Hyperlink Detection in a Simple Table: Qlik Sense

ALERT
This module is for testing purposes only and is a proof of concept for extension editing in Qlik Sense.
***This module is not supported by Qlik.**

INTRODUCTION

One of Qlik's largest differentiators is that it offers open APIs. This allows for the platform to be customized, extended, and embedded easily. A great place to visualize this is through Qlik Sense's "Developer Hub" which is accessible from both the desktop and server versions.

In this tutorial, we will take some sample code that was created for the "Simple Table" extension and modify it to allow for optional hyperlink detection. This example illustrates how easy it is for a developer to get started creating/editing extensions.

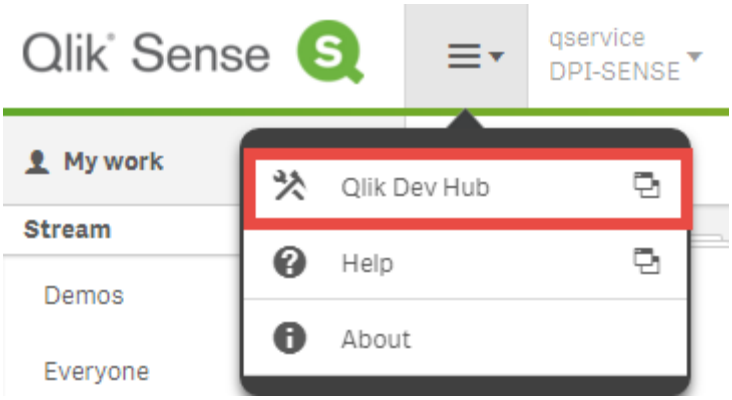
REQUIREMENTS

- Qlik Sense 2.x, Qlik Sense Desktop 2.x

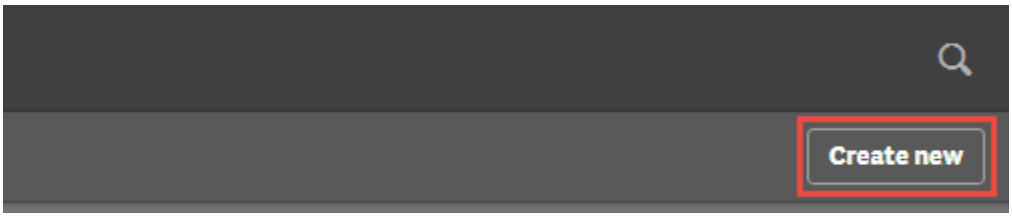
LAYOUT

CREATE A NEW EXTENSION

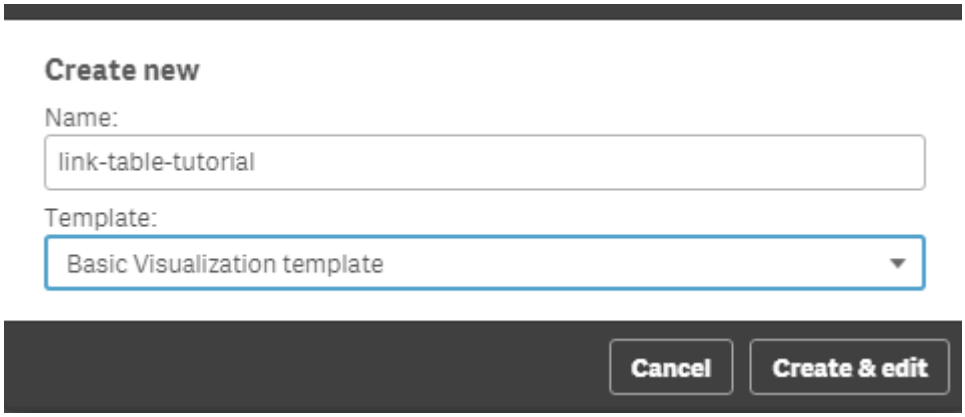
- Navigate to the Hub (Desktop or Server)
- Click on the drop down in the top left, and select “Qlik Dev Hub”



- In the top right, select “Create new”



