Quickly Quasar

Quasar, a Vue Widget Framework

—

Haris Hashim

© 2018 harishashim@gmail.com

Overview

A step by step training manual style instruction to accompany mentor and mentee discussion and learning session. This is a simple introduction to Quasar.

IMPORTANT This document does not include explanation that is done during one to one mentorship session.

Table of Contents

Part 1: Getting Started

Prerequisite

1. VS Code installed
2. Node JS and NPM installed

Steps

1. Install Quasar command line interface by executing in command prompt

   ```
   npm install -g quasar-cli
   ```
2. List all Quasar starter kit just for giggles

   ```
   quasar list
   ```
3. Install the default starter kit

   ```
   quasar init quickly
   ```
4. Follow the suggestion by changing directory and executing

   ```
   npm install

    quasar dev
   ```
5. Try to understand the codes
   a. src/main.js
   b. src/router.js
   c. src/components/Hello.vue
   d. What is templates folder?
6. Execute this to relate and understand point (5.d) above

   ```
   quasar new --list
   ```
7. We will replace the default Hello.vue by doing the following
   a. Create a new .vue file by doing

   ```
   quasar new layout HelloWorld
   ```

   b. Change the route for path '/' to point to HelloWorld

   c. Save file to see hot reload in action.
8. Change title in HelloWorld.vue to "Hello World!", save file to see hot reload in action.

Exercise

1. Add any type of button and properly label it as "OK". Check documentation at this URL for button. And this URL about adding component.
2. Do code so that when the "OK" button is pressed, title change from "Hello World!" to "Bye!".

Part 2: Layout, Declaring Data, and Handling Click

Prerequisite

This part is continuing part 1 and as the answer to previous exercise

Workaround

There is lots of problem with eslint and code formatter provided plugin in VS Code. The only solution (at this point of time) is to disable eslint.

1. Open build/webpack.base.conf.js
2. Comment out the following codes

```
36    module: {
37      rules: [
38        // { // eslint
39        //   enforce: 'pre',
40        //   test: /\.(vue|js)$/,
41        //   loader: 'eslint-loader',
42        //   include: projectRoot,
43        //   exclude: /node_modules/,
44        //   options: {
45        //     formatter: require('eslint-friendly-formatter')
46        //   }
47        // },
48        {
49          test: /\.js$/,
50          loader: 'babel-loader',
51          include: projectRoot,
52          exclude: /node_modules/
53        },
```

3. If quasar dev already started, restart by doing CTRL-C and execute the command again.

Steps

1. **Layout**. Change the layout to add a button.
   a. Put below codes somewhere on top of

```
53
54    <div>
55      <q-btn round color="secondary" @click="btnClick">
56        <q-icon name="card_giftcard" />
57      </q-btn>
58    </div>
59
60    <router-view />
61
```

b. Take note that if saved, no button will be shown. Since q-btn and q-icon tag need to be defined first. To do that, add below codes in <script>

```
70   <script>
71     import {
72       QBtn,
73       QIcon
74     } from 'quasar'
75
76     export default {
77       components: {
78         QBtn,
79         QIcon
80       },
81       data() {
```

c. Save and check in browser that button is shown below the title.

2. **Declaring Variable.** Rather than "hard coding" the title. We will use a variable.

   a. Add title variable to data() with previously specified value.

```
81       data() {
82         return {
83           title: 'Hello there, world!'
84         }
85       }
86     }
87   </script>
```

   b. Add the variable to the layout.

```
10         <q-toolbar-title>
11           {{title}}
12         </q-toolbar-title>
13       </q-toolbar>
14
```

3. **Handling Click**. Nothing will happen if button is clicked now. We need to code the handler.

   a. Add methods section and btnClick function to <script>. The code will toggle title to "Bye!" just so that we can see change and verify that it works.

```
75       data() {
76         // ...
77       },
78       methods: {
            Complexity is 3 Everything is cool!
79         ■ btnClick() {
80           if (this.title == 'Bye!') {
81             this.title = 'Hello there, world!'
82           } else {
83             this.title = 'Bye!'
84           }
85         }
86       }
87     }
88   </script>
```

   b. Save and test the button.