UNOBTRUSIVE JAVASCRIPT WITH

jQuery UI Dialog

January 26, 2011
Hamed Foroozanfard
jQuery UI Dialog API

- Feel Rusty on JS? see lecture notes from UCSD Extension: [http://classes.pint.com/javascript1/](http://classes.pint.com/javascript1/)
- UXT Sample App: [https://pascal.ucsd.edu/sample/dialog/index.htm](https://pascal.ucsd.edu/sample/dialog/index.htm)

Common Questions

- Q: Why does my dialog open automatically? See [This Blog Post](http://blog.nemikor.com/) & UXT Sample App
  - ✓ A: set option: `autoOpen: false`

- Q: How to load a page into a dialog? See [This Blog Post](http://blog.nemikor.com/)
  - ✓ A: load contents of the linked page into a `div` and create a dialog

  Possible side-effect: the page contents are loaded immediately after the document is ready, even if the user never opens the dialog. To solve use `.one()` or define `.click()` on the target element.

  ```javascript
  var $dialog = $('<div></div>').load($link.attr('href') + '#content') //.load() loads only when the content is ready
  .dialog({
      autoOpen: false,
  });
  ```

- Q: How to load a page on demand? See [This Blog Post](http://blog.nemikor.com/) & UXT Sample App
  - ✓ A: use jQuery `.one()` function (.one() binds a handler to an event, but the handler is only ever run once.)
  - ✓ A: use create a dialog variable and define a `click` function on the target element

See Demo
Problems & Solutions: Target Content Caching In General

- Q: How to overcome caching of dialog content?
  - A1: (Client Side) turn-off caching all-together using jQuery Ajax API (not recommended)
    
    
    \$\text{.ajaxSetup} \{
    \text{cache: false }
    \};
    
    - A2: (Client Side) add a random # or a timestamp to the target link being loaded
      - side-effect: resources are fetched fresh every time now (server load goes up)!!! See next slide for solution
      
      \$\text{dialog.dialog\{......\}.load(}$
      \$\text{\$\text{link.attr('href')}\"\&\text{random=\"}\text{+Math.random()}\text{\"*999999}$
    
    - A3: (Server Side) modify response header
      
      \text{response.setHeader("Cache-Control","no-cache"); //HTTP 1.1}
      \text{response.setHeader("Pragma","no-cache"); //HTTP 1.0}
      \text{response.setDateHeader("Expires", 0); //prevents caching at the proxy server}
    
    - A4: Give me another one for a prize!
Problems & Solutions: Target Resources Caching

• Q: How to overcome caching of JS/CSS Files in the Target Content Page?
  ✓ A1: (Server Side) Version JS/CSS files (i.e. myjs1.x.x.js) w/new releases; new names will force a new load on the client side
    - side-effect: evaluate maintenance costs
  ✓ A2: (Client Side) Load scripts/styles using JavaScript w/delay after page loads (similar to including js files/scripts at the bottom of the page but w/more control to you! ;)
    - Advantages: (a) Faster Initial Page Loads + (b) You Decide when to Cache!

```javascript
function importJS(src) {
    var scriptElement = document.createElement('script');
    scriptElement.setAttribute('src', src);
    scriptElement.setAttribute('type', 'text/javascript');
    document.getElementsByTagName('head')[0].appendChild(scriptElement);
}
function importJSNoCache(src) {
    var ms = new Date().getTime().toString();
    importJS(src + '?' + ms);
}
var delay = 1;
setTimeout("loadExtraFiles();", delay * 1000);

function loadExtraFiles() {
    importJS('/uidialog/js/custom.js');
    ........
}

✓ A3: Give me another one for a prize!
```
Problems & Solutions: Library Loading Conflicts

• Q: Close or Any of the Predefined Dialog button(s) don’t work!
  ✓ A1: most likely loading jQuery more than once i.e. jQuery included in the dialog content as well as the source page which will cause a library conflict see A2 for solution

  ✓ A2: exclude site-mesh decorator application from the target page in decorator.xml
    why? UXT decorators load jQuery by default which causes a conflict w/dialog buttons since the source page had already loaded jQuery (google “jQuery noConflict mode” if you have to use multiple instances of jQuery)

    <decorators>
      <excludes>
        <pattern>*.json</pattern>
        <pattern>*/loadDialogContent</pattern>
      </excludes>
    ......
    </decorators>

