Qlik Sense SaaS demo tool setup instructions

Contents

[Introduction 3](#_Toc488060497)

[Overview of architecture 3](#_Toc488060498)

[Integrated architecture overview 6](#_Toc488060499)

[Start of Demo tool setup instructions 8](#_Toc488060500)

[Setup Qlik Sense 11](#_Toc488060501)

[Insert redirect URL helper page for DIV tag integration single sign on 16](#_Toc488060502)

[Security rules 16](#_Toc488060503)

[Users 17](#_Toc488060504)

[Setup streams 17](#_Toc488060505)

[Setup apps 17](#_Toc488060506)

[Extensions 19](#_Toc488060507)

[Publish apps 20](#_Toc488060508)

[Export certificate 21](#_Toc488060509)

[City bike demo app 21](#_Toc488060510)

[Setup Meteor project environment variables to refer to Qlik Sense 23](#_Toc488060511)

[QRS Meteor 23](#_Toc488060512)

[Setup web integration Meteor server 25](#_Toc488060513)

[Start the two meteor servers (they simulate the SaaS platform) 26](#_Toc488060514)

[Appendix 1: What is the link between Meteor QRS and Qlik Sense 27](#_Toc488060515)

[Appendix 2: Insert redirect URL helper page for DIV tag integration single sign on 30](#_Toc488060516)

[Appendix 3: Troubleshooting 31](#_Toc488060517)

[QRS Meteor Issues 31](#_Toc488060518)

[Slide generator issues 34](#_Toc488060519)

[Sense Web integration issues 35](#_Toc488060520)

[Meteor issues 38](#_Toc488060521)

[Bcrypt issues on windows 39](#_Toc488060522)

[Hardcore meteor resets 39](#_Toc488060523)

[Appendix 3: Captcha security 40](#_Toc488060524)

[Appendix 4: Background documentation 41](#_Toc488060525)

[Login screen, User acount creation (not used at the moment) 41](#_Toc488060526)

[Change the look and feel using Semantic UI (for QRSMETEOR) 41](#_Toc488060527)

[Change layout (optional for QRSMETEOR) 42](#_Toc488060528)

[Example project to run your first meteor webserver and mongoDB 42](#_Toc488060529)

[Install meteor workaround 43](#_Toc488060530)

# Introduction

This manual describes how you can download the source code of SaaSDemo.qlik.com from Github, and how you can run your own demo server.

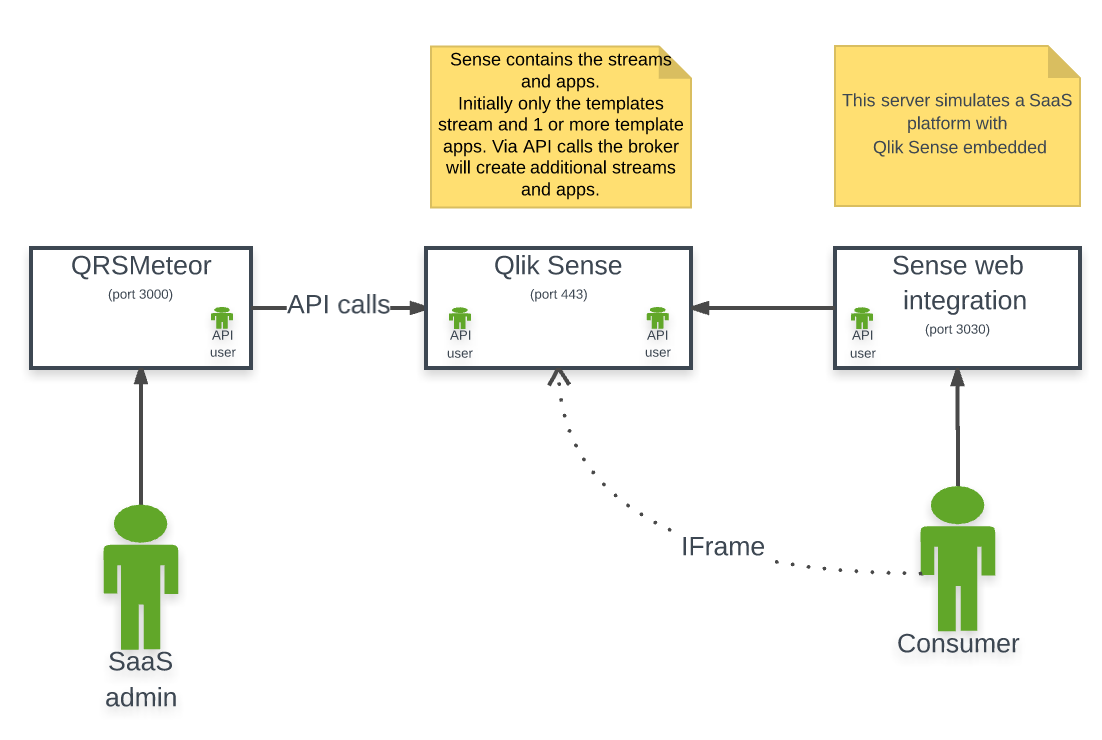
This demo platform provides you with a wrapper around the Qlik Sense QMC, so you can send API calls to Qlik Sense by pressing certain buttons. Using these calls you can, for example, create streams and apps or complete any other activity that you’d be able to do in the QMC or app. Your SaaS platform most likely uses rest APIs already, so just copy the code and start automating. The tasks we have automated are what we refer to as the "generation.” This demo platform uses MeteorJS as the basis to simulate your SaaS platform. You can adjust the source to suit your needs. This manual explains how you can setup meteor to communicate with Qlik Sense.

The steps

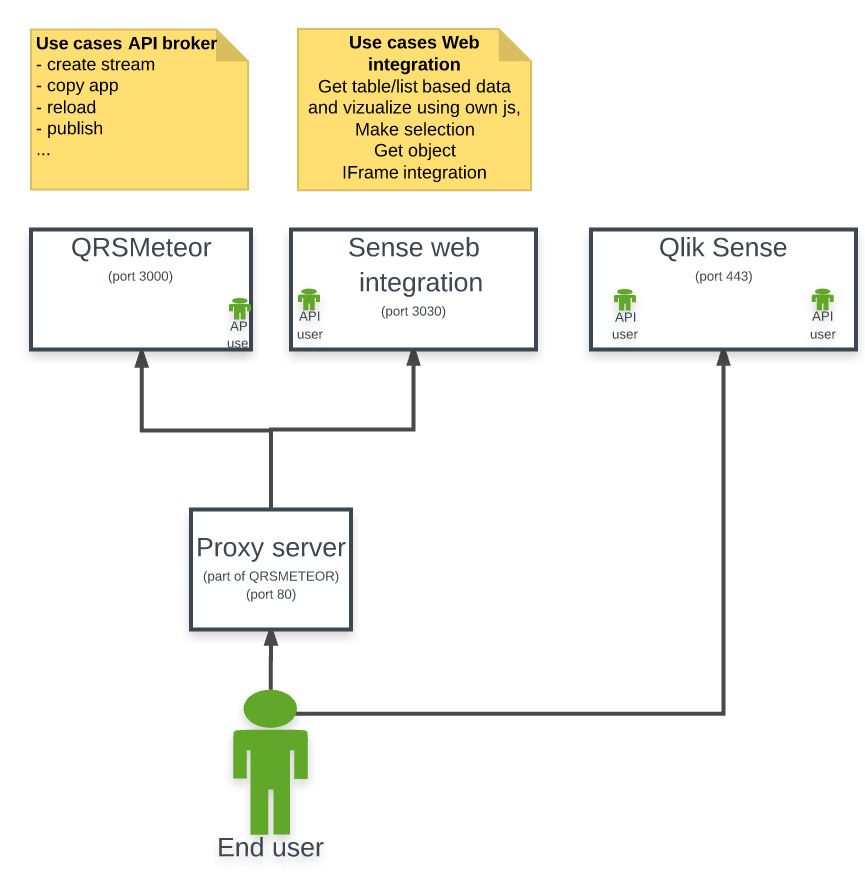
1. This SaaS demo platform is based on MeteorJS (a combination of NodeJS and MongoDB). So we first have to install that.
2. Next we are going to install and setup Qlik Sense (install some template apps, and install some extensions)
3. Next we have to put some settings in a text file so our 2 meteor servers know where Qlik sense is located.

# Overview of architecture

In the picture below you can see what the 3 servers are doing. Each has a different purpose.



