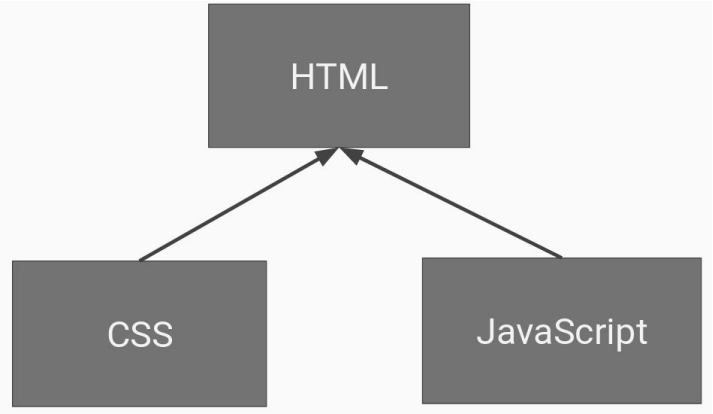
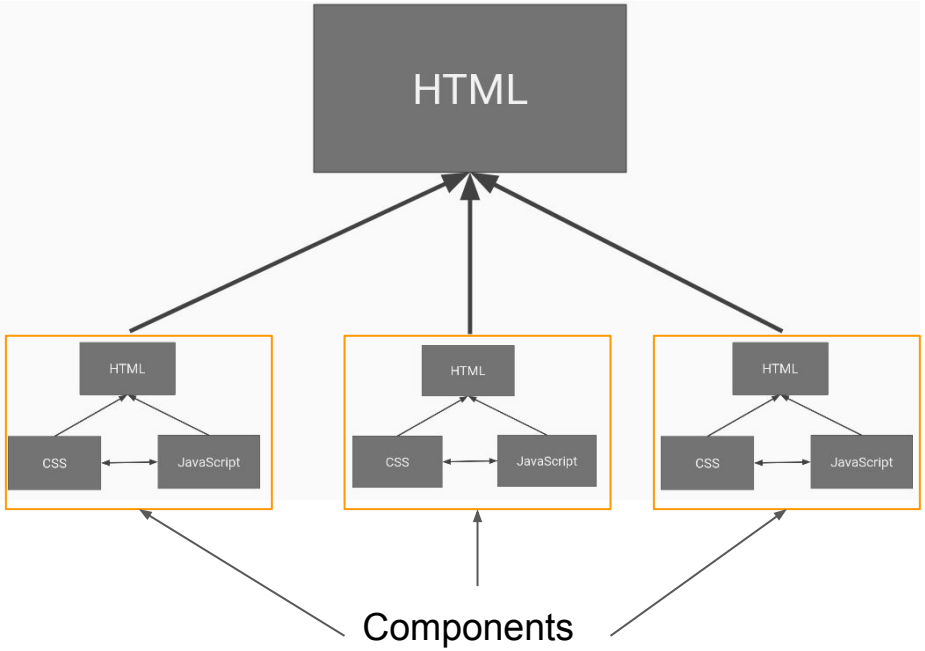
Vue Router

