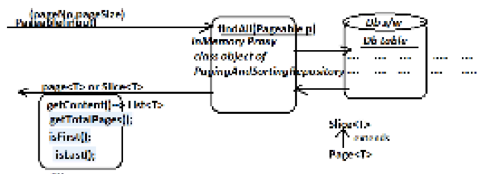
Spring Boot MVC with spring data jpa Pagination

Page<T> findAll(Pageable pageable); (PagingAndSortingRepository)

=> This method takes pageNo(0 based),pageSize as inputs in the form of Pageable obj and returns output as Page<T> /Slice<T> obj having requested page records , pages count , current page numbers ,total records and etc..



If total records count is 20 .. and pagesize is 5 then it gives 20 records in to 4 pages(5,5,5,5).. if we ask for 3 page (indirectly 4 page) records then it gives 16 to 20 records. if ask for 2nd page (indirectly 3 page) records then it gives 11 to 15 records.

note:: Pageable object can have both Paging and Sorting info

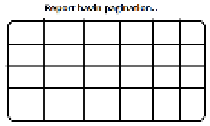
Pageable pageable=PageRequest.of(2,5);
(2 → pageNo, 5 → pageSize)

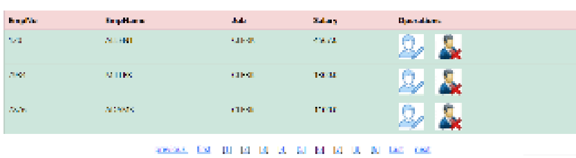
=> DispatcherServlet can prepare Pageable object and can give as handler method arg value if the parameter type is Pageable ... in this process it prepares that Pageable object having pageNo , pageSize gathered from the request parameters "page","size".
=>We can give default pageNo, pageSize to the Pageable object of handler method parameter using @PageableDefault annotation as shown below

```java
@GetMapping("/emp_report")
public String showEmployeeReport(
    @PageableDefault(page=0,size=3,sort="job",direction=Sort.Direction.ASC) Pageable pageable,
    Map<String,Object> map) {


}
```

Report having pagination..





steps for coding

step1) Keep the existing Mini Project ready

step2) Make Sure that the Repository interface pointing to "Employee" model class is extending from PagingSortingRepository(I).

```java
//IEmployeeRepository.java
package com.nt.repository;
import org.springframework.data.repository.PagingAndSortingRepository;
import com.nt.model.Employee;

public interface IEmployeeRepository extends PagingAndSortingRepository<Employee, Integer> {

}
```

step3) Add pagination related report method in service interface and its service impl class

In service Interface

```java
public Page<Employee> getEmployeeReportDataByPage(Pageable pageable);
```

In service Impl class

```java
@Override
public Page<Employee> getEmployeeReportDataByPage(Pageable pageable) {

    return empRepo.findAll(pageable);
}
```

step4) place the following handler method in controller class

```java
public String showEmployeeReport(@PageableDefault(page=0,size=3,sort="job",direction=Sort.Direction.ASC) Pageable pageable,
                                 Map<String,Object> map) {
    System.out.println("EmployeeOperationsController.showEmployeeReport()");
    //use service
    Page<Employee> page=empService.getEmployeeReportDataByPage(pageable);
    // put result in model attribute
    map.put("empData",page);
    //return LVN
    return "show_employee_report";
}
```

step5) Write the following code in show_employee_report.jsp

```jsp
<%@ page isELIgnored="false" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"/>

<div class="container">

<c:choose>
  <c:when test="${!empty empsData.getContent()}">
   <!-- <table border="1" class="table table-striped" > -->
   <!-- <table border="1" class="table table-hover" > -->
   <table border="1" class="table" >
```

=>Bootstrap contains set of libraries representing the readymade Styles, Javascript operations

provides the readymade CSS and JS

=> Bootstrap gives lots of readymade external styles which can be applied in our html files and jsp file by specifying their style names..

