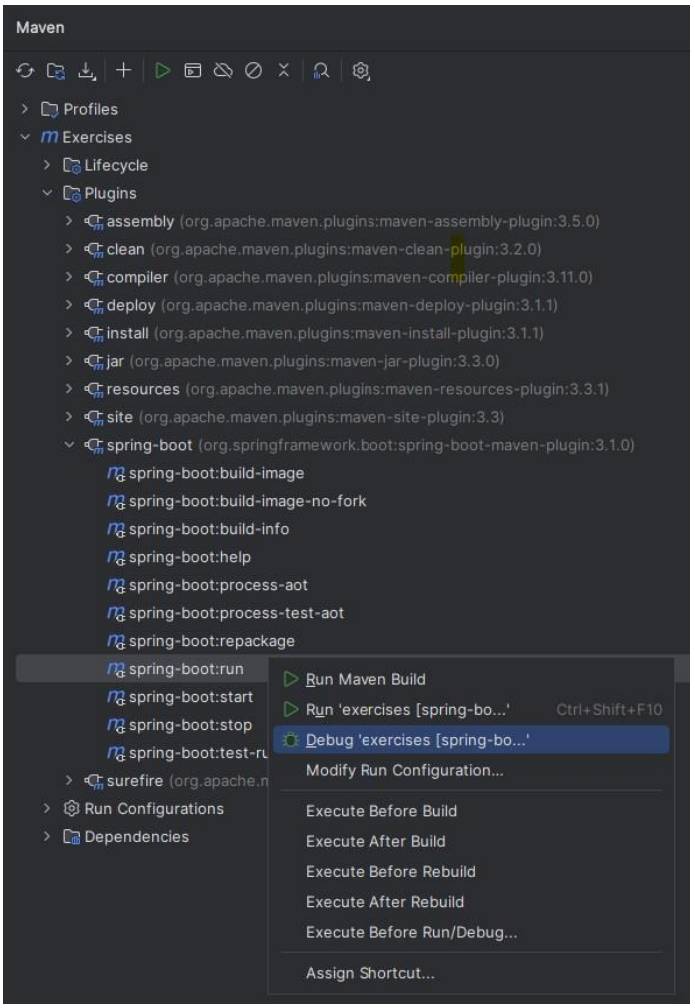
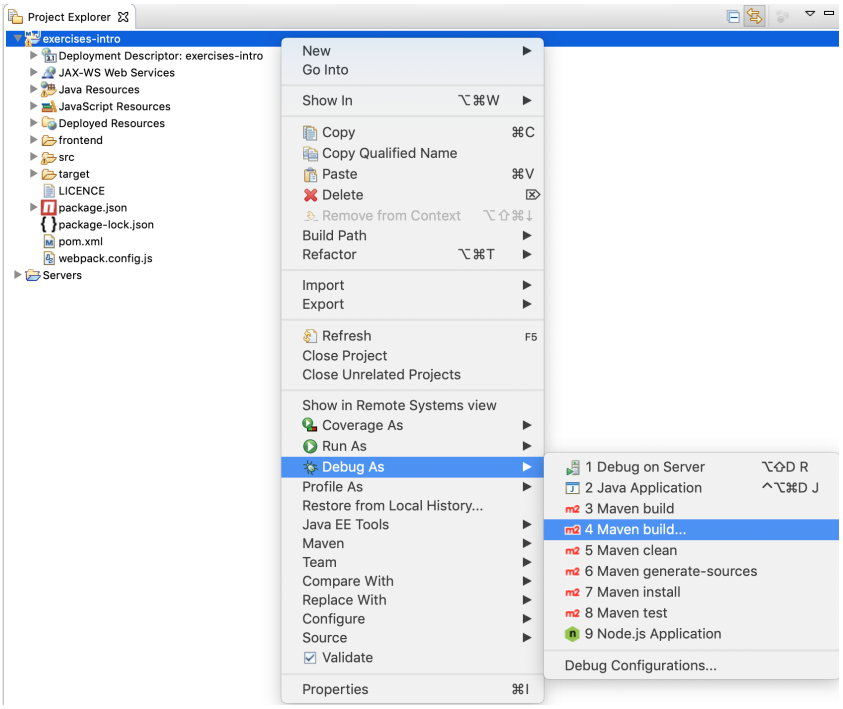