Use case overview:



|  |  |
| --- | --- |
| QRS Meteor server | * + - Custom tool which uses the Qlik Sense APIs. The app generation tool or "the broker"     - Also serves as authentication module |
| Web integration Meteor server | * + - The web portal. Simulates the website of the OEM partner |
| Qlik Sense server |  |

In the end you can put everything on 1 machine or create 2 separate machines. Via the settings-production.json files in the project root of each meteor project you can configure the link. Also some settings are set via the startup environment variables in the bat files when you start meteor.

In the end you will get:

|  |  |
| --- | --- |
| A Qlik sense server | Running on the port you can set in the proxy settings, make sure you link the host and port in the settings.json files in the project root of both meteorQRS and webintegration. (see the settings-example.json file in the project root)  "mbj@qlik.com" ,  main.js  FOLDERS  V B QRSMeteor  .meteor  client  b imports  node modules  public  C) Sense Demo materials  server  •gitattributes  •gitignore  Client.pem  client key.pem  packagejson  qrs.sublime-project  README.md  settings-development - example.json  E •settings-development.json  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  'adminEmai1s " :  'public" :  "host  "PRESALESI",  'port" •  . 81,  "UDC": "OEM HOST" ,  'virtualProxyC1ientUsage"  'meteor"  "weblntegrationHost": "n1sch-mbj3" ,  "weblntegrationDemoPort" :  3030,  "facebook" :  'clientld  ". "1754933311453043"  "templateAppId" :  "5658d4a5-01e2-4b74-9c5d-1b01c2430c8f"  private" •  "host  "PRESALESI",  'port" •  . 81,  'certificatesDirectory"  "C : \ \Users \ \mbj \ \GitHub\  'notificationURL": "http://n1sch-mbj3:3000/updateSenseInfo" ,  "SenseServerInterna1LanIP" :  " PRESALESI " ,  'useSSL": false,  passphrase" :  null,  'authentication": "header" ,  'virtualProxy": "hdr" ,  "headerKey"  "hdr-usr",  "UDC". "OEM HOST" ,  "headerVa1ue  'isSecure": true,  'engineHeaders " :  " UserDirectory=PRESALES1; UserId=Q1ik" ,  'enginePort": 4747, |
| A meteor server for [meteorQRS](https://github.com/QHose/QRSMeteor). This server is needed for the   * + documentation, videos and   + the app generation and   + Serves as the authentication module for the SaaS platform demo. | Apps  C O saasdemo.qlik.com  Bookmarks sensedemo.qlik.com  S  SaaS with Sense  DBQRSMeteor I  Documentation  Com  knsv.github.io/merm  Cases: Home Salesf  loc SaaS  O  TODO app  X The power of  Multi-tenant Publisher  Your SaaS platorm  QMC API  Videos  The power of API's; the SaaS with QIil< Sense demo platform  What is the concept which we try to explain in this demo? Well  today we will show you how you can easily demonstrate the power  of the QIik Sense APIs to customers and partners to fully extend and  take advantage of QIik Sense to embed analytics within a multi-  tenant SaaS application environment.  Read more  or Introduction video  How to use the demo?  To put it simply, this demo shows you how you can copy an app  (dashboard) which serves as a 'template layout" for each of your  customers. Next to creating a copy, we will also reload the app with  data from the customers database. Because you can insert a custom  script you can also specify customer specific data models like  "custom fields and tables" or create them on the fly in order to  connect to big data sources. In the end we make a stream per  customer and publish the app into this stream.  .4  SaaS API automation — a deep dive  scnsc  iii  Template (app)  Your broker  (REST  Your configuration database  Customer table  Connection strings  user & passwords  user rights  Definition of data model  • including custom fields and tables  Olik@  In the demo site that belongs to this video you will see the  broker in action. But what are the activities the SaaS demo  broker automates?  Read more  "How to" Video or  the demo  Click here to see videos about the concept and the technology used. |
| A [meteor server](https://github.com/QHose/senseWebIntegration) for the web integration demo. This simulates the SaaS platform of the OEM partner.    This is a separate web server running on the web integration port you specifiy in the settings-production.json file (in the project root). This project shows the iframe and div tag integration. | On step 4  C O saasdemo.qlik.com/generation  . Apps Bookmarks sens emo.qlik.com  SaaS with Sense  S  DBQRSMeteor I  Documentation  Com  Multi-tenant Publisher  Your SaaS platorm  QMC API  I nsert Customers  For demo purposes; Insert customers and users  (In real life you forward them dynamically  from your SaaS platform to QIik Sense)  4: Test Single Sign On  Select a user to test Single Sign On.  Read more...  Simulate a user login  Videos  Select the template  This is just a regular app you cre  and published in the templates s  Try different users and notice the difference in apps and streams visible. Test row level  security (by making use of the country group of the user), in the "row level security  app" in the "Everyone" Stream. Notice that a user with role  • Consumer can only display  • Contributor can create sheets in existing apps (but no apps)    If you click  • Developer can create apps and load data. (Use our client via the hub)  Admin can maintain their customer apps in the QMC  You can maintain users and groups, logout or switch user, and notice the changed  authorizations. Some users have access to create sheets or apps, others can only  display and select.  Users and groups  John  Levels of self service  See Qik ense embedded    You go to this separate website, in a new tab  SaaS with Sense  x  saasdemo.qlik.com:808C  x  O saasdemo. lik.com:8080/hu  . Apps Bookmarks sensedemo.qlik.com  SaaS with Sense DBQRSMeteor I Com  knsv.github.io/merma Cases: Home  S  Salesf  loc SaaS  O  TODO app  x  Your SaaS Platform with QIik embedded  Via IFrame •  Via Div tags  Back to SaaS multi tenant demo  'Frame Integration: Hub  This demo shows that you can integrate the QIik Sense Hub using an IFrame  With an HTML IFrame you can show a website inside another website.  This is the most simple and loosely coupled integration. You will use our QIik Sense client (the Hub) to display and modify content like apps and sheets  • In the example below we included: http://saasdemo.qlik.com:81/meteor/hub  Note that using this generic link, the hub automatically shows only authorized data for the logged in user. (This is possible because we have sent a ticket to QIik  Sense with all the information about the user (userld and group memberships). Based on this we have created 1 security rule in the QMC: user.group =  resource.name (filter: Stream *) that ensures a flexible authorization concept  Single sign on (ticket request) redirect demo page  To slow things down for demo purposes we have put a "page in the middle".  between the user trying to access QIik Sense  and the demo site (your SaaS platform), in which the user is already logged in. So this is a perfect place to get the users id and groups (or any  array of fields you would like to use to authorize on) and communicate these values to the QIik Sense Proxy via the ticketing process.  This allows QIik Sense to re-use these groups in the resource and "row/column level" authorizations.  Why do you see this page?  Read more about Security or  Request a ticket for the lozed in user |

# Integrated architecture overview

In the picture on below you will find all integration aspects together in 1 overview for a typical SaaS provisioning architecture, in which we have:

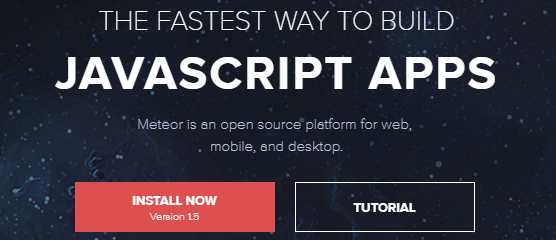
* Your SaaS platform's server (our senseWebIntegration server)
* Qlik Sense server
* A database per customer (the collections in our mongoDB)
* A master configuration database
* Your current API "broker" (meteorQRS project)

