

Using Vue with ASP.NET Core By Example

Lab 6 Instructions

Shawn Wildermuth, Instructor

1. Add Vue-Router to the Project
    a. Open the project in Visual Studio again.
    b. Find the **package.json** and open it.
    c. Add **vue-router** to the list of dependencies:

```
    "vee-validate": "2.0.8",
    "vuex": "3.0.1",
    "vue-router": "3.0.1"
  }
```

    d. Find the **/Views/Root/Index.cshtml** file.
    e. Add **vue-router** to the list of scripts:

```
@section scripts {
  <script src="~/node_modules/vue/dist/vue.js"></script>
  <script src="~/node_modules/axios/dist/axios.js"></script>
  <script src="~/node_modules/vuex/dist/vuex.js"></script>
  <script src="~/node_modules/vue-router/dist/vue-router.js"></script>
  <script src="~/js/store.js"></script>
  <script src="~/js/cart.js"></script>
  <script src="~/js/index.js"></script>
}
```

    f. Create a new file in the **wwwroot/js** directory called **app.js**
    g. In the new file, call **Vue.use** with **VueRouter** object:

```
// app.js
Vue.use(VueRouter);
```

    h. Back in the **index.cshtml** file, add the **app.js** as the last file in the list of scripts:

```
@section scripts {
  <script src="~/node_modules/vue/dist/vue.js"></script>
  <script src="~/node_modules/axios/dist/axios.js"></script>
  <script src="~/node_modules/vuex/dist/vuex.js"></script>
  <script src="~/node_modules/vue-router/dist/vue-router.js"></script>
  <script src="~/js/store.js"></script>
  <script src="~/js/cart.js"></script>
  <script src="~/js/index.js"></script>
  <script src="~/js/app.js"></script>
}
```

    i. Now you're ready to start creating your SPA.

2. Create Landing Route
    a. Open the **app.js** if it's not open.
    b. Add a new **array** called '**routes'**.
    c. Create the first element of the array as an object with two properties: **path** and **component**.
    d. Set the properties to **"/"** and a new component called **ProductList**:

```
// app.js
Vue.use(VueRouter);

let routes = [
  { path: "/", component: ProductList }
];
```

    e. After the routes, create a new instance of **Vue** with an object for the settings.
    f. Set the **el** to **"#index-view"** like we did for the old **index.js** file:

```
// app.js
Vue.use(VueRouter);

let routes = [
  { path: "/", component: ProductList }
];

var vm = new Vue({
  el: "#index-view"
});
```

    g. Add a new member of **Vue** called **store** to inject the store into the Vue and all child components:

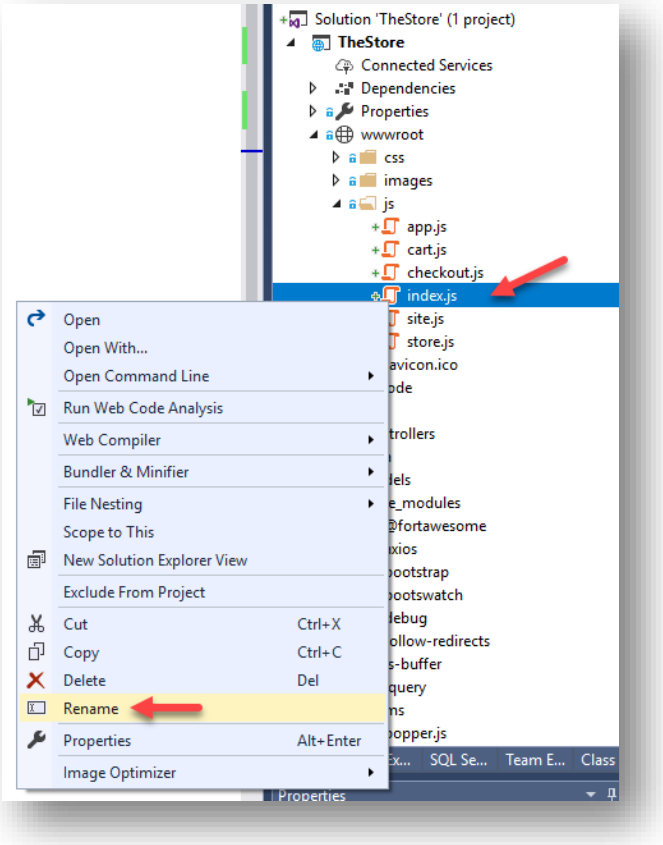
```
var vm = new Vue({
  el: "#index-view",
  store,
});
```

