

Components 4:

Wrapping jQuery Widgets

You can access the docs for the jQuery UI widgets at <http://jqueryui.com/demos/>.

1. The `cc:attribute` element has “method-signature” and “targets” attributes that you use when you are referring to a regular method instead of a bean property. The method-signature attribute gives the return type and arguments (the method name is arbitrary), and the targets attribute refers to the element of the implementation section that will use the method. Using these two attributes lets you easily make a composite component that generates a command button:

```
<cc:interface>
  <cc:attribute name="action" required="true"
    methodSignature="java.lang.String action()"
    targets="button"/>
</cc:interface>
<cc:implementation>
  <h:commandButton value="Click Me"
    action="#{cc.attrs.action}" id="button"/>
</cc:implementation>
```

Given this idea, make a component called `utils:commandButton1` that uses “value” and “action” just like `h:commandButton`, but results in the stylized jQuery UI button. This is a very basic reimplementaion of the PrimeFaces `p:commandButton`. Make a test case with a button that just runs an action controller method that navigates to another page. Hint: easy because you can turn the `h:commandButton` into a stylized jQuery UI button without knowing the ID of the `h:commandButton`. Do this by automatically giving your `h:commandButtons` a particular CSS name.

2. Make a better version called `utils:commandButton` that allows the page author to pass in the “disabled” attribute, in order to determine if the button is displayed normally (default) or grayed out and inactive (`disabled="true"`). You should pass this attribute to jQuery, not to `h:commandButton`. For example, in jQuery itself, to make it disabled, you do this:

```
$(patternMatchingButton).button({ disabled: true });
```

This is more difficult than the previous version because you cannot turn all the `h:commandButtons` into jQuery buttons in the same manner. Instead, you have to focus on a particular `h:commandButton`, which means you need to use `#{cc.clientId}` as discussed in the lecture.

Super-Fast jQuery Review

1. Find all elements that have a particular CSS class name (notice the “.” below)

```
$(".your-css-class-name")
```

2. Find all h1 elements (notice the lack of a “.” below)

```
$("h1")
```

3. Find “all” (0 or 1) elements that have a given id (notice the “#” below)

```
$("#your-id")
```

4. Find all h1 elements that use class="foo" and are also inside a div that has id="bar"

```
$("#div#bar h1.foo")
```

5. Turn all matching elements into styled buttons

```
$(".some-css-selector-pattern-like-the-ones-above").button();
```

6. Run JavaScript code after page load (technically on DOM load)

```
$(function() {  
    someJavaScriptCall(...);  
    $(".some-pattern").somejQueryCall();  
    ...  
});
```

7. Read about buttons

- <http://jqueryui.com/button/>
- <http://api.jqueryui.com/button/>

8. See the names of the standard jQuery UI icons

- <http://api.jqueryui.com/theming/icons/>
- <http://www.petefreitag.com/cheatsheets/jqueryui-icons/>