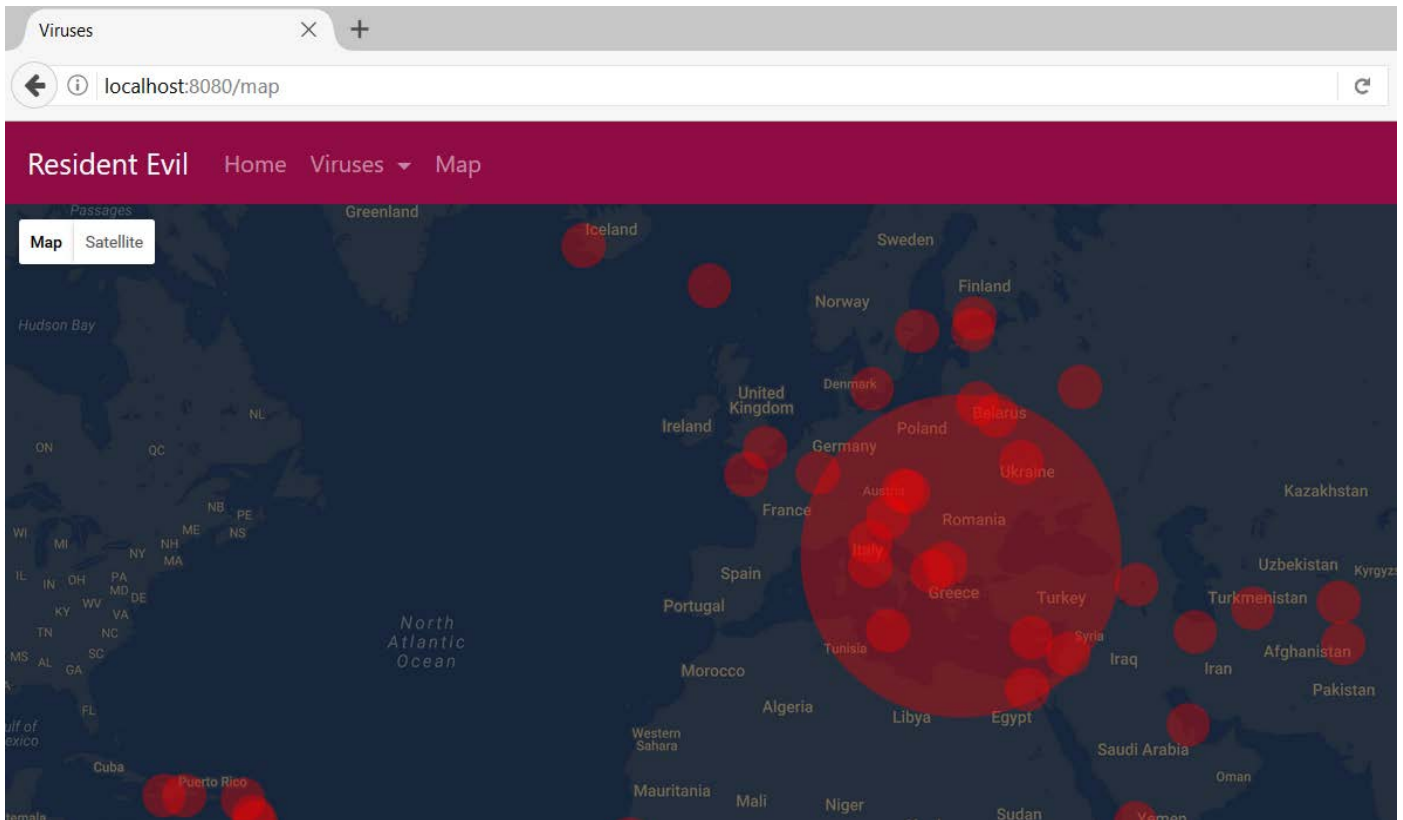


Project: Resident Evil

Resident Evil is a system that registers virus spreads across the world. It is a significantly big project, and as such it will have several parts. In this exercise you will land the basis of the application. You will also create the majority of the visual design.



Exercises: Thymeleaf Engine

Problems for exercises and homework for the [“Java MVC Frameworks - Spring” course @ SoftUni](https://softuni.org/).

