disabled" th:if="${currentPage + 2 < totalPages}">
     <a class="page-link" href="#">...</a>
     </li>
     <li class="page-item font-weight-bold" th:classappend="${currentPage == totalPages} ? 'disabled'">
     <a th:replace="~{fragments/paging :: paging(${currentPage + 1},'Next', 'Next Page')}"></a>
     </li>
     <li class="page-item" th:classappend="${currentPage == totalPages} ? 'disabled'">
     <a th:replace="~{fragments/paging :: paging(${totalPages}, '>>', 'Last Page')}"></a>
     </li>
     </ul>
</nav>
```

Pagination – Changes in the View (2/2)

/fragments/paging.html

```
<!--pagination code -->
<a th:fragment="paging(pageNum, label, tooltip)" class="page-link"
th:href="@{'/student?' + ${keyword!=null && keyword!=''? 'keyword=' + keyword +
'&' : ''} + 'page=' + ${pageNum} + '&size=' + ${pageSize}}"
th:title="${tooltip}" rel="tooltip">
[[${label}]]
</a>
```

Referenzen

https://www.bezkoder.com/thymeleaf-pagination/
https://microservice-api-patterns.org/patterns/quality/dataTransferParsimony/Pagination

https://getbootstrap.com/docs/4.0/components/pagination/