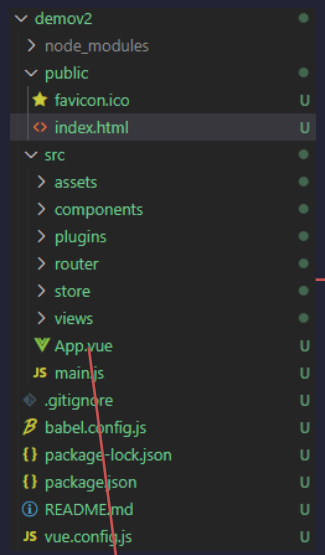
(디지털 컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

훈련기간 : 2021.05.07 ~ 2021.12.08

KH정보교육원 | 전승리

Vue의 기본 구조 왜 SPA인지?



1.side bar를 보면 public-index.html이 있다. 열어보자 그럼 아래 와 같은 코드가 존재한다.
아래 파일을 렌더링하고 이 안에서 모든것이 이루어진다. (로딩이 따로 없다)

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <title><%= htmlWebpackPlugin.options.title %></title>
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700,900">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@mdi/font@latest/css/materialdesignicons.min.css">
  </head>
  <body>
    <noscript>
      <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>
    </noscript>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

2.흐름 index.html -> src -> main.js 에서 Vue 인스터스가 생성되고 html에있는 id='app'과 Vue에 #app이 연결이된다.

```js
import Vue from 'vue'
import App from './App.vue'
import router from './router'
import store from './store'
import vuetify from './plugins/vuetify'

Vue.config.productionTip = false


new Vue({
  router,
  store,
  vuetify,
  render: h => h(App)
}).$mount('#app')
```

3.렌더링과 함께 App이라는 component를 보여주게된다.

```js
import Vue from 'vue'
import VueRouter from 'vue-router'
import Home from '../views/Home.vue'

Vue.use(VueRouter)

const routes = [
  {
    path: '/',
    name: 'Home',
    component: Home
  },
  {
    path: '/about',
    name: 'About',
    // route level code-splitting
    // this generates a separate chunk (about.[hash].js) for this route
    // which is lazy-loaded when the route is visited.
    component: () => import(/* webpackChunkName: "about" */ '../views/About.vue')
  }
]
```

path: 주소
name: 이름
component: import 한 컴포넌트를 넣어준다.

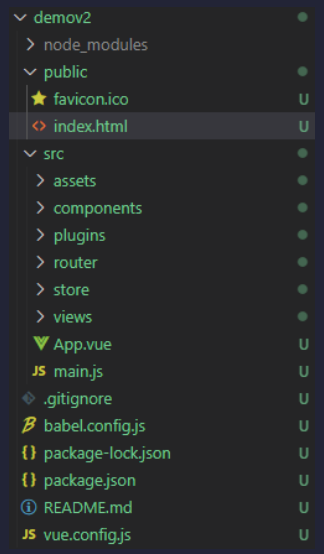
그럼 페이지 이동은 어떻게 되는건가?

router-view로 클릭했을때 router에서 해당하는 페이지를 보여준다.
router-link가 렌더링된 페이지에서는 a태그로 바뀌어나타나게된다.
버튼을 누르면 화면이 바뀌는데 사실상 component가 바뀌지면서
그렇게 보이는것뿐실제로는 spa다.
<router-view/>를 통해 router-link to="XX"를 누르는 순간
router-view가 path: XX에 해당하는 component로 바꿔준다.

Vue의 기본 구조 왜 SPA인지?

```vue
<template>
    <div>

</template>

<script>
export default {

}
</script>

<style scoped>

</style>
```

<template> html 코드가 들어간다.
<script> export defult {} data, method같은 데이터가 들어간다.
<style scoped> css가 들어가는데 scoped는 현재 template에만 적용하겠다는 뜻이다.



view 디렉토리에는 하나의 페이지가 있는데 이 페이지를 쪼개고싶다면 components디렉토리에 생성하면된다.

전역과 지역 component

전역

```
Vue.component('Jeonbutton', {
    template:
        `<div>
        {{ name }}<br>
        <helloworld>
            </helloworld>
        <button @click='changeText'>Click</button>
        </div>`,
    // 여기서는 작은 따옴표가아닌 백틱을 사용하도록한다.
    // 템플릿안에는 html태그가 들어간다.

    // 다음과같이 오브젝트형태인 data: 를 사용시 reference로 다른 여러곳에서 component사용시에 변경할때 동시에 업데이트가
    // 되버린다. 따라서 함수형으로 바꿔주어야한다. data()

    data() {
        return {
            name: 'ree'
        }
    },
    methods: {
        changeText(){
            this.name = 'updated Jeon'
            // this.name = app1.name
            // app1.name = 'updated deffent instance'도 가능
        }

    },
})
```

또한 전역컴포넌트는 component안에서도 component를 사용할수 있다.

