

Below is an email from Andreas Zientek, @andreasz on splunk-usergroups slack!

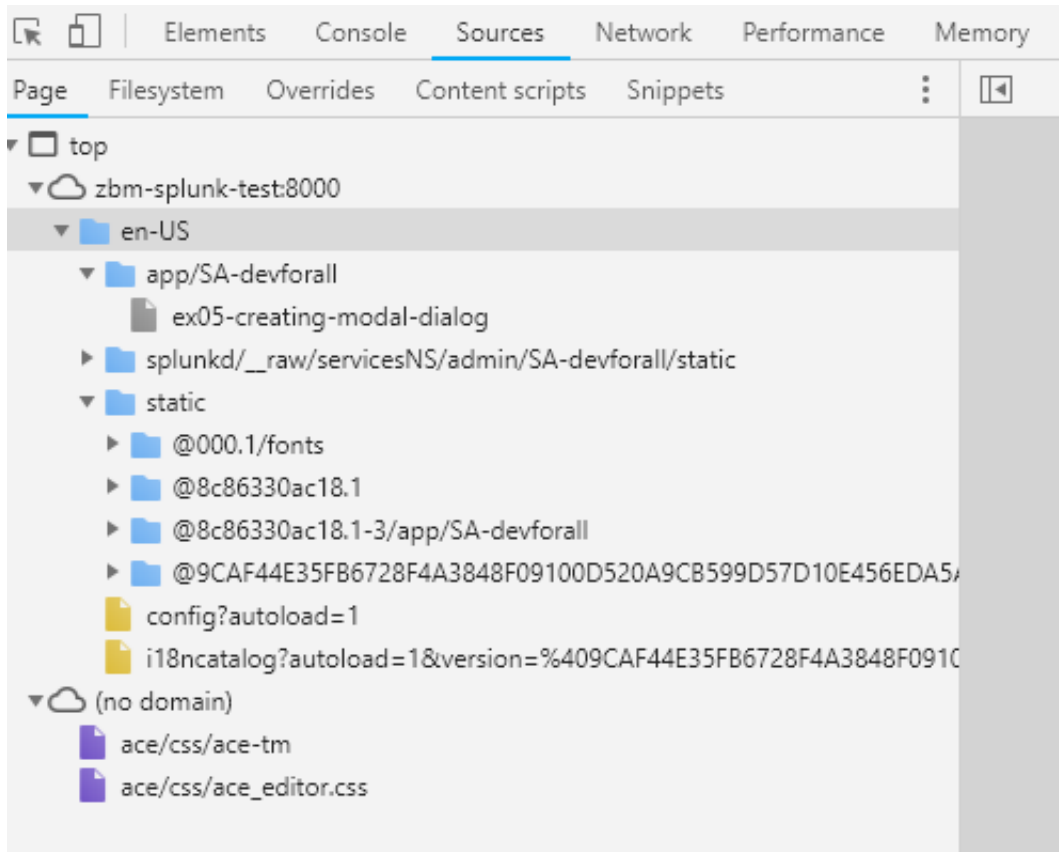
---

Hello David,

First of all the app is great. Big Thx! Maybe I can contribute a little. There is a very helpful trick to debug JavaScript code. I think it would be a big help for the users of the app.

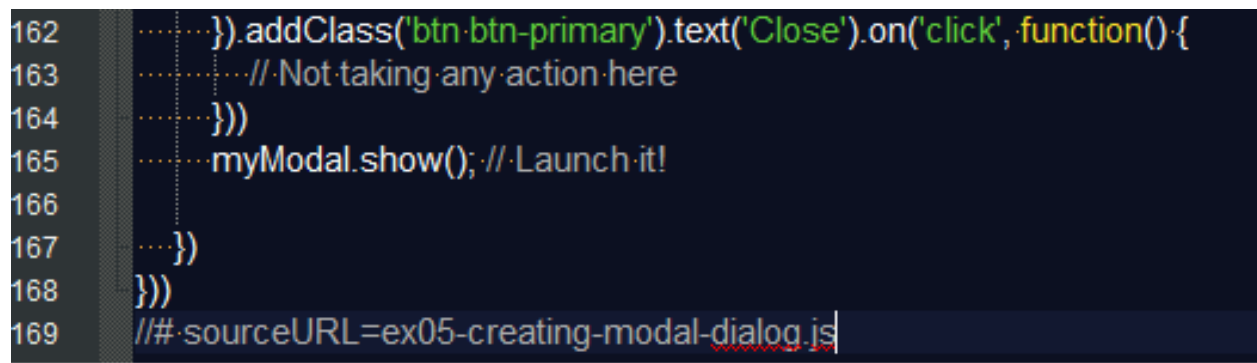
Let's take ex05-creating-modal-dialog.js as example.

