

Route Outlets & Links

NEXT: NESTED & DYNAMIC ROUTES

We have just created a route in the previous section. The route is not so useful yet because we are yet to mount it in the browser. That's where the Route Outlet comes in.

#Route Outlets

These are entry points where Vue can mount the configured routes. A route outlet is simply a component:

```

<!-- ./src/App.vue -->
<template>
  <div id="app">
    

    <!-- Route Outlet -->
    <router-view></router-view>
  </div>
</template>

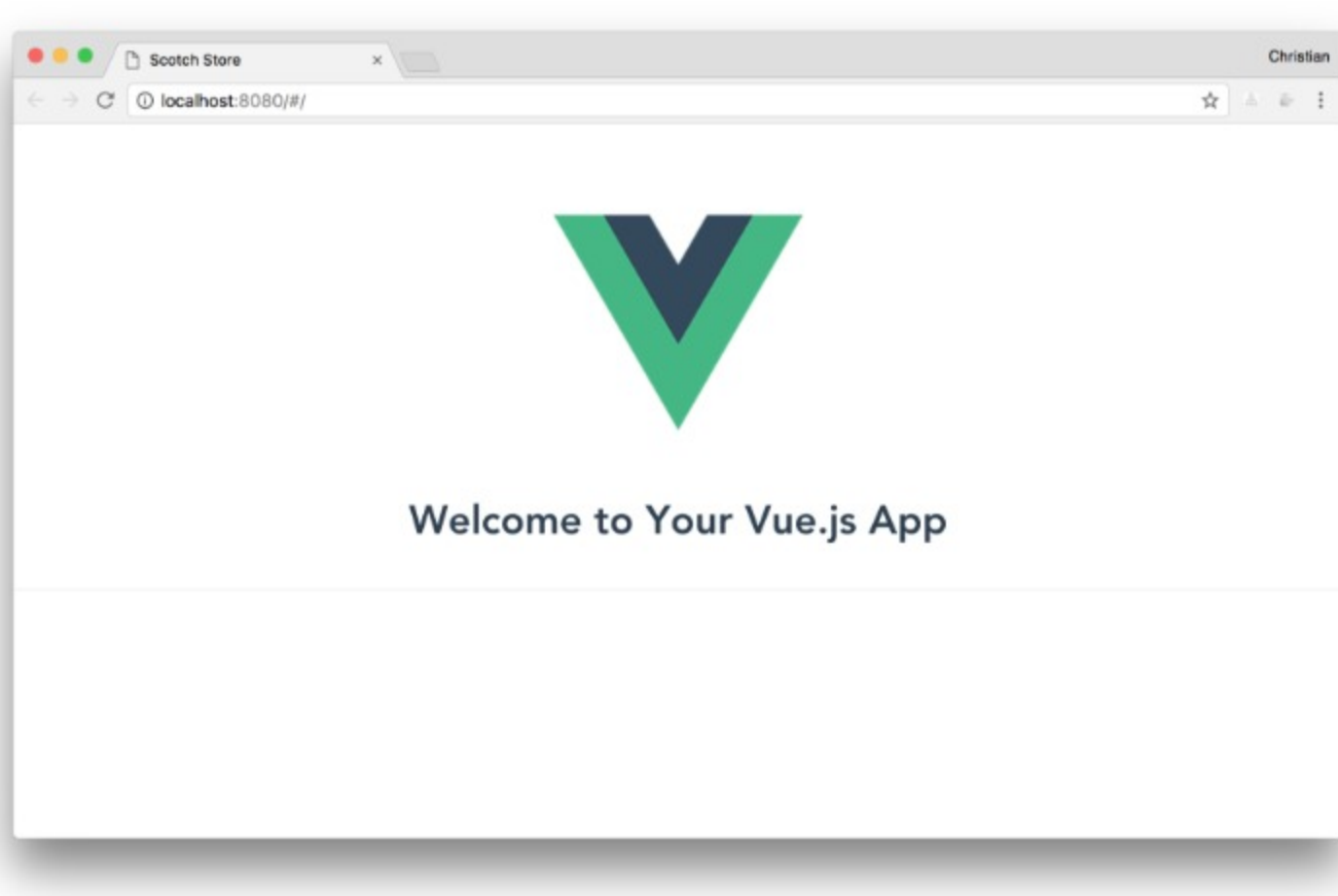
<script>
export default {
  //...
};
</script>

```

At this point, if you run your app, you will be served with the Home page route. It should look fairly like the image below:

Table of Contents

- 1 Route Outlets
- 2 Route Links



The styles are truncated from all code samples to make it easier to navigate and digest the code. You can always view these styles from the demo repository.

#Route Links

Navigation is a primary characteristic of routing. Let's create more routes and use Vue's built-in navigation component to access these routes.

Before we do that, move the existing `Home` component from `components` folder to a `pages` folder. This is so that the `components` folder can contain only components that have a tendency to be reused while `pages` folder contains components that are used with routes.