Your SaaS platform 
IF rame 
DIV-tag 
Authentication - Single Sign on 
re-use your existing way of authenticating users. 
Insert a small piece of code inside your existing "security 
module". In this module you already know who the user is 
(authentication) and what his rights are (authorization). 
The only thing we need to do is create a trust relationship 
between your server and the new QIik Sense server (via 
an SSL certificate exchange), and send this information 
about the user (user ID, group membership) in a JSON 
object to Sense (we call this ticketing). 
Exarnple ticket code 
var profile { 
'UserDirectory: 'Sunny Petrol', 
'Userld': 'John' 
'Group': Sunny Petrol', 
'Group': 'Manager, 
'Group': 'Germany 
qlikauth.requestTicket(req, res. profile); 
OEM Customer 
You can integrate the hub, an app, a sheet or 
individual chats. In most cases you have the 
flexibility to control whether you would like to 
see our menu, our selection bar, or whether or 
not you would like to provide initial selections as 
opposed to making the charts static so that no 
selections are possible at all. 
DMZ Firewall 
rust via QMC certificate exchange 
Your SaaS platform 
Your goal is to create embedded 
analytics using the alik platform within 
a multi-tenant SaaS environment, 
where each of your customers would 
only have access to their own data 
Trust via QMC certificate exchange 
updates to apps and custom content 
VVe normally create a temp ate app per business process or 
area In your software How can you update this template? 
Your SaaS platform can update the template app whenever 
you want The generated apps are all pubished into the 
stream with the name of the customer. The sheets created 
by Your SaaS platform are •the base sheets" and are read 
only for the customer. Custom content stored separately 
and won't be overwritten. 
Spoily - 
Cloudy - 
Sunny 
Petrol - 
Sales.QVF 
Authorization: resources 
Based on the user ID and group membership streams, apps and 
sheets (called "resources" in our rule based security model) are 
either shown or not. Lets say we start simple and each customer 
can only view their own dashboards. Since we gave each 
customer their own stream, we can grant access if the name of 
the stream (the company name) matches the name of the 
company we received from the website in the ticket. This can be 
controlled by adding 1 row of security configuration: user-group - 
resource. name. 
Row level security 
Using 'section access' in the script we enforce row/column level 
security Here we also use the group of the ticket and match its 
value to a field value in your data In our example case we only 
show sales data where from country is Germany' 
Your SaaS platforms user repository 
(authentication module with 
users and groups) 
Customer 1 
Your configuration database 
Customer table 
Connection strings 
User & passwords 
User rights 
Customer 20.001 
Definition of customer specific data model 
including custom fields and tables 
Ternplate 
app - Sales 
DMZ Firewall 
API Broker. Lets explain how you can use our AP s to make a copy Of the template for each Of your 
customers and load it with customer-specific data 
n this diagram, we've made a few assumptions, 
• we've assumed you already have a multi-tenant SaaS platform and 
• thus assume you have a piece of software which acts as a broker; a software capability that you 
already use today to maintain your SaaS applications for all your customers. 
The principle of SaaS automation is quite simple. You first create a template app, also called a dashboard, 
and you make a copy for each of the customer instances on your SaaS patform The temp ate app 
a ready contains a template with 80% of the functionality and we provide some predefined sheets, charts, 
dimensions and measures to get you started. 
Autonuted use cases: 
Create stream 
Copy app 
Replace script 
Reload app (load the data hside the app) 
Publish app intro <customer name> stream 

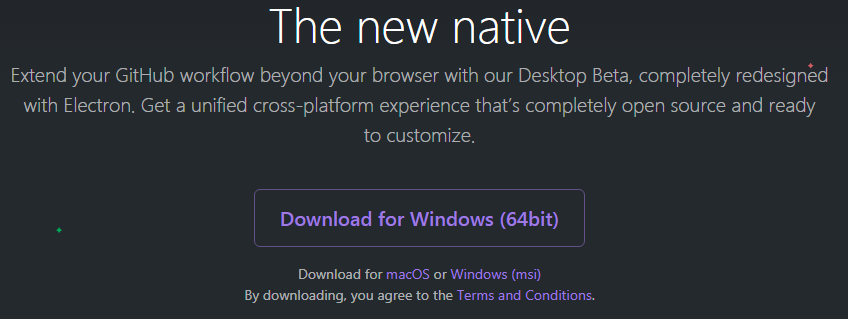
# 

# Start of Demo tool setup instructions

* + Install MeteorJs from www.meteor.com (Install under an administrator account. This is the account you will always use for your environment)



* + Install Qlik Sense
  + Note that Qlik Sense services have a startup delay. So wait 10 minutes if you don’t see the QMC/Hub.
  + Run CMD, services.msc to see if you services have started (Qlik Sense ….)
  + Install GitHub [Desktop](https://desktop.github.com/) .



|  |  |
| --- | --- |
| Open github desktop and clone the 2 repositories. (After you restarted your computer, you can also click on the links above, and select open in desktop (Make sure you are logged in on the GitHub website).  After you opened the project in your GitHub desktop you can press Fetch Origin.  In this way you always get the most recent version of the site. | Go to [QRSMETEOR](https://github.com/QHose/QRSMeteor.git)    Repeat for [senseWebIntegration](https://github.com/QHose/senseWebIntegration.git)    After you opened the 2 projects, you will see them in github desktop.    You should have this directory structure (this is the **project root folder)**:  OOT  QRSMataor  . meteor  client  imports  u  comp atibility  node modules  public  a  Sense Demo materials  server  .gitattributes  •gitignore  package.json  qrs.sublime- project  qrs.sublime-workspace  README.md  settings-development - example.json  settings- development.json  Local Certificates  Date modfied  9,'18/2016 AM  8/18/2016 1:17 PM  9,'17/2016 3:42 PM  7/30/2016 1200 AM  PM  8/14/2016 1049 AM  9/21/2016 7:00 PM  407 PM  6/1/2016231 PM  7/28/2016 ill PM  5/18/2016 1:28 PM  9/20/2016 3.0 PM  9,'18/2016 AM  9/20/2016 1:55 PM  3:33 PM  Type  File folder  File folder  File folder  File folder  File folder  File folder  File folder  File folder  GITATTR18UTES Fil  GITIGNORE File  JSON File  SUBLIME-PROJEC.  SUBLIME-WORKS.  MD File  JSON File  JSON File |
| Go to the **project root** (where you still are), open the command prompt and run “meteor npm install --save” |  |

# Setup Qlik Sense

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ports and firewalls | Make a table of the relevant ports you need to open on your firewall.  Go to the proxy and enable http, so you don’t have to configure certificates.  (In this example we used 443 for http) This port 443 and others you will refer to in the settings.json files (When you start the server, you point to this file. This makes sure all servers can find each other)   |  |  | | --- | --- | | Sense | 443 | | Meteor qrs | 3000 | | Meteor web integration | 3030 | | Port to external mongo db (if you use this, not needed for a simple demo) | 15417 | | Engine port (EnigmaJS uses this) | 4747 | | Proxy port ­­ | 4243 | |
| Setup header authentication Virtual Proxy. | Machine generated alternative text: X Virtual proxy edit  IDENTIFICATION  Description  Prefix  Session inactivity timeout (minutes)  Session cookie header name  AUTHENTICATION  Anonymous access mode  Authentication method  Header authentication header name  Header authentication dynamic user directory  Header auth  hdr  The prefix must be unique for all virtual proxies u  the LIRL (https://[nodel/[prefixl/)  30  X-Qik-Session-hdr  The session cookie header name must be uniquE  No anonymous user  Header authentication dynamic user directory  hdr-usr   |  |  | | --- | --- | | Make sure you link the virtual proxy to an engine. | Machine generated alternative text: LOAD BALANCING  Load balancing nodes  Server node  Add new server node  Central | | Optional: Add the server host names in the whitelist.  Make sure you insert the external hostname, ip address of the server as well as the internal ones. | Host white list  C Add new value  172.31 .22.248  52.50.134.241 | | Link it to the central proxy | X Virtual proxy: associated items  X Associated proxies Showing: 1  Node  Central  Selected: 0  Status  running | |
| Setup ticket Virtual Proxy | Create a meteor virtual proxy (make sure you point to the SSO route!, see below), just change 2008ENT with the name of your meteor server host. Make sure you put the correct port number where meteor runs. The end result must be like this  Machine generated alternative text: AUTHENTICATION  Anonymous access mode  Authentication method  Authentication module redirect URI  No anonymous user  Ticket  http://2008ENT 3000/sso    Put your sense server hostname in the redirect url.    Only for demo purposes:    Link the Virtual Proxy to an Engine.  Insert host whitelist.  Link the Virtual Proxy to a Proxy.    **Background ui/router.js**  // Single sing on integration route for dummy users of step 4 and SSBI demo  Router.route('/SSO', {}); |

|  |  |
| --- | --- |
| Setup presentationSSO Virtual Proxy | Put your sense server hostname in the redirect URI.    Only for demo purposes:    Link the Virtual Proxy to an Engine.  Insert host whitelist.  Link the Virtual Proxy to a Proxy.    **Background**  //users for the slide generator have their own virtual proxy redirect path  Router.route('/presentationsso', {}); |
| Setup anon Virtual Proxy (for the JSFIddle exercises) | Link the Virtual Proxy to an Engine.  Insert host whitelist.    Link the Virtual Proxy to a Proxy. |

## Insert redirect URL helper page for DIV tag integration single sign on

See appendix for more info on how to handle the authentication flow.