If you open the Chrome DevTools Source tab, you won't find the JS file in the tree

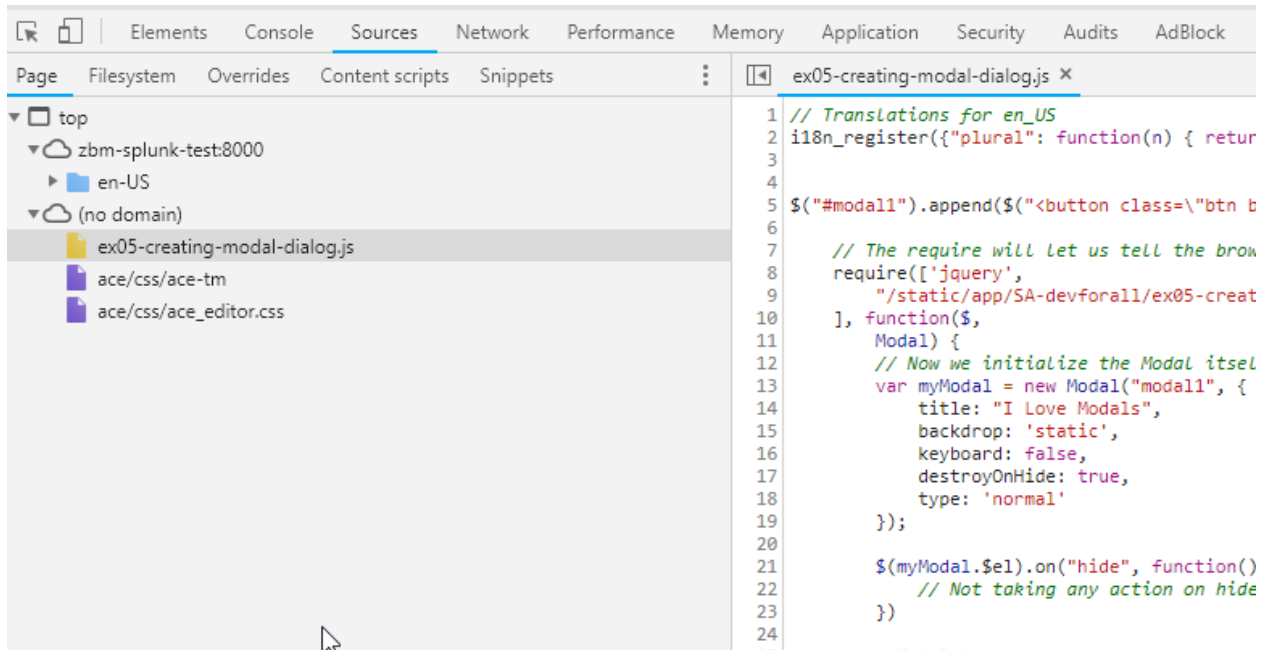


But if you put the following statement as the last line in the JS file:

```
///sourceURL=ex05-creating-modal-dialog.js
```