Make sure that you update the link to that in the `./src/router/index.js` file. Next up we'll create our other pages.

```

<!-- ./src/pages/Admin.vue -->
<template>
  <div>
    <div class="title">
      <h1>{{msg}}</h1>
    </div>
  </div>
</template>

<script>
export default {
  name: 'home',
  data () {
    return {
      msg: 'Welcome to the Admin Page'
    }
  }
}
</script>

```

```

<!-- ./src/pages/Cart.vue -->
<template>
  <div>
    <div class="title">
      <h1>{{msg}}</h1>
    </div>
  </div>
</template>

<script>
export default {
  name: 'home',
  data () {
    return {
      msg: 'Welcome to the Cart Page'
    }
  }
}
</script>

```

Just two basic components with welcome pages that differ. Let's add them to our routes config:

```

// ./src/router/index.js
import Vue from 'vue';
import Router from 'vue-router';

// Moved to pages from components
import Home from '@pages/Home';
import Cart from '@pages/Cart';
import Admin from '@pages/Admin';

Vue.use(Router);

export default new Router({
  routes: [
    {
      path: '/',
      name: 'Home',
      component: Home,
    },
    {
      path: '/admin',
      name: 'Admin',
      component: Admin,
    },
    {
      path: '/cart',
      name: 'Cart',
      component: Cart,
    }
  ]
});

```

Then you can add navigation links to take us to the pages:

```

<!-- ./src/App.vue -->
<template>
  <div>
    <nav>
      <div class="container">
        <ul class="nav_left">

          <!-- Navigation Links -->
          <li><router-link to="/">Home</router-link></li>
          <li><router-link to="/admin">Admin</router-link></li>
          <li><router-link to="/cart">Cart</router-link></li>

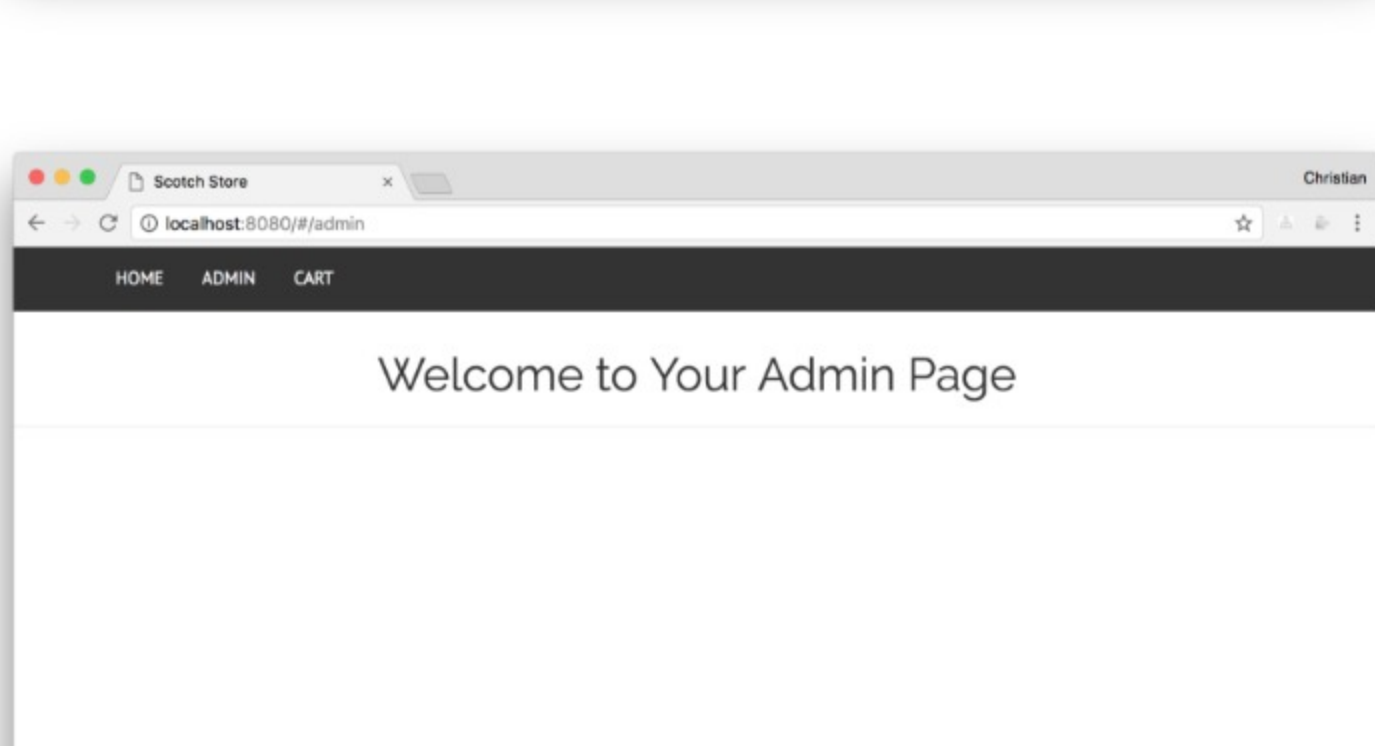
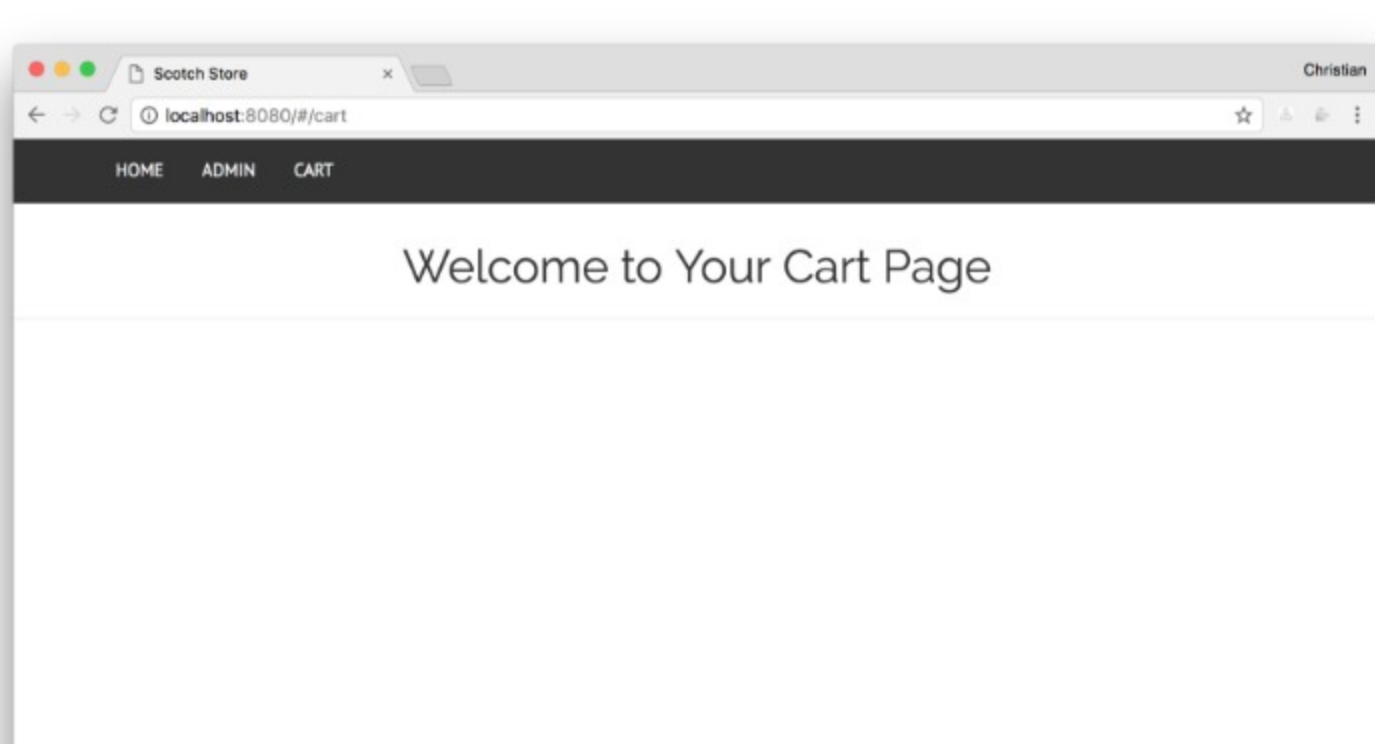
        </ul>
      </div>
    </nav>

    <router-view></router-view>
  </div>
</template>

...

