Refactor JSP structure #75

Edit New issue

Closed **ppyordanov** opened this issue on Jan 19 · 1 comment

PY **ppyordanov** commented on Jan 19

The .jsp file structure needs refactoring as the main file "home.jsp" has started to increase in size and needs to be decomposed into smaller chunk components that could then be potentially reused on different pages, e.g. the navigation bar will be extracted in a separate file for easy inclusion.

PY **ppyordanov** added the **enhancement** label on Jan 19

PY **ppyordanov** self-assigned this on Jan 19

PY **ppyordanov** added this to the **Mobile Web Application** milestone on Jan 19

PY **ppyordanov** referenced this issue from a commit on Jan 19

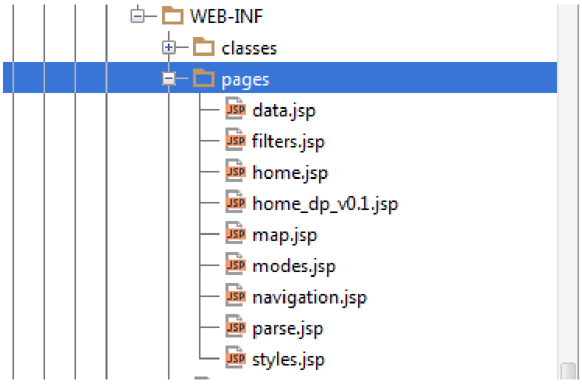
`[BUGFIX+REFACTORING] fixed jquery & bootstrap conflict + jsp decompos…` … c472313

PY **ppyordanov** commented on Jan 19

The *JSP* files have been split into separate components facilitating further template development. The main component types are as follows:

- pages - main views on the web server
- page components - sections from pages that have been extracted in order to enable reusability

Here is the file list after refactoring, accompanied by a brief documentation of each component's role:



```
WEB-INF
  classes
  pages
    data.jsp
    filters.jsp
    home.jsp
    home_dp_v0.1.jsp
    map.jsp
    modes.jsp
    navigation.jsp
    parse.jsp
    styles.jsp
```

- *data.jsp* - this is a utility page, included as a main view at present; however, it is not planned as a part of the final interface; it serves as a convenient access point for the */data* controller method that allows for quick database population with sample data (using a *json* resource file with 5000 data readings located in the *resource* directory) in order to demonstrate Mongo DB's response time with different size of input data

- *parse.jsp* - this file is used for direct parsing of raw models to javaScript structures; currently, the main view of the web server returns a *List* (a list of models) that is converted to a dictionary on the client side code to improve processing;JavaServer Pages facilitates the use of logical operations that allows for efficient client data parsing ;the other option is to serialize the return parameter to json using the Jackson or Google's Gson (which I have installed and using in the *DataPopulation* class at the moment) serializer by creating a custom serializing method. The *json* can then be parsed to an associative array:

Labels

enhancement

Milestone

Mobile Web Applic...

Assignee

PY ppyordanov

Notifications

Unsubscribe

You're receiving notifications because you modified the open/close state.

1 participant

PY

Lock issue

```
var obj = JSON.parse(text);
```

- *home.jsp* - this is the main view of the web server. It contains all of the necessary stylesheet and javaScript file attachments:

```
<link type="text/css" href="${pageContext.request.contextPath}/resources/external/bs/css/bootst
      rel="stylesheet"/>
<link type="text/css" href="${pageContext.request.contextPath}/resources/custom/main.css" rel="
<link type="text/css" href="${pageContext.request.contextPath}/resources/external/jquery/jquery
      rel="stylesheet"/>

<script src="${pageContext.request.contextPath}/resources/external/jquery/jquery-1.11.1.min.js"
<script src="${pageContext.request.contextPath}/resources/external/bs/js/bootstrap.js"></script
<script src="${pageContext.request.contextPath}/resources/external/jquery/jquery-ui.js"></scrip

<script type="text/javascript"
        src="https://maps.googleapis.com/maps/api/js?v=3.8&libraries=geometry,visualization&sen
<script src="${pageContext.request.contextPath}/resources/custom/map_styles.js"></script>
<script src="${pageContext.request.contextPath}/resources/custom/map.js"></script>
```

As well as injections of the rest of the components:

```
<c:set var="parser" value="parse.jsp"/>
<c:set var="navigation" value="navigation.jsp"/>
<c:set var="map" value="map.jsp"/>
<c:set var="styles" value="styles.jsp"/>
<c:set var="modes" value="modes.jsp"/>
<c:set var="filters" value="filters.jsp"/>

<jsp:include page="${parser}"></jsp:include>
<jsp:include page="${navigation}"></jsp:include>
<jsp:include page="${map}"></jsp:include>
<jsp:include page="${styles}"></jsp:include>
<jsp:include page="${modes}"></jsp:include>
<jsp:include page="${filters}"></jsp:include>
```

Note: comments to the relevant scripts will be included in the code

- *map.jsp* - this page contains the Google map component (*div* tag structures); it can be easily included in any page of the web server
- *navigation.jsp* - navigation bar (map configuration options, other links)
- *styles.jsp* - map style configuration
- *modes.jsp* - map mode confuguration
- *filters.jsp* - map filters application

Note: what has been documented here is a temporary configuration and this post will be updated as the project hierarchy changes.

PY **ppyordanov** closed this on Jan 19

PY **ppyordanov** referenced this issue from a commit on Jan 19

PY `[DIR] directory structure` 5a16079

PY

Write Preview Markdown supported Edit in fullscreen

Leave a comment

Attach images by dragging & dropping, selecting them, or pasting from the clipboard.

Reopen issue Comment