## Security rules

Create the following security rules (See the most recent security rules implemented [here](http://saasdemo.qlik.com/securityRules))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Rule | Resource filter | Actions | Comment |
| Z\_SLIDE\_GENERATOR | resource.name="API apps" and ( user.environment.group="TECHNICAL" or user.environment.group="GENERIC" ) | Stream\_\* | Read |  |
| Z\_Root\_ALL | ((user.roles="RootAdmin")) | \* | All actions | Root: Hub and QMC full access |
| Z\_QMC\_ADMINS\_CUSTOMER2 | user.environment.group = "Admin" and ( resource.HasPrivilege("read") or resource.app.HasPrivilege("read") ) | App\_\* | Update Delete Publish Export Data | Administrators are only allowed to maintain the apps and app.object in the QMC |
| Z\_QMC\_ADMINS\_CUSTOMER | ((user.environment.group="Admin")) | QmcSection\_Stream, QmcSection\_App\* | Create  Read  Update  Delete | QMC Menu only. Only give access to stream if the name of the stream match the AD group. Note, we disabled the default Stream Rule |
| Z\_OEM\_STREAM\_ACCESS | ((resource.name=user.environment.group or (resource.name="Everyone") )) | Stream\_\* | Read |  |
| Z\_OEM\_DEVELOPER | user.environment.group = "Developer" and ((resource.owner = user and resource.stream.Empty()) or (resource.app.HasPrivilege("read") and resource.published ="false" )) | App\* | Create Update Delete Export Publish Export Data |  |
| Z\_OEM\_APP\_ACCESS (CONSUMER) | resource.stream.HasPrivilege("read") or (resource.resourcetype = "App.Object" and resource.published ="true" and resource.app.HasPrivilege("read")) | App\* | Read |  |
| Z\_CONTRIBUTOR | user.environment.group="Contributor" and resource.app.HasPrivilege("read") and resource.published ="false" | App\* | Create Update Delete Publish | Can create own sheets in predefined apps. |
| Z\_AUDITOR | user.environment.group="GLOBAL AUDITOR" | \* | Read | display all for auditors |

Disable these rules (Disable export is optional):

* Stream
* CreateAppObjectsPublishedApp
* CreateApp
* ExportAppData
* StreamEveryoneAnonymous

## Users

|  |  |
| --- | --- |
| Setup license access rule. | Machine generated alternative text: User access rule edit  IDENTIFICATION  Name  Disabled  Description  BASIC  Actions  user  value  License rule to grant user access  Rule to set up automatic user access  Allow access  name  like    Also add login passes for anon users. |
| Add a Qlik admin | Create a local user Qlik in windows with a local administrator role and assign the root admin role in the QMC. You can also use this user in the settings.json file for QRS meteor. |

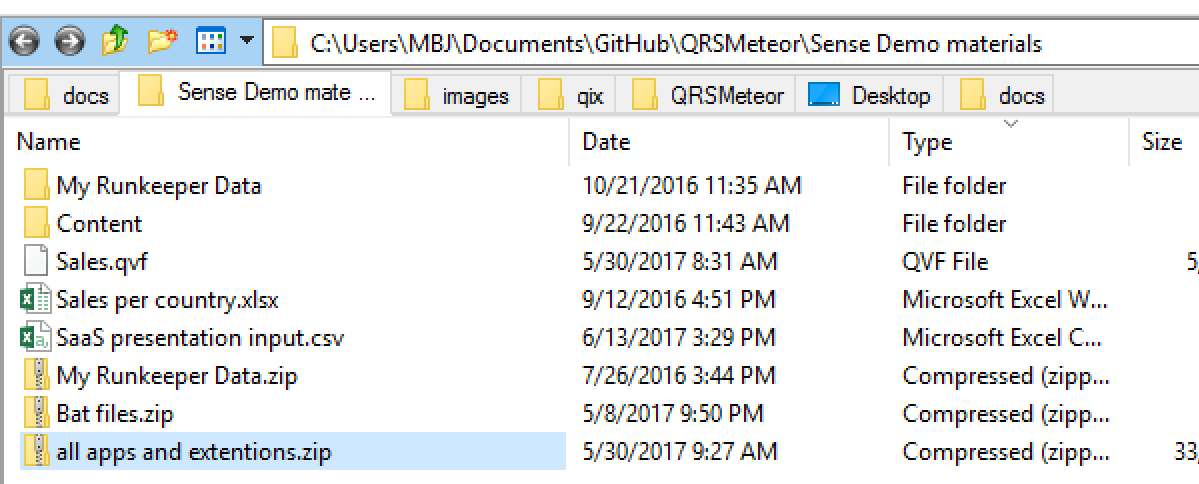
## Setup streams

Create the following streams (without adding a security rule) :

* + Everyone stream for the div tag integration app (in a default installation of Qlik Sense, this stream is already created).
  + Templates stream in which we store the template apps.
  + API apps stream for the app to store the slide generator presentation.

## Setup apps

After you have installed GitHub desktop and synced it, you should have a populated QRSMeteor folder.



(Please do not change the directory structure or files. Qlik sense for example loads the files in the Runkeeper data folder or a binary load from the Sales.qvf (do not use this app yourself))

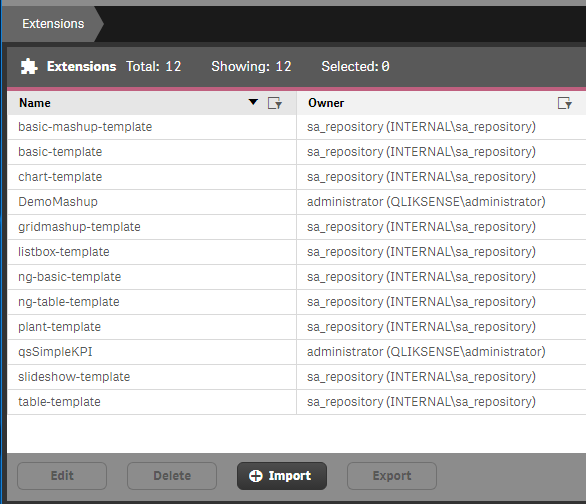
In this folder you will find the apps needed to import in: (all apps and extentions.zip)

|  |  |
| --- | --- |
| Import apps via QMC | * + 1. Sales.qvf     2. Row level security.qvf     3. My first template (Runkeeper.qvf)     4. Slide Generator |
| Create new app from Sales.qvf | * Duplicate the Sales app. * Rename the app to “Sales and Customers” . |
| Update data connections | Create a new data connection "Import demo" and let it point to your GitHub sync folder.  In my case: C:\Users\Qlik\Documents\GitHub\QRSMeteor\QRSMeteor\Sense Demo materials). |
| Update the data connection in the script so the loads refer to the new folder. | For the Apps:   * Sales * My first template (Runkeeper) * Slide Generator (Open App without data)   Examples:   * Runkeeper: the app will try to reload the EXTRACTED files of in the GitHub folder. * Sales : (after we published into the templates stream and after you created a copy for the customer) will try to reload via a binary load.   Perform a reload and save the apps after the update of the script. |
| Row level app: Update users | Open row level security app (open App without data).  Update the section access script, insert the correct domain names/users.  Perform a reload and save the apps after the update of the script. |
|  |  |
| Slide generator JSFiddles (only for integration.qlik.com | Update the app GUIDs in the fiddles. |

You have already imported the demo apps (see all apps and extentions.zip in the QRSMeteor\Sense Demo materials folder). So you now need to put these GUIDS later on in the settings-\*.json files. In the root folder of the github project. Also make sure you extract the My Runkeeper Data zip. All in the GitHub\QRSMeteor\Sense Demo materials folder.

## Extensions

In order for the applications to work, you need to import 2 extensions on the server:



* Import the simple [KPI extension](https://github.com/alner/qsSimpleKPI).
* Install the Demo Mashup Extensions ( In order to use the City Bike Demo app)

## Publish apps

|  |  |  |  |
| --- | --- | --- | --- |
| APP | STREAM | GUID | Purpose |
| My First Template(Runkeeper).qvf | Templates | 85da155b-bd30-487c-955d-f729f91e9306 | Used for the generation. Services as template app. |
| Sales and Customers.qvf | Templates | 0d4a1439-a921-4978-902d-433d1d38cb7e | Used for the generation. Services as template app. |
| Row level security.qvf  (SSBI app) | Everyone | b43115d3-648c-4f75-830a-6f416334baec | Demo row level security, also called self service BI app. Make sure you update its GUID in the settings.json file |
| Sales.qvf | Everyone | ­­­ 3f8418a7-0449-4375-af09-0aff80ca4f3a | Used for all the Iframe and div integration tag examples in web integration meteor server. |
| Slide generator.qvf (IntegrationPresentationApp) | API apps | 9b5091fb-5715-4bfb-a948-6d708d56da72 | Provides data and some sheets for the slide generator. Make sure you update its GUID in the settings.json file. |

## Export certificate

The Meteor server communicates with the Qlik Sense server via a couple of methods, but one of them is using certificates.