Vue: Evolution of Concepts - HTML to Vue

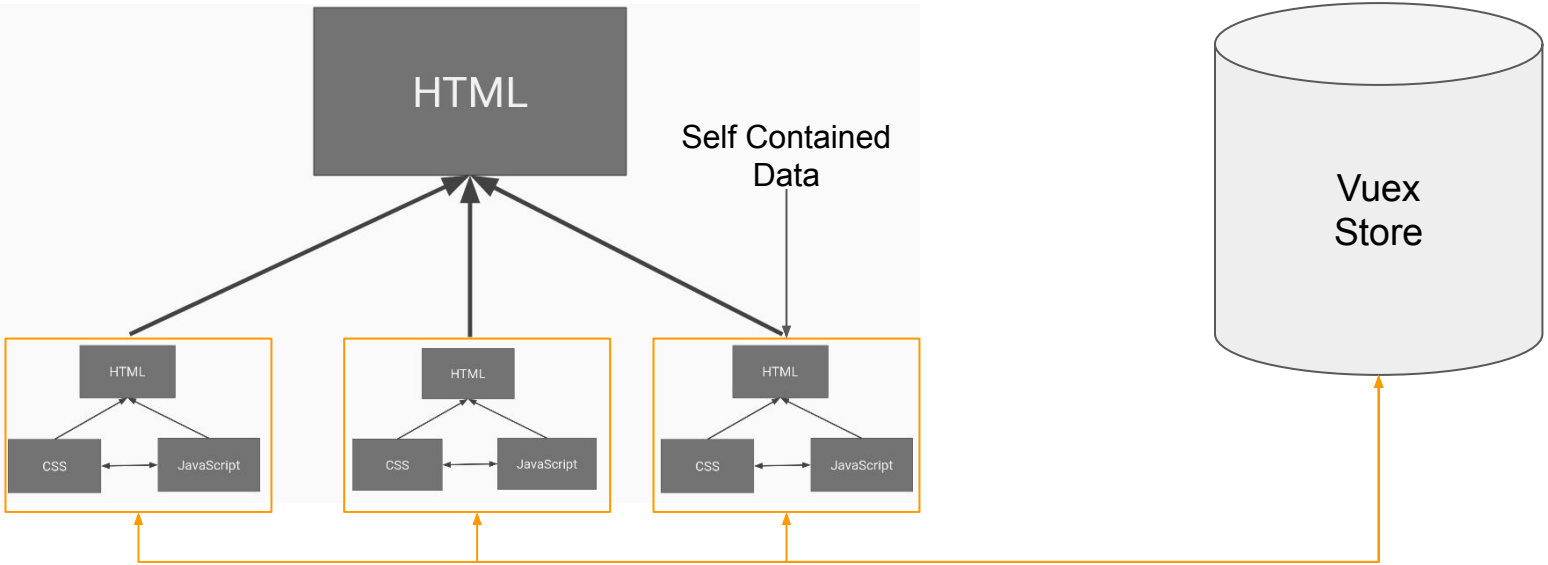
Index.html & App.vue





Vue: Evolution of Concepts - Vue to Vuex

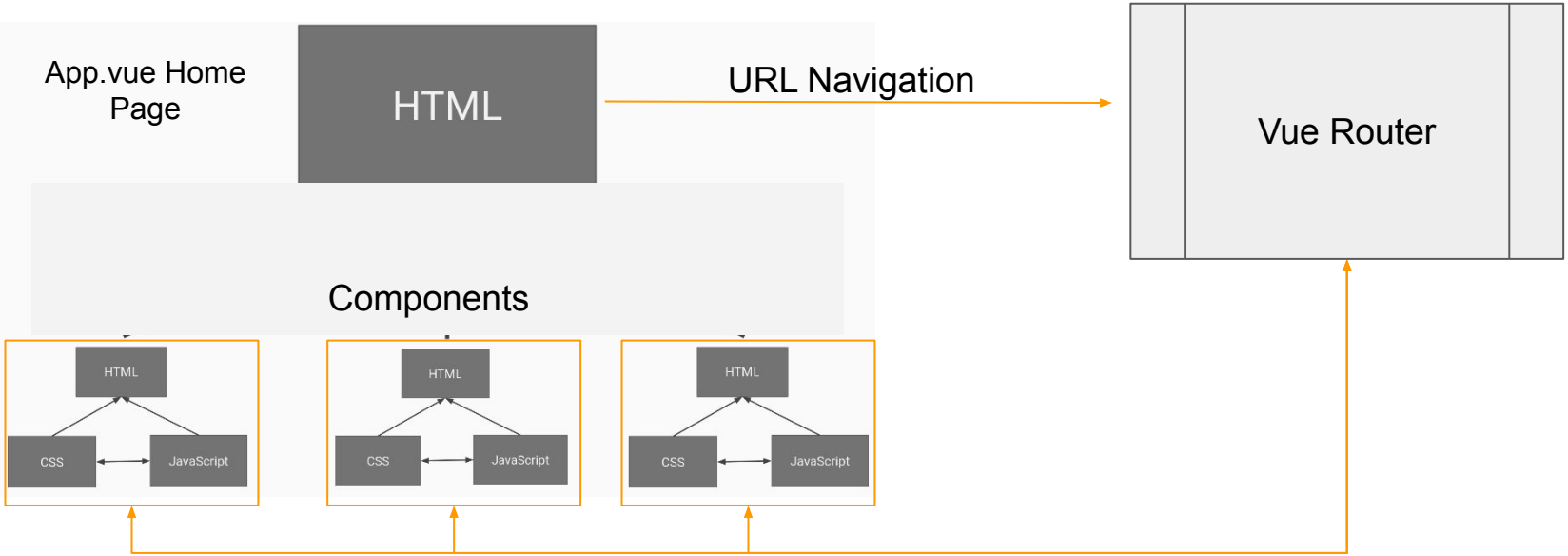
Index.html & App.vue



Vue: Evolution of Concepts - Components as Site Pages

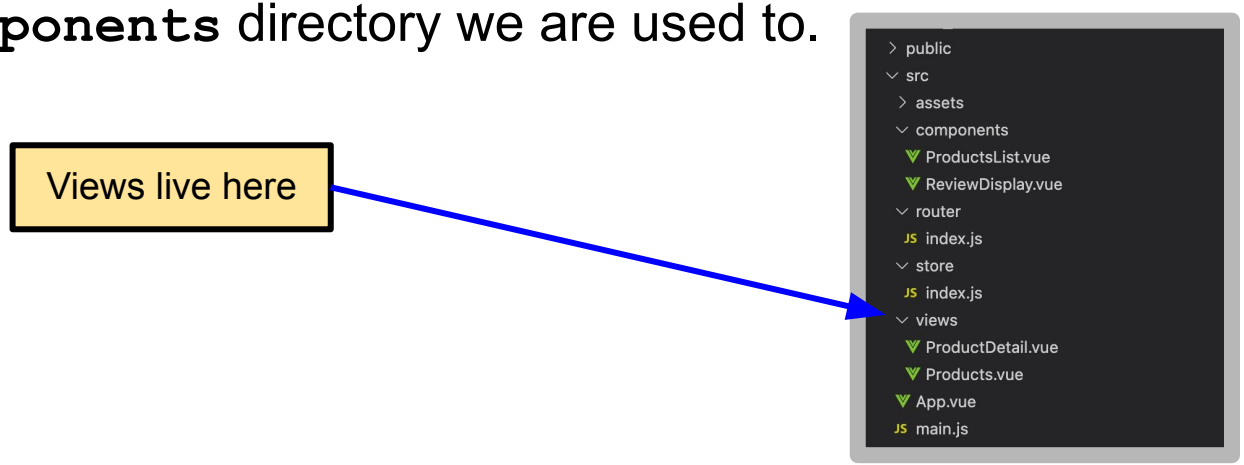
- Single Page Applications allow you to write complete applications that do not require reloading of the full page.
- Even though we only have one physical page, we can create components that serve as virtual site pages.
- Components that serve as site pages are still on same physical page and can be loaded and unloaded on the page as needed.
- Physical pages are specified by URLs, in Single Page Applications, we use **routes** to apply this paradigm to components acting as virtual site pages.

Vue: Evolution of Concepts - Routing to Site Pages



Components vs. Views

- **<u>Views</u>** are just components that serve a special purpose: acting as virtual pages.
- Difference between a View and Component is conceptual, aside from the fact that Views live in a **`views`** directory rather than in the **`components`** directory we are used to.



Using RouterView

- **`RouterView`** is a functional component that renders components (views) for a given path.
- **`RouterView`** does not require view components to be in the views directory, but they should be.

The **`App`** component includes the **`RouterView`** component in its **`<template>`** section.

`RouterView` handles all other rendering of components which are views.

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

Defining Routes

- Routes used by **`RouterView`** are defined in src/router/index.js

The **`routes`** array holds objects representing routes.

`RouterView` route objects require the route **`path`** and the **`component`** being routed to.

Optionally, a route can be given a **`name`**. *This is best practice*.

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import Products from '@/views/Products'

Vue.use(VueRouter)

const routes = [
  {
      path: '/',
      name: 'products',
      component: Products
  }

]

