Qlik Sense

Mashup & Extensions Overview

Ajay Kakkar

Contents

1) What Is Visualization API 2
What is needed to make our mashup work 2
2) What files do we have in basic qlik mashup 2
3) What does CSS file include 2
4) What does HTML file include 2
5) What does JS file include 2
6) What does QEXT file include 2
7) What does WBL file include 3
8) Extension and Mashup 4
9) Mashup Connection with Qlik Sense Data 4
What to know before making a Qlik Sense Mashup 5
APIs used for building mashups. 5
10) CSS frameworks 6
Updating CSS links in existing mashups 6
11) How to make a basic Mashup in qlik sense 7
Overview of mashup editor 8
12) How to make a basic Extension in qlik sense 9
Definition of property in extension 10
Sample Log Output 11
APIs used for building extension 12
More links to know about extensions 12
13) Authentication solutions 12
14) Mashup License & Access 12
What Does On Fly Means 13
15) Tips & Tricks 13
How to identify which extension are you using in qlik sense dashboard using chrome browser 13
Improve Presentation of your Code 13
16) Important Methods or Frequently used Methods 14

What Is Visualization API

The visualization interface is the external interface to Qlik Sense visualizations. It allows developers to get visualizations defined in an app and to create **temporary visualizations** on the fly.
visualization api consists of MASHUP, Extension, Widget.

What is needed to make our mashup work

1. Qlik sense publish application
2. A Final **Base Template** or **UI** Design for your mashup
3. Basic HTML & CSS knowledge
4. Bootstrap / Custom CSS knowledge

What files do we have in basic qlik mashup

5. CSS file
6. HTML file
7. JS file
8. QEXT file
9. WBL file

What does CSS file include

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language, i.e it has style defined for every tag in HTML file based on our UI or Template Design.

What does HTML file include

Hypertext Markup Language (**HTML**) is the standard markup language for creating web pages and web applications with Cascading Style Sheets (CSS) and JavaScript. It has all the basic tags based on our UI or Template Design.

What does JS file include

JavaScript file which contains the **JavaScript** code used to validate the **form** & used to define some function in a specific structure to use with our qlik mashup, we can add our own **JavaScript** code in that same JS file.

What does QEXT file include

Meta data about the mashup/extension like

10. Type : mashup/visualization

11. Name : name of mashup/visualization(can be changed)
12. Version : version number(can be changed)
13. Description : of mashup/visualization(can be changed)
14. author : author of mashup/visualization
15. keywords : qlik-sense, mashup/visualization
16. license : -
17. repository : git repository link
18. dependencies : qlik-sense": ">=2.1.x"

and many more.

What does WBL file include

This file has all the files included in mashup/extension.

Example :

