

DocMUNCHER SYSTEM MANUAL

Overview

DocMuncher is written in HTML5, CSS3, AngularJS, jQuery and Highcharts. The skeleton of the website is provided by the `index.html` file alongside `controllers.js`. In `index.html` the content is injected via AngularJS in order to reload only a specific part of the screen rather than the whole page.

As such, the files that control the content are all separate HTML files that can be found under the folder `Partials`. Each HTML file is injected into `index.html` at a different time, depending on what the user clicks on.

In order to keep the information about the master project, we created an AngularJS service (`UserService`, found in `controllers.js`). This service holds:

- `currentProject` - the project the user is currently working on
- `currentCluster` - the cluster the user is currently working on
- `currentDoc` - the document the user is currently working on
- `currentTab` - the tab that is currently active (used for highlighting the navigation bar at the top)
- mock objects - used instead of a database as only teams 39 and 40 are to use the database for the back-end
 - tabs
 - documents
 - projects

Home Screen (`partials/home.html`)

- Controller: `HomeCtrl` in `resources/js/controllers.js`
- Projects are displayed using `ng-repeat`, going through the list of projects in `UserService`.
- When clicking a project, with an `ng-click` event, `setMaster(project)` is called. That is, `currentProject` in `UserService` is changed to the project that the user clicked on.
- To display the new project creation form, `ng-if` logic was implemented in the two divs at the bottom of the file:
 - if the selected document is "new" - so if the user clicked on "New Project", the form is displayed
 - otherwise, the overview project box is displayed

Project Overview ([partials/projectOverview.html](#))

- Controller: `ProjectCtrl` in `resources/js/controllers.js`
- The toggle logic for the **Matrix View** / **Pie Chart View** buttons is created using `ng-show`. `selected` is a binary variable that flags which view is selected, and depending on it one of the divs is shown
- The drill-down pie chart is created using the Highcharts API. The constructor method for the pie chart can be found in the controller. The data fed into the constructor is a mock-up object, but works similarly for production data.
- Similarly, the heat map in the **Matrix View** is created with the same API. The constructor can also be found in the controller.
- The heat map uses colour axis stops to dynamically colour the cells based on the numbers given in the data. The axis is adjusted to have a small variety of colours on purpose, in order to keep the map easy to read.
- Redirecting to the **Cluster Overview** happens in the constructors of the charts, using a `select` event when the user clicks on a cluster (pie slice or matrix cell). The event triggers a function that updates `UserService.currentCluster` and redirects to the page.

Cluster Overview ([partials/clusterOverview.html](#))

- Controller: `ClusterCtrl` in `resources/js/controllers.js`
- In order to be shown by category, the document objects are pulled from `UserService` using `ng-repeat` and `ng-if` combined, to show the documents within a risk threshold.
- The three tables shown in **Cluster Overview** are styled using three classes each:
 - `.clusterOverviewTable` - gives the look of the table
 - `.column-left`, `.column-centre`, `.column-right` - positioning
 - `.table-risky`, `.table-borderline`, `.table-risk-free` - colours and other specific styling
- The redirect to **Document Overview** happens using an `ng-click` event that triggers a function that sets `UserService.currentDoc` as the selected item and redirects to the document view.

Document Overview ([partials/documentOverview.html](#))

- Controller: `DocCtrl` in `resources/js/controllers.js`
- Embedding `.pdf` documents is done in the directive `embedSrc` at the bottom of `controllers.js`. The document is displayed in the `embed` tag in the HTML view.
- The link for the current document in the `embed` tag is injected via AngularJS with `{{currentDoc.link}}`.

Saved Documents (`partials/savedDocuments.html`)

- Controller: `SavedCtrl` in `resources/js/controllers.js`
- The pdf preview is done using the same directive as in **Document Overview**.
- At the beginning of the controller, `$scope.documents` is instantiated with a mock list of objects. This should be replaced with the list of saved documents once the back-end is linked.