Exercises: Java EE: JavaServer Pages

Problems for exercises and homework for the "Java Web Development Basics" course @ SoftUni. Submit your solutions on the **course page** of the **current instance**.

We will implement a simple application, only with Servlets and JSP, exploring the most interesting exploits of the Servlet API nad JavaServer pages.

MeTube Application v1

MeTube is an application in which you create tubes, with several properties. It has many versions, and you will most probably see it several times.

You will have to create a simple web application which has several pages and 1 object entity.

1. Data

This is the data layer of the application. There is 1 data object for you to implement.

So let's create our **Tube**. The **Tube** is a data object which stores data about an abstract **tube**. You will see later what it will be used for.

First implement a class **Tube** – which has these properties:

- **`Name`** – a **String**.
- **`Description`** – a **String**.
- **`YouTubeLink`** – a **String**.
- **`Uploader`** - a **String**.

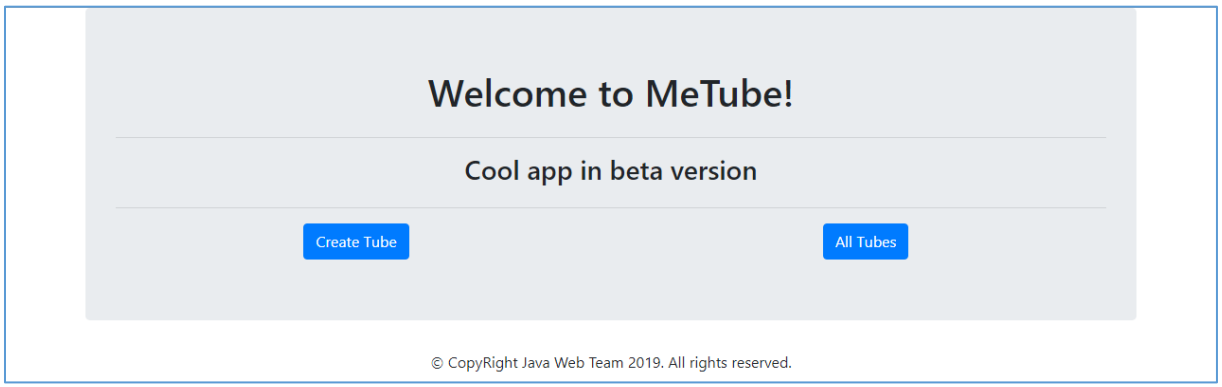
2. Views

In this application you must create several views, similar to the pictures below:

index.jsp

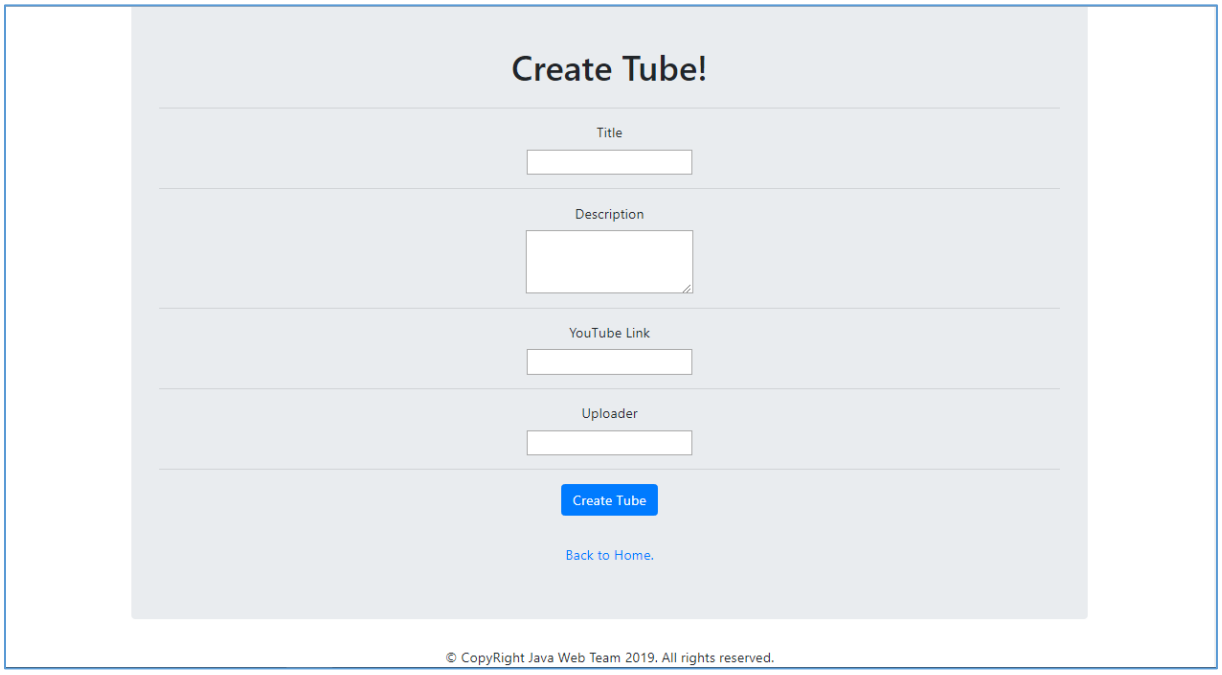
This is the home page, in other words the page that should be visualized when the application starts.