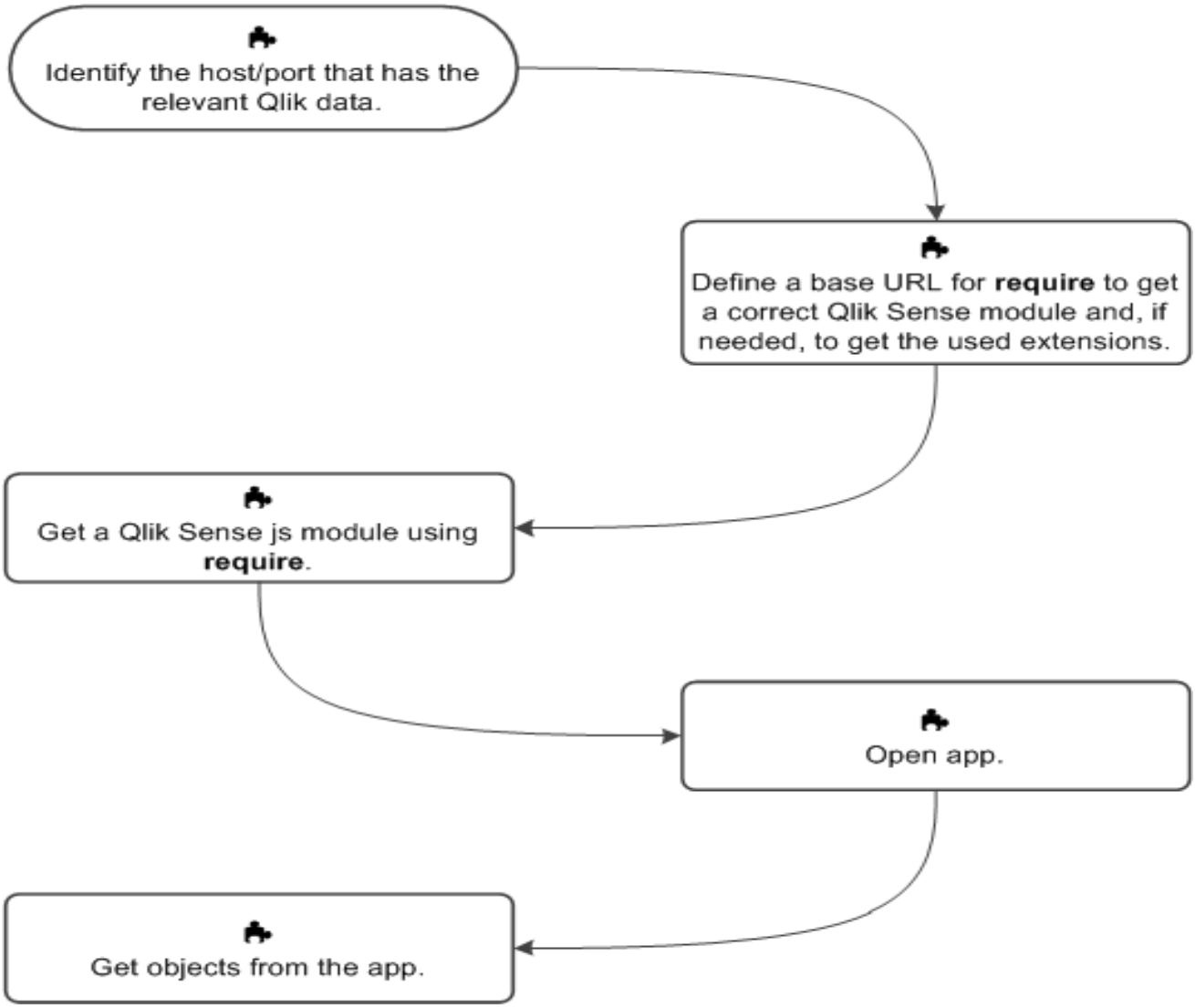
```
test.js;
test.qext;
test.html;
test.css
```

Note : WBL & QEXT file is important in Qlik Sense Mashup/Extension.

Extension and Mashup

EXTENSION	MASHUP
It is used for adding more functionalities/module's in to qlik sense, Can be used in mashup pages as object	Uses all abject from app's sheet into the mashup template
It consists of wbl, qext, js file (if angular js is used it will have template.html file in it)	It consists of wbl, qext, js, css & html file
Can make your application slow depending on complex code written in js file	Doesn't get effected by any of the scripts as its web based it takes a few seconds to render/show charts from qlik sense app

Mashup Connection with Qlik Sense Data



What to know before making a Qlik Sense Mashup

19. All Base file such as **PORT**, **URL**, **REQUIRE.js** are there by default in **head** tag when you start making a mashup
20. **Open App** function is nothing but APP ID (Only in server version) / App Name (Only in desktop version with full path of QVF file)
21. The **Qlik Sense object** is typically a visualization from an app,
22. **Visualization Objects** can be charts/tables/map/extensions etc
23. **Single configurator** is a Qlik Sense tool that provides an easy way of creating simple mashup pages without writing any code. It returns a **Qlik Sense object**, which is identified in an URL.
24. The **Single configurator** creates a **URL** that returns a **complete HTML page** that contains all embedded visualization. This URL can be embedded in a web page by including it in an iframe.

Note : **Single configurator** technique is not that customizable.

Example :

```
<iframe src="https://sense-demo.qlik.com/single/?appid=133dab5d-8f56-4d40-b3e0a6b401391bde&obj=hRZaKk&select=Year,2012" frameborder="0"></iframe>
```

Click here to open example by QLIK

APIs used for building mashups.