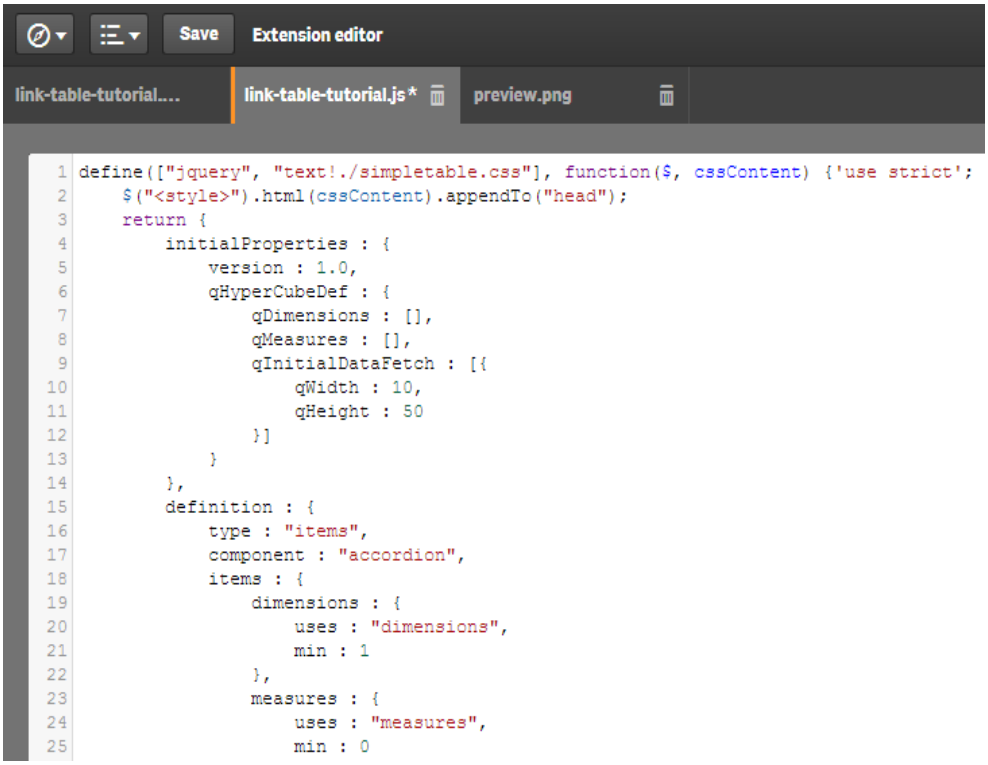
- Give it the name “hyperlink-table-example”



SOURCE SOME "SIMPLE TABLE" EXTENSION CODE

- In a new tab/window, navigate to https://help.qlik.com/sense/2.1/en-us/developer/#../Subsystems/dev-hub/Content/Examples/simple-table-extension-example.htm
- This page gives example JavaScript, CSS, and .qext code to create the Simple Table extension.
- Copy the code beneath "com-qliktech-simpletable.js"

- Then, go back to the dev-hub, select the JavaScript code tab and paste the contents overwriting the old



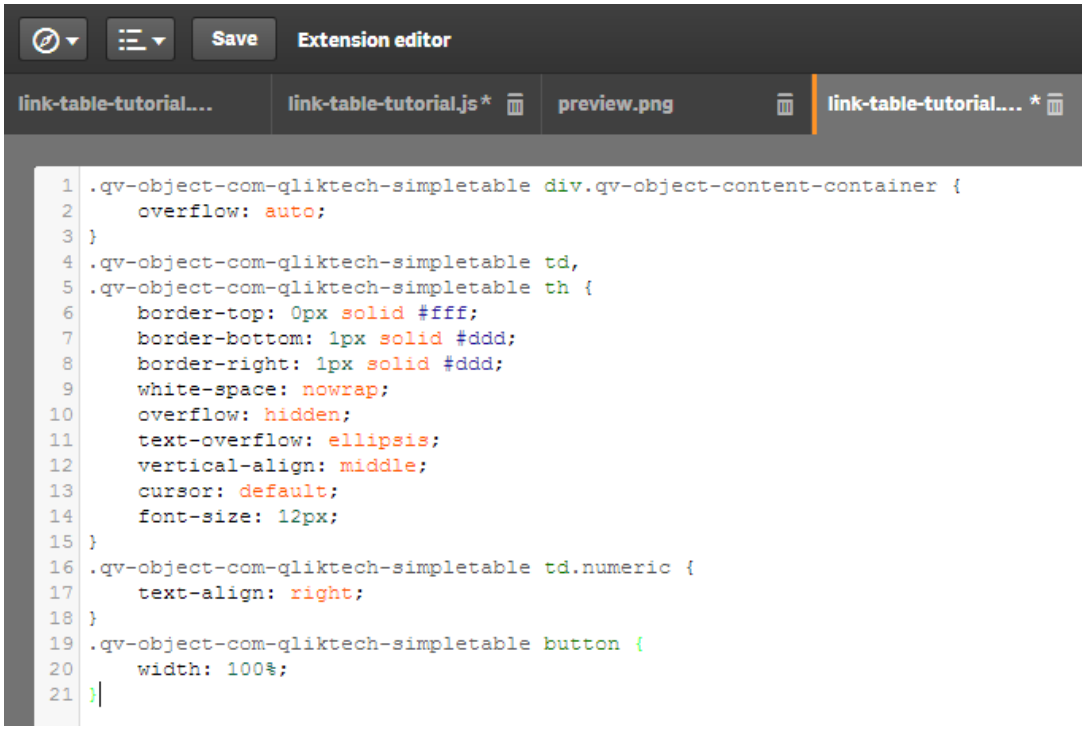
- In the dev-hub, click the "+" button on the top right and create a new file called "link-table-tutorial.css"