```

Rather than the `a` tag, we use Vue's `router-link` component. `router-link` is just a wrapper for the `a` tag with some optimization logic.



Getting Started

- 1 Introduction [Free](#)
- 2 Tools [Free](#)
- 3 Our Task [Free](#)
- 4 Hello World [Free](#)

Vue Basics

- 5 Vue CLI [Free](#)
- 6 Components and the Vue Instance [Free](#)
- 7 Template Syntax [Free](#)
- 8 Conditional & List Rendering [Free](#)
- 9 Styles & Classes [Free](#)

Routing

- 10 Single Page Apps [Free](#)
- 11 Creating Routes [Free](#)
- 12 Route Outlets & Links [Free](#)
- 13 Nested & Dynamic Routes [Free](#)

Forms

- 14 Two Way Binding (Reactivity) [Free](#)
- 15 Form Validation [Free](#)
- 16 Handling Form Submissions [Free](#)

API Backend

- 17 Prelude [Free](#)
- 18 Setup [Free](#)
- 19 Provisioning a MongoDB Database [Free](#)
- 20 Enabling CORS [Free](#)
- 21 Schemas & Models [Free](#)
- 22 Routes and Controllers [Free](#)
- 23 Testing with POSTman [Free](#)

Vuex

- 24 The Current Problem [Free](#)
- 25 State Management [Free](#)
- 26 Getters [Free](#)
- 27 Mutations [Free](#)
- 28 Actions [Free](#)

Using Store In Components

- 29 State and Actions on Components [Free](#)
- 30 LAB: Product List [Free](#)
- 31 LAB: Product Details [Free](#)
- 32 LAB: Admin Features [Free](#)
- 33 LAB: Spinner, Cart & Store Subscription [Free](#)