API Name	Example	API Type
Engine API	This can give you all sheet name, app name, objects in sheet etc	JSON RPC (**Remote Procedure Call** protocol encoded in JSON)
Backend API	All backend activities	JavaScript library
Root API		JavaScript library
App API	App level operations	JavaScript library
Bookmark API	Bookmark handling create, delete, activate	JavaScript library
Field API	All field activities	JavaScript library
Global API		JavaScript library
Table API	Can make custom Table	JavaScript library
Selection API	Can make selection bar , can use functions to make visualizations	JavaScript library
Variable API	Create, read variables	JavaScript library
Visualization API	Create own visualization	JavaScript library
App Integration API	Integrates app in mashup or extensions	URL integration

Single Integration API	Single configurator	URL integration
qlik-visual	One way to make Visualization on fly	Web component

CSS frameworks

The following CSS frameworks work's with Qlik Sense Mashups, which means there should be no clashes between the Qlik Sense CSS styles and the styles included in the common CSS framework:

- Bootstrap (http://getbootstrap.com/)
- Foundation (http://foundation.zurb.com/)
- jQuery UI (https://jqueryui.com/)
- Material Design (https://design.google.com/)
- Semantic UI (http://semantic-ui.com/)
- Pure CSS (http://purecss.io/)

Add the following reference to your mashups:

```
<link rel="stylesheet" href="../../resources/autogenerated/qlik-styles.css">
```

Updating CSS links in existing mashups

If your mashups were built using Qlik Sense 2.1 or earlier, you should replace the following references:

```
<link rel="stylesheet" href="../../resources/autogenerated/qlikui.css">
<link rel="stylesheet" href="../../resources/assets/client/client.css" media="all">
```

with the following:

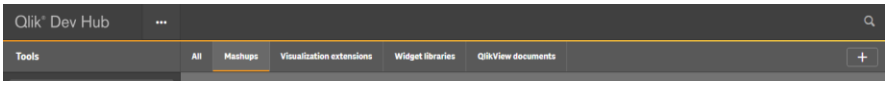
```
<link rel="stylesheet" href="../../resources/autogenerated/qlik-styles.css">
```

Note :

- These API/CSS frameworks are are used while building your mashup, can be used for more customizations in your page.
- Almost every thing comes out of the box while you start building a fresh mashup.
- We have example of Bootstrap css to start with.

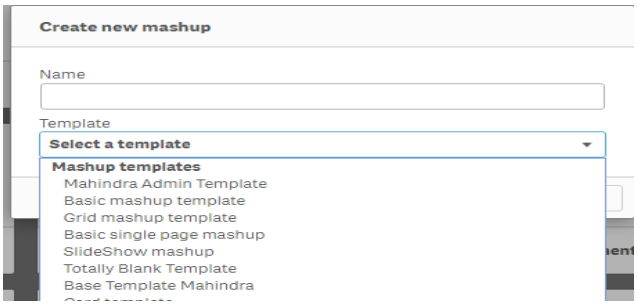
How to make a basic Mashup in qlik sense

25. Qlik sense publish application
26. A Final **Base Template** or **UI** Design for your mashup
27. Basic HTML & CSS knowledge
28. Bootstrap / Custom CSS knowledge
29. Go to http://localhost:4848/dev-hub/
30. Then navigate to **create new button/plus sign** on right hand Conner