  ✓ A3: Give me another one for a prize!
Problems & Solutions: Pre-defined Buttons

- Q: How to add generic buttons to a dialog?
  ✓ A1: use buttons: { ... } property when defining the dialog

  ```javascript
  $dialog.dialog({
    ............
    buttons: { "Cancel": function() {
      $(this).dialog("close");
      $(this).dialog("destroy");
    }
  },
  //you can also override the pre-defined close prototype
  close: function(ev, ui) {
    $(this).remove();
  }
  ............
  }).......
  ```

- Q: How to remove the dialog completely when closed?
  ✓ A: see previous question override close function of the Dialog box.
Problems & Solutions: Dialog w/Form Plug-in

• Q: When using Dialog w/jQ Form Plug-in in the Form plug-in does not kick-in and form submits as usual hence Causing a redirect pre-form submission takes place?

✓ A1: see previous question, also make sure you delay loading of extra libraries using the techniques in the previous slide.
✓ A2: could be a timing issue see next Q

• Q: When using Dialog w/jQ Form Plug-in and debugging everything works as expected but not otherwise?

✓ A1: could be a timing issue. Remember AJAX requests are asynchronous by default i.e. pre-form submission, If you need to do some kind of server-validation (for example) you need to make a synchronous AJAX request or have a callback function to see if there is a response yet, here is my solution:

```javascript
function validateJSON(){
    var res = $('#formSearch').ajaxSubmit(
        {target: '#content',
         beforeSubmit: resetErrCounter,
         dataType: 'json',
         url: '/uidialog/postJsonValidator',
         async: false
         }, success: processJson
        //,delay: {callback: res, time: 5000 // delay timeout in ms} //<= alt. using a callback instead of async option
        );
    // alert(res);
    return errors();
}

✓ A3: Give me another one for a prize!

Presenter: Hamed Foroozanfard
Useful: jQuery + Spring 3 → Rock Your World!

• Spring 3 now has a JsonView built in just add Jackson Libraries to the Classpath and add a couple of lines to your config files and you are good to go.

```xml
<bean class="org.springframework.web.servlet.view.ContentNegotiatingViewResolver">
  <property name="mediaTypes">
    <map>
      <entry key="json" value="application/json"/>
    </map>
  </property>
  <property name="defaultViews">
    <list>
      <bean class="org.springframework.web.servlet.view.json.MappingJacksonJsonView">
        <!-- property name="prefixJson" value="true"*/-->
      </bean>
    </list>
  </property>
  <property name="order" value="0"/>
</bean>
```

• Now your controller will recognize if you are requesting a JSON response and will automatically return a JSON mapping of the Model if requested.

```javascript
$('#formSearch').ajaxSubmit({
  target: '#content',
  beforeSubmit: resetErrCounter,
  dataType: 'json',
  url: '/uidialog/postJsonValidator',
  async: false
  ,success: processJson
  // ,delay: {callback: res, time: 5000 // delay timeout in ms} //<= alt. using a callback instead of async option
});
```

Presenter: Hamed Foroozanfard
IxEdit: jQuery by Example

http://www.ixedit.com/

IxEdit is a JavaScript-based interaction design tool for the web. With IxEdit, designers can practice DOM-scripting without coding to change, add, move, or transform elements dynamically on your web pages. Especially, IxEdit must be useful to try various interactions rapidly in the prototyping phase of your web application.

What is Interaction?
Interactions are visual changes of the screen which occur when users act on user interface elements. In other words, interactions are the behaviors of the user interface. For instance, when a user clicks a button, an image switches to another, or when a user drags an edge, the viewpoint expands. Those are interactions. Users are doing their jobs with a computer through various interactions generally.

IxEdit solves the problem
To implement interactions on a web page, programming with JavaScript is needed. However, it is hard to manage JavaScript for many designers. Therefore, making well-designed web interactions is difficult in general. IxEdit solves this problem. If you have basic knowledge about HTML and CSS, you can create interactions as you like. JavaScript coding is no longer needed.
Useful: JavaScript Development

- IDEs
  - Aptana Studio
  - MyEclipse (PAID)
  - NetBeans
  - Cloud9 IDE ← cloud IDE just announced by Mozilla

- Bottom line, don’t rely on IDEs for JS development too much, firebug (or alike for different browsers) Maybe your best friends 😊

THANKS & ENJOY!

Presenter: Hamed Foroozanfard