단점으로는 웹팩같은 빌드 시스템을 사용하고 모든 컴포넌트를 전역등록 했으면 설사 어떤 컴포넌트를 더 이상 사용하지 않더라도 최종 빌드에는 들어가있게된다 사용자가 받아야할 js양이 불필요하게 커진다는뜻

이러한 문제를 해결하기위해 지역등록을 할 component는 지역등록을하자

지역

```
const app = new Vue({
    el: '#app',
    components: {
        'seungbutton': seungreebutton
    },
    data: {
        name: "Jeon"

    },
    methods: {
        changeText(){
            this.name = 'updated Jeon'
            // this.name = app1.name
            // app1.name = 'updated deffent instance'도 가능
        }

    },
    })
```

지역에서 지역 component가 사용하고싶은경우 사용할 지역 컴포넌트에 지역 components: {} 안에 이름과 object를 적어주면된다.

GlobalComponent 설정 (전역)

```vue
<template>
    <button @click="addCounter">{{ counter }}</button>
</template>

<script>
export default {
    name: 'global-component',
    props: ['initialTest'],
    template: '<button @click="addCounter">{{ counter }}</button>',
    data () {
        return {
            counter: this.initialTest
        }
    },
    methods: {
        addCounter () {
            this.counter += 1
        }
    }
}
</script>
```

props: 에 배치해야만 컴포넌트간 데이터 전달이 가능함

템플릿의 경우엔 컴포넌트를 배치했을때 자동의 생성되는 HTML 태그를 의미힌다.

(참고로 main.js에서도 Vue.component(원하는이름,옵션)형식으로 전역설정가능)

명명법: 컴포넌트의 이름을 작성한다.
보편적으로 LetsGo 라면 Lets-go 같은 형태의 이름으로 작성한다.
마찬가지로 Java의 명명법같은 형식이라 생각하면된다.
component의 경우 - 표시가 필요하다.
문법이 관습적인 명명법을 따를 것들이 있으므로 되도록 지켜주자

```
Vue.component(GlobalComponent.name, GlobalComponent)
```

Vue.component(원하는 이름, 옵션)

GlobalComponent 설정 (전역) 3.X Componen

3.X Component 7.1 패치로 component 선언 방식이 변경됨 아직 ing기때문에 2.X로 버전 다운

실제 Vue 객체를 생성하는 중앙(main)에서 사용할 컴포넌트의 등록까지 모두 처리함
가장 최근 Vue (7월 1일)의 경우 컴포넌트 등록을 중앙에서 하도록 변경함
중앙에서 컴포넌트를 등록하는 방법은 아래와 같이
app.component(컴포넌트 이름, 연결되는 실제 컴포넌트 객체) 형식으로 등록할 수 있다.
밑에 붙어 있는 use(store).use(router).mount('#app')은 향후 Vuex 까지 잠시 보류한다.

3.X는 main에서 선언 (21.7.1 기준)

main.js

```
import GlobalComponent from './components/GlobalComponent.vue'
const app = createApp(App)

app.component(GlobalComponent.name, GlobalComponent)
.use(store).use(router).mount('#app')
```

3.X에선 scipt에서 import가 안됨

```
<template>
 <global-component v-bind:initial-test='counter'>
 </global-component>
```

Test.vue

```
<script>
import LocalComponent from '../components/LocalComponent.vue'

export default {
    components: {
        'local-component': LocalComponent
    },
```

GlobalComponent 설정 (전역) 2.X Componen

2.X Component

2.X는 전역설정을 아래와같이 하면된다.

3.X와 달리 main에서 건드릴 부분 없음

main.js

```
import Vue from 'vue'
import App from './App.vue'
import router from './router'
import store from './store'
import vuetify from './plugins/vuetify'

Vue.config.productionTip = false

new Vue({
  router,
  store,
  vuetify,
  render: h => h(App)
}).$mount('#app')
```

2.X는 Component를 아래와 같이 import 해주고 사용해야함

```
<template>

<global-component v-bind:initial-test="counter">
</global-component>
```

Test.vue

```
<script>
import Vue from 'vue'
import GlobalComponent from '../components/GlobalComponent.vue'
Vue.component(GlobalComponent.name, GlobalComponent)
```