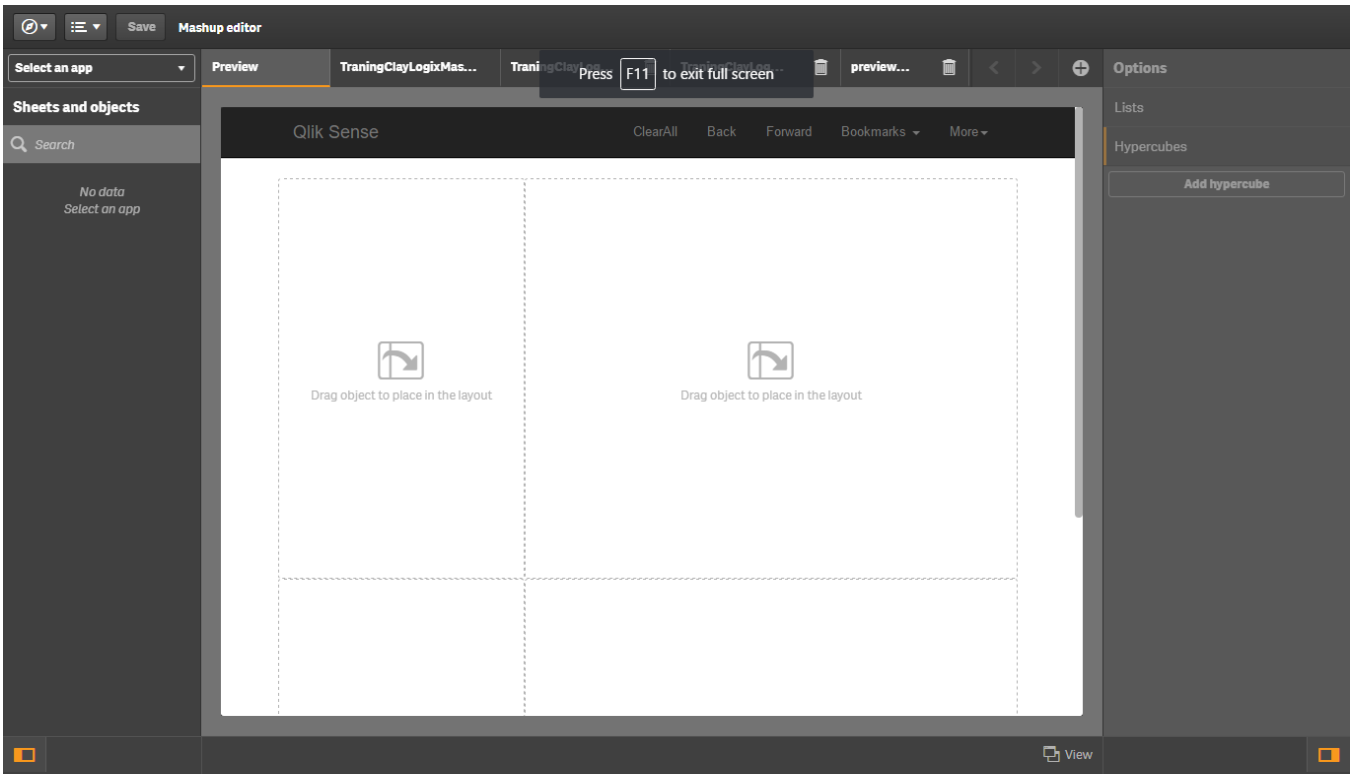


31. Either way is to click on create new mashup button

32. Type the preferred name of the mashup then select the type from option provided



33. I will choose Grid Mashup and name it as TraningMashup



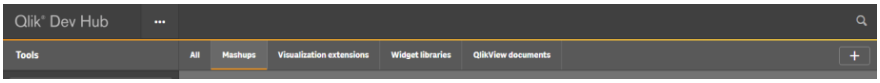
Overview of mashup editor

34. Select the publish / developed app
35. You will see all the sheets and object inside the left pane
36. Search can be used to find the specific object (title is used to search i.e adding title is recommended so it is easier to identify the object)
37. All graph/table/extension or any type of visualization can be added in to mashup by simply dragging and dropping it into **qvplaceholder**
38. **qvplaceholder** becomes **qvobject** when a **object** is added in to it
39. qvplaceholder & qvobject is a important part of mashup which lets us identify where we want to drag and drop our object
40. **qvplaceholder** is where we drag our object
41. **qvobject** is where our object is finally placed
42. In html file there are division tags with some attributes
    - Class (**qvplaceholder**/**qvobject** can add other custom classes if you want)
    - Id (this is important for adding the object it works is unique identity, two same id in single mashup cannot be used, if used one of them will be blank)
    - Style (optional)
43. This above process automatically fills the **JavaScript** file for us
44. It can be done manually but it is not recommended for learner's
45. You can use a lot of IDE's like **Notepad++, NetBeans, Eclipse** etc, to **edit** all these files as this editing screen is not yet developed or is not user friendly for developing of coding

