

Exercises: Templating

Problems for exercises and homework for the [“JavaScript Applications” course @ SoftUni](#).

The following tasks have no automated test – submit your solutions for peer review on the course page.

1. List Towns

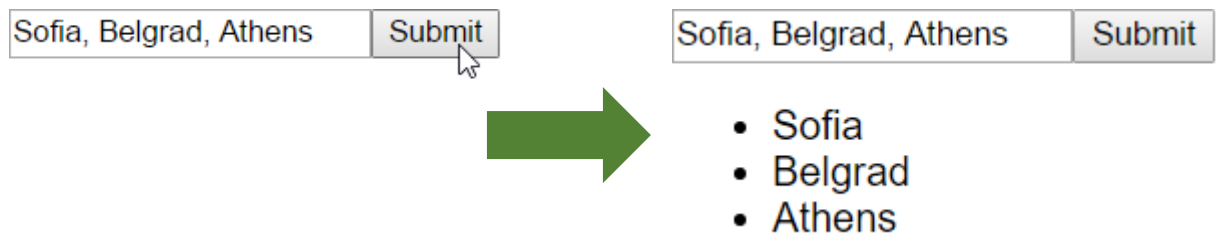
You are given an **input field** with a **button**. In the input field you should enter **elements separated** by comma and whitespace (", "). Your task is to create a simple **template** that defines a **list** of towns. Each **town** comes from the **input** field.

listtown.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>List Town</title>
  <script src="node_modules/handlebars/dist/handlebars.min.js"></script>
  <script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
  <script src="towns.js"></script>
</head>
<body>
  <input id="towns" type="text"/>
  <button id="btnLoadTowns">Load</button>
  <div id="root"></div>
</body>
<script>
  $((() => attachEvents()))
</script>
<script type="text/x-handlebars-template" id="towns-template">
  // TODO: Create the template here
</script>
</html>
```

In your **attachEvents()** function you **should** attach a click event to the **button** with id **“btnLoadTowns”** and **render** the **towns** that come from the input field in the html **template** with id **“towns-template”**

Screenshots



This is how the html look like:

```

<input id="towns" type="text">
<button id="btnLoadTowns">Load</button>
<div id="root">
  <li>Sofia</li> == $0
  <li>Varna</li>
  <li>Plovdiv</li>
</div>

```

2. HTTP Status Cats

[Download project skeleton.](#)

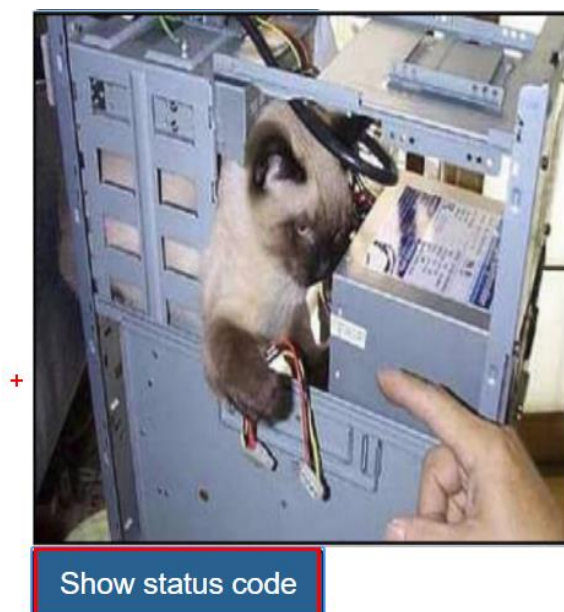
We all love cats. They are also a fun way to learn all the HTTP status codes.

Your task is to **refactor** the given **html** and to create a **template** to represent **each** cat card block on it's own. After you have **created** the template **render** it into the div with id **"allCats"**.

A **cat** has an **id**, **statusCode**, **statusMessage** and **imageLocation**. The cats are **seeded** using the **function** from the js file named **catSeeder.js**

Each card block has a **button** that **unveils** status code information **connected** to each cat. You should **toggle** the button and change it's text from **"Show status code"** to **"Hide status code"**.

Screenshots



3. Refactor "Prodavachnik"

Take the Prodavachnik application from the previous exercise ([Exercise: Single Page Applications](#)) and **refactor** it to use **Handlebars templates**. Take the markup for each relevant section and place it in its own template. When the user navigates around the site, instead of hiding and showing content, only the needed template will be rendered.

Try to **reuse** as much of the markup as possible – for example, the login and register pages can use the same template, as well as create and edit ad. Use a **partial template** for each individual ad.

For the ads catalog, use whichever style you prefer – as rows in a table or as individual divs.

Hints

The ads array that you fetch from the server can be rendered using an **each** helper. For the header, use a **conditional statement** that only shows the links and user greeting that apply to the current session state (logged in or not). Whenever the session state changes (on page load, when the user logs in, registers, or logs out), re-render the header.