|  |  |
| --- | --- |
| If you are on the same server as Qlik Sense | You can just use the certificates that are in the standard folder of Qlik Sense. C:\ProgramData\Qlik\Sense\Repository\Exported Certificates\.Local Certificates  But you can also export certificates in QMC in pem format to a folder you like.  EXPORT  hchine narræ  PRESALESI  Retype g  Include key  Export for  e Add  P E V-fmt  to this diEk on    In qrs meteor settings.json point to the certificates folder. Make sure you put double \\ for each \ (See example settings.json file) |
| If you are on a different server as Qlik Sense for Meteor QRS. | Create a folder local certificates (meteor tries to load the exported certificates from QMC) in a specific folder. Export them without password in PEM format. Ensure you put the name of the machine where meteor runs. (check: windows command prompt, type hostname) |

## City bike demo app

If you want to use the citybike div tag integration demo in sense web integration you need to do some steps.

|  |  |
| --- | --- |
| Insert content in the QMC content library |  |
| Specify the correct /meteor/ proxy in the config object of the div tag. | var config  window . 1 oca ti on . ho stname ,  host:  prefix: " /meteor/ "  window . 1 oca ti on . ,  port :  issecure :  window. location. prot ocol  "https:"  re guile . config( {  baseTJr1: ( config.issecure ? "https •J/" : "http:/    File: DemoMashup.js in ‘C:\Shared\StaticContent\Extensions\DemoMashup’ |
| Edit the .js file in de dev-hub after you imported the demo qvf, make a note of the guid and insert it here | in  •wcr ti mc  DunoMa*  k. ( 7Ef7 . |
| (should be done automatically when you imported the zip file) Import the extension "network chart" |  |
| If you get an error when opening a sheet with an extension, you have to put a correct virtual proxy in the extension source code |  |

# Setup Meteor project environment variables to refer to Qlik Sense

Now Qlik Sense has been installed, we have to tell our 2 meteor servers where Qlik Sense is located and which ports to use etc.

## QRS Meteor

After meteor starts up using the bat file, it needs to know where Sense is, we configure this using the [variables](https://github.com/QHose/QRSMeteor/blob/master/settings-development%20-%20example.json) in settings-development.json or settings-production.json files. This is the main file that configures the host and ports of Qlik Sense etc.

|  |  |
| --- | --- |
| Variables | Description |
| Host | The host of the Qlik Sense server |
| Port | Port of Qlik Sense |
| UDC and userdirectory | You make a user Qlik in Qlik Sense. This user belongs to a UDC, normally the computername, update it everywhere. (Case Sensitive !). |
| virtualProxyClientUsage | The name of the Meteor Virtual Proxy . |
| webIntegrationHost | Where does the Meteor webintegration run? Normally on the same host as Meteor QRS. |
| webIntegrationDemoPort | The port you specifify in the web integration startup.bat file. |
| templateAppId | This is the guid of the runkeeper after you have imported into Sense. |
| SSBIApp | GUID of the Demo row level security app, also called self service BI app. |
| IntegrationPresentationApp | GUID of the Sales app. |
| IntegrationPresentationProxy | Name of the Integration Presentation Virtual Proxy. |
| certificatesDirectory | Location of the Qlik Sense certificates needed to communicate with the Qlik Sense server. |
| notificationURL | Not implemented at this time, can remain empty, but can be used if you want to get a signal (an incoming http call) if something in sense changed). |
| headerValue | Value of the header Autentication. UserDirectory\\Local Admin User |
| engineheaders | UserDirectory=**UDC**;UserId=**Admin user** |
| Google and facebook | Enter your API credentials if you want to use google auth. Otherwise disable the packages in (put a # in front of it)  main.js  FOLDERS  V B QRSMeteor  v .meteor  local  .finished-upgraders  •gitignore  .id  oackages  9  10  11  12  13  14  15  16  17  18  19  20  'meteor add' and 'meteor remove' will edit this  but you can also edit it by hand.  Check this file (and the other files in this dil  Meteor packages used by this project, one per 1:  accounts-password@l.3.e  accounts-goo le  service- conf uration@l.0.10  useraccounts : iron -routing  useraccounts : semantic-ui  aldeed : autoform  aldeed : collection2 |
|  |  |
| [Example settings file](https://github.com/QHose/QRSMeteor/blob/master/settings-development%20-%20example.json) | OPEN FILES  UlHelpers.js  QPSFunctionsjs  X config.js  APILogs.html  X introduction.html  x SSO.html  main.js  FOLDERS  V B QRSMeteor  .meteor  client  Imports  node modules  public  C) Sense Demo materials  server  •gitattributes  •gitignore  Client.pem  client key.pem  packagejson  qrs.sublime-project  README.md  settings-development - exa  "Gettings-development.json  UlHelpers.js  QPSFunctions.js  config.js  settings-development.json  APILogs.html  introd  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  galaxy . meteor . com" :  'env": {  "MONGO URL" :  "ROOT URL" : '  'mongodb : / /qlik : QIik123@aws-us-east-1-portal.9.dblayer.com : 15417 /DBQRSMeteor " ,  http://saasdemo.qlik.com"  'adminEmai1s " :  'public" :  "mbj@qlik.com" ,  "host  "PRESALESI",  'port" •  . 81,  "UDC": "OEM HOST" ,  'virtualProxyC1ientUsage"  'meteor"  "weblntegrationHost": "n1sch-mbj3" ,  "weblntegrationDemoPort" .  3030,  "facebook" :  'clientld  ": "1754933311453043"  "templateAppId" :  "5658d4a5-01e2-4b74-9c5d-1b01c2430c8f"  private" •  "host  "PRESALESI",  'port" •  . 81,  'certificatesDirectory"  "C : \ \mbj \QRSMeteor" ,  'notificationURL": "http://n1sch-mbj3:3000/updateSenseInfo" ,  "SenseServerInterna1LanIP" :  "PRESALESI " ,  'useSSL": false,  passphrase" :  null,  'authentication": "header" ,  'virtualProxy": "hdr" ,  "headerKey"  "hdr-usr",  "UDC": "OEM HOST" ,  "headerVa1ue "  'isSecure": true,  'engineHeaders " :  " UserDirectory=PRESALES1; UserId=Q1ik" ,  'enginePort": 4747,  'proxyPort": 4243,  'google" :  1 |
| QRS Meteor | See settings-development - example.json in the project root in github |

## Setup web integration Meteor server

[Clone](https://github.com/QHose/senseWebIntegration.git) sense web integration github to your github folder (make sure you use the standard location, in your documents folder) (How to clone a Github Project, click here)

If you start sensewebintegration, make sure you meter run <name of your settings file.json>.

|  |  |
| --- | --- |
| Variables | Description |
| virtualProxyClientUsage | The name of the Meteor Virtual Proxy . |
| host | The host of the Qlik Sense server |
| portMeteorQRS | The port where the Meteor server runs on |
| port | Http(s) port defined in de Proxy. |
| isSecure | Flase – Don’t change |
| multipleDivAppGuid | GUID of the Sales app in the Everyone stream |

# Start the two meteor servers (they simulate the SaaS platform)

We have 2 servers to start, we use 2 separate .bat files to do this. You can put them in your windows startup folder. (Or configure as service via e.g. non sucking package manager)

* + QRS Meteor
  + Sense web integration

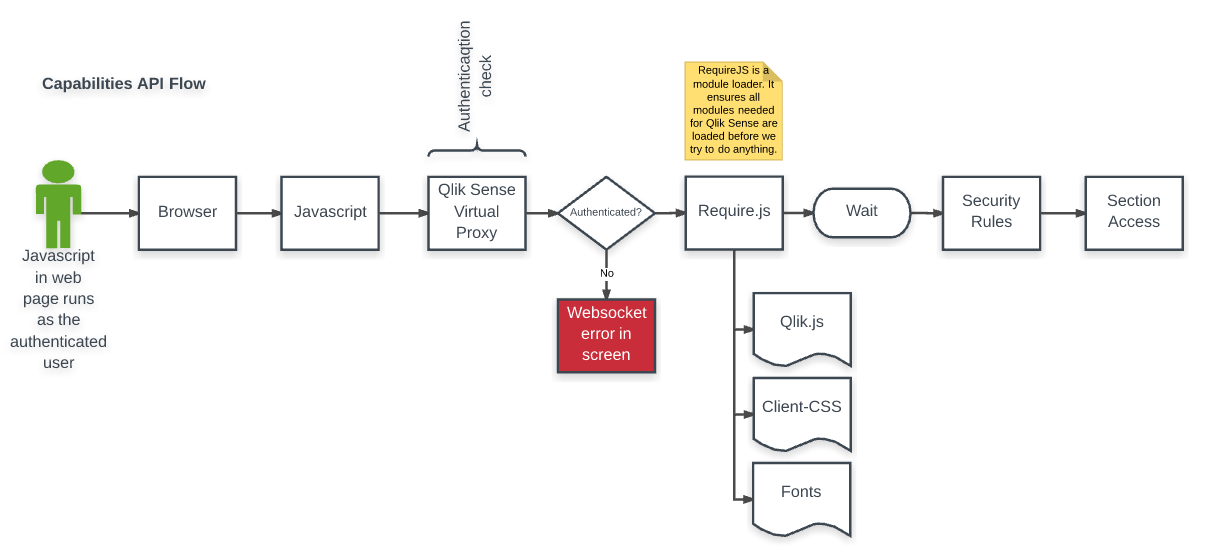
Ensure your windows machine [starts up](http://www.softwareok.com/?seite=faq-Windows-10&faq=28) the 2 meteor projects when the server starts. Create a bat file, and set windows environment variables. (edit the bat file below). Run as administrator.

