

eCommerce Systems / Professor H Roumani

PROJECT E

Executive Summary

In this project we take the second step in our view migration journey by creating a webapp whose views are generated via Javascript. The project will expose the key elements of SPA (Single Page Application) without using any framework by relying only on a minimal set of Javascript / DOM instructions.

Reference Implementation

This project re-implements a subset of the services of Project-D, and as such, its functionality is 100% identical to that of Project-D. Hence, aside from internal and response-time changes, the reference implementation of D acts as one for E.

The Chosen D Subset

This project re-implements the following subset of D's services: S0, S1, and S5. Specifically:

- **Dashboard**
Serving as a main menu for the webapp.
- **Prime Number Finder**
Find the next prime in a given range one by one (prompting in between).
- **SIS Report Generator**
Generate a report from a Student Information System filtered based on a name prefix, a major, and a minimum GPA. The output is sorted on a custom column.

Naming Convention

It is critical that you adhere to the following naming convention:

- The model package is named `model` and it contains:
 - The model's singleton class: `Engine.java`.
 - The bean: `StudentBean.java`.
 - The data access object: `StudentDAO.java`.
- The controller package is named `ctrl` and it contains:
 - The `Dash.java` servlet with URL mapping `/Dash.do`.
 - The `Prime.java` servlet with URL mapping `/Prime.do`.
 - The `Sis.java` servlet with URL mapping `/Sis.do`.
- There is only one html file under `WebContent` named `Dash.html`. It contains:
 - Three divisions with IDs `dashDiv`, `primeDiv`, and `sisDiv` all belonging to the `view` class.
 - The `dashDiv` division contains a menu for the two services. Each menu is a hyperlink (anchor tag) that triggers the javascript `showPrime` and `showSis` functions.
 - The `primeDiv` division contains a form (id: `primeForm` onsubmit: `prime(this); return false;`) and a nested division with id: `primeResult`.
 - The `sisDiv` division contains a form (id: `sisForm` onsubmit: `sis(this); return false;`) and a nested division with id: `sisResult`.

- A folder under **WebContent** named **res** containing the Javascript source file **e.js**.

Implementation Notes

1. Export you Project-D into a war; rename the war to ProjE; import the renamed war.
2. You may want to start by amalgamating the html contents of the existing Dash.jspx, Prime.jspx, and Sis.jspx into one file named Dash.html and then deleting all the jsp files.
3. Create the needed divisions.
4. For the forms, remove the action and method attributes; add an ID; and add onsubmit.
5. Include a nested division for the result after the form.
6. For the input form elements, replace **name** with **id**. The button need not have neither a name nor an id.
7. Link the Javascript file to Dash.html by including this line in its head section:

```
<script type="text/javascript" src="res/e.js"></script>
```

8. In your servlets, remove the if statement checking for fresh visit versus "doit". Instead, assume it is always a "doit" request.
9. Use [this Javascript source file](#) as a starting template.

Persist Your Work

Follow these steps to persist / backup your work:

- Right-click your project (in the Project Explorer) and select Export.
- Select **WAR** File.
- Provide a destination as **ProjE.war** on the Desktop).
- Check the box to *Export the source files* (extremely important)

Now copy the created war file to your Google Drive, DropBox, or some other cloud service, or copy it to a USB drive.

• EECS4413 • Professor Roumani • Don't miss the forest for the trees •