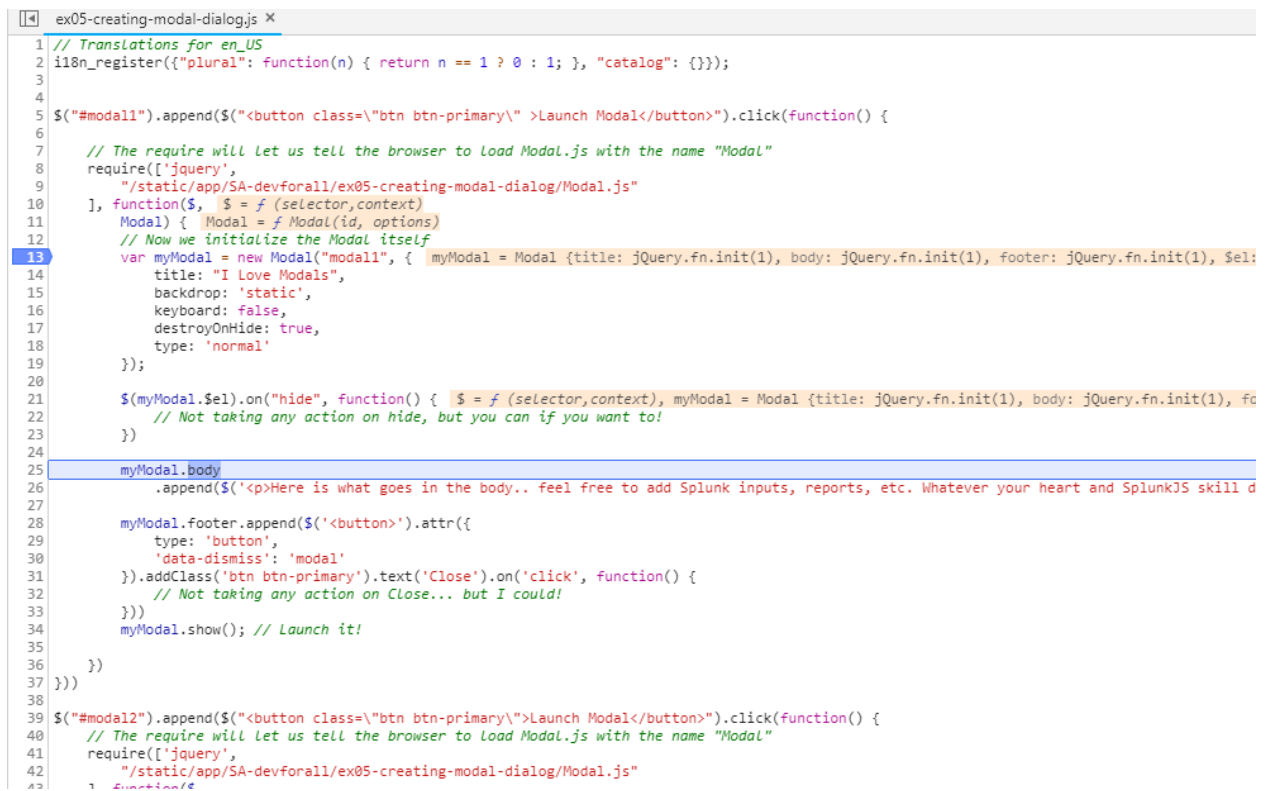


After a `_bump` load the page again. Now you can find the JS file under the “(no domain)” node.



The screenshot shows the Chrome DevTools Sources tab. On the left, the file tree is expanded to show the file `ex05-creating-modal-dialog.js` under the "(no domain)" node. The main pane displays the JavaScript code for this file, starting with a comment `// Translations for en_US` and a function `i18n_register`. The code includes a jQuery click handler for a button with class `btn btn-primary` that triggers a modal. The modal is initialized with the following options: `title: "I Love Modals", backdrop: 'static', keyboard: false, destroyOnHide: true, type: 'normal'`.`

After setting a breakpoint, refresh the page and click on the “Launch Modal” Button. The debugger stops the process and we can go step by step:



The screenshot shows the same Chrome DevTools Sources tab, but with a breakpoint set at line 13 of `ex05-creating-modal-dialog.js`. The code is the same as in the previous screenshot, but the breakpoint is highlighted in blue. The code shows the modal initialization and the click handler. The modal is initialized with the following options: `title: "I Love Modals", backdrop: 'static', keyboard: false, destroyOnHide: true, type: 'normal'`.`

When I hover over the objects I can see the actual values:

```
5 $("#modal1").append("<button class='btn btn-primary' >Launch Modal</button>").click(function() {
6
7 // The require will let us tell the browser to Load Modal.js with the name "Modal"
8 require(['jquery',
9 "/static/app/SA-devforall/ex05-creating-modal-dialog/Modal.js"
10 ], function($, $ = f(selector, context)
11 Modal) { Modal = f Modal(id, options)
12 // Now we initialize the Modal itself
13 var myModal = new Modal("modal1", { myModal = Modal {title: jQuery.fn.init(1), body: jQuery.fn.init
14 + "I Love Modals"
15
16 Modal
17 ▶ $el: jQuery.fn.init [div#modal1.modal.h:
18 ▼ body: jQuery.fn.init(1)
19 ▼ 0: div.modal-body
20   accessKey: ""
21   align: ""
22   assignedSlot: null
23   attributeStyleMap: StylePropertyMap
24   attributes: NamedNodeMap {0: class,
25   autocapitalize: ""
26   baseURI: "https://zbm-splunk-test:80
27   childElementCount: 0
28   childNodes: NodeList []
29   children: HTMLCollection []
30
31 }
32 // Not taking any action on close... but I could!
33 })
34 myModal.show(); // Launch it!
35
36 })
37 })
38 })
```

It makes the debugging so much easier and you can see the code flow (if..else statements, etc.). I hope you like the idea!

Mit freundlichen Grüßen / Kind Regards

Andreas Zientek