UI Documentation

Conference Management System

Reference list for includes in the head tag

Always make sure includes follow the order below:

```
<link rel="icon" href="http://www.cssreset.com/favicon.png">
<link rel="stylesheet" type="text/css" media="all" href="css/fonts.css">
<link rel="stylesheet" type="text/css" media="all"
href="css/themes/base/style.css">
<link rel="stylesheet" type="text/css" media="all"
href="css/themes/base/jquery.ui.core.css">
<link rel="stylesheet" type="text/css" media="all"
href="css/themes/base/jquery.ui.theme.css">
<link rel="stylesheet" type="text/css" media="all"
href="css/themes/base/jquery.ui.datepicker.css">
<link rel="stylesheet" type="text/css" media="all"
href="css/themes/base/jquery.ui.selectmenu.css">
<link rel='stylesheet' type="text/css" href='fullcalendar-
1.6.3/fullcalendar/fullcalendar.css'>
<script type="text/javascript" src="http://code.jquery.com/jquery-
1.9.1.js"></script>
<script type="text/javascript" src='fullcalendar-
1.6.3/fullcalendar/fullcalendar.js'></script>
<script type="text/javascript" src="js/modernizr.custom.79639.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.core.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.widget.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.position.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.selectmenu.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.datepicker.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.stepper.js"></script>
<script type="text/javascript" src="js/ui/jquery.ui.dataTable.js"></script>
<script type="text/javascript" src="js/componentController.js"></script>
```

Starting a new page

- Use “template.jsp” for creating a new page. Duplicate it and rename the file for your desired page name.
- Change the title (`<title>Seneca | Template</title>`) of your page.
- Add your page link to the breadcrumb changing the destination pointed out by the *href* attribute (`"#"`) and also the *text* of the breadcrumb last link (`page name`).
- Change header section (`<h1>Page Name</h1>`) to have the title of your page on the screen and to make sure the respective menu is set for showing witch page user is accessing.
- Change the section title (`<h2>Section Title</h2>`) regarding the first section of your page.

Adding a new page section

- Inside `<form/>`, insert the following snippet after the last `</article>` on the page:

```
<article>
	<header>
		<h2>Section Title</h2>
		<img class="expandContent" width="9" height="6" 
src="images/arrowDown.svg" title="Click here to collapse/expand content"/>
</header>
	<div class="content">
		<fieldset>
		</fieldset>
	</div>
</article>
```

- Change the section title text (`<h2>Section Title</h2>`) regarding the new section.

Adding a new table component

- Add the following includes to the head: (See reference list for includes in the head tag)
    - `<link rel="stylesheet" type="text/css" media="all" href="css/themes/base/jquery.ui.core.css">`
    - `<link rel="stylesheet" type="text/css" media="all" href="css/themes/base/jquery.ui.theme.css">`
    - `<link rel="stylesheet" type="text/css" media="all" href="css/themes/base/jquery.ui.selectmenu.css">`
    - `<script type="text/javascript" src="js/ui/jquery.ui.core.js"></script>`
    - `<script type="text/javascript" src="js/ui/jquery.ui.widget.js"></script>`
    - `<script type="text/javascript" src="js/ui/jquery.ui.position.js"></script>`
    - `<script type="text/javascript" src="js/ui/jquery.ui.selectmenu.js"></script>`
    - `<script type="text/javascript" src="js/ui/jquery.ui.dataTable.js"></script>`
- Add the following script to the head:

```
<script type="text/javascript">
$(screen).ready(function() {
	/* TABLE */
	$('#tableName').dataTable({"sPaginationType": "full_numbers"});
	$('#tableName').dataTable({"aoColumnDefs": [{ "bSortable": false, 
"aTargets":[5]}], "bRetrieve": true, "bDestroy": true});
	$.fn.dataTableExt.sErrMode = 'throw';
	$('.dataTables_filter input').attr("placeholder", "Filter entries");
});
$(function(){
	$('select').selectmenu();
});
</script>
```

- Inside `<fieldset/>`, insert the following snippet on the page:

```
<div class="tableComponent">
	<table id="tableName" border="0" cellpadding="0" cellspacing="0">
		<thead>
			<tr>
```

```
                <!-- First field -->
                <th width="100" class="firstColumn" tabindex="15"
title="First field">First field<span> <!-- Sorting arrow --> </span></th>
                <!-- Field with expandable width -->
                <th title="Field with expandable width" <span> <!--
Sorting arrow --> </span></th>
                <!-- Regular field -->
                <th width="230" title="Regular field">Regular
field<span> <!-- Sorting arrow --> </span></th>
                <!-- Action field -->
                <th width="65" title="Action" class="icons"
align="center">Action</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td class="row">First field</td>
                <td>Field with expandable width</td>
                <td>Regular field</td>
                <td class="icons" align="center"><a href="#"
class="action"><img src="images/iconPlaceholder.svg" width="17" height="17"
title="Action description" alt="Action"/></a></td>
            </tr>
        </tbody>
    </table>
</div>
```