- Go back to the help site and copy/paste the code under "simpletable.css" to your new tab in the dev-hub "link-table-tutorial.css"

Full code example

+ com-qliktech-simpletable.qext
+ com-qliktech-simpletable.js
− simpletable.css

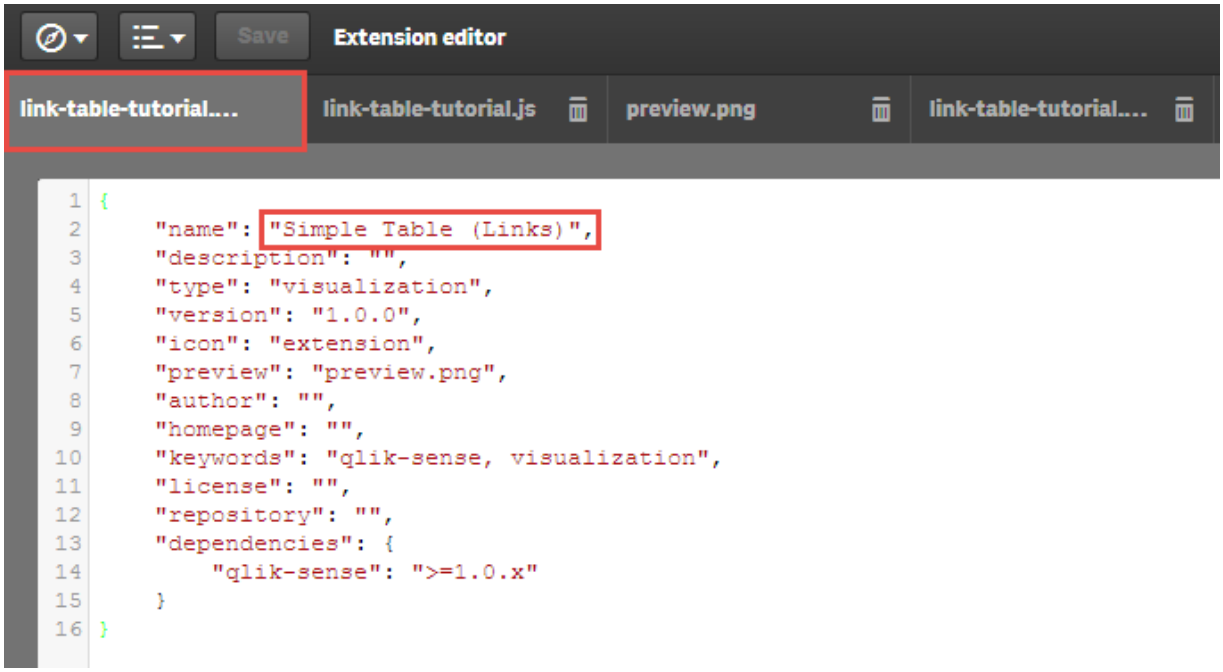
```css
.qv-object-com-qliktech-simpletable div.qv-object-content-container {
    overflow: auto;
}
.qv-object-com-qliktech-simpletable td,
.qv-object-com-qliktech-simpletable th {
    border-top: 0px solid #fff;
    border-bottom: 1px solid #ddd;
    border-right: 1px solid #ddd;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    vertical-align: middle;
    cursor: default;
    font-size: 12px;
}
.qv-object-com-qliktech-simpletable td.numeric {
    text-align: right;
}
.qv-object-com-qliktech-simpletable button {
```



