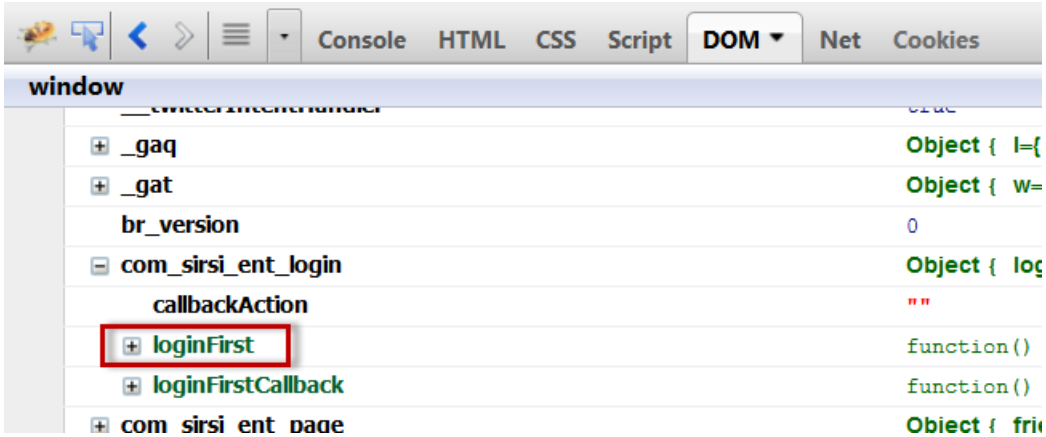
Cool Stuff That Was There All Along

- A handy function that presents an Enterprise login
- Checking for the logged-in state
- Built-in table sorting you can re-use

The Enterprise loginFirst() function

- The “DOM” tab of firebug reveals this:

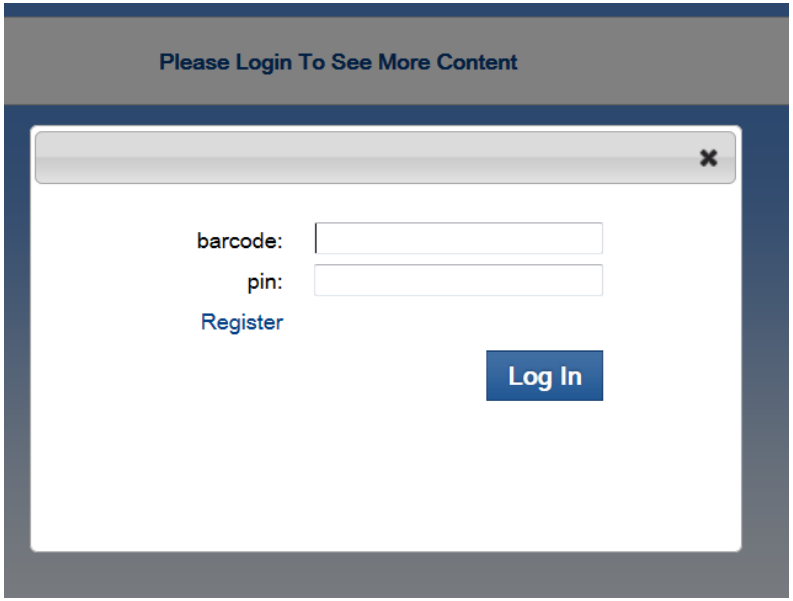


- This function will present Enterprise’s login dialog if the user isn’t logged in.
  - If the user is already logged in, the function does nothing

A Quick Test Of loginFirst()

For a quick test, let's put this in a Rooms content area as HTML:

```
<p align="center"><strong><a
href="javascript:com_sirsi_ent_login.loginFirst()">Please
Login To See More Content</a></strong></p>
```



COSUGI 2014

Testing for the logged-in state

- An AJAX call to ask Enterprise if the user is logged in:

```
/*create a URL to submit to getJSON*/
var entUrl="/client/"+encodeAsTapestry(com_sirsi_ent_page.friendlyUrl)+"/search/patronlogin:loggedIn";
jQuery.getJSON(entUrl ,function(data){
var isLoggedIn=data.loggedIn;
if(isLoggedIn) {Write code between these braces to do something if they are logged in}
)};
```

