**Section 40 – The Stock Trader**

Chapter 608 – Chapter 632

1. Project Setup

Npm install –save-dev babel-preset-stage-2

In .babelrc add this code

[“stage-2”]

1. Determine all the components, give the name using template.

<template>

    <h1>The Stocks Component</h1>

</template>

1. Determine the routes.js

Npm install –save vue-router

1. We create data dummy for stock in Stocks.vue

<template>

    <div>

        <app-stock v-for="stock in stocks" :stock="stock"></app-stock>

    </div>

</template>

<script>

import Stock from './Stock.vue';

export default {

    data() {

        return {

            stocks: [

                { id: 1, name: 'BMW', price: 110 },

                { id: 2, name: 'Google', price: 200 },

                { id: 3, name: 'Apple', price: 250 },

                { id: 4, name: 'Twitter', price: 8 },

            ]

        }

    },

    components: {

        appStock: Stock

    }

}

</script>

1. Get and print the data in Stock.vue

<template>

    <div class="panel panel-success">

        <div class="panel-heading">

            <h3 class="panel-title">

                {{ stock.name }}

                <small>(Price: {{ stock.price }})</small>

            </h3>

        </div>

        <div class="panel-body">

            <div class="pull-left">

                <input

                    type="number"

                    class="form-control"

                    placeholder="Quantity"

                    v-model="quantity"

                    >

            </div>

            <div class="pull-right">

                <button

                        class="btn btn-success"

                        @click="buyStock"

                        :disabled="quantity <= 0 || !Number.isInteger(quantity)"

                        >Buy

                </button>

            </div>

        </div>

    </div>

</template>

<script>

export default {

    props: ['stock'],

    data() {

        return {

            quantity: 0

        }

    },

    methods: {

        buyStock() {

            const order = {

                stockId: this.stock.id,

                stockPrice: this.stock.price,

                quantity: this.quantity

            }

            console.log(order);

            this.quantity = 0;

        }

    }

}

</script>

1. Create Header.vue

<template>

    <nav class="navbar navbar-default">

        <div class="container-fluid">

            <div class="navbar-header">

                <router-link to="/" class="navbar-brand">Stock Trader</router-link>

            </div>

            <div class="collapse navbar-collapse">

                <ul class="nav navbar-nav">

                    <router-link to="/portfolio" activeClass="active" tag="li"><a>Portfolio</a></router-link>

                    <router-link to="/stocks" activeClass="active" tag="li"><a>Stocks</a></router-link>

                </ul>

                <ul class="nav navbar-nav navbar-right">

                    <li><a href="#" @click="endDay">End Day</a></li>

                    <li class="dropdown">

                        <a

                                href="#"

                                class="dropdown-toggle"

                                data-toggle="dropdown"

                                role="button"

                                aria-haspopup="true"

                                aria-expanded="false">Save & Load <span class="caret"></span></a>

                        <ul class="dropdown-menu">

                            <li><a href="#">Save Data</a></li>

                            <li><a href="#">Load Data</a></li>

                        </ul>

                    </li>

                </ul>

            </div><!-- /.navbar-collapse -->

        </div><!-- /.container-fluid -->

    </nav>

</template>

1. Install vuex as a development, as a production dependency.

Npm install --save vuex

1. Create actions, mutations, state, and getters.

import stocks from '../../data/stocks';

const state = {

    stocks: []

};

const mutations = {

    'SET\_STOCKS' (state, stocks) {

        state.stocks = stocks;

    },

    'RND\_STOCKS' (state) {

    }

};

const actions = {

    buyStock: ({commit}, order) => {

        commit();

    },

    initStocks: ({commit}) => {

        commit('SET\_STOCKS', stocks);

    },

    randomizeStocks: ({commit}) => {

        commit('RND\_STOCKS');

    }

}

const getters = {

    stocks: state => {

        return state.stocks;

    }

};

export default {

    state,

    mutations,

    actions,

    getters

};