const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})
```

Using Routes

- The router-link tag can be used to define a link to a route in another component's **`<template>`** section.

```
<template>
  <div id="app">
    <nav>
      <router-link v-bind:to="{ name: 'home'}"> Home  </router-link>
    </nav>
    <router-view />
  </div>
</template>
```

Dynamic Routing

- We can specify dynamic values in a router path. much like we did with `@PathVariable` in Java.
- A route in the router/index.js file can use a dynamic value by using a placeholder preceded by a :
  - `path="/product/:productId"`
- Any variables in the router path will be accessible to the component that was routed to via the `$route.params` object.
  - `const product = this.$route.params.productId;`

Anatomy of a router-link

```
<router-link v-bind:to="{ name: 'home'}" tag="div">Home</router-link>
```

The **`to`** attribute can take a literal path (**`to="/"`**) or bind to an object with properties. This one has the **`name`** of the route to use as the property. Using the **`name`** of a route is the best practice method of specifying the route.

The **`tag`** attribute is optional. It allows you to specify that the router link should be rendered as a different type of tag. In this instance it will render as a **`div`**.

Using Route Parameters with router-link

- The properties object passed in to the **`router-link`** tag can have a params property containing any dynamic router params.

```
<router-link :to="{ name: 'product-details', params: {productId: product.id} }">
    {{ product.name}}
</router-link>
```

The **`productId`** property in the **`params`** object will be used to populate the productId in the dynamic route (when **`path="/product/:productId"`**, for instance).

Using Query Parameters with router-link

- The properties object passed in to the **`router-link`** tag can have a query property containing any query params.

```
<router-link :to="{ name: 'homepage', query: { 'active': true  } }">Home</router-link>
```

The **`query`** object can be used to pass query params in a router link

/home?active=true

Let's Work Through Another Example