Combining the login concepts

- Show rooms content only if user is logged in (JavaScript upload, not widget!)

```
function hideRooms(){
var entUrl="/client/"+encodeAsTapestry(com_sirsi_ent_page.friendlyUrl)+"/search/patronlogin:loggedIn";
jQuery.getJSON(entUrl ,function(data){
var isLoggedIn=data.loggedIn;
if(isLoggedIn){
jQuery('.content_container').css('display','block');
jQuery('#taxonomyContainer').css('display','block');
}
else{
var loginHtml = '<b><a href="javascript:com_sirsi_ent_login.loginFirst()">Please Login To See More Content</a></b>';
jQuery('.content_container').css('display','none');
jQuery('#taxonomyContainer').html("");
jQuery('#taxonomyContainer').append(loginHtml);
}
});
}

function customJavaScript(){
/*this is necessary for IE.  Otherwise, it will cache previous ajax requests
for the logged-in state, leading to incorrect values*/
jQuery.ajaxSetup({ cache: false });
jQuery(document).ready(function(){
hideRooms();
});
}
```

Caveat!

- The code sample inserts the login link in the #taxonomy_container ID. This only exists if you have a taxonomy of rooms (more rooms displayed in your profile than just the Home room, in other words).
- If just have *just* a Home room, you'll need to rethink the following:

/*you won't need this next line*/

~~jQuery('#taxonomyContainer').html("");~~

/*you'll have to pick a different target than the #taxonomy_container, since you won't have one if you have nothing but a home room*/

~~jQuery('#taxonomyContainer').append(loginHtml);~~

/*there are many other places you might stick the login link. Example: below the main menu links like 'login, my account, ada, helps, etc. The line above could be replaced with:*/

jQuery('#mainMenuContainer').append(loginHtml);

Cool Variables That Were Already There

- Ever been taken away by nice men in white coats after trying to dissect an Enterprise URL for a profile or language code?
  - Enterprise URLs “shift” a bit depending on page, whether a language code has been selected, etc.
  - Profile/language codes can be yanked with a few lines of code. Let’s shoot for one line of code, and one that might last beyond the next upgrade.

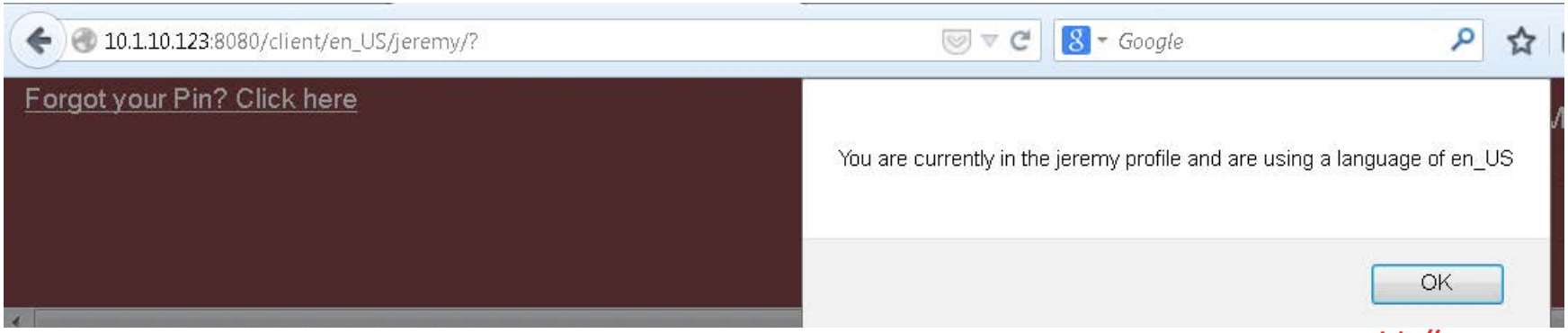
- The “DOM” tab of Firebug exposes this nifty object containing a couple of cool properties:

COSUGI 2014

Cool Variables That Were Already There(2)

- The applications of profile/language codes in scripts are huge (i.e. policy translations by branch, translating for multiple languages in one script)
- Try the following code in a console like Firebug:

```
var curProfile= com_sirsi_ent_page.friendlyUrl;
var curLang = com_sirsi_ent_page.localeCode;
var myMsg='You are currently in the ' + curProfile + ' profile';
myMsg += ' and are using a language of ' + curLang;
alert (myMsg);
```



Cool Variables That Were Already There(3)

- It is recommended that you query these variables in an event like jQuery(document).ready() or a timeout delay. Otherwise, in some cases your script may attempt to query the com_sirsi_ent_page object before it really exists.
- It doesn't work in Internet Explorer's compatibility mode. But then again, Enterprise doesn't either.

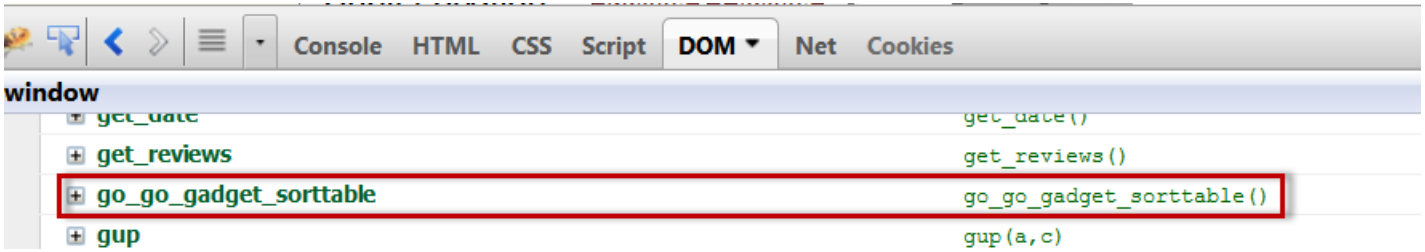
Sortable Search Result Items

- There are many JavaScript/jQuery scripts/plug-ins that can sort table data. Use whatever you like. But…
- Engineering already uses a 3rd-party one called ‘sortable.js’.
  - Really cool work by Stuart Langridge, http://www.kryogenix.org/code/browser/sorttable/
  - SD hacked it a bit, building in the nifty arrows, styling, other fixes
- Many customers have implemented an “items on search results” widget. Why not make it sortable?

Library	Call No.	Item ID	Item Type	Current Location	Home Location
ARROWOOD	JF ROW	310000001	BOOK	INTRANSIT	STACKS
ARROWOOD	JF ROW	310000002	BOOK	CHECKEDOUT	STACKS
ARROWOOD	JF ROW	310000000	BOOK	STACKS	STACKS

Sortable Search Result Items (2)