|  |  |  |
| --- | --- | --- |
| Modify QRS.bat and webintegration.bat so it point to your github meteor root directory. | **Example content of bat file for QRSMeteor**  cd C:\New folder\QRSMeteor  set ROOT\_URL=http://integration.qlik.com  meteor run --settings settings-production.json --port 3000  **Example content of bat file for webIntegration** | //GO TO METEOR PROJECT ROOT FOLDER  //SET A WINDOWS ENVIRONMENT VARIABLE, SO METEOR KNOWS WHERE IT RUNS (YOU NEED THIS FOR GOOGLE AUTHENTICATION FOR EXAMPLE, AND METEOR NEEDS TO KNOW WHERE TO LOAD THE JS FILES FROM)  //THE ACTUAL COMMAND TO START THE APP, with --port 80 you can change the port on which it runs |
| If you run the bat file (for both meteor and web integration you create 2 bat files), the screen will look like this on the right. Watch this screen for errors. See problem solving below to resolve. | Publi  Privae  Avail*  11  Proa  Google  Chrome  Mozilla  Firefox  Micros...  GitHub  QRS start  : ROOT _URL=http://saasdemo .  : MONGO_URL=mongodb://q1ik:  ——settings setting  : run  arn ing: The  ——production flag should only be used to simulate production bundli  => Started proxy.  Building local packages |  |

# Appendix 1: What is the link between Meteor QRS and Qlik Sense

|  |  |
| --- | --- |
| . The real API calls between Meteor and Sense happen in these [pieces of code](https://github.com/QHose/QRSMeteor/tree/master/imports/api/server). | QHose / QRSMeteor  < > Code  (D Issues o  n Pull requests o  Projects o  QRSMeteor / imports / api / server  Branch: master  QHose intro page text, sso page andere text and buttons  QPSFunctions.js  QRSFunctionsApp.js  QRSFunctionsStream.js  QRSFunctionsSystemRules.js  intro page text, ss  disable api loggin  new doc links  security rules upd |
| Which apps and streams does the QRS Meteor show when you login? | The meteor server side (See github code: [publications](https://github.com/QHose/QRSMeteor/blob/master/server/publications.js)) only give the streams and apps back to the client that belong to the "Templates" stream, or which are generated by the user that is currently logged in in QRS meteor (with the login button).    **Example Meteor server side code, that return the apps and streams**  return Apps.find({  $or: [{ "id": { "$in": generatedAppsFromUser } },  { "stream.name": "Templates" }//, { "stream.name": "Everyone" }]  });      From <<https://github.com/QHose/QRSMeteor/blob/master/server/publications.js>>  return Streams.find({  $or: [{ "id": { "$in": generatedStreamsFromUser } },  { "name": "Templates" }//, { "name": "Everyone" }]  }); |
| The link between meteor and qlik sense is documented in a configuration file called settings-production.json. If you start meteor with the bat file, you will make a reference to this settings file.    The config file (see path in the picture), links the source code to this settings file.    You will see a public part (which all clients should know), otherwise they can't use the system. And private stuff which only the meteor server would be aware off. | V B QRSMeteor  .meteor  client  v imports  V B api  server  QPSFunctions.js  QRSFunctionsApp.js  QRSFunctionsStream.js  QRSFunctionsSystemRulesjs  APILogsjs  apps.js  -5 Skonfig.js  customers.js  Streams.js  B sso  10  11  12  13  14  15  16  17  18  19 v  20  21  22  23  24 v  25  26  27  28  (Meteor. isC1ient) {  var _ senseConfig {  "host": Meteor. settings . public . host,  port": Meteor. settings . public . port,  'virtualProxyC1ientUsage": Meteor. settings. public. virtualProxyC1ientUsage,  "UDC": Meteor. settings. public. UDC,  "weblntegrationDemoPort": Meteor. settings. public.webIntegrationDemoPort,  if  (Meteor. isServer) {  console. log( 'This Sense SaaS demo tool uses this config as defined in the settings-XYZ.-  import crypto from ' crypto'  import fs from 'fs' ;  var _ senseConfig {  "host": Meteor. settings . public . host,  "SenseServerInterna1LanIP": Meteor. settings . private. SenseServerInterna1LanIP,  port": Meteor. settings . private. port,  'useSSL": Meteor. settings. private.useSSL, |
| **Certificates authentication**  In order to login, request tickets and make a connection to the engine API we used certificates to authenticate. A certificate is nothing more than a very long String (a piece a text). So if we talk to the engine api we need a specific port and https and give some headers.  On the meteor server, make sure you make a note of the hostname. Then you go to QMC-Export certificates. Enter the hostname, and leave all other fields blank  **Header authentication**  The tool also uses the HDR virtual proxy, e.g. if you get the list of apps, we use header authentication to authenticate. | Export certificates in QMC  EXPORT  hchine narræ  PRESALESI  Retype g  Include key  Export for  e Add  P E V-fmt  to this diEk on    Link to the certificates  export const _certs  // server _ key: fs.readFi1eSync( 'C: / ProgramData/Q1ik/Sense/Repository/Exported Certificates/ . Loca  // server _ cert: fs.readFi1eSync('C:/ProgramData/Q1ik/Sense/Repository/Exported Certificates/ . Loc  key: fs. readFi1eSync (Meteor. settings . private . certificatesDirectory+ ' /client_key. pem' ) ,  cert: fs. readFi1eSync(Meteor. settings . private. certificatesDirectory+' /client. pem' ) ,  // ca: fs. readFi1eSync( 'C:/ProgramData/Q1ik/Sense/Repository/Exported Certificates/ . Local Certif    Here you see the javascript object we use to communicate via certificate authentication. E.g. in qpsFunctions.js  FOLDERS  V B QRSMeteor  .meteor  client  v imports  V B api  server  QPSFunctions.js  QRSFunctionsAp  QRSFunctionsStre  QRSFunctionsSyst  APILogsjs  apps.js  • L&config.js  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  key: fs. readF11eSync(P1eteor.settings. private. certificatesDirectory+' /cllent  _ key. pem •  fs . readFi1eSync (Meteor. settings . private. certificatesDirectory+ ' / client . pem ' ) ,  cert:  fs. readFi1eSync( 'C:/ProgramData/Q1ik/Sense/Repository/Exported Certificates/ . Local Cel  // ca:  // console. log( 'The certificates used for QIik Auth:  export var certicate_communication_options  rejectUnauthorized:  false,  hostname: _ senseConfig. SenseServerInterna1LanIP,  headers: {  ' x-qlik-xrfkey '  senseConfig . xrfkey ,  certs);  'X-Q1ik-User' : Meteor. settings. private. engineHeaders,  ' application/json '  ' Content -Type '  key :  cert:  certs . key,  certs . cert  exoort const securitvlnfo    QPS functions example code  FOLDERS  V B QRSMeteor  .meteor  client  v imports  V B api  server  QPSFunctions.js  QRSFunctionsAp  QRSFunctionsStre  QRSFunctionsSyst  APILogsjs  apps.js  config.js  customers.js  Streams.js  V BSSO  7  8  9  10  12  13  14  15  17  18  19  20  21  22  23  24  25  26  27  28  import  Iodash  from  Iodash;  ' Iodash '  When communicating with the QPS APIs, the URL is as follows:  https://<QPS machine name>:4243/<path>  Each prox has its own session  cookie, so you have to logout the users per proxy used.  export  if  function logoutUser(UDC,  / [console . log(  (name) {  // // console. log( 'Make  name  QPS Functions: logout the current:  + name +  on proxy:  + senseConfig.virtua1ProxyC1ientUsage);  QPS-10gout call, We authenticate to Sense using the options (including a certificate) object in the HTTPs call •  • ' ) ; // , cert  // // console. log( 'Meteor tries to logout the user on this URL: https://' + senseConfig.SenseServerInterna1LanIP + ':4243/qps/' + senseConfig.virtua  try  const call  call . action  call . request  call . response  ' logout user:  name ;  senseConfig . SenseServerInterna1LanIP  senseConfig. virtualProxyC1ientUsage  ' https://'  ' :4243/qps/ '  HTTP.ca11( 'DELETE',  call. request, { 'npmRequestOptions' :  certicate_communication_options } )  ' / user/'  UDC |