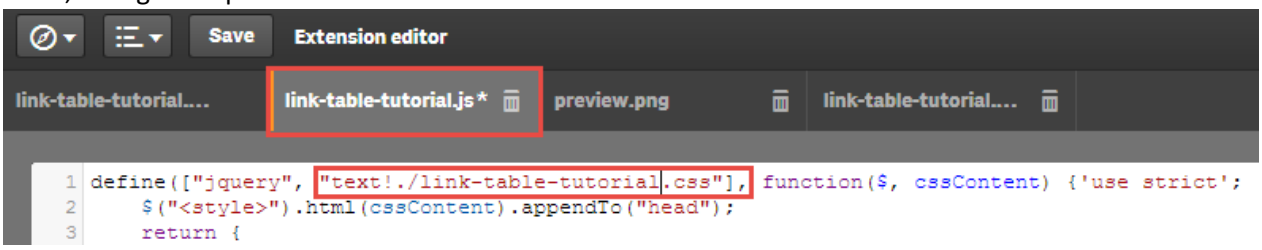
- Click “Save” in the top left

EDIT THE CODE TO LINK THE SHEETS

- In the dev-hub, click on the top left tab (the .qext) change the "name" to "Simple Table (Links)"

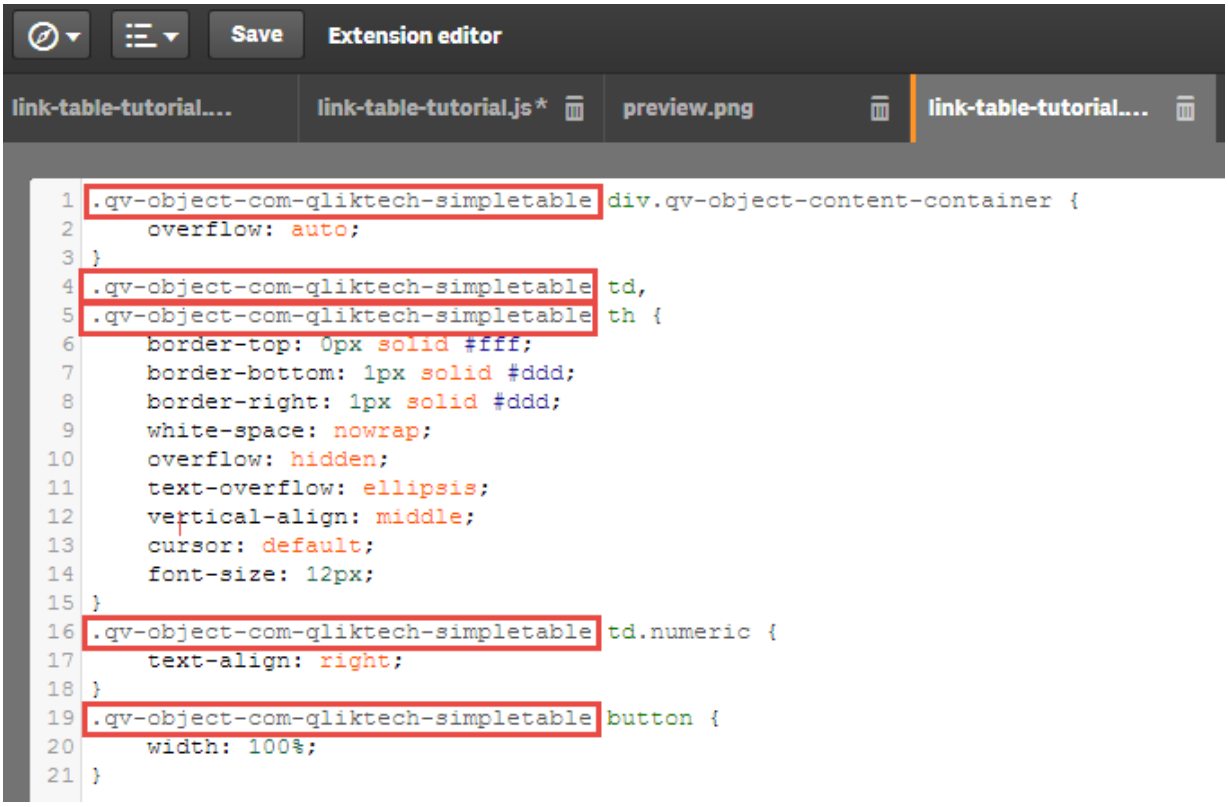


- Click on the JavaScript tab
- On line 1, change "simpletable.css" to "link-table-tutorial.css"