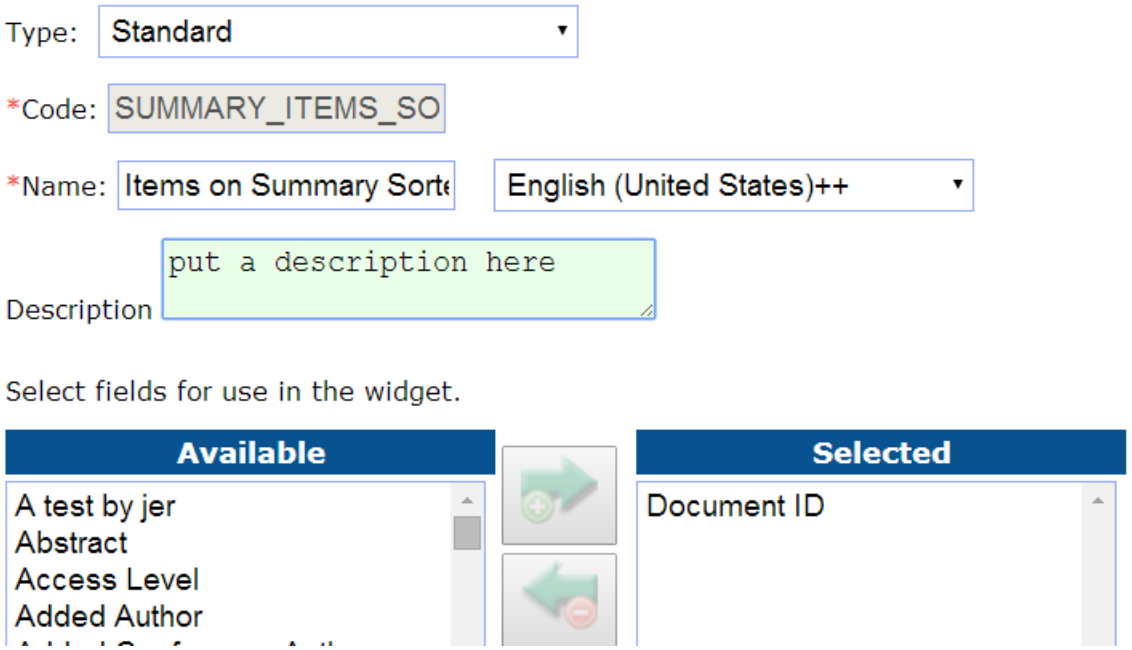
- DOM inspection showed that SD Engineering wrapped the sortable.js into a function (go_go_gadget_sorttable):



On a page that doesn't normally have sortable tables (like the search results page), you have to invoke this function before you can access the sortable.js' methods.

After you do that, it works largely as described at:
http://www.kryogenix.org/code/browser/sorttable/

Sortable Search Result Items (3)



Sortable Search Result Items (4)

HTML content:

```
<script type="text/javascript">
 getSummaryItems('$RESULT_ID');
</script>
```

Custom JavaScript

```
function getSummaryItems(rId){
go_go_gadget_sorttable(); //invoking SDs wrapper function
var hitString = rId.split('hitlist');
var hitNum=hitString[1];
var tableId='table_'+rId;
var catKey = (jQuery ('#'+rId+'_DOC_ID').text().split(":"))[1];
showSummaryItems(catKey,hitNum,tableId);
}

function showSummaryItems(catKey,hitNum,tableId){
var numRows=0;
jQuery.getJSON('http://10.1.10.122:8080/symws/rest/standard/lookupTitleInfo?clientID=DS_CLIENT&titleID=' + catKey+'&includeItemInfo=true&json=true&callback=?',function(tr){
callInfo=tr.TitleInfo[0].CallInfo;
var htmlItemOutput='';
jQuery.each(callInfo,function(entryIndex,entry){
var libID = this.libraryID;
var callNum = this.callNumber;
jQuery.each(callInfo[entryIndex].ItemInfo,function(itemIndex,item){
numRows++;
htmlItemOutput += '<tr><td>'+libID+ '</td><td>'+callNum+'</td><td>'+ this.itemID+'</td><td>' + this.itemTypeID
+'</td><td>'+this.currentLocationID+'</td><td>'+this.homeLocationID+'</td></tr>';
});
});
htmlHeaderOutput='<thead><tr style="background-color:#EEEEEE"><th>Library</th><th>Call No.</th><th>Item ID</th><th>Item Type</th><th>Current Location</th><th>Home
Location</th></thead>';
/*any table you want to be sortable with this plug-in must have a class of sortable*/
htmlFinalOutput= '<table class="sortable" style="width:145%;margin-left:-80px;border:1" id="'+tableId + '">'+ htmlHeaderOutput + htmlItemOutput + '</table>';
jQuery('#results_bio'+hitNum).append(htmlFinalOutput);
jQuery(document).ready(function(){
if(numRows>1){
sorttable.makeSortable(document.getElementById(tableId));
}
});
});
}
```

LET'S GET REAL

Using Widgets and Web Services in Enterprise

Michael Robinson, Head of Systems, University of Alaska Anchorage
http://consortiumlibrary.org/blogs/mcrobinson/