# Appendix 2: Insert redirect URL helper page for DIV tag integration single sign on



Challenge: if the user is not yet authenticated you well get a websocket error (the red box on previous image)

To make sure the user is always authenticated you can put a small HTML file on the Qlik Sense server. (C:\Program Files\Qlik\Sense\Client\returnUrl.html)

Create the returnUrl.html with the following content:

<!doctype html> <html><head><script>window.location=decodeURIComponent((RegExp('returnUrl=(.+?)(&|$)').exec(location.search)||[,null])[1]);</script></head></html>

Here’s the point of returnUrl.html if the user is unauthenticated: The user is sent to whatever Authentication module you have defined in the QMC – this can be google/salesforce/local authentication module/default windows authentication. The user authenticates and is then sent back to whatever mashup page the user came from, this can be a portal, a regular webpage or whatever.

# Appendix 3: Troubleshooting

## QRS Meteor Issues

|  |  |
| --- | --- |
| Cannot read property when starting up | Do you get "TypeError: Cannot read property 'google' of undefined", this means you started meteor without specifying the settings.json file. Or your settings file is not read correctly… |
| No connection to Qlik Sense | Try if you can connect to server/hub, if not check you proxy whitelist  Use the name or the server you use to connect the hub. If you go to <http://server.com>, then server.com should be in your proxy whitelist. (because there your virtual proxy runs)  Machine generated alternative text: Websocket origin white list  Add new value  92-111-9-190.statjc.cnello.nl  presalesl |
| License assigned to users? (no access pass) | Machine generated alternative text: IDENTIFICATION  Name  Disabled  Description  BASIC  Actions  user  value  License rule to grant user access  Rule to setup automatic user access  •J Allow access  userDirectory  like |
| Proxy and engine attached to Virtual Proxy?  s it active? | Machine generated alternative text: sa!X0'd  swau |
| Access issues | Security rules ok? (see if you can login in the hub, and see and open your app).   * + - * Ensure the user you login with has access to the hub, stream and app. Test this with the audit function in the security rules.       * Create the Z rules (from the example vm).       * Create local admin users (display\_QMC etc). |
| Ticketing issues | Extra meteor login screen? And wrong demo user login?  Ensure all the hostnames in your settings-production.json and in the Qlik Sense virtual proxy redirect are the same.  Are you getting a page not found error after you requested a ticket? (with redirect url: undefined in the api lots). This means the website you host meteor on does is not equal to the url you have put in the redirect url of the Qlik Sense proxy.      server2.domain.IocaI  'Frame Integration: Hub  p SaaS with Se nse  Yun-SaaS Platform Via IFræ-ne  Via Oiv taæ  q likserver2.domain.IocaI  Back to SaaS multi d«no  >  • Thi. can integrate Hub Æing sn IF.  HTML s'. 8 ins i &  • This is simple Y.' will Ü_lr Hub) to disp&y end  • the http:.4qKk—1  • Note thst Æing this g—ic link. he d&ts Ecket  in QMC = s Äpt  Iti-tawlt  Docu-r—ltation •  1  Sign In  Sign In with Google  1  Sign In  Must be in  Don't have an account? Register      Change the SenseServerInternalLanIP": "LOCALHOST", (if you are running meteor and sense on the same machine) |
| No streams or apps showing up? | Make sure you use the right usernames and domains in your settings.json files. (these calls go via header authentication)  Stream  App  Received O streams via QRS API from QIik Sense  Receiving the Streams from QIik Sense via the QRS API...  '/qrs/stream/full'  rows per page  Received O apps via QRS API from QIik Sense  Receiving the apps from Qik Sense via the QRS  '/qrs/app/full'  Templ  eration?  Stream  rows per page  of O  of 0 |
| No data in template app error | Make sure you have updated the script in the template app, are the data connections correct?  C:\C73B9CA5\5BBCB0A6-7DF1-4165-B4DC-97134F1B25ED_files\image044.png |
| When starting the Meteor server, an error appears when downloading the necessary packages. | Remove your Project GitHub folder and clone again.  Start de Meteor server again.  (If previous method does not work: uninstall and reinstall Meteor). |

## Slide generator issues

|  |  |
| --- | --- |
| Access denied | F12- console:  Check if you created a security rule for the ‘presentation’ virtual proxy for the slide generator. |
| No engine found | You have the wrong id of the slide generator.qvf in your settings.json file: |
| 403 issues | See below. Ensure whitelist is filled, and add CORS header in the presentation virtual proxy. |
| Make sure your slide generator app uses correct users and groups in section access. |  |
| 403 in server log | Cause: unauthenticated (wrong header user or certificate)  Make sure you have used the correct client and client key certificates which you exported from Qlik Sense QMC. Also make sure you refer to the directory where you have put them in the settings-\*.json file in the meteor project root. |

## Sense Web integration issues

|  |  |
| --- | --- |
| Undefined/undefind url | Machine generated alternative text: 92-111-9-190.static.cheI o.nl:3030/hub  )kmarks @ loc SaaS Q 10k. hub Q 10k. QMC 10k. Dev Hub @ Server SaaS Q S dev hub Q server QMC Creating an app  Methods I Meteor GL  Sense integatimderno Via 'Frame • Via Div tags  'Frame Integration: Hub  Back to SaaS nwlti tenant den-vo  Demo which shows a that you can integrate I Sense chart in jour page using an IFreme  With an HTML IFrame you cen show a website inside another website  • In the example Below we included: http://undefned/undefined/hub    Then make sure you start meteor web integration correctly.  Machine generated alternative text: : \Users ikext ernal \Documents senseweblntegration>meteor  --settings settings-production. json - -port 30B -production    (use a .bat file, that includes the meteor settings file.) |
| Does your mashup not load? | Check the virtual proxy redirect, maybe you changed the port or host of the SSO module in meteor.  Machine generated alternative text: AUTHENTICATION  Anonymous access mode  Authentication method  Authentication module redirect URI  No anonymous user  Ticket  http://2008ENT 3000/sso |
| User does not change | Make sure you logout your user in the hub. Most of the time this happens when you reset your environment. And have already tested web integration with ticketing. If you then delete all. Meteor QRS can’t logout the user and it keeps logged in. So logout manually. |
| Gray screen | Update your settings.json file of sensweb integration, it can’tt find your sense host address. Also it could be that the authentication redirect URL of the virtual proxy meteor is not correct. It should point to the QRS meteor server host and port /sso.  IFrame Integration: Hub  •Sam. t2•iik an  mat rd loo•ey ed You will QJik S•nu (the t' di•gauy Md modify like •nd  for i: ;OEiZJe we  Wt.— We in QMC: -  Strum_•)  saaswithqlik.com•s server DNS address could be found. |
| Not found, site can’t be reached | Make sure you run both server bat files  This site can't be reached  presalesl refused to connect.  Search Google for presalesl 8080 hub  ERR CONNECTION REFUSED http:((localhost:?U0/hub |
| Object id not found | Check the ui helpers.js file, and check the guids |
| Iframe app not found | Ensure you have the correct guid from the Sales.qvf file you published into the Everyone stream. |
| DIV tags and on the fly remains empty | is authenticated in QIik Sense  HTML page created, check user  O Failed to load resource: the server responded with a status  0 uncaught Er  ervices 'Shou •Service/show-seruice  htt p: . ore/docs 'errors . html#scripterror  at nakeError  at HTML ScriptEIement.onScriptError e.  O Failed to load resource: the server responded with a status  of (Not Found)  of (Not Found)  0 uncaught Er  sets/ [q  http:/'requireis.ore/docs'errors.html#scripterror  Restart your server after installation or upgrade.  Ensure you have copied the correct require.js from your "C:\Program Files\Qlik\Sense\Client\assets\external\requirejs" folder. |
| Undefined error | Press f12 – console see if you see an error like this. Most likely you have the wrong app ID in your mashup code. In the sales.qvf this is the main.html file, it refers to DemoMashup.js. you can edit the source in the extensions folder ‘C:\Shared\StaticContent\Extensions’ |
| Icons are missing | * Because of this specific MeteorJS based setup, I needed to copy some files in my project folder of Meteor. Therefore the version of Qlik Sense needs te be in sync with the version of Meteor. * Ensure you copied the fonts and requireJS to the meteor folder. * It can also be that your browser had a connection issue, try to reload the page. (and check the F12-console for errors)      * Copy our requireJS, CSS and fonts to your webserver. (if you can’t make a normal ref. in the HTML to these files, which we can’t with MeteorJS).      |  |  | | --- | --- | | RequireJS | C:\Program Files\Qlik\Sense\Client\assets\external\requirejs  Copy to: C:\...\GitHub\senseWebIntegration\client\compatibility | | CSS | C:\Program Files\Qlik\Sense\Client\autogenerated    Copy to: C:\...\GitHub\senseWebIntegration\client\stylesheets | | Fonts | Go to C:\Program Files\Qlik\Sense\Client\fonts on your Sense server.  In your external webserver create a resources/fonts directory (in your web server root) and copy the fonts. In our case that is “Copy to folder: C:\...\GitHub\senseWebIntegration\public\resources\fonts” | | You include a reference to these **local** files (to prevent cross site scripting); The require.js, the css and the fonts from above. | E.g. I have put them in the server\public folder on my external web  server.  GitHub senseWebIntegration public  resources fonts  Name  font.icons.html  Iui- icons.ttf  qlikview-sans.ttf  qlikview-sans- bold.ttf  qlikview-sans-bold italic.ttf  q likview-sans- italic. ttf    In our case: copy them to : C:\...\GitHub\senseWebIntegration\client\stylesheets | |

