

## Java Exercise – Paging

© 2017 Edument AB

### Part 1 – The repository

1. Pagination of data is an important pattern when we present information on the web. For example, Google uses pagination to navigate the results like this:



Read about web pagination here: <https://en.wikipedia.org/wiki/Pagination>

2. In this exercise we will implement paging a list of customer where we define customers as follows:

```
public class Customer {  
    public int id;  
    public String customerName;  
    public String address;  
    public String zipcode;  
    public String city;  
}
```

3. To simplify the exercise, we use a generic list of customers as an in-memory database:
  - a. Create a new class Repository
  - b. It should contain a private list of customers
  - c. In the constructor initialize the list using this code:

```
for (int i = 1; i <= 100; i++) {  
    Customer cust = new Customer();  
    cust.id = i;  
    cust.customerName = "Customer " + i;  
    cust.address = "Main street " + i;  
    cust.zipcode = "" + (20000 + i);  
    cust.city = "Stockholm";  
    customers.add(cust);  
}
```

4. Add a method to the repository with this signature:

```
public List<Customer> getPageOfCustomers(int page, int pageSize)
```

Now we need to implement the logic to return a page of data from the customers list, to do this see this page:

- <http://stackoverflow.com/questions/12214894/java-paging-util>

5. We also need a method that returns the number of pages, like:

```
public int numberOfPages(int pageSize) {  
    return (int) Math.ceil(customers.size() / pageSize);  
}
```

## Part 2 – Spring Boot and paging

1. Now create a new Spring Boot web project
2. Our task here is to create a page that allows us to navigate across the customer database, one page at the time. Each page should contain 10 customers.
3. Create a new Customer controller and the corresponding view that maps to **/Customers**
4. Add so that you can pass in the current page as an optional query string like **/Customers?page=123** and if the parameter is missing then use **1** as the default value.
5. On the page, display the correct customer page
6. Add the necessary **previous** and **next** navigation links to navigate the customer list.

Add so that the previous link is not visible on page 1 and the next link is not visible on the last page.

7. Most of the time we also want to see the current position in the data set, like how Google does it here:



8. Add the numbering between the links, as the picture above shows and make the current page number not be a hyper-link.
9. But what happens if we have thousands of customers in the dataset? Then a common pattern is to only show x pages before and after the current page number, like:



Modify the repository so it contains 1000 customers and try to implement the necessary logic so that the user gets a nice user-experience for the numbering above.

## Part 3 – Bootstrap

(This part is optional and if you have time)

1. Bootstrap the most popular library for giving your web site a nice, professional-looking style.
2. First read about what it is here:
  - a. [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
  - b. <http://getbootstrap.com/>
3. There's a nice tutorial here: <http://www.w3schools.com/bootstrap/> and especially a chapter about making nicer pagination [http://www.w3schools.com/bootstrap/bootstrap\\_pagination.asp](http://www.w3schools.com/bootstrap/bootstrap_pagination.asp)
4. Review the getting started and try to style your customer page and navigation links using Bootstrap. Use the Bootstrap CDN links as shown here: <http://getbootstrap.com/getting-started/#download> and paste it into the <head></head> part of your page.
5. What is a CDN? Read more about it here: [https://en.wikipedia.org/wiki/Content\\_delivery\\_network](https://en.wikipedia.org/wiki/Content_delivery_network)

## If you have time

Paging is hard, read more about it here to learn about the best practices around paging

- Pagination Best Practices  
<https://gist.github.com/brajeshwar/2802235>
- Pagination and JDBC  
<http://stackoverflow.com/questions/2771439/jdbc-pagination>
- Why Most Programmers Get Pagination Wrong  
<https://blog.jooq.org/2016/08/10/why-most-programmers-get-pagination-wrong/>
- What is pagination (SEO focused)  
<http://www.webseoanalytics.com/blog/what-is-pagination/>