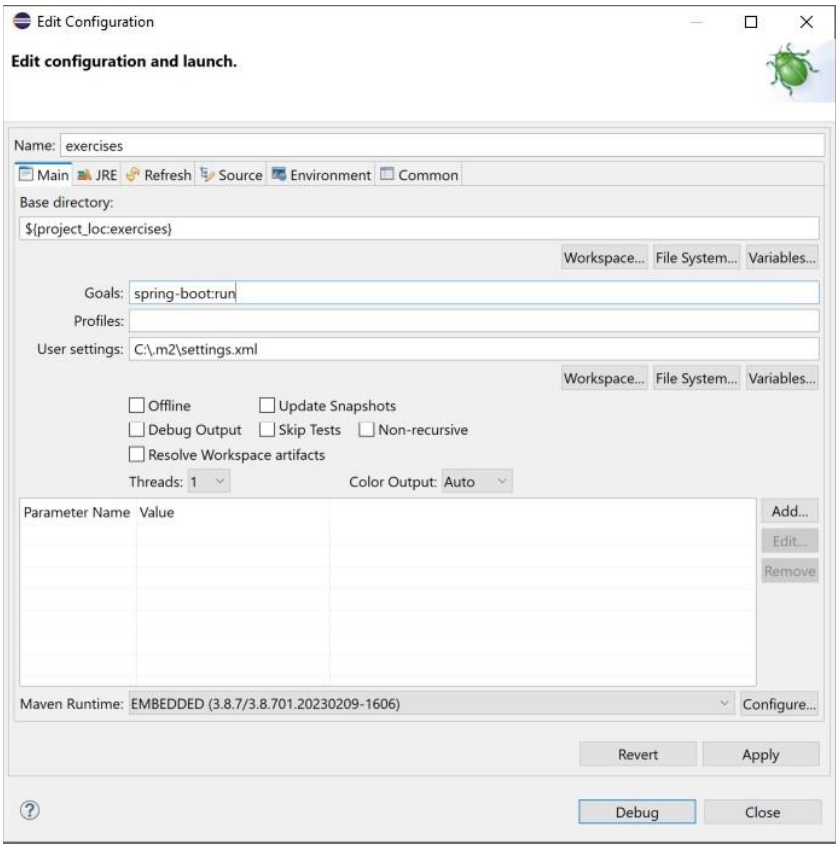
Instructions