1. Views

You would need couple of views. Separate the **menu** and **the forms** in html files and include them. Use a design that you find appropriate. The examples use **Bootstrap 4**.

Show

Add

Project: Resident Evil

Resident Evil is a system that registers virus spreads across the world.

- **Home**
 - Entry point of the application
- **Viruses**
 - Dropdown menu with 2 buttons – **[Show]** and **[Add]**
 - **[Show]** – All the viruses are shown here. You can **edit** and **delete** each virus.
 - **[Add]** – You can add.

NOTE: #900C45 this is the color you need 😊

2. Data Entities

Create the required **entities**. Use the appropriate **data types**.

- **Virus**
 - Name – Cannot be empty, should be between **3** and **10** symbols.
 - Description – Cannot be empty, should be between **5** and **100** symbols.
 - Represented as Text in the database
 - Side Effects – Should have a maximum of **50** symbols.
 - Creator – Should be either **Corp** or **corp**.
 - Is Deadly – Boolean
 - Is Curable – Boolean
 - Mutation – Cannot be null. Should hold one of the following values:
 - **ZOMBIE**
 - **T_078_TYRANT**
 - **GIANT_SPIDER**
 - Turnover Rate – Number, between **0** and **100**.
 - Hours Until Turn (to a mutation) – Number, between **1** and **12**.
 - Magnitude – Cannot be null. Should hold one of the following values:
 - **Low**
 - **Medium**

- **High**
 - Released On – Date, should be before the “**today**” date.
 - Capitals – A **collection** of **Capitals**.
- **Capitals**
 - Name
 - Latitude
 - Longitude

3. Establish the Back-End

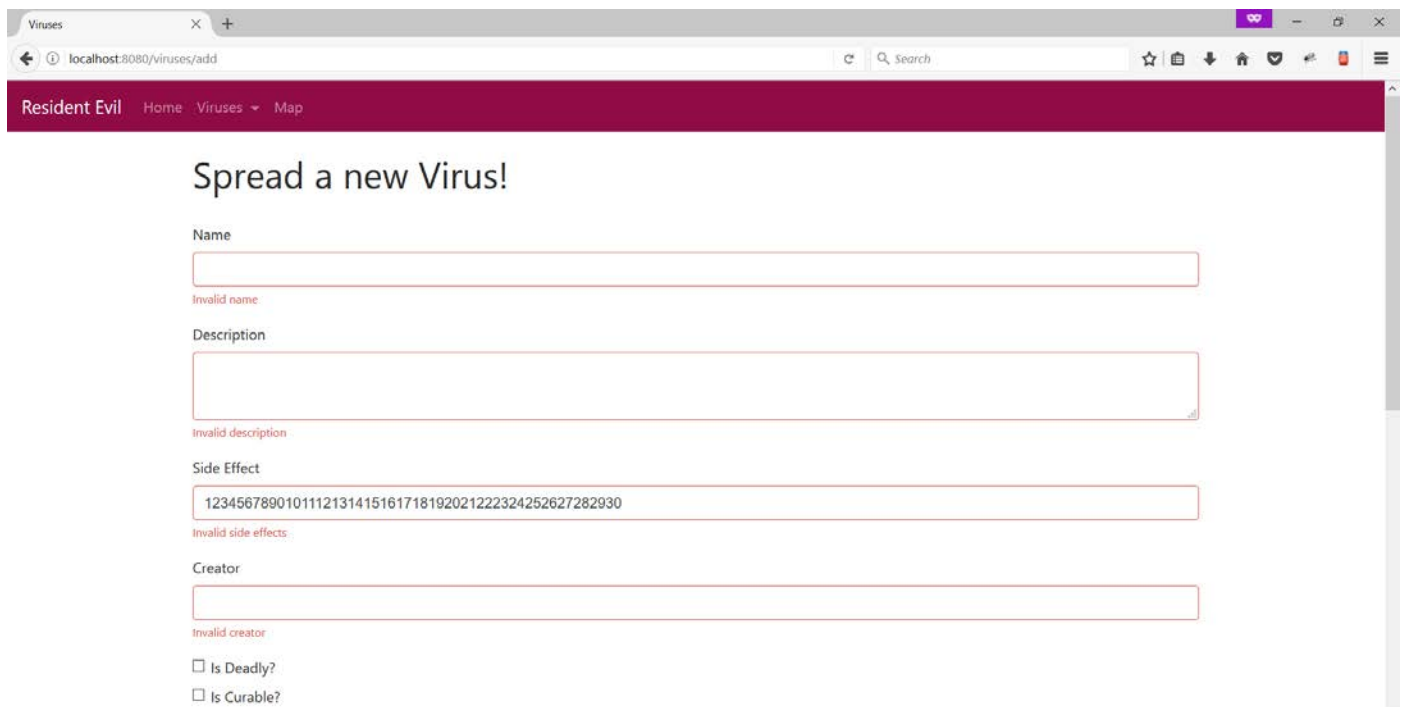
Create the required:

- **Entities**
- **Models**
- **Repositories**
- **Services**
- **Controllers**

Load the **capitals** by the provided **SQL**.

4. Add Viruses

Create a functionality to **add** Viruses. Make the necessary **validations**. Create a custom annotation to validate the Release Date.



The screenshot shows a web browser window with the URL `localhost:8080/viruses/add`. The page has a dark red header with the text "Resident Evil" and navigation links "Home", "Viruses", and "Map". The main content area is titled "Spread a new Virus!". Below the title is a form with four text input fields and two checkboxes. Each text input field has a red border and a red error message below it: "Invalid name", "Invalid description", "Invalid side effects", and "Invalid creator". The "Side Effect" field contains the text "1234567890101112131415161718192021222324252627282930". The checkboxes are labeled "Is Deadly?" and "Is Curable?".

☐ ZOMBIE
☐ T_078_TYRANT
☐ GIANT_SPIDER
☐ Save Humanity
 Mutation cannot be null

Turnover Rate

Invalid turnover rate

Hours Until Turn

Invalid turnover rate

Magnitude

Release Date

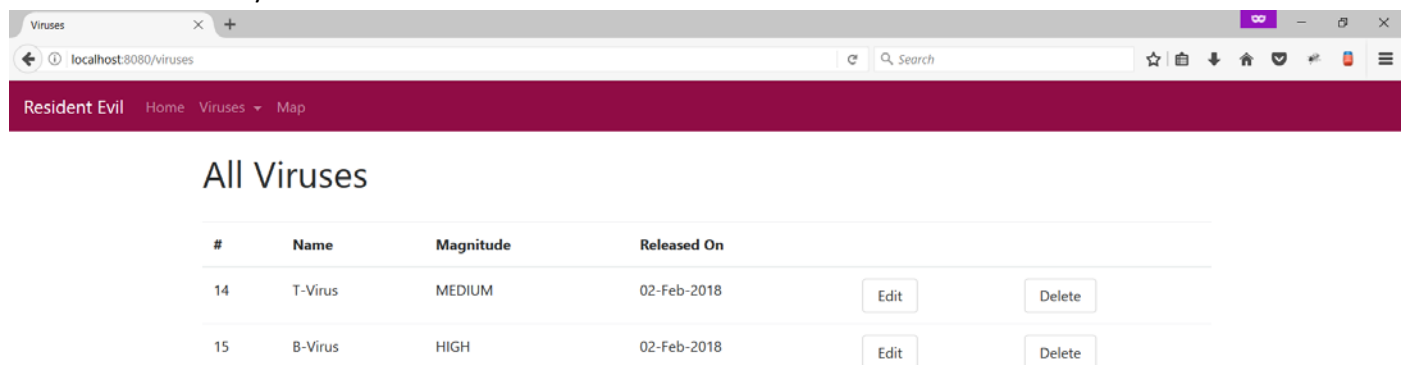
Invalid Date

Affected Capitals

You must select capitals

5. Show Viruses

Create a functionality that **shows** all of the created viruses.



#	Name	Magnitude	Released On		
14	T-Virus	MEDIUM	02-Feb-2018	Edit	Delete
15	B-Virus	HIGH	02-Feb-2018	Edit	Delete

6. Edit Viruses

Create a functionality to **edit** Viruses. You should be able to edit everything **except** the **release date**. Make the necessary **validations**.

7. Delete Viruses

Create a functionality to **delete** Viruses