**Web/mashup development with Qlik sense desktop**

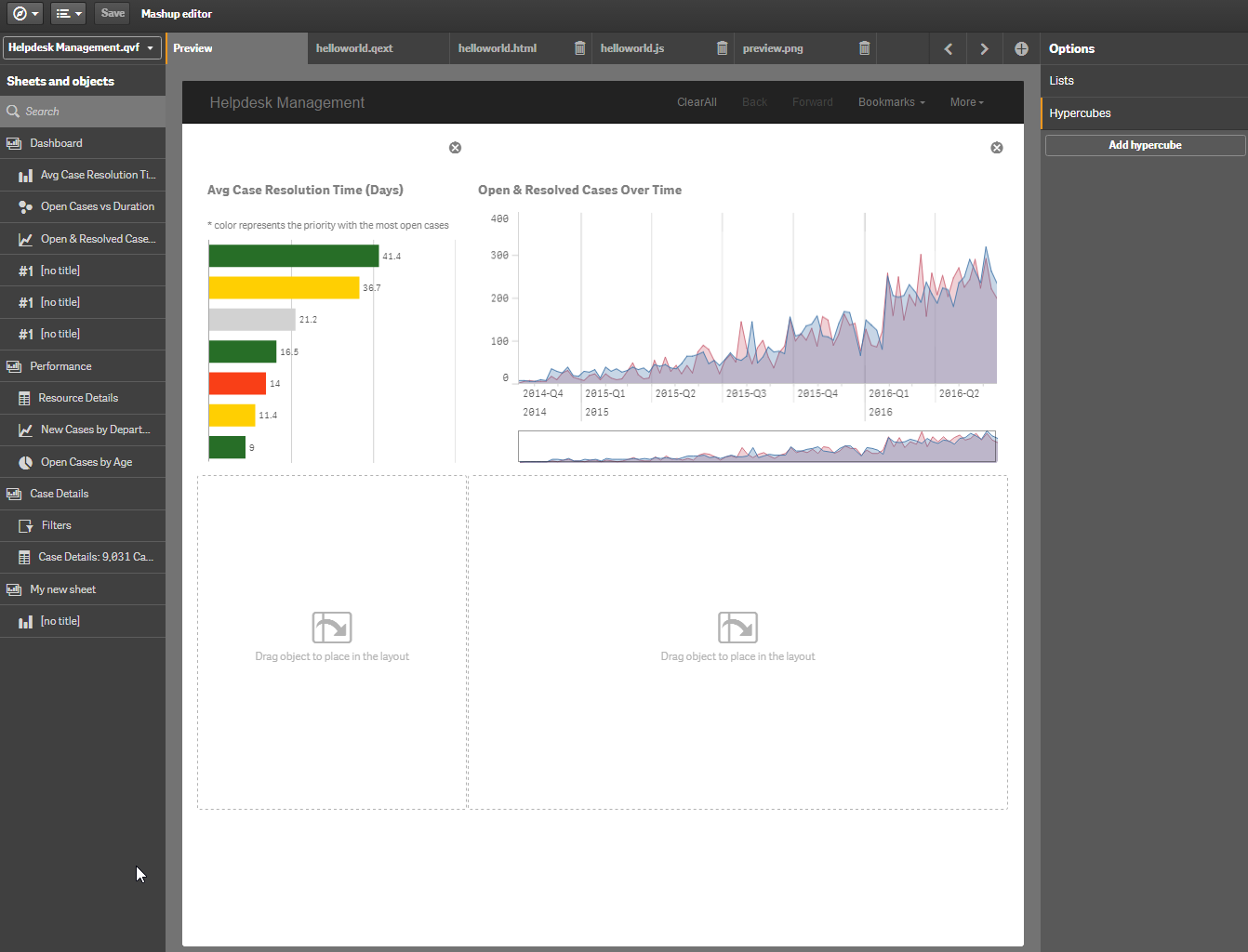
How to turn Qlik sense desktop into data and visualization server