## Meteor issues

Start the bat file, and most of the times it suggest a workaround to solve the issue. Open a command line and execute that command (e.g. meteor npm install –save).

After you do a new sync with github

* See which error you get in the console window (F12)
  + Most of the time you miss new NPM packages
* Go to your project root folder and start command window and run: "meteor npm install --save"
* Restart your meteor server

New settings.json settings

Insert the missing parameters in the settings.json file in your project root. (you will need to do this if you get undefined for an appId for example). Look in the root folder for the example settings file.

## Bcrypt issues on windows

* Launch PowerShell as administrator
* npm install -g node-gyp
* npm install --global --production windows-build-tools
* npm install -g node-pre-gyp
* npm install -g chokidar
* From root of your app folder:
* meteor npm install --save bcrypt

<https://forums.meteor.com/t/how-to-fix-the-bcrypt-error-in-windows-7/28856/25>

## Hardcore meteor resets

* Stop your project, and type: meteor reset, then try to start your app again. This will remove all data from your database.
* In your *existing* application, remove node\_modules completely and run meteor npm install to rebuild it. Then see if the project works.
* Try a meteor reset which will wipe away many of the caches and things which could be broken. Please note that this will delete your development database. To avoid this, you can accomplish a similar reset by removing all directories in your .meteor/local/ directory except for db.
* Ultimately, try completely reinstalling Meteor.
  + For Windows, [use these instructions](https://github.com/meteor/meteor/issues/7524#issuecomment-236166061)
  + For OS X or Linux, you can simply run: curl <https://install.meteor.com/> | sh

From <https://github.com/meteor/meteor/issues/7766>>

# Appendix 3: Captcha security

This site uses a proxy server (included in QRS meteor on port 80), and we configured [Sikka](https://github.com/meteorhacks/sikka) as a firewall for both Meteor projects. The only thing you need to do is to Visit [Google's Recaptcha](https://www.google.com/recaptcha/intro/index.html)website to get Captcha keys for your domain. We've added a default set of keys works on locally, meteor.com and onmodulus.net to make your development experience simpler. But, you should get a new pair of keys for a production deployment.

Put your new keys in the settings.json file in your project root.

{

"sikka": {

"captcha": {

"siteKey": "your\_new\_site\_key",

"secret": "your\_new\_secret\_key"

}

}

}

# Appendix 4: Background documentation

## Login screen, User acount creation (not used at the moment)

If you login into QRS Meteor, you can edit the files below to change the behavior.

* You can play around with enabling facebook and google authentication
* Or activate client side account creation (imports\startup folder)

AccountsTemplates.configure({

confirmPassword: false,

enablePasswordChange: true,

forbidClientAccountCreation: true,

overrideLoginErrors: true,

sendVerificationEmail: false,

lowercaseUsername: false,

focusFirstInput: true,

## Change the look and feel using Semantic UI (for QRSMETEOR)

[Semantic](http://semantic-ui.com) is like Bootstrap. It helps with the look and feel of the website. (If you miss symbols, or the site does not look well, execute the things below)

After you moved meteor to another machine, you might have to reconfigure semantic UI

|  |  |
| --- | --- |
| Stop meteor (press control C 2x, or close the command window) | { group:  ' CONTRIB  entered server side rel  ' Linda'  Attributes :  { group:  'SUNNY P  { group:  ' TED  { group:  ' CONTRIB  The passport for reque  ' Linda'  Attributes :  { group:  'SUNNY P  { group:  ' TED  { group:  ' CONTRIB  Target I d:  ' 6dc335c?-I  Meteor server side cre,  erminate  batch |
| Remove the .custom.semantic.json in your project folder of qrs meteor | FOLDERS  V B QRSMeteor  .meteor  V Bclient  compatibility  V Blib  definitions  site  themes  on  custom.semantic.json  semantic.less  B theme.config.import.less  theme.import.less  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  'colorize" :  true,  ' comment "  true,  'container": true,  "dimmer": true,  "divider" :  true,  "dropdown " :  true,  "embed" :  "feed" :  "flag" :  "form" .  'grid" :  true,  true,  true,  true,  true,  "header": true,  icon"  ' image "  'input "  'item" .  "label " :  " list" :  true,  true,  true,  true,  true,  true,  "loader": true, |

* Restart meteor, this will ensure all semantic things work again
* Go to the project root (QRSMeteor)
  + Start command window
  + Meteor npm install --save
  + C:\C73B9CA5\5BBCB0A6-7DF1-4165-B4DC-97134F1B25ED_files\image028.png

## Change layout (optional for QRSMETEOR)

|  |  |
| --- | --- |
| If you like to adjust the site theming you can look here | FOLDERS  B QRSMeteor  .meteor  B client  compatibility  B lib  definitions  B Site  collections  elements  B globals  eset. overrides. import. less  eset.variables. import. less  Site. overrides. import. less  site.variables.import.less    User Global Variables  @primaryCoIor : #5EA819;  @green : #5EA819;  *range: #F8951C;  *ray: #F2F2F2; |

## Example project to run your first meteor webserver and mongoDB

|  |  |
| --- | --- |
| If you want to get your first hello world example with meteor.js, then try this. | Next you can start your browser and go to <http://localhost:3000> and see your new app! |
| Want to learn to build your first site? Try the meteor todo app (blaze) | <https://www.meteor.com/tutorials/blaze/creating-an-app> |
| If you would like to see the end result. Go to a directory on your pc, and execute this code, to create all html, js, and css files for you. | Meteor create --example todos |

## Install meteor workaround

* Install [7-Zip](http://www.7-zip.org/) or any other program that knows how to extract tar.gz files.
* Download the installation archive from <https://packages.meteor.com/bootstrap-link?arch=os.windows.x86_32>.
* In a command prompt, run echo %LocalAppData%\.meteor -- this is the directory in which Meteor should be installed.
* Extract the installation archive into the directory above.
* Add this directory to your PATH environment variable.
* You should now be able to open a new command prompt and run meteor. Some versions of Windows may require restarting your machine first.

From <<https://github.com/meteor/docs/blob/version-NEXT/long-form/alternate-windows-installation.md>>

* Remove unsecure

Remove the package “insecure” for production scenarios, and implement allow [deny server   
 side methods](https://www.discovermeteor.com/blog/allow-deny-challenge-results/). Otherwise the client can do database operations.

## Setup login using social accounts

|  |  |
| --- | --- |
| Enable the code in server\accounts.js | // ServiceConfiguration.configurations.upsert({  // service: "facebook"  // }, {  // $set: {  // appId: Meteor.settings.private.facebook.clientId,  // loginStyle: "popup",  // secret: Meteor.settings.private.facebook.secret  // }  // });  Etc.. |
| Ensure you update your settings.json file and create the needed OAUTH API keys in for example facebook, google etc. | "google": {  "clientId": "",  "secret": ""  },  "facebook": {  "clientId": "",  "secret": ""  },  "twitter": {  "clientId": "",  "secret": ""  },  "linkedin": {  "clientId": "",  "secret": ""  },  "github": {  "clientId": "",  "secret": ""  } |