



# Qvu

## Getting Started

1. Ensure java 8 or higher is installed on the server that will run Qvu.
2. Download the Qvu application from <http://myqvu.com>
3. Create the server-side folder that will house the Qvu repository.
4. Start the Qvu application by running the following command:  

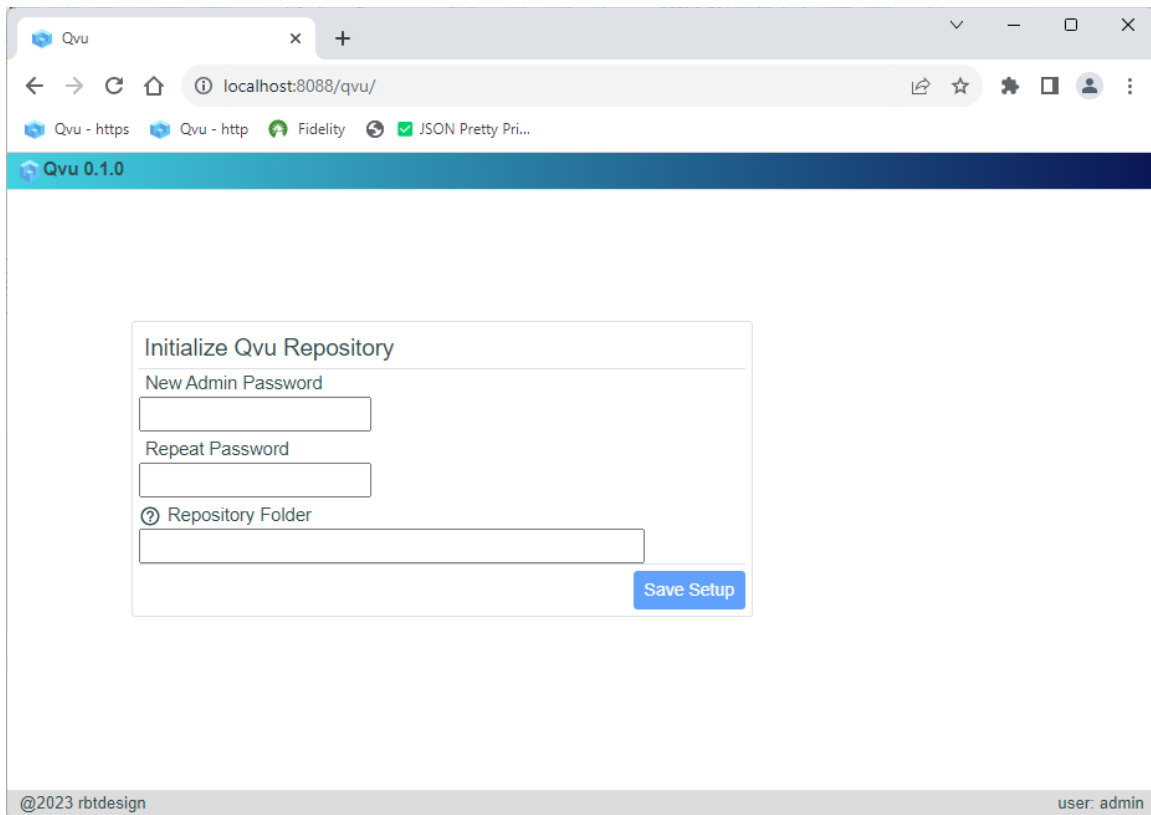
```
java -jar qvu.jar
```
5. Once the application starts, pull up the initialization page by going to  
<http://localhost:8088/qvu>

and logging in with:

username: admin

password: admin

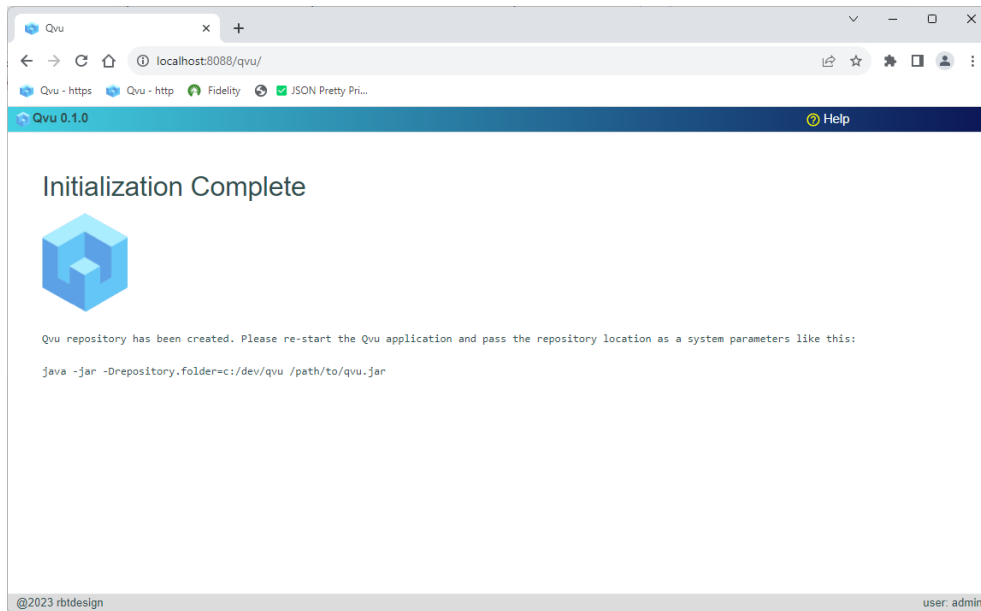
- the initialization screen shown below should display:



The screenshot shows a web browser window with the address bar set to `localhost:8088/qvu/`. The page title is "Qvu 0.1.0". The main content area displays a form titled "Initialize Qvu Repository". The form contains three input fields: "New Admin Password", "Repeat Password", and "Repository Folder" (which has a help icon to its left). A blue "Save Setup" button is located at the bottom right of the form. The footer of the page shows "@2023 rbtdesign" on the left and "user: admin" on the right.

6. Enter a new admin password and the repository folder created in step 3.

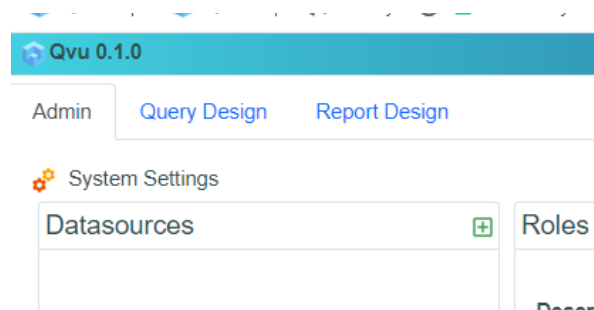
- Click “Save Setup” – if Qvu was successfully initialized you should see a message similar to the following:



- Once the initialization complete screen displays, stop the application and restart passing the repository folder location as a system parameter as follows:  

```
java -jar -Drepository.folder=/my/repository/location qvu.jar
```
- Pull up the application at <http://localhost:8088/qvu>, login with username=admin and password=<new password> and you should see the administration page displayed:

By default basic authentication is used and users and roles are stored as json in the file <repository.folder>/config/qvu-security.json. Qvu supports SAML, OIDC and Basic authentication. You can change this setup by clicking the “System Settings” icon in the admin tab.



Default Security Type: basic

Basic SAML Oidc

? \*IDP URL

? \*SP Entity ID

☐ Sign Assertions

? Signing Cert File

? Signing Key File

☐ Enabled

\*indicates required field

Cancel Save

Once changed you will have to restart the application. You can also implement your own customized security. See the help documentation for more information on this process.

If you wish to enable SSL, modify the <repository.folder>/config/application.properties file then uncomment and add values for the following properties:

#ssl - complete entries below to use https

#server.ssl.key-store=

#server.ssl.key-store-type=

#server.ssl.key-alias=

#server.ssl.key-store-password=

#server.ssl.key-password=

Again, you will have to restart the application. You will probably also want to change the port property at the same time:

server.port=8443

08/24/2023