# Overview

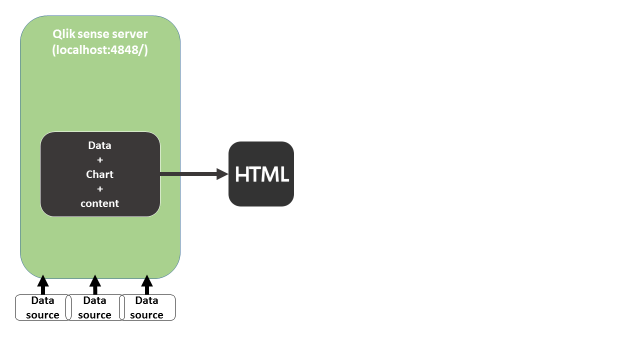
Using mashup editor is definitely the easiest and most intuitive way to create a web page with qlik visualization. Mashup editor comes with Qlik sense. It is extremely easy to use even people with zero web development knowledge can build a decent web application in less than 1 minute.

More information on mashup editor:

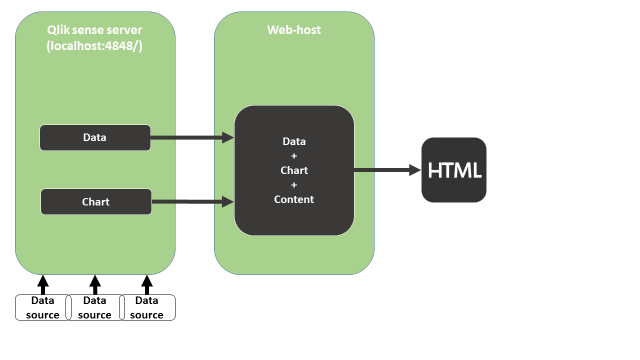
https://help.qlik.com/en-US/sense-developer/3.0/Subsystems/Dev-Hub/Content/MashupEditor/mashup-editor-overview.htm



Mashup editor is a great tool for beginner, powerful, no coding involved, no learning curve. It is perfect for small-medium scale implementation with simple structure. However advance users will probably discover limitations along the way using mashup editor. The web application is hosted on sense server so it is not possible to be accessed through common ports like port 80; needless to say you will not be able to develop on your favourite backend suites, QMC (Qlik management console) is the only server management tool you have and it is not available on Qlik sense desktop version; And most importantly, system integration is unnecessarily complicated in this case.



A much better alternative is using Qlik sense as standalone server for data and charts, another host for content delivery. Advantage of this approach is obvious: Scalability; Modularization; Clean system integration. But there are drawbacks on this approach too, licensing and user control may be one of the road blocks. In this article we use qlik sense desktop version. It is free and doesn’t support authentication so licensing is not an issue. For commercial implementation, QAP is definitely the right product to consider.