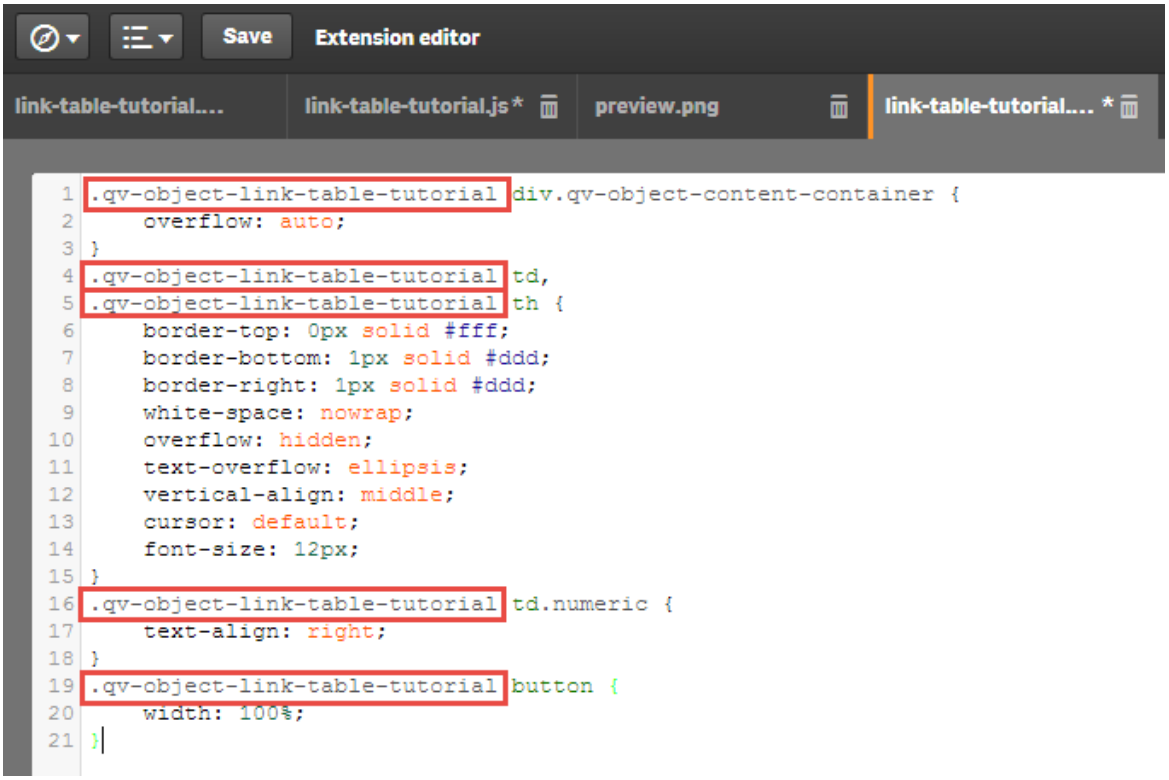
- Click on the css tab
- Change all instances of ".qv-object-com-qliktech-simpletable" to ".qv-object-link-table-tutorial" (on lines 1,4,5,16,19)

**Original



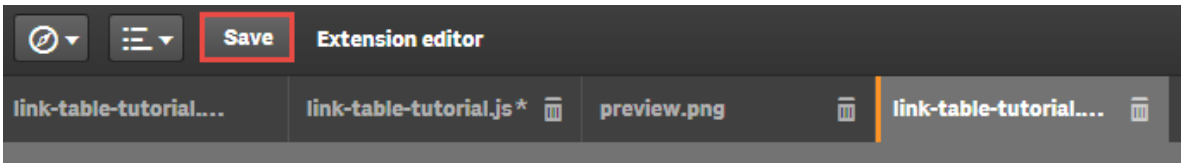

```
.qv-object-com-qliktech-simpletable div.qv-object-content-container {
    overflow: auto;
}
.qv-object-com-qliktech-simpletable td,
.qv-object-com-qliktech-simpletable th {
    border-top: 0px solid #fff;
    border-bottom: 1px solid #ddd;
    border-right: 1px solid #ddd;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    vertical-align: middle;
    cursor: default;
    font-size: 12px;
}
.qv-object-com-qliktech-simpletable td.numeric {
    text-align: right;
}
.qv-object-com-qliktech-simpletable button {
    width: 100%;
}
```

