

# Exercise (Instructions): Accordion, Scrollspy and Affix

## Objectives and Outcomes

In this exercise we explore the use of the collapse Javascript plugin together with panel component and panel group classes to create an accordion to show/hide content in a web page. We also leverage the nav pills class to create a side navigation menu. We use the scrollspy plugin to highlight the scroll position within this nav component. In addition we use the Affix plugin to enable the nav to scroll with the page initially, but get fixed shortly thereafter. At the end of this exercise, you will be able to:

- Design an accordion using the collapse plugin together with the panel component and panel-group class
- Use a scrollspy to highlight the current scroll position in a nav
- Use the affix plugin together with the scrollspy on the side nav component

## Adjusting the Content Rows

- For the first row of the content, adjust the column classes for the first column to *col-xs-12 col-sm-6 col-lg-8*. Then adjust the column classes for the second column to *col-xs-12 col-sm-6 col-lg-4*.
- For the remaining two rows, delete the second columns and adjust the column classes for the first column of content to *col-xs-12* so that it occupies the entire 12 columns.

## Converting Tabs to Accordion

- First delete the `<ul>` class that was introduced for the tabbed navigation.
- Then the turn the *tab-content* div into a *panel-group* div. Use the code structure as shown below:

```
<div class="panel-group" id="accordion"
      role="tablist" aria-multiselectable="true">
    . . .
</div>
```

- Then, replace the first tab-pane into a panel such that the name appears as a panel heading, and the `<p>` will be in the panel body. Use the structure of the code as shown below:

```

        <div class="panel panel-default">
            <div class="panel-heading" role="tab" id="headingPeter">
                <h3 class="panel-title">
                    <a role="button" data-toggle="collapse"
                        data-parent="#accordion" href="#peter"
                        aria-expanded="true" aria-controls="peter">
                        Peter Pan <small>Chief Epicurious Officer</small>
                    </a>

                </h3>
            </div>
            <div role="tabpanel" class="panel-collapse collapse in"
                id="peter" aria-labelledby="headingPeter">
                <div class="panel-body">
                    <p> . . . </p>
                </div>
            </div>
        </div>
    </div>

```

- For the remaining three leaders, use the same structure as above, with the appropriate ids set up for the panels, as shown in the code structure below:

```

        <div class="panel panel-default">
            <div class="panel-heading" role="tab" id="headingDanny">
                <h3 class="panel-title">
                    <a class="collapsed" role="button" data-toggle="collapse"
                        data-parent="#accordion" href="#danny"
                        aria-expanded="false" aria-controls="danny">
                        Dhanasekaran Witherspoon <small>Chief Food Office
                    </small></a>

                </h3>
            </div>
            <div role="tabpanel" class="panel-collapse collapse"
                id="danny" aria-labelledby="headingDanny">

```

pse"

>

pse"

```
        <div class="panel-body">
            <p> . . . </em></p>
        </div>
    </div>
</div>
<div class="panel panel-default">
    <div class="panel-heading" role="tab" id="headingAgumbe">
        <h3 class="panel-title">
            <a class="collapsed" role="button" data-toggle="colla

                data-parent="#accordion" href="#agumbe"
                aria-expanded="false" aria-controls="agumbe">
                Agumbe Tang <small>Chief Taste Officer</small></a>

        </h3>
    </div>
    <div role="tabpanel" class="panel-collapse collapse"
        id="agumbe"    aria-labelledby="headingAgumbe">
        <div class="panel-body">
            <p> . . . </em></p>
        </div>
    </div>
</div>
<div class="panel panel-default">
    <div class="panel-heading" role="tab" id="headingAlberto">
        <h3 class="panel-title">
            <a class="collapsed" role="button" data-toggle="colla

                data-parent="#accordion" href="#alberto"
                aria-expanded="false" aria-controls="alberto">
                Alberto Somayya <small>Executive Chef</small></a>

        </h3>
    </div>
    <div role="tabpanel" class="panel-collapse collapse"
```

```

        id="alberto"    aria-labelledby="headingAlberto">
        <div class="panel-body">
            <p> . . . </em></p>
        </div>
    </div>
</div>

```

- After completing the update, check the behavior of the accordion on the web page.

## Adding Scrollspy with Affix

- First we will adjust the content so that we create some space on the right side of the page to accommodate the nav component for the scrollspy behavior. To do this, we will nest the entire content of the web page into a row first, and move the existing content into a column with a column class of col-sm-10. The remaining two columns will be left for the nav. To do this, add the following code right after the container div:

```

<div class="row">
  <div class="col-xs-12 col-sm-10">
    . . .
  </div>
</div>

```

- To introduce the scrollspy, we first apply some data-\* attributes to the body of the page as follows:

```
<body data-spy="scroll" data-target="#myScrollspy" data-offset="200">
```

This prepares the body to activate the scrollspy using the *data-spy="scroll"* attribute. The target of this is set using *data-target="#myScrollspy"* thus specifying that the element with the id *myScrollspy* will act as the scrollspy.

- To introduce the scrollspy with the affix plugin, add the following code to the end of the container, inside the outer row:

```
<nav class="hidden-xs col-sm-2" id="myScrollspy">
```

```

00">
    <ul class="nav nav-pills nav-stacked" data-spy="affix" data-offset-top="4
    <li><a href="#history">Our History</a></li>
    <li><a href="#corporate">Corporate</a></li>
    <li><a href="#facts">Facts</a></li>
    </ul>
</nav>

```

This nav element is introduced with the classes *hidden-xs* to indicate that this element will be hidden for extra small screen sizes. It will occupy the remaining two columns (specified through the *col-sm-2* class) for the small to large screen sizes. We see that the *nav-pills* is applied to the *<ul/>* along with *nav-stacked* to create a vertically stacked nav component. Also the affix plugin is applied to this nav component. The list items use the ids of the three rows. We need to apply the ids to the three rows.

- Apply the appropriate ids for the three content rows as follows:

```

<div id="history" class="row row-content"> ... </div>
<div id="corporate" class="row row-content"> ... </div>
<div id="facts" class="row row-content"> ... </div>

```

- Finally, add the following CSS styles to the mystyles.css file to support the affix:

```

.affix {
    top:100px;
}

```

- Now that all updates are complete, you can now view the web page to see the scrollspy and affix in action.

## Conclusions

In this exercise we constructed the accordion using the collapse plugin together with the panel components. We also explored the use of scrollspy and affix plugins.