# Talking to Qlik sense server

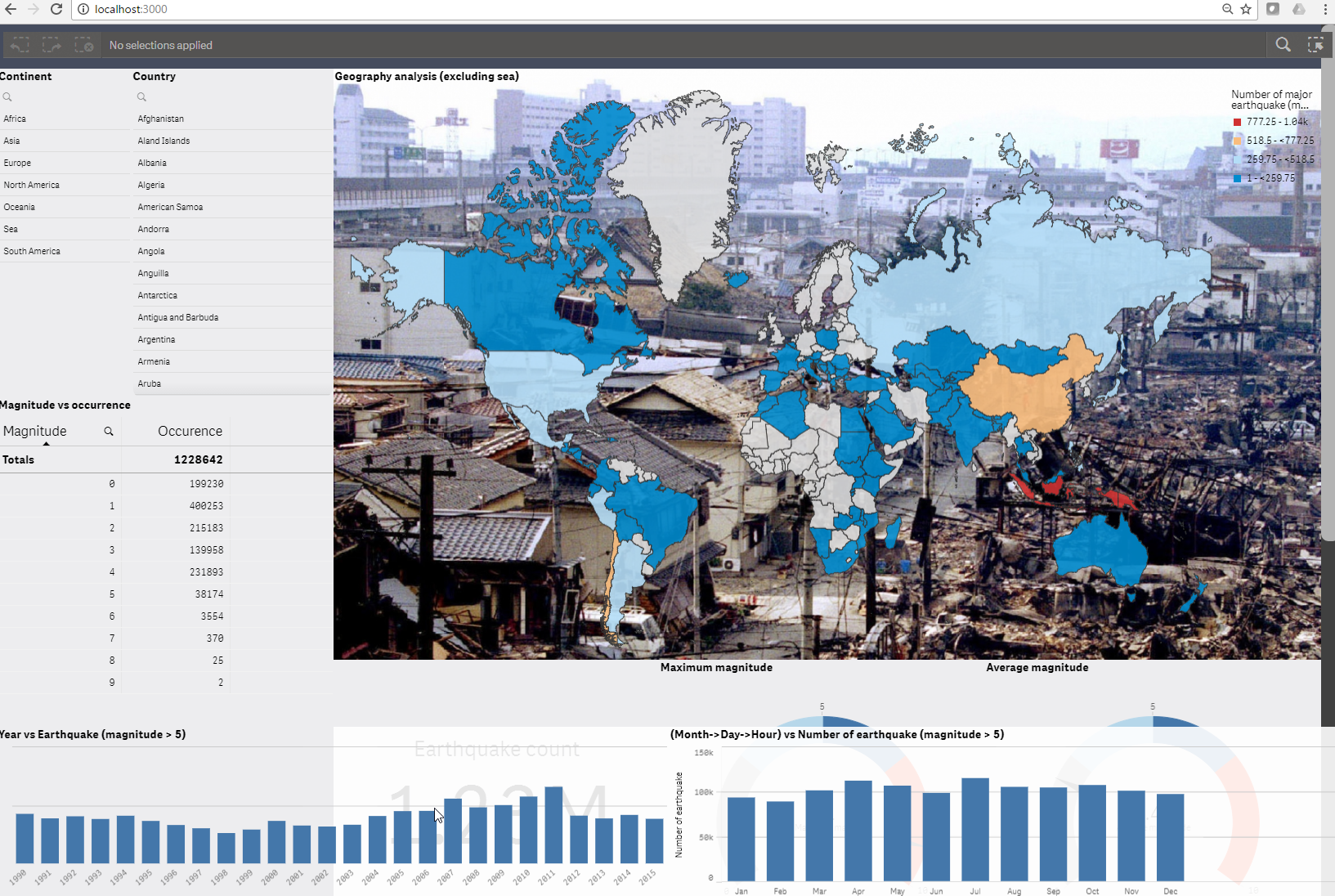
Below are all the APIs available in Qlik sense 3.1

|  |  |
| --- | --- |
| [Root API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/MashupAPI/qlik-interface-interface.htm) | JavaScript library (Capability API) |
| [App API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/MashupAPI/qlik-app-interface.htm) | JavaScript library (Capability API) |
| [Bookmark API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/MashupAPI/qlik-bookmark-interface.htm) | JavaScript library (Capability API) |
| [Field API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/MashupAPI/qlik-field-interface.htm) | JavaScript library (Capability API) |
| [Global API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/MashupAPI/qlik-global-interface.htm) | JavaScript library (Capability API) |
| [Table API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/TableAPI/qlik-table-interface.htm) | JavaScript library (Capability API) |
| [Selection API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/SelectionAPI/qlik-selectionState-interface.htm) | JavaScript library (Capability API) |
| [Variable API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/MashupAPI/qlik-variable-interface.htm) | JavaScript library (Capability API) |
| [Visualization API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/VisualizationAPI/VisualizationAPI.htm) | JavaScript library (Capability API) |
| [App Integration API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/AppIntegrationAPI/app-integration-api.htm) | URL integration |
| [Single Integration API](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/single-integration-api.htm) | URL integration |
| [qlik-visual](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/QlikVisual/qlik-visual.htm) | Web component |
| [Leonardo UI](http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/LeonardoUI/LeonardoUI.htm) | UI library |