46. Go to Improve Presentation go to **Page 10** (**Tips & Tricks**)

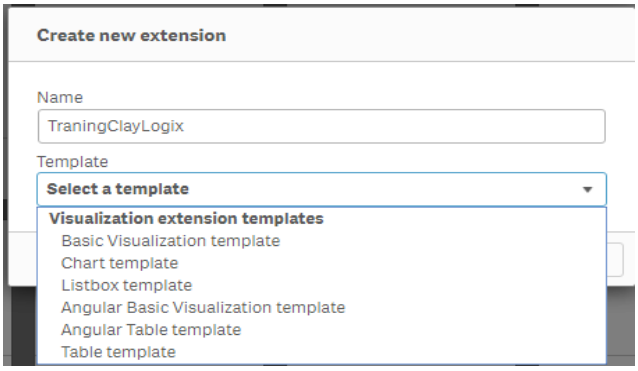
How to make a basic Extension in qlik sense

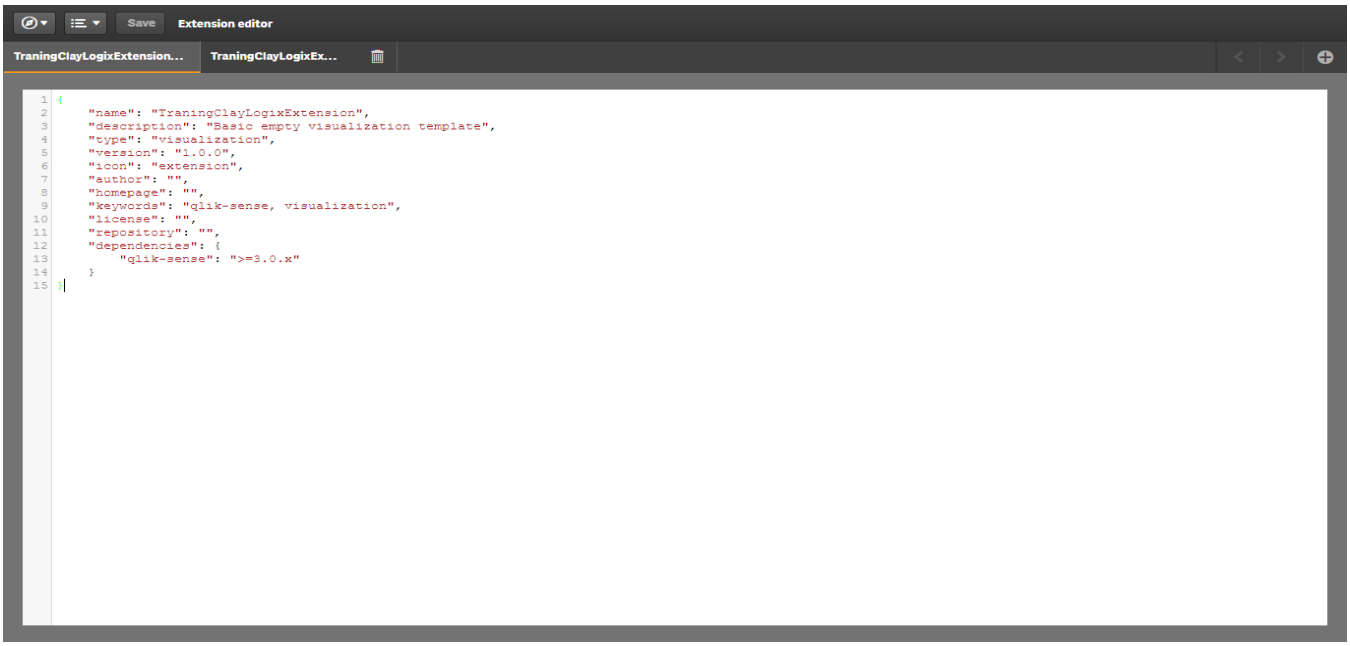
47. Qlik sense publish application
48. A Final **Base Template** or **UI** Design for your mashup
49. Basic HTML & CSS knowledge
50. Bootstrap / Custom CSS knowledge
51. Go to http://localhost:4848/dev-hub/
52. Then navigate to **create new button/plus sign** on right hand Conner



53. Either way is to click on create new extension button

54. Type the preferred name of the extension then select the type from option provided
55. I will choose basic visualization template and name it as TraningExtension





Definition of property in extension

- **name** = TraningExtension,
- **description** = Basic empty visualization template,
- **type** = visualization,
- **version** = 1.0.0,
- **icon** = extension,
- **preview** = Logo.png
- **author** = Ajay.R.Kakkar,
- **keywords** = qlik-sense, visualization,
- license = 1.0,
- **repository** = repository link here,
- dependencies : { qlik-sense=>=3.0.x }

Note :

qlik-sense=>=3.0.x means current version is equal to or higher than 3.0

custom preview should be 140 px by 140 px in size.