**Revised



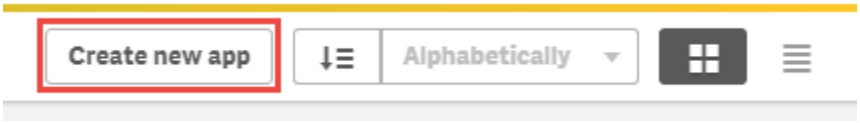

```
.qv-object-link-table-tutorial div.qv-object-content-container {
    overflow: auto;
}
.qv-object-link-table-tutorial td,
.qv-object-link-table-tutorial th {
    border-top: 0px solid #fff;
    border-bottom: 1px solid #ddd;
    border-right: 1px solid #ddd;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    vertical-align: middle;
    cursor: default;
    font-size: 12px;
}
.qv-object-link-table-tutorial td.numeric {
    text-align: right;
}
.qv-object-link-table-tutorial button {
    width: 100%;
}
```

- All of your sheets are now linked properly.
- Click “Save”



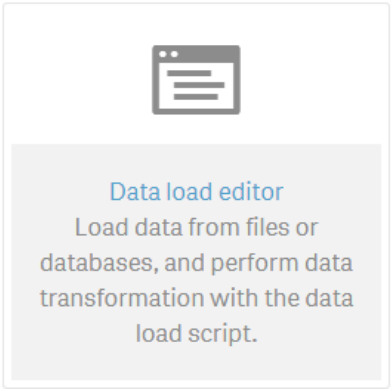
TEST THE EXTENSION SO FAR

- Navigate to the hub and create a new app



- Name it “SimpleTableTest”

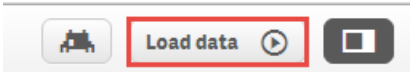
- Click on the “Data load editor”



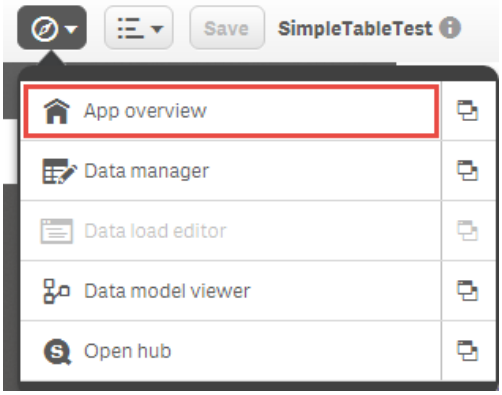
- Paste in the following code:

```
[Test Data]:
LOAD * INLINE [
        Site, URL
        Qlik, http://www.qlik.com
        Google, www.google.com
        MSDN, https://msdn.microsoft.com
        Beer Advocate, www.beeradvocate.com
];
```

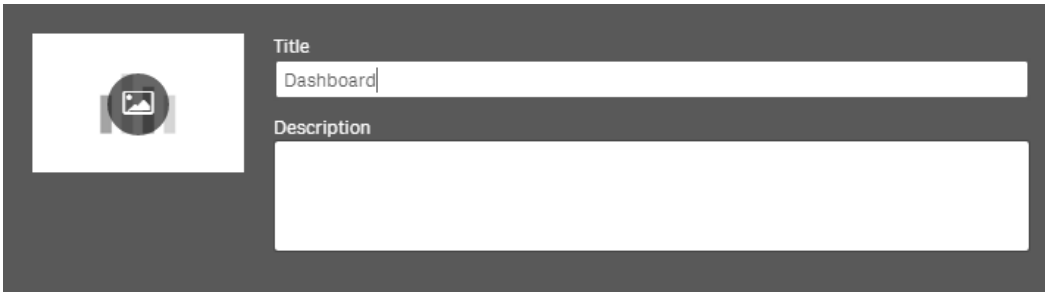
- Load the data



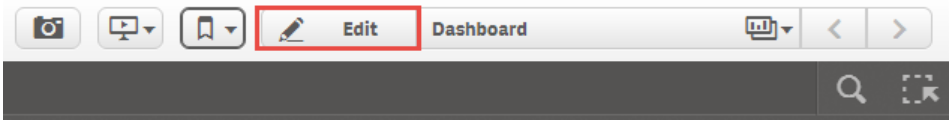
- Navigate to the “App Overview”