    h. Add a new member called '**router'** and assign it a **new** instance of **VueRouter** with an object with **routes** inside it as the parameter:

```
var vm = new Vue({
  el: "#index-view",
  store,
  router: new VueRouter({ routes })
});
```

    i. Now you're ready to create your first 'view'.

3. Create First View
    a. Find the **index.js** file in the Solution View.
    b. Right-click and pick rename. Rename it to **productList.js**:



    c. Open the **index.cshtml** file again, and rename the **index.js** to **productList.js** too:

```
@section scripts {
  <script src="~/node_modules/vue/dist/vue.js"></script>
  <script src="~/node_modules/axios/dist/axios.js"></script>
  <script src="~/node_modules/vuex/dist/vuex.js"></script>
  <script src="~/node_modules/vue-router/dist/vue-router.js"></script>
  <script src="~/js/store.js"></script>
  <script src="~/js/cart.js"></script>
  <script src="~/js/productList.js"></script>
  <script src="~/js/app.js"></script>
}
```

d. Open the newly renamed **productList.js** file.
e. Change the call to **new Vue** with a call to **Vue.component** using **"product-list"** as the component name.
f. Store the return value from **Vue.component** in a new variable called **ProductList**:

```
// productList.js
let ProductList = Vue.component("product-list", {
  el: "#index-view",
  store,
  data: {
    appName: "Product List",
    error: ""
  },
```

g. Remove the **el** and **store** properties.
h. Change the **data** property to be a **function** that returns the data:

```
// productList.js
let ProductList = Vue.component("product-list", {
  data: function () {
    return {
      appName: "Product List",
      error: ""
    };
  },
```

i. Add a new member called **template** and assign it an empty string with the backwards quote `` so you can have a multi-line string.
j. Inside the backwards quotes, create a **div**:

```
// productList.js
let ProductList = Vue.component("product-list", {
  template: `<div>
</div>`,
  data: function () {
    return {
      appName: "Product List",
      error: ""
    };
  },
```

k. Open the **index.cshtml** file.

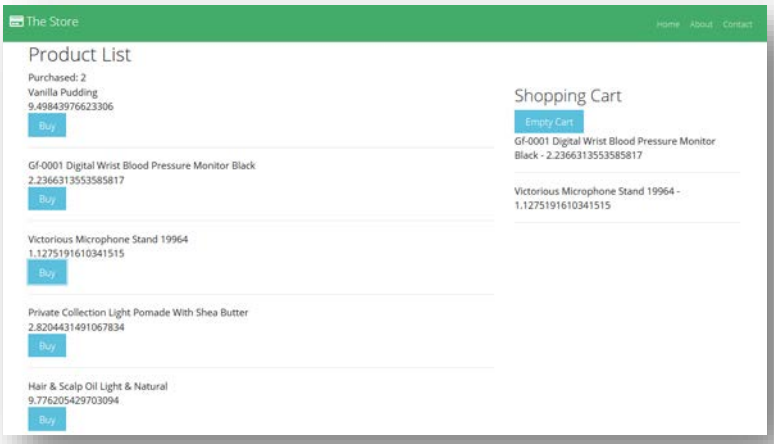
l. Cut everything inside the **#index-view div** and paste it inside the **productList.js** template **divs**:

```
// productList.js
let ProductList = Vue.component("product-list", {
  template: `<div>
  <h2>{{ appName }}</h2>
  <div>Purchased: {{ $store.state.cart.length }}</div>
  <div class="row">
    <div class="col-md-8">
      <div v-for="p in $store.state.products">
        <div>{{ p.name }}</div>
        <div>{{ p.description }}</div>
        <div>{{ p.listPrice }}</div>
        <button class="btn btn-info" v-on:click="onBuy(p)">Buy</button>
        <hr />
      </div>
    </div>
    <div class="col-md-4">
      <the-cart v-bind:items="$store.state.cart" v-on:empty-cart="onEmptyCart()">
      </the-cart>
    </div>
  </div>
</div>`,
```

m. Back in the **index.cshtml**, add a **<router-view>** inside the **#index-view div**:

```
<div id="index-view">
  <router-view></router-view>
</div>
```

n. Run the project to see the first page works now:



4. Create Second View
    a. Open the **checkout.js** file.
    b. Cut the call to use **VeeValidate** and move it to **app.js**:

```
// app.js
Vue.use(VueRouter);
Vue.use(VeeValidate);
```

    c. Open the **index.cshtml** file.
    d. Add **vee-validate** to the list of scripts.
    e. Add **checkout.js** to the list of scripts too:

```
@section scripts {
  <script src="~/node_modules/vue/dist/vue.js"></script>
  <script src="~/node_modules/axios/dist/axios.js"></script>
  <script src="~/node_modules/vuex/dist/vuex.js"></script>
  <script src="~/node_modules/vue-router/dist/vue-router.js"></script>
  <script src="~/node_modules/vee-validate/dist/vee-validate.js"></script>
  <script src="~/js/store.js"></script>
  <script src="~/js/cart.js"></script>
  <script src="~/js/checkout.js"></script>
  <script src="~/js/productList.js"></script>
  <script src="~/js/app.js"></script>
}
```

    f. Open the **checkout.js** file.
    g. Like we did with the **productList**, change the call to **new Vue** to use **Vue.component**:

```
// checkout.js
let Checkout = Vue.component("checkout", {
  el: "#theForm",
  data: {
    customer: {}
  },
  methods: {
    onSave: function () {
      alert(JSON.stringify(this.customer));
    }
  }
});
```

h. Like with the **productList**, remove the **el** member, and change the **data** member to a **function** that returns the **data**:

```
// checkout.js
let Checkout = Vue.component("checkout", {
  data: function () {
    return {
      customer: {}
    };
  },
  methods: {
    onSave: function () {
      alert(JSON.stringify(this.customer));
    }
  }
});
```

i. Add a new member called **template** with the **backwards quotes** and include an empty **div** to contain the **template**:

```
// checkout.js
let Checkout = Vue.component("checkout", {
  template: `<div>
</div>`,
  data: function () {
    return {
      customer: {}
    };
  },
```

j. Open the **checkout.cshtml** file in the **Views/Root** folder.
k. Copy the entire **form** element.
l. Paste it in the **checkout.js** inside the **div** in the **template**:

```
// checkout.js
let Checkout = Vue.component("checkout", {
  template: `<div>
  <form v-on:submit.prevent="onSave()">
    <div class="form-group">
      <label>Name</label>
      <input class="form-control"
             v-model="customer.name"
             name="customerName"
             v-validate="{ required: true }" />
      <span class="text-danger">{{ errors.first("customerName") }}</span>
    </div>
    <div class="form-group">
      <label>Email</label>
      <input class="form-control"
             v-model="customer.email"
             name="email"
             v-validate="{ required: true, email: true }" />
      <span class="text-danger">{{ errors.first("email") }}</span>
    </div>
    <div class="form-group">
      <input type="submit" class="btn" value="Save" />
    </div>
  </form>
</div>`,
  data: function () {
    return {
      customer: {}
    };
  },
  methods: {
    onSave: function () {
      alert(JSON.stringify(this.customer));
    }
  }
});
```

m. Open the **app.js**.