Sample Log Output

extensionMeta:Object
author:""
dependencies:Object
description:"Basicemptyvisualizationtemplate"
homepage:""
icon:"extension"
isLibraryItem:true
isThirdParty:true
keywords:"qlik-sense,visualization"
license:""
name:" TraningExtension"
repository:""
template:" TraningExtension"
templateIconClassName:"icon-extension"
type:"visualization"
version:"1.0.0"
__proto__:Object
footnote:""
permissions:(...)
qInfo:Object
qId:"CzXPEw"
qType:" TraningExtension "
__proto__:Object
qSelectionInfo:Object
showDetails:false
showTitles:false
subtitle:""
title:""
version:"1.0.0"
visualization:" TraningExtension"
getpermissions:()
__proto__:Object

Note : To see more details add extension to **Documents\Qlik\Sense\Extensions** and open in dev-hub

APIs used for building extension

API Name	API Type
Engine API	JSON RPC (**Remote Procedure Call** protocol encoded in JSON)
Backend API	JavaScript library
Root API	JavaScript library
App API	JavaScript library
Bookmark API	JavaScript library
Field API	JavaScript library
Global API	JavaScript library
Table API	JavaScript library
Navigation API	JavaScript library
Selection API	JavaScript library
Variable API	JavaScript library

More links to know about extensions

- Getting Started
- Extensions debugging & web developer tools
- Custom components introduction
- Examples

Authentication solutions

1. Ticket solution
2. Session solution
3. Header solution
4. SAML
5. Anonymous users

Mashup License & Access

1. There is no License if you are developing it on the same host or domain
2. License apply when you go out of host or domain (for public use or for world to access it), if 1000 people are hitting or using your Qlik server and using all the memory your qlik mashup will tends to get slow

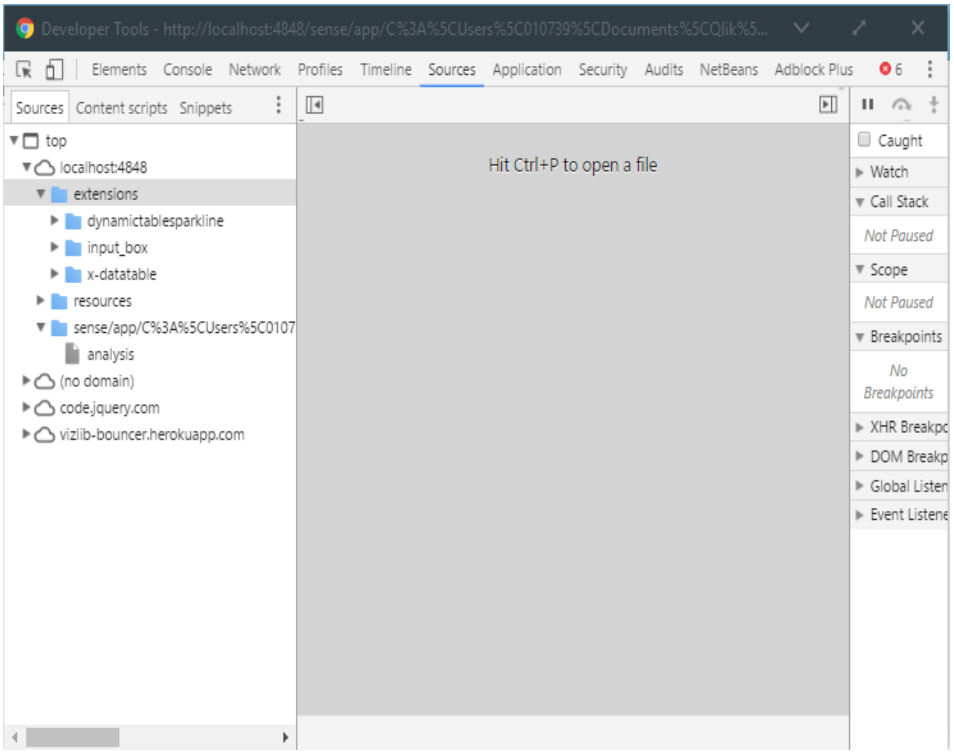
What Does On Fly Means

It means we can develop maps or charts directly in mashup pages which is rendered **temporarily** and destroyed when page or section is closed. This method takes time to build or code our charts / `visualization` compared to qlik sense hub.