More information on APIs can be found on official website: <http://help.qlik.com/en-US/sense-developer/3.1/Content/APIs-and-SDKs.htm>

API is the foundation of system integration; it makes communication with sense engine possible.

# Embedding charts



There are four APIs for visualization integration. They are:

* Visualization API
* App integration API
* Single Integration API
* Qlik-visual

And app.getObject method in app API.

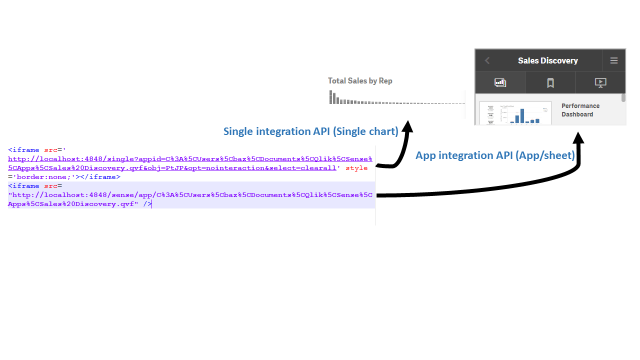
# App integration API and Single integration API

App integration API and single integration API are two similar APIs and they are relatively easy. Every “Object” in Qlik sense: charts, extensions, app and worksheet, can be accessed through URL; App and worksheet are referenced using app integration API; Individual charts are referenced using single integration API.

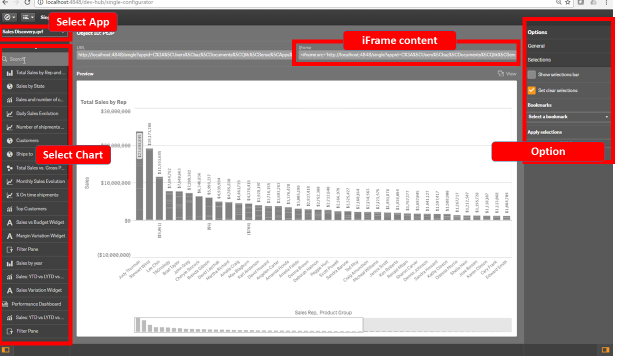
When I said integration API and single integration API are relatively easy I actually meant EXTREMELY easy. Create an empty html file and past the following code inside. Open Qlik sense desktop, open the html file, there you go.

<iframe src='http://localhost:4848/single?appid=C%3A%5CUsers%5Cbaz%5CDocuments%5CQlik%5CSense%5CApps%5CSales%20Discovery.qvf&obj=PtJP&opt=nointeraction&select=clearall' style='border:none;'></iframe>

<iframe src="http://localhost:4848/sense/app/C%3A%5CUsers%5Cbaz%5CDocuments%5CQlik%5CSense%5CApps%5CSales%20Discovery.qvf" />

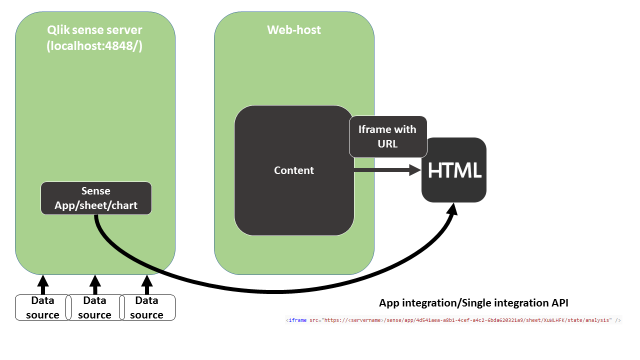


If the iframe is broken, open single configurator page at <http://localhost:4848/dev-hub/single-configurator> or <http://localhost:4848/single>. Select your app, your chart and option, copy the text in iframe content.



