**Section 37 – Connecting To Servers Via HTTP**

Chapter 543 – Chapter 555

1. Put vs Post

Put will override the data

Post will create a new data

1. Final Result

Main.JS

import Vue from 'vue'

import VueResource from 'vue-resource';

import App from './App.vue'

Vue.use(VueResource);

Vue.http.options.root = 'https://vuejs-http.firebaseio.com/';

Vue.http.interceptors.push((request, next) => {

  console.log(request);

  if (request.method == 'POST') {

    request.method = 'PUT';

  }

  next(response => {

    response.json = () => { return {messages: response.body} }

  });

});

new Vue({

  el: '#app',

  render: h => h(App)

})

App.vue

<template>

    <div class="container">

        <div class="row">

            <div class="col-xs-12 col-sm-8 col-sm-offset-2 col-md-6 col-md-offset-3">

                <h1>Http</h1>

                <div class="form-group">

                    <label>Username</label>

                    <input class="form-control" type="text" v-model="user.username">

                </div>

                <div class="form-group">

                    <label>Mail</label>

                    <input class="form-control" type="text" v-model="user.email">

                </div>

                <button class="btn btn-primary" @click="submit">Submit</button>

                <hr>

                <input class="form-control" type="text" v-model="node">

                <br><br>

                <button class="btn btn-primary" @click="fetchData">Get Data</button>

                <br><br>

                <ul class="list-group">

                    <li class="list-group-item" v-for="u in users">{{ u.username }} - {{ u.email }}</li>

                </ul>

            </div>

        </div>

    </div>

</template>

<script>

    export default {

        data() {

            return {

                user: {

                    username: '',

                    email: ''

                },

                users: [],

                resource: {},

                node: 'data'

            };

        },

        methods: {

            submit() {

//                this.$http.post('data.json', this.user)

//                        .then(response => {

//                            console.log(response);

//                        }, error => {

//                            console.log(error);

//                        });

//                this.resource.save({}, this.user);

                this.resource.saveAlt(this.user);

            },

            fetchData() {

//                this.$http.get('data.json')

//                        .then(response => {

//                            return response.json();

//                        })

//                        .then(data => {

//                            const resultArray = [];

//                            for (let key in data) {

//                                resultArray.push(data[key]);

//                            }

//                            this.users = resultArray;

//                        });

                this.resource.getData({node: this.node})

                        .then(response => {

                            return response.json();

                        })

                        .then(data => {

                            const resultArray = [];

                            for (let key in data) {

                                resultArray.push(data[key]);

                            }

                            this.users = resultArray;

                        });

            }

        },

        created() {

            const customActions = {

                saveAlt: {method: 'POST', url: 'alternative.json'},

                getData: {method: 'GET'}

            };

            this.resource = this.$resource('{node}.json', {}, customActions);

        }

    }

</script>

<style>

</style>