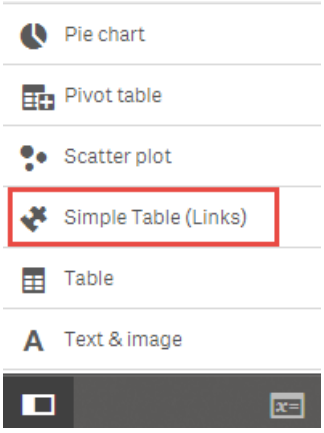
- Click “Create new sheet” and name it whatever you’d like



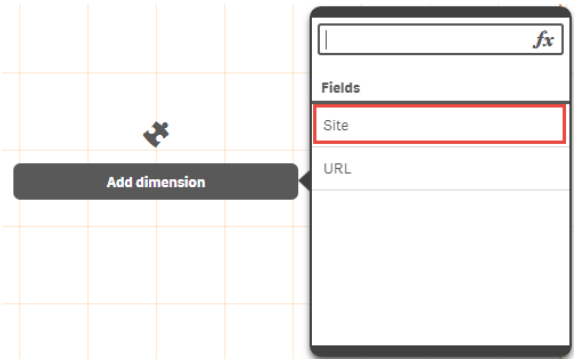
- Click on the sheet, then click “Edit” on the top right



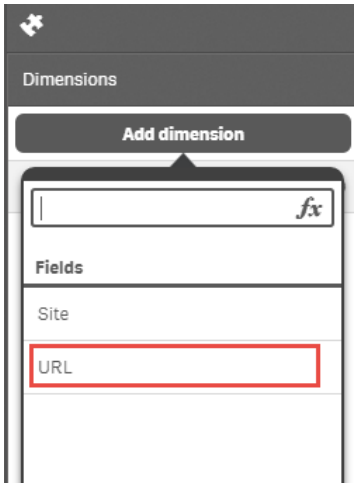
- Select the “Simple Table (Links)” Extensions from the Charts list and drag it onto the sheet



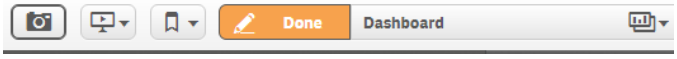
- Add the dimension “Site”



- In the upper right, click on “Dimensions”
- Click “Add dimension” and select “URL”



- Select “Done”



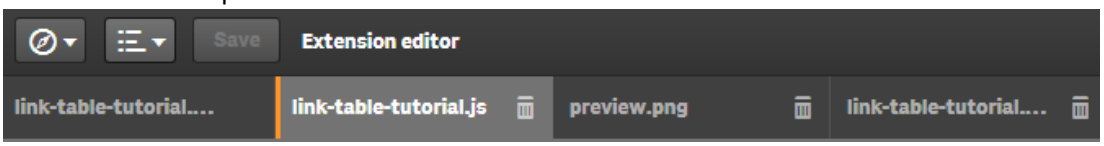
- You'll see that the table is populated but the links have not been detected yet.

Dashboard

Site	URL
Beer Advocate	www.beeradvocate.com
Google	www.google.com
MSDN	https://msdn.microsoft.com
Qlik	http://www.qlik.com

CREATE A SWITCH

- Navigate back to the dev-hub
- Select the JavaScript tab



- Insert a new line after line 38

```
30                  settings : {
31                      uses : "settings",
32                      items : {
33                          initFetchRows : {
34                              ref : "qHyperCubeDef.qInitialDataFetch.0.qHeight",
35                              label : "Initial fetch rows",
36                              type : "number",
37                              defaultValue : 50
38                          },
39
40                      }
41                  }
42              }
43          },
```

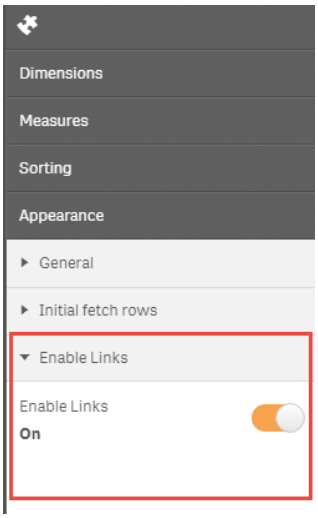
- Enter the following code on the new blank line (39). This will insert the code for the Boolean Switch:

```
linkColumns:{
        type: "boolean",
        component: "switch",
        translation: "Enable Links",
        ref: "linkColumns",
        defaultValue: true,
        trueOption: {
          value: true,
          translation: "properties.on"
        },
        falseOption: {
          value: false,
          translation: "properties.off"
        },
        show: true
},
```

