

Best Practice: How to use an extension to host JavaScript in Custom Controls

August 2020

You can add JavaScript to a report to provide your own user interfaces when the report is run in HTML. For more information, see [Adding JavaScript to a report](#) and [JavaScript samples](#).

Best Practice: Use an [extension](#) to host and upload JavaScript. Extensions can be used to host custom control and page module JavaScript.

Advantage: If you bundle your JavaScript file(s) into an extension, you can easily upload the extension to your IBM Cognos Analytics environment without having to access your server's file system directly to place the JS files.

How to bundle JavaScript in an Extension

1. Create your JavaScript file(s) as you normally would
2. Extensions are defined in a spec.json file that is contained in the root of the extension .zip file. In a text editor, create a new file called spec.json using this template (see [Slide 4](#)).
3. Create a zip file that contains the JavaScript file(s) from Step 1 and the spec.json file from Step 2.
4. [Upload](#) the extension to IBM Cognos Analytics
5. Create a report, drag in a Custom Control from the Toolbox, and set the *Module path* property to match your extension (see [Slide 5](#)).

spec.json Template

```
{
  "name": "AZIndexedPromptBar",
  "extensions": [
    {
      "perspective": "common",
      "features": [
        {
          "id": "AZIndex"
        }
      ]
    }
  ]
}
```

The only value that needs to be filled in is the "name" property, which is used to reference the JavaScript files contained in the extension.

Do not use spaces in the name.

Optional: The "id" property helps you identify the name of the JS file that the extension is using

In the report, select the Custom Control, open the Properties, and set the *Module path* property as follows:



GENERAL	
Description	Datalist Prompt
UI type	UI without event propagation
Module path	../v1/ext/AZIndexedPromptBa...
Configuration	{ "height": "200px", "multiple": "...
Height	
Width	
MISCELLANEOUS	
Name	Datalist Prompt

The "name" property from spec.json

Module path

../v1/ext/AZIndexedPromptBar/AZIndex

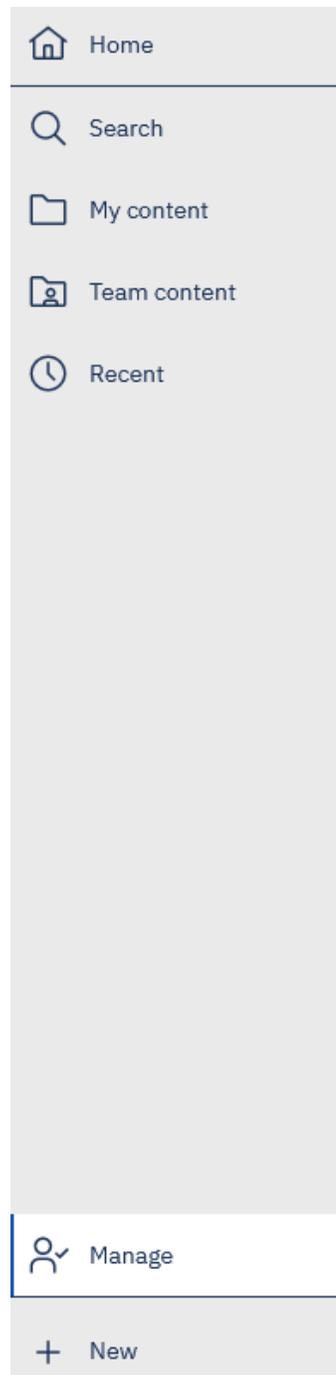
Standard URL prefix

OK

Cancel

The filename of the associated JavaScript file, without the .js at the end

Tip: The "name" property is also shown on the *Extensions* tab under: *Manage > Customization*



< Customization

Themes Extensions Palettes Views Profile Parameters Custom Visuals

Name	Modified ^
AZIndexedPromptBar	6/18/2020 7:10 AM
BasicMultiselectHTMLList	6/18/2020 7:10 AM
HTMLDataListWithTextFilter	6/18/2020 7:10 AM
MultiStyledHTMLDropdown	6/18/2020 7:10 AM