The index page must contain buttons that redirect to **create-tube.jsp** and **all-tubes.jsp**.



create-product.jsp

This is the page where you create your tubes. It must contain **input** field for **title**, **textarea** for **description**, inputs for **YouTube link** and **Uploader** and a **button** which creates a **Tube** with the given values and a **link** that returns the **home** page.

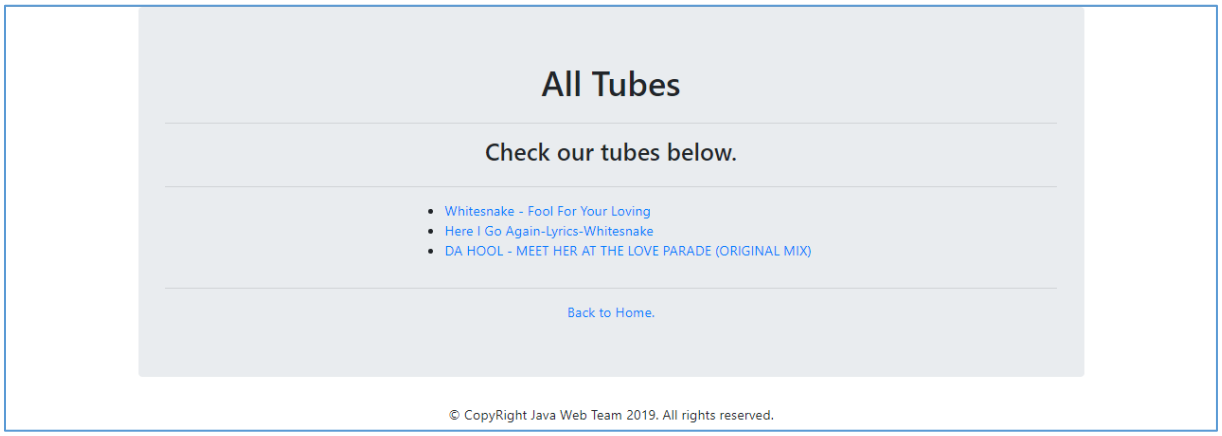


details-tube.jsp

This is the page which **renders full data** about the **selected Tube**. The selected **Tube** should be extracted from the **database**, using the **`title`** from the **query parameters**. If tube does not exist – write an apropriate message.

all-tubes.jsp

Renders only the **`titles`** of the **`Tubes`**. Upon clicking a **`title`** of a **`Tube`**, you should be **redirected** to a **details page**, with **query parameter** – the **`title`** of the **`Tube`**. If no tubes are added yet – write "No tubes – Create some!"



3. Servlets

Tube Create

Implement a **Servlet** – **`TubeCreateServlet`**, which listens on route "**`/tubes/create`**".

Upon a **GET** request, it should return a form which accepts **4 inputs** – a **`title`**, a **`description`**, a **`YouTubeLink`** and an **`uploader`**.

The **form** should send a **POST** request to the **`same route`**.

Upon a **POST** request, you should **redirect** to "**`/tubes/details`**".

Tubes All

Implement a **Servlet** – **`TubeAllServlet`**, which listens on route "**`/tubes/all`**".

Upon a **GET** request, the **Servlet** should render a page with the **`names`** of all created **`Tubes`**. Upon clicking a **`title`** of a **Tube**, you should be **redirected** to a "**`/tubes/details`**", with **query parameter** – the **`title`** of the **Tube**.

Tube Details

Implement a **Servlet** – **`TubeDetailsServlet`**, which listens on route "**`/tubes/details`**".

Upon a **GET** request, the **Servlet** should extract the **Tube** with the given **`title`** in the **query parameters**, and render a page with **full information** about it.

4. Constraints

You must use the following things while implementing your application:

- Servlets
- Hibernate
- Repository layer
- Service layer
- Web layer