- Confirm that your code looks like the following:

```
30                  settings : {
31                      uses : "settings",
32                      items : {
33                          initFetchRows : {
34                              ref : "qHyperCubeDef.qInitialDataFetch.0.qHeight",
35                              label : "Initial fetch rows",
36                              type : "number",
37                              defaultValue : 50
38                          },
39                          linkColumns:{
40                              type: "boolean",
41                              component: "switch",
42                              translation: "Enable Links",
43                              ref: "linkColumns",
44                              defaultValue: true,
45                              trueOption: {
46                                value: true,
47                                translation: "properties.on"
48                              },
49                              falseOption: {
50                                value: false,
51                                translation: "properties.off"
52                              },
53                              show: true
54                          },
```

- Click “Save”
- You’ve now successfully added a switch to your extension. The switch is not linked to anything yet, so will not have any effect on anything.
- You can choose navigate back to the hub and view the application. If you do, refresh the page, click on the table, and you will see a new section has appeared on the right-hand options called “Enable Links.”



ADD CONDITIONAL LOGIC

- Go back to the dev-hub and go to the JavaScript tab
- On line 62, immediately after "$element" add ", layout"

```
62          paint : function($element, layout) {
63              var html = "<table><thead><tr>", self = this, lastrow
64              //render titles
65              $.each(this.backendApi.getDimensionInfos(), function(
66                  html += '<th>' + value.qFallbackTitle + '</th>';
67              });
```

- Then, REPLACE line 90 with the following code:

```
if(layout.linkColumns){
        if(cell.qText.slice(0,4)==='http'){
                html += '> <a href="' + cell.qText + '" target="_blank">' + cell.qText + '</a></td>';
        }
        else if(cell.qText.slice(0,3)==='www'){
                html += '> <a href="http://' + cell.qText + '" target="_blank">' + cell.qText + '</a></td>';
        }
        else{
                html += '>' + cell.qText + '</td>';
        }
}
else{
        html += '>' + cell.qText + '</td>';
}
```

***This code is identifying whether the field value begins with 'http' or 'www' and if so, wraps it in the appropriate HTML. This could of course be customized/optimized much further.*

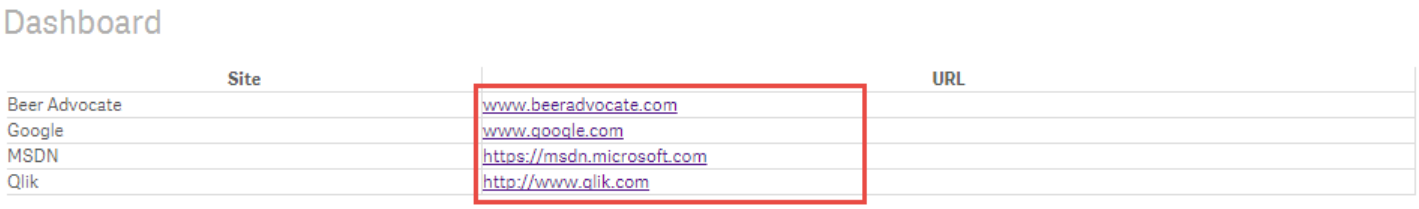
- Confirm that the code looks like the following:

```
                if(layout.linkColumns){
                      if(cell.qText.slice(0,4)==='http'){
                        html += '> <a href="' + cell.qText + '" target="_blank">' + cell.qText + '</a></td>';
                      }
                      else if(cell.qText.slice(0,3)==='www'){
                        html += '> <a href="http://' + cell.qText + '" target="_blank">' + cell.qText + '</a></td>';
                      }
                      else{
                        html += '>' + cell.qText + '</td>';
                      }
                    }
                    else{
                        html += '>' + cell.qText + '</td>';
                    }
                });
                html += '</tr>';
            });
            html += "</tbody></table>";
            //add 'more...' button
```

- Click “Save”

EXTENSION IS FINISHED – GO TEST

- Navigate back to your App and refresh the page
- You will see that there are now links in the table for the values that are detected by the JavaScript conditional logic
    - If you don't see the links enabled, try toggling the "Enable Links" switch, or recreate the table if necessary



- Toggle the "Enable Links" switch to turn the hyperlink detection on or off