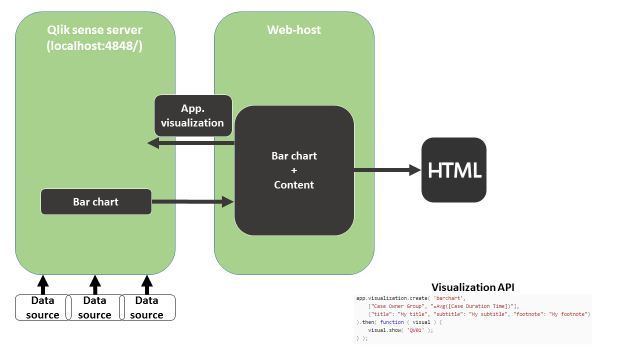
Visit <http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/AppIntegrationAPI/app-integration-api.htm> and <http://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/single-integration-api.htm> for more information.

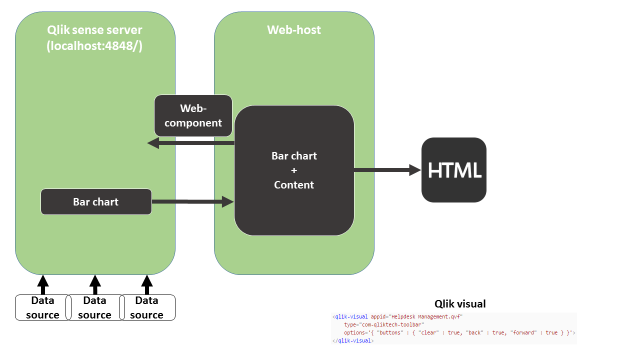
App integration API and single integration API are simple to implement because their resource pool is limited to static objects. The web host provides nothing but a URL, the browser visit localhost:4848 (Qlik sense desktop server) to fetch content.



# Visualization API and App.getObject

Visualization API and Qlik visual API are similar, Qlik visual is a web component version of visualization API. This approach is slightly more complicated than iframe embedment because communication with Qlik server is invovled. But they are definitely more powerful.



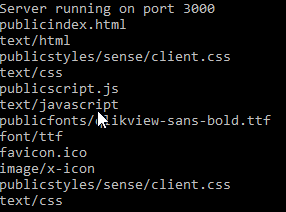


# Coding example

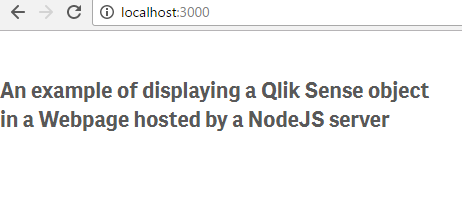
I use nodejs-qliksense by Websy (http://branch.qlik.com/#!/project/56728f52d1e497241ae69815) as a starting point.

Download the file from Github, unzip it; open command prompt; cd <your folder location>.

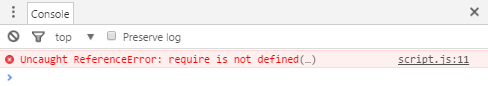
Perform a npm install: . Now we have the server ready. Start the server using  , the follow text is shown and your server is running.



Open your browser, Chrome is used in this tutorial. Go to localhost:3000 you will see this page.



Open developer portal by pressing F12, “require is not defined” error message is shown.