- Change the id (#tableName) in the script and in the table tag (id="tableName") regarding the new table component.
- Change all other necessary attributes, fields, and text regarding the new component.
- NOTES:
    - There must always be a field with expandable width.
    - Set width for regular fields.
    - Add a new css rule for extra actions (if does not exist), including the path for the respective icon in the *style.css* file. Replace “newAction” with the new one:

      ```
      .tableComponent table tbody tr td.newAction img {
          border: none;
          background-image: url('../../../images/newAction.svg');
      }
      ```
    - Change the *class* attribute for the new action on the page:

      ```
      <td class="icons" align="center"><a href="#" class="newAction"><img
      src="images/iconPlaceholder.svg" width="17" height="17" title="Action
      description" alt="Action"/></a></td>
      ```

Adding a new input component

- Inside <fieldset/>, insert the following snippet on the page:

  ```
  <div class="component">
  ```

```
	<label for="inputId" class="label">Input label:</label>
	<input name="inputName" id="inputId" class="input" tabindex="1"
title="Input description." type="text">
</div>
```

- Change *id*, *name*, *title* and *for* attributes regarding the new component.
- Make sure *tabindex* attribute follows the page sequence.
- Change all other necessary attributes and text regarding the new component.

Adding a new checkbox component

- Inside `<fieldset/>`, insert the following snippet on the page:

```
<div class="component">
	<div class="checkbox" title="Checkbox description"> <span class="box"
role="checkbox" aria-checked="true" tabindex="1" aria-
labelledby="checkboxId"></span>
		<label class="checkmark"></label>
		<label class="text" id="checkboxId">Checkbox description.</label>
		<input type="checkbox" name="checkboxName" checked="checked"
aria-disabled="true">
	</div>
</div>
```

- Change *id*, *name*, *labelledby* and *title* attributes regarding the new component.
- Make sure *tabindex* attribute follows the page sequence.
- Change all other necessary attributes and text regarding the new component.

Adding a new select box component

- Add the following includes to the head: (See reference list for includes in the head tag)
  - `<link rel="stylesheet" type="text/css" media="all" href="css/themes/base/jquery.ui.core.css">`
  - `<link rel="stylesheet" type="text/css" media="all" href="css/themes/base/jquery.ui.theme.css">`
  - `<link rel="stylesheet" type="text/css" media="all" href="css/themes/base/jquery.ui.selectmenu.css">`
  - `<script type="text/javascript" src="js/ui/jquery.ui.core.js"></script>`
  - `<script type="text/javascript" src="js/ui/jquery.ui.widget.js"></script>`
  - `<script type="text/javascript" src="js/ui/jquery.ui.position.js"></script>`
  - `<script type="text/javascript" src="js/ui/jquery.ui.selectmenu.js"></script>`
- Add the following script to the head:

```
<script type="text/javascript">
$(function(){
	$('select').selectmenu();
});
</script>
```

- Inside `<fieldset/>`, insert the following snippet on the page:

```
<div class="component">
	<label for="selectboxId" class="label">Select box label:</label>
	<select name="selectboxName" id="selectboxId" title="Select box
description. Use the alt key in combination with the arrow keys to select an
option." tabindex="1" role="listbox" style="width: 402px">
		<option role="option" selected>Default selected option</option>
		<option role="option">Regular option</option>
	</select>
</div>
```

- Change *id*, *name*, *title* and *for* attributes regarding the new component.
- Make sure *tabindex* attribute follows the page sequence.
- Change all other necessary attributes and text regarding the new component.

Adding a new button component

- Inside `<fieldset/>`, insert the following snippet on the page:

```
<div class="component">
	<div class="buttons">
		<button type="button" name="buttonName" id="buttonId"
class="button" title="Button Description" tabindex="1"
onclick="window.location.href='#'">Button text</button>
	</div>
</div>
```

- Change *id*, *name* and *title* attributes regarding the new component.
- Make sure *tabindex* attribute follows the page sequence.
- Change *href* value for the *onclick* attribute regarding the desired destination.

Adding a new search box component

- Inside `<fieldset/>`, insert the following snippet on the page:

```
<div class="component">
	<label for="searchBoxId" class="label">Search box Label:</label>
	<button type="submit" name="searchBoxButtonName" class="search"
tabindex="1" title="Search box button description"
onclick="window.location.href='#'"></button>
	<input type="text" name="searchBoxInputName" id="searchBoxId"
class="searchBox" tabindex="2" title="Search box input description">
</div>
```

- Change *id*, *name*, *title* and *for* attributes regarding the new component.
- Make sure *tabindex* attribute follows the page sequence.
- Change *href* value for the *onclick* attribute regarding the desired searching reference.
- Change all other necessary attributes and text regarding the new component.
- Implement ajax method to reload table with respective results.

Adding action buttons

- Inside `<form/>`, insert the following snippet after the last `</article>` on the page:

```
<article>
	<h4></h4>
	<fieldset>
		<div class="actionButtons">
			<button type="submit" name="submit" id="save" 
class="button" title="Click here to save inserted data">Save</button>
			<button type="button" name="button" id="cancel" 
class="button" title="Click here to cancel" 
onclick="window.location.href='calendar.jsp'">Cancel</button>
		</div>
	</fieldset>
</article>
```

- Change *href* value for the *onclick* attribute (`'calendar.jsp'`) regarding another possible desired destination.