To use bootstrap in our Apps (html files /jsp files) add this

```html
<link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"/>
```

get this from bootstrapcdn.com

Spring Boot MVC with spring data jpa Pagination

Page<T> findAll(Pageable pageable); (PagingAndSortingRepository)

=> This method takes pageNo(0 based),pageSize as inputs in the forn. of Pageable obj and returns output as Page<T>/Slice<T> obj

having requested page records, pages count, currentmbers, total records

and etc..

(pageNo,pageSize)

Pageable(input)

Db s/w

findAll(Pageable p)

InMemory Proxy class object of

Db table

PagingAndSortingRepository....

page<T> or Slice<T>

getContent()-- List<T>

getTotalPages();

isFirst(); isLast();

....

Slice<T>

extends Page<T>

if total records count is 20 .. and pagesize is 5 then it gives

20 records in to 4 pages(5,5,5,5).. if we ask for 3 page (indirectly 4 page) records then it gives 16 to 20 records. if ask for 2nd page (indirectly 3 page) recors then it gives 11 to15 records.

Pageable pageable=PageRequest.of(2,5);

pageNo

note:: Pageable object can have

both Paging and Sorting info pageSize

=> DispatcherServlet can prepare Pageable object and can give as handler method arg value

if the parameter type is Pageable ... In this process it prepares that Pageable object having pageNo, pageSize gathered from the request parameters "page","size".

=>We can give default pageNo, pageSize to the Pageable object of Handler method parameter using @PageableDefault annotation as shown below

```
@GetMapping("/emp_report")
public String showEmployee Report(
@PageableDefault(page=0,size=3,sort="job",direction=Sort.Direction.ASC) Pageable pageable,
Map<String, Object> map) {
```

EmpNo

120

}

Report havin pagination..

first next [1] [2] [3] [4] previous last

EmpName

ALLEN1

Job

Salary

Operations

CLERK

4567.0

7934

MILLER

CLERK

1300.0

7876

ADAMS

steps for coding

CLERK

1100.0

previous first [1] [2] [3] [4] [5] [6] [7] [8] [9] Last next

step1) Keep the existing Mini Project ready

step2) MakeSure that the Repository Inteface pointing to "Employee" model class

is extending from PagingSortingRepository(I).

```
//IEmployeeRepository.java
package com.nt.repository;
import org.springframework.data.repository.PagingAndSortingRepository;
import com.nt.model.Employee;
public interface IEmployee Repository extends PagingAndSortingRepository<Employee, Integer> {
}
```

step3) Add pagingation related report method in service interface and ins service impl class

In service Interface

```
public Page<Employee> getEmployees ReportDataByPage(Pageable pageable);
```

In service Impl class

```
@Override
public Page<Employee> getEmployees ReportDataByPage(Pageable pageable) {
return empRepo.findAll(pageable);
```

}

step3) place the following handler method in controller class

public String showEmployee Report(@PageableDefault(page=0,size = 3,sort = "job", direction = Sort. Direction.ASC) Pageable pageable,

}

Map<String, Object> map) {

System.out.println("EmployeeOperationsController.showEmployee Report()");

//use service

Page<Employee> page=empService.getEmployees ReportDataByPage(pageable);

// put result in model attribute

map.put("empData",page);

//retunr LVN

return "show_employee_report";

step3) Write the following code in

show_employee_report.jsp

<%@ page isELIgnored="false" %>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"/>

<c:choose>

<c:when test="${!empty empsData.getContent()}">

<!-- <table border="1" class="table table-striped" > -->

<!-- <table border="1" class="table table-hover" > -->

<table border="1" class="table">

<tr class="table-danger">

<th>EmpNo </th>

<th>EmpName</th>

<th>Job </th>

<th>Salary</th>

<th> Operations </th>

</tr>

=>Boostrap contains set of Libraries representing the readymade Styles, Javascript operations provides the readymade CSS and JS

=> Bootstrap gives lots of readymade external styles which can be applied in out html files and jsp file by specifying their style name..