Tips & Tricks

How to identify which extension are you using in qlik sense dashboard using chrome browser

1. open sheet in which you want to identify the extension
2. once the sheet is loaded press **F12** and you will get a tab where there will be lot of option tabs
3. select **SOURCES** and you will get all files in the url loaded as tree/explorer view
4. there will be a folder named extension in that you will get all **extraction names and the files rendered in folder format**



Improve Presentation of your Code

5. To beautify your JavaScript file

   http://jsbeautifier.org/
6. To beautify your CSS file

http://www.cleancss.com/css-beautify/

7. To beautify your HTML file
   http://www.cleancss.com/html-beautify/

Important Methods or Frequently used Methods

1. **qlik.currApp() :** Reference to the current app.
2. **qlik.getAppList() :** Gets a list of Qlik Sense apps which you have access too.
3. **qlik.getExtensionList()** : Get a list of Extension in your qlik sense server.
4. **qlik.resize() :** Refresh all the objects or particular object by adding object ID as parameter.
5. **qlik.app.back() :** Select last or previous selection.
6. **app.clearAll()** : Clear all selection.
7. **app.forward() :** Select step forward selection.
8. **app.field('[Field Name]') :** Gets a field reference and add use Field API Methods.
9. **app.field('[Field Name]').clear()** : Gets a field reference and clear the field selection.
10. **app.lockAll()** : Lock all selections in the app.
11. **app.unlockAll()** : Unlock all selections in the app.
12. **app.field('[Field Name]').clearOther(true)** : Clear other fields add true parameter if locked fields is to be cleared
13. **app.field('[Field Name]').select() :** Select field values, parameters select(Array:[select value index number], toggle:[true/false], softlock:[true/false])
14. **qlik.app.field('[Field Name]').selectAll()** : Select all values in the field, for softlock add parameter true.
15. **qlik.app.field('[Field Name]').selectAlternative()** : Selects alternative values in a field, for softlock add parameter true.
16. **qlik.app.field('[Field Name]').selectExcluded() :** Selects excluded values in a field, for softlock add parameter true.
17. **qlik.app.field('[Field Name]').selectPossible()** : Selects possible values in a field, for softlock add parameter true.
18. **qlik.app.field('[Field Name]').toggleSelect(Value, softlock:[true/false]) :** Toggle field selection, add Value as string.
19. **qlik.app.field('[Field Name]').selectMatch(Value, softlock:[true/false]) :** Select Matching Values form the field.
20. **qlik.navigation.getCurrentSheetId() :** Get Current Sheet ID.
21. **qlik.navigation.getMode()** : Get sheet mode Edit/Analysis.
22. **qlik.navigation.setMode()** : Set sheet mode to Edit [qlik.navigation.EDIT] /Analysis [qlik.navigation.ANALYSIS].
23. **qlik.navigation.gotoSheet(Value):** Navigate to sheet replace Value with SheetID.
24. **qlik.navigation.gotoStory(Value):** Navigate to story replace Value with StoryID.

25. **qlik.navigation.isModeAllowed(mode:[edit/analysis])** : Checks if a given mode is allowed.
26. **qlik.navigation.nextSheet()** : Go to next sheet.
27. **qlik.navigation.prevSheet()** : go to previous sheet.
28. **app.variable.getByName('var_name')** : Get Variable metadata.
29. **qlik.app.variable.getContent('var_name', callbackfunction)** : Gets variable content.
30. **qlik.app.variable.setContent('var_name', 'value')** : Set variable content.
31. **qlik.app.variable.setNumValue('var_name', 100)** : Set variable Number value.
32. **qlik.app.variable.setStringValue('var_name', 'value')** : Set variable string value.
33. **exportData()** : this method helps in exporting data default export is in excel.