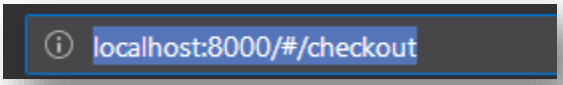
n. Add a new route for **/checkout** with the component name of **Checkout**:

```
// app.js
Vue.use(VueRouter);
Vue.use(VeeValidate);

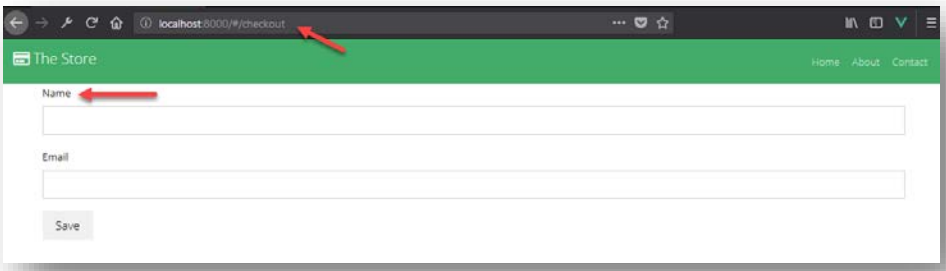
let routes = [
  { path: "/", component: ProductList },
  { path: "/checkout", component: Checkout }
];
```

o. Run the project.

p. When you open it, change the Url to …#/checkout and see if the checkout shows up:



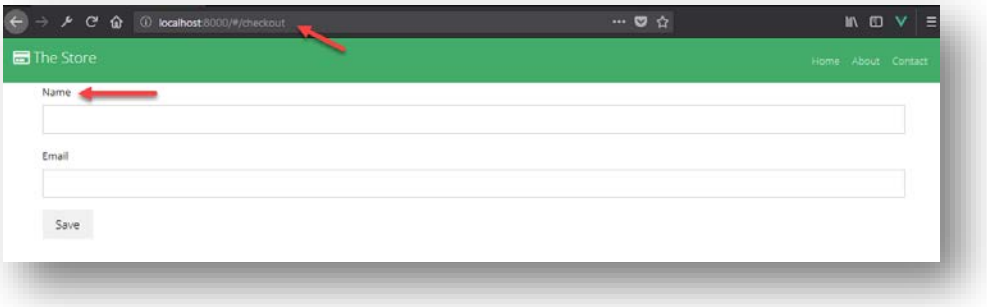
q. Should show the Checkout page:



5. Use router-link to Switch Between Views
    a. Open the **cart.js** file.
    b. In the template, add a new line after the "**Empty Cart**" line.
    c. Add a **<router-link>** specifying the **to** attribute to **/checkout**.
    d. Add **class** to the **router-link** for "**btn btn-success**".
    e. Inside the **router-link**, type **Checkout**:

```
// cart.js
Vue.component("the-cart", {
  template: `
<div>
  <h3>Shopping Cart</h3>
  <button class='btn btn-info' v-on:click="emptyCart">Empty Cart</button>
  <router-link to="/checkout" class="btn btn-success">Checkout</router-link>
  <div v-for="i in items">
    <div>{{ i.name }} - {{ i.listPrice }}</div>
    <hr/>
  </div>
</div>
`,
```

    f. Run the project.
    g. If you aren't at the home page, click the name of the store at the top-left to return you to the home page.
    h. Click on the Checkout button and watch it switch to the checkout page:



i. Return to the **checkout.js** file.
j. Add a new **router-link** before the **submit** button.
k. Set the inside of the **router-link** to say "**Cancel**".
l. Set it's **to** attribute to **'/'** and add a class for **'btn btn-info'** to make it look like a button:

```
// checkout.js
let Checkout = Vue.component("checkout", {
  template: `<div>
  <form v-on:submit.prevent="onSave()">
    <div class="form-group">
      <label>Name</label>
      <input class="form-control"
             v-model="customer.name"
             name="customerName"
             v-validate="{ required: true }" />
      <span class="text-danger">{{ errors.first("customerName") }}</span>
    </div>
    <div class="form-group">
      <label>Email</label>
      <input class="form-control"
             v-model="customer.email"
             name="email"
             v-validate="{ required: true, email: true }" />
      <span class="text-danger">{{ errors.first("email") }}</span>
    </div>
    <div class="form-group">
      <router-link to="/" class="btn btn-info">Cancel</router-link>
      <input type="submit" class="btn" value="Save" />
    </div>
  </form>
</div>`,
```

m. Run the project again.
n. You should be able to flip back and forth between the two views.
o. Try adding items to the cart and see that they persist between the two views:

