NOKIA Developer

Mobile Web Design Pattern: Tabbed List

This design pattern is part of the Mobile Web Design Patterns series.

Description

A series of list items arranged horizontally to form tabs which when clicked, reveal a pane of associated content. **Note:** There are many ways to create a tabbed list in HTML. Some of these require the creation of two lists—one for the tabs themselves and the other for the associated content. This reveals one of the weaknesses of tabs on the mobile web as it can be tricky to create a tabbed interface that will remain semantic and degrade gracefully on less capable browsers.

Associating a badge with a specific tab allows you to connect the information in the badge with a particular mode in your application, even when that mode is not the current one. The main purpose of Badge is to convey information in tab bar.

You can display a badge on a tab to communicate with users in a non intrusive, understated way. This type of feedback is suitable for communicating information that isn't critical to the user's task or context, but that is useful to know. The badge looks similar to the one Phone displays on the Voice mail tab to indicate the number of unheard messages: it is a red oval that appears near the upper right corner of the tab. The white text inside the oval provides the information.

Use when

- For top level navigation, as an alternate styling to a horizontal list.
- When combined with JavaScript to show/hide small amounts of related data.

Rationale

In this case, the rationale for use includes a cautionary tale. Tabs are often used within native apps to navigate between views. In these scenarios, the entire view is mapped to the tab. In other words, clicking Left and Right within the view will (typically) only map to switching from one tab view to the other. This mapping is persistent throughout the view enabling users to scroll at will, then switch views from anywhere within the page with a simple Left or Right click.

On the mobile web, the Left and Right keys cannot be custom mapped so in effect, the tabs are merely links that happen to look like tabs. To switch from one view to another, the user must actually navigate to the tab then click it. For this reason, the two examples provided above are the few that really make sense. Let's look at them in more detail:

1. In lieu of top level navigation.

Here the tabs are just an alternative styling of a top nav bar that could have been a horizontal list, or a horizontally aligned series of buttons. When a link is clicked, a new page loads.

As a best practice, this type of use should include a 'back to top' link and/or repeat the navigation options at the bottom of the page. Without these, users will have to scroll back to the top of the page to move onwards.

2. To show/hide small amounts of data.

Here, clicking the tab is designed to alternate between different content views. This is most elegantly achieved asynchronously, using JavaScript to show/hide a div containing the content in question. This view change happens immediately and does not require a page refresh.

The reason it's recommended to keep the content small is that you otherwise end up with the same problem noted above—the user has to scroll back up to the tab to switch the view.

Note: This is also why tabs are ideally suited to simple widgets which often do not scroll and make wide use of JavaScript to show/hide data and switch views.

Figure: The Yahoo mobile web site uses a small tabbed widget to display news headlines. Unfortunately, this particular implementation requires a full reload of the page each time a tab is clicked. This makes for a disjointed experience (especially as the tabs are midway down the page).

Design Tips

- The selected tab should be highlighted (using contrasting colour, size or type style), and the deselected tabs should be legible enough to appear clickable.
- The selected tab should appear visually connected to its content area. This establishes a clear hierarchy between the tab label and the content below.
- Avoid using multiple rows of tabs. These can be difficult to read and confuse the visual hierarchy described above. Double tab navigation, while useful on the desktop can be particularly confusing on mobile and will often suffer from chronic lack of space. In these cases, it is best to use a traditional drill-down list approach instead.
- Keep tab labels short. If the labels are long and/or begin to resemble small sentences, the content you are alternating between may benefit from a different navigation scheme.

Two excellent resources for tab design (albeit not mobile focussed) are Tabs, Used Right and the section on tabs in the book Don't Make me Think.