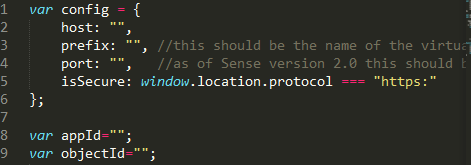


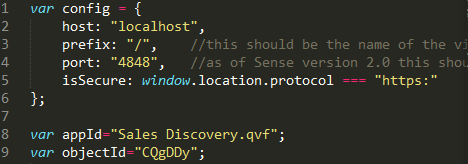
This is because requireJS is not presented. Navigate to nodejs-qliksense-master/public/index.html, Change the content of script tag to “localhost:4848”.



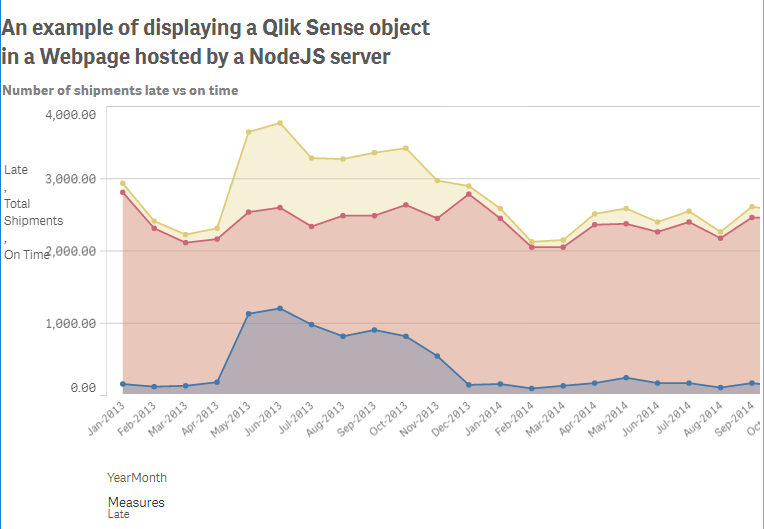


Navigate to nodejs-qliksense-master/public/script.js, Change config, appID and objectID.



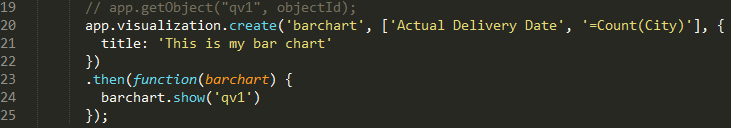


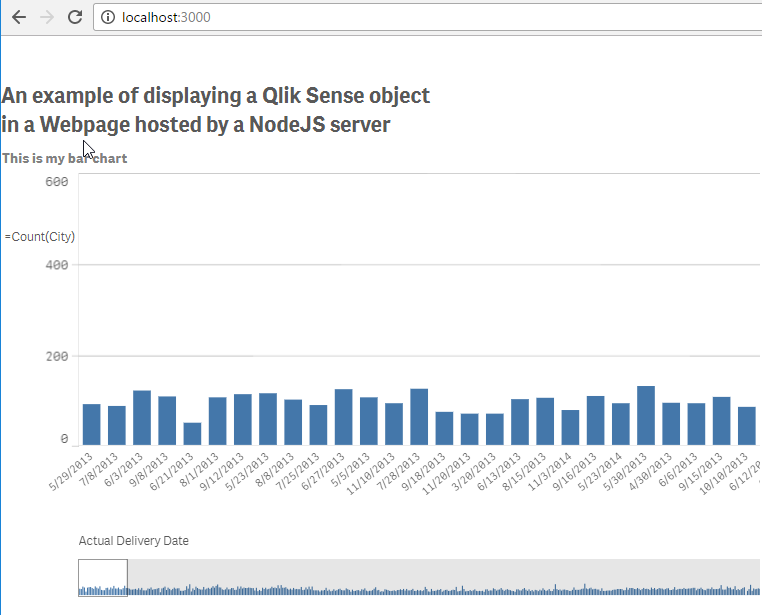
There you go.



You can replace getObject with visualization API to Create charts dynamically.





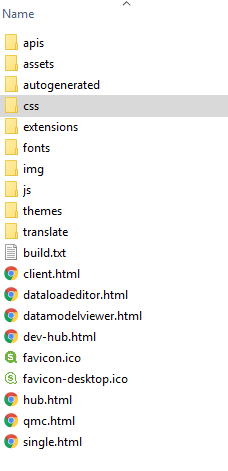


## Behind the code

If you take a look at the header, requireJS module is imported directly from qlik server.



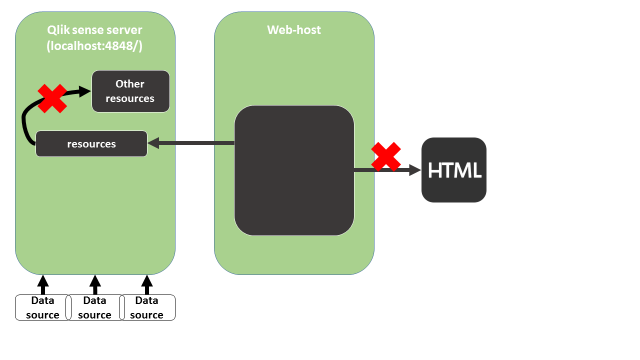
For qlik sense desktop, all resources are allocated in this folder: C:\Users\<Your user name>\AppData\Local\Programs\Qlik\Sense\Client, can be accessed through <http://localhost:4848/resources/>

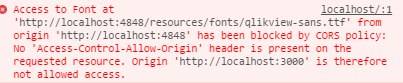


These are fonts, images, css files, js modules. Referencing them may be tricky.

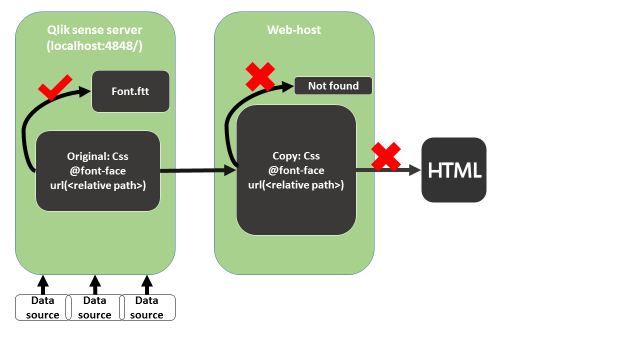
For example, referencing qlik-style.css directly from server will most likely result in CORS policy error. This is because Chrome blocks indirect referencing.



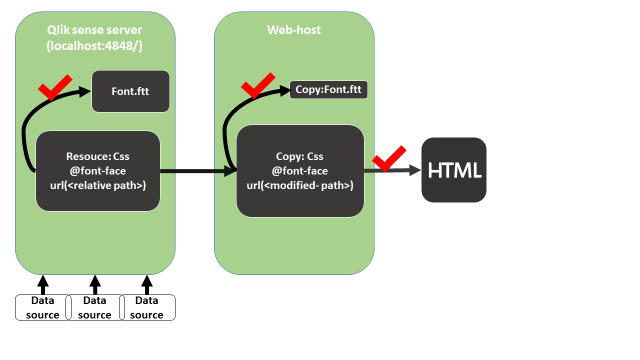




The most intuitive solution is to make a local copy of the file.



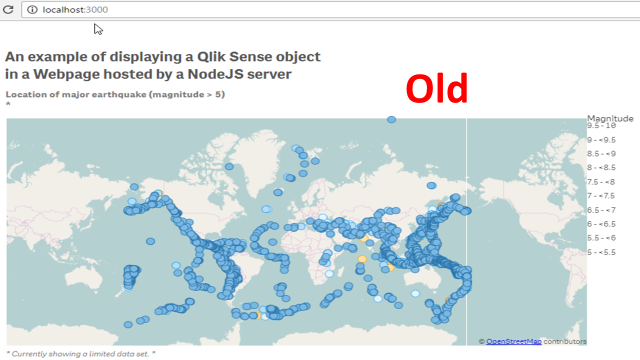
Don’t forget to copy the indirect resources and modify referencing path.