1. Import the Maven project to your Favorite IDE.
2. Run the Maven goal spring-boot:run
    a. If you have command line Maven installed you can run mvn spring-boot:run in terminal OR
    b. Debug or run the application in IntelliJ



    c. Debug or run the application in Eclipse





3. Go to localhost:8080

Exercise 1: Application layout

In this exercise you will learn to create a typical layout structure for a Vaadin application. The goal is to create an application layout containing a header, a footer and a main area that consists of a navigation area and a content area. The screenshot below illustrates the desired result.

The header marked with a red color should be 100% in width and 150px in height. The footer (in blue) should be 100px in height and 100% in width. The area between the header and the footer should take up the rest of the space in the application. You will need to add a horizontal layout, and put the navigation area and content inside the horizontal layout, so that they will be on the same line. The navigation area (in yellow) should be 25% of the main area's width, while the content area (in green) should be 75% of the content area's width.

The stub application contains four s each with a text and the corresponding color. Your task is to complete the layout. In order to complete the application layout, you need to add one extra layout to the application and adjust the size of the components.

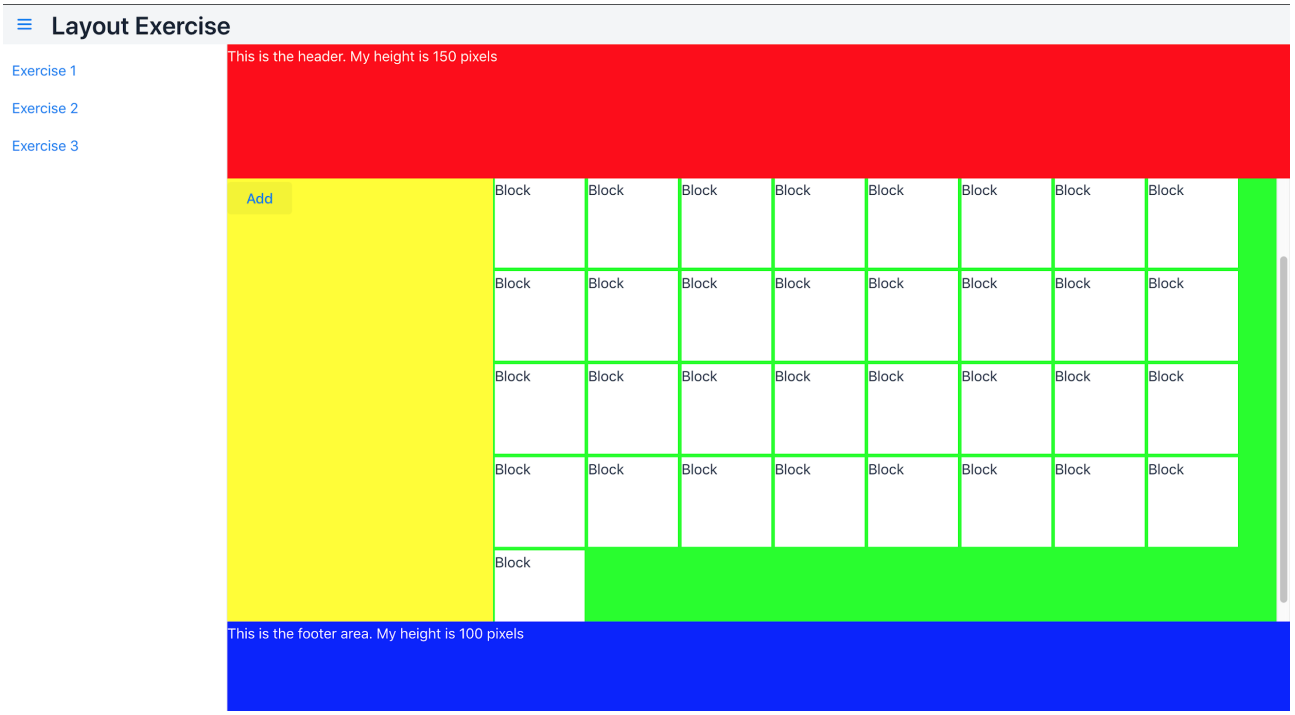
Remember to verify (e.g. with Chrome Dev Tool) that header is 150px and footer is 100px even if the browser window resizes.

Exercise 1 bonus task: Enabling scrollbars

You can continue with this task once you've completed the first part of the application layout. This exercise's purpose is to enable scrollbars in the content area. Start by replacing the navigation text with a button. Implement a click listener which will add a new Div block to the content area. You can use the createBlock() helper method for creating the blocks for the content area. After this, enable scrolling for the content div.

Note that if you resize the window, the amount of blocks shown on one row in the content area should adjust to browser size so that at any given time, the maximum number of buttons are shown on one row.

The challenge of this task is to figure out how to enable scrollbars and how to make the buttons wrap according to the browser size.