To use bootstrap in our Apps (html files /jsp files) add this

<link rel="stylesheet"

href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"/>

v5.2.2

CSS

get this from bootstrapcdn.com

https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css

Click to copy

JavaScript

https://cdn.jsdelivr.net/npm/bootstrap_5.2.2/dist/js/bootstrap.min.js

<c:forEach var="emp" items="${empsData.getContent()}">

<tr class="table-success">

<td>${emp.empno} </td>

<td>${emp.ename} </td>

<td>${emp.job} </td>

<td>${emp.sal} </td>

Click to copy

JavaScript Bundle

https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js

note:: In CSS we have 3 types of styles

a) Inline CSS (specific to one tag)

b) Embedded CSS (Specific to one UI comp) c) External CSS (Common for multiple UI Comps)

note:: Bootstrap offers lots of external CSS

<td>

**   <img src="images/delete1.jpg" width="50" height="50"> </td>**

</tr>

</c:forEach> </table>

<p style="text-align:center">

<c:if test="${empsData.hasPrevious()}">

** previous   **

</c:if>

<c:if test="${!empsData.isFirst()}">

**first   **

</c:if>

<c:forEach var="i" begin="1" end="${empsData.getTotalPages()}" step="1">

**[${i}]   **

</c:forEach>

<c:if test="${!empsData.isLast()}">

**Last   **

</c:if>

<c:if test="${empsData.hasNext()}">

** next**

</c:if>

</p>

</c:when>

<c:otherwise>

<h1 style="color:red;text-align:center"> Records not found </h1>

</c:otherwise> </c:choose>

<c:if test="${!empty resultMsg}">

<h3 style="color:green;text-align:center"> ${resultMsg}</h3> </c:if>

**

**

<hr>

<hr>

<h1 style="text-align:center"><img src="images/home1.png" width="50" height="50">Home</h1>

**<h1 style="text-align:center"><img src="images/add.png" width="50" height="50"> Add Employee </h1> **

Page<T> docs

R<

Sir where we can find

Method Summary

All Methods

Modifier and Type

Static Methods Instance Methods Method and Description

isFist() and getPagabl

Madhav Tamaje to Me (Dire

Abstract Methods Default Methods

MT

yes sir

Akshay G to Me (Direct Mes

```
static <T> Page<T>
empty()
```

Creates a new empty Page.

```
static <T> Page<T>
empty (Pageable pageable)
```

AG

yes sir

Who can see your

Creates a new empty Page for the given Pageable.

To: ROHIT AV

(Direct Me

long

Type message here...

`int`

<U> Page<U>

`getTotalElements()`

Returns the total amount of elements.

`getTotalPages()`

Returns the number of total pages.

`map(Function<? super T,? extends U> converter)`

Returns a new Page with the content of the current one mapped by the given Function.

Methods inherited from interface org.springframework.data.domain. Slice

`getContent, getNumber, getNumberOfElements, getPageable, getSize, getSort, hasContent, hasNext, has Previous, isFirst, islast, nextOrLastPageable, nextPageable, previousOrFirstPageable, previousPageable`

note:: if we run the Eclipse IDE web application in an external Tomcat Server, the changes done in the source code of the Application will reflect to the output automatically

note:: if we run the Eclipse IDE web application in an embedded Tomcat Server, that changes done in the source code of the Application will not reflect to the output automatically though we refresh the the webpage in browser.. To make this happen add "devtool starter" to Eclipse IDE maven/gradle based spring boot web /Disributed Application

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope> <optional>true</optional>

</dependency>

=>we can add this starter while creating the Spring boot Project by select dev tool starter

=> we can add this starter after creating the Project

right click on the Project ---> spring ----> add starters ---> select pom.xml/build.gradle --> select dev tools starter

=> Modify the source code of java classes like service class, controller class and etc.. and observe the server console for automatic code changes and those changes will reflect in the browser output