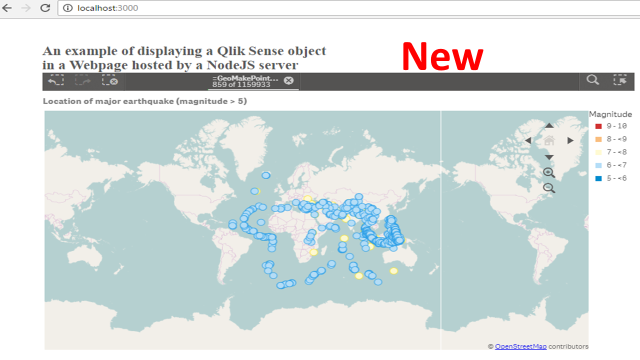


That is why in the base code, author migrated the content of qlik-style.css to public/styles/sense/client.css. And font style files are referred locally in public/fonts.

The code was developed base on previous version of Qlik sense, newer features in sense 3.x for example Leonardo UI and font-italic are missing. That is why you may encounter error with a few APIs. for example, appending a current selection bar may result in ERR\_CONNECTION\_REFUSED error. You may use the updated sample code attached with this code. When you finally resolve all conflicts/error, you will see minor differences.



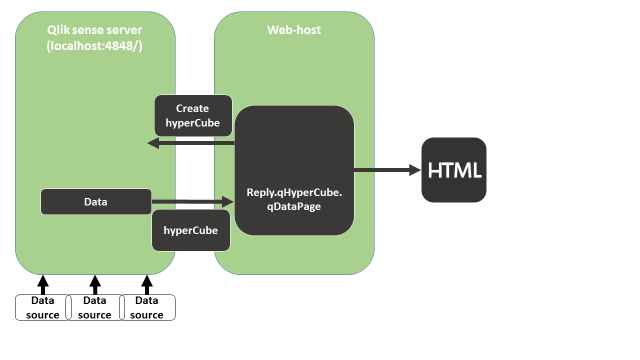


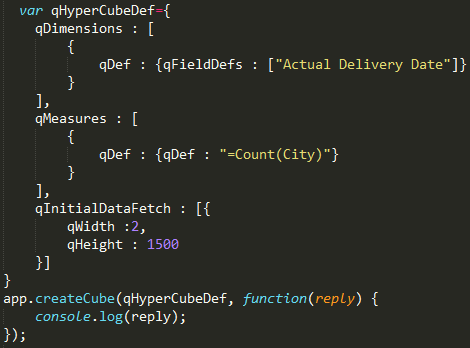


# Getting data

Getting data is much easier than getting charts. For qlik sense, data is framed and accessed as hypercube. For more information on hyperCube, please refer to <Apendix2 dimension measure lisObj and hyperCube.docx>.

To access data from qlik engine, it is recommended to create a hypercube.

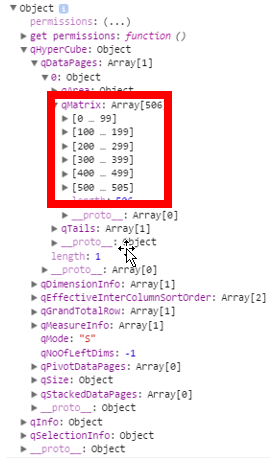




More information on app.createCube function, refer to https://help.qlik.com/en-US/sense-developer/3.1/Subsystems/APIs/Content/MashupAPI/Methods/createCube-method.htm.

In the call back method, we print the returned argument.

Take a look into reply argument, data is in reply.qHyperCube[0].qMatrix[index].



For any question, join our Slack channel: qlik-branch.slack.com!

Ping me(@Wu\_zhu) or Brian(@munz) or Alex(@Alex) or Rey(@rey) in the channel.