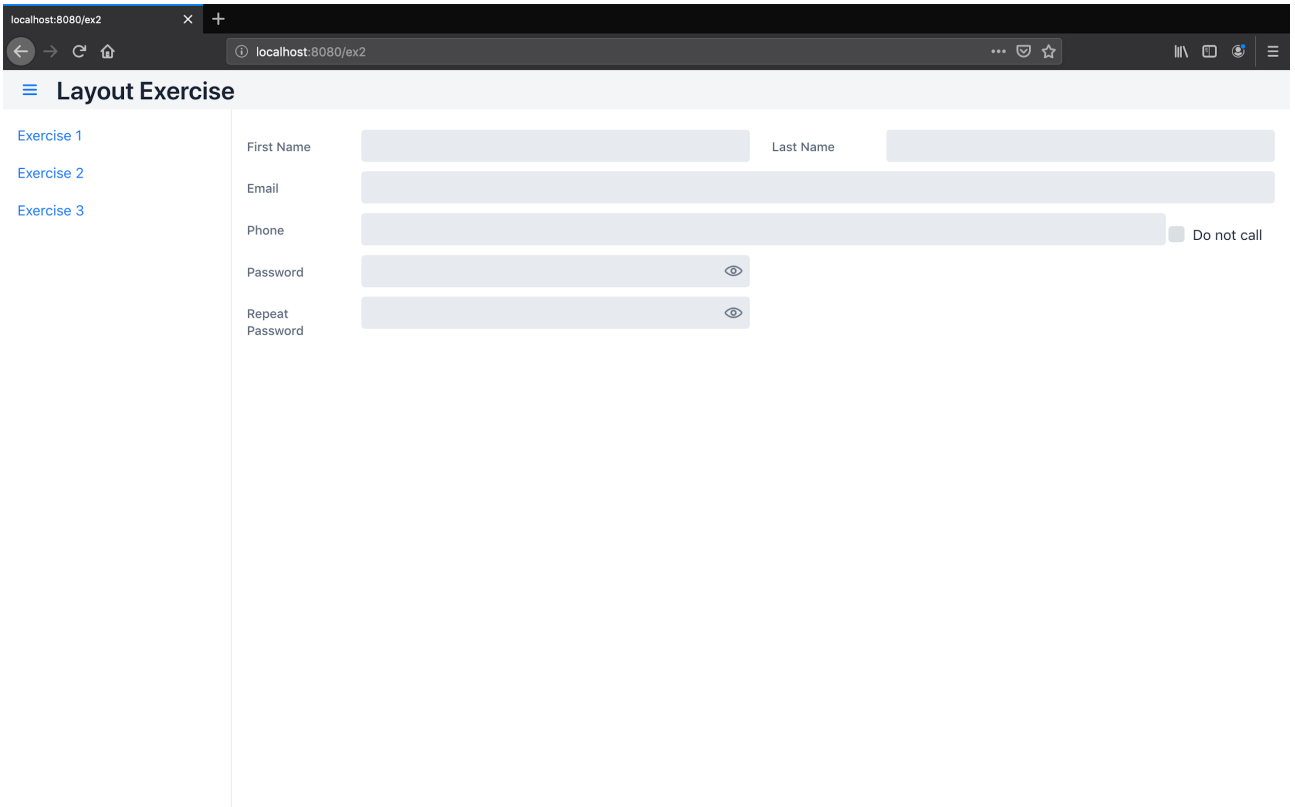


Exercise 2: Form Layout

In this exercise you will learn to create a typical form layout. The goal is to create a form layout containing:

- A text field for first name
- A text field for last name
- A text field for email which spans 2 columns
- A complex form item for phone number which contains a text field and a checkbox
- A password field for a password
- Another password field for repeating the password, the repeating password field should start from a new line.

The screenshot below illustrates the desired result.



Exercise 3: Vaadin Board

Vaadin Board can be used to create a responsive layout easily, especially when you are building a dashboard view to show some charts.

In this exercise you will learn to create a responsive layout with Vaadin Board. The goal is to create a board with 2 rows:

- The first row has two pie charts
- The second row has a column chart

There are already methods for creating those charts, your task is just to add them to a Vaadin Board.

Note that Vaadin Board is a commercial tool, so you need to have a license. You can follow this link to check how to validate licenses. On a local machine, the licenses for commercial components and tools are checked automatically using your Vaadin account. The first time you use a commercial component or tool, you are asked to log in to vaadin.com to validate your license.