<오늘부터 내일까지 꼭 해야 할 일>

서버 접근이 안된다…

로컬은 되나.? 로컬은 되네..

1. Swagger부분 Front쪽 있어보이게 보강..

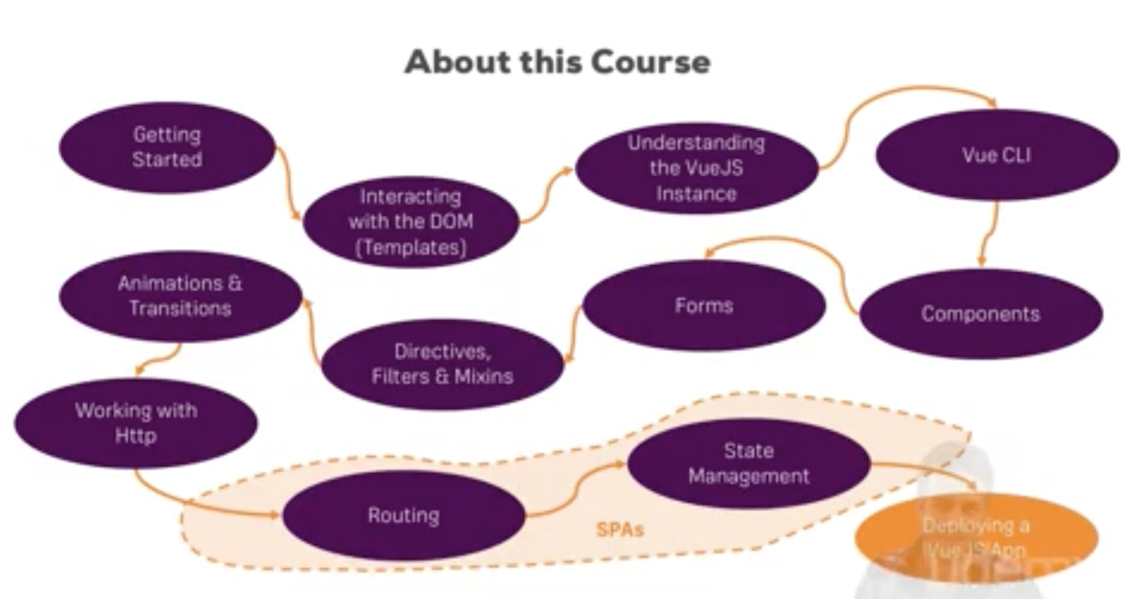
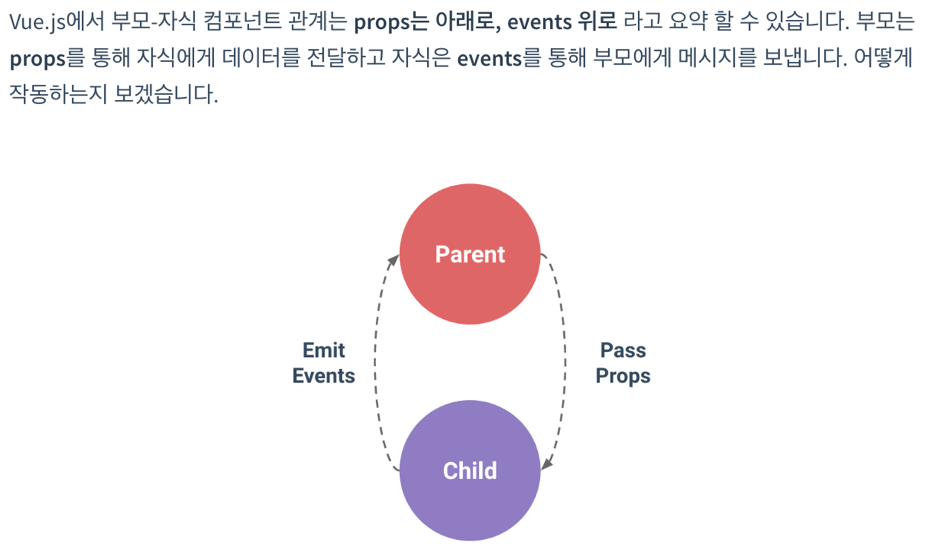
* ~~Component 어떻게 착착 쌓았었지?~~

: 한번 태그로 감싸고 position: absolute로

* ~~Input 쪽 완벽하게~~

Input은 무조건 v-model로 처리해야 한다!

* ~~부모 컴포넌트, 자식 컴포넌트 간 데이터 처리!!! 대박 ㅠㅠ~~



Vue 다시 공부

이건 자식컴포넌트

<template>  
 <div class="my-switch">  
 <div id="on" :class="{active: value}" v-on:click="switched(true)">ON</div>  
 <div id="off" :class="{active: !value}" v-on:click="switched(false)">OFF</div>  
 </div>  
</template>  
  
<script>  
  
**export default** {  
 name: **'my-switch'**,  
 props: [**'value'**],  
 data: **function** () {  
 **return** {  
 realValue: **this**.value  
 }  
 },  
 methods: {  
 switched (isOn) {  
 **this**.realValue = isOn  
 **this**.$emit(**'input'**, **this**.realValue)// input은 그냥 있는 이벤트인가?  
 }  
 }  
}  
</script>  
  
<style scoped>  
 #on, #off {  
 width: 40px;  
 height: 20px;  
 background-color: lightgray;  
 padding: 2px;  
 display: inline-block;  
 margin: 10px -2px;  
 box-sizing: content-box;  
 cursor: pointer;  
 text-align: center;  
 }  
  
 #on:hover, #on.active {  
 background-color: lightgreen;  
 }  
  
 #off:hover, #off.active {  
 background-color: lightcoral;  
 }  
</style>

자식 그냥 이렇게 해도 되고

**export default** {  
 name: **'my-switch'**,  
 props: [**'value'**],  
 methods: {  
 switched (isOn) {  
 **this**.$emit(**'input'**, **i**sOn)// input은 그냥 있는 이벤트인가?  
 }  
 }

이건 부모컴포넌트

<template>  
 <div class="rs-import-section">  
 <section>  
 <h2>Import from Swagger</h2>  
 <p>Swagger의 Base URL을 copy & paste해주세</p>  
  
 <div>  
 <input type="email" id="email" v-model="userdata.email">  
 <input type="password" id="password" v-model.lazy="userdata.password">  
 <input type="text" id="age" v-model.number="userdata.age">  
 </div>  
  
 <div>  
 <textarea id="message" rows=5 v-model="msg"></textarea>  
 </div>  
  
 <div>  
 <!-- 체크박스 v-model에 똑같은거 2개 걸면 자동으로 array로 값 들어간다! 엄청엄청 신기하다-->  
 <label for="sendemail"><input type="checkbox" id="sendemail" value="sendMail" v-model="sendMail">SendMail</label>  
 <label for="sendInfoemail"><input type="checkbox" id="sendInfoemail" value="sendInfoMail" v-model="sendMail">Send Info Mail</label>  
 </div>  
  
 <div>  
 <!-- 라디오 v-model에 똑같은거 2개 걸면 하나만 선택해서 값 들어간다! 엄청엄청 신기하다-->  
 <label for="male"><input type="radio" id="male" value="male" v-model="gender">Male</label>  
 <label for="female"><input type="radio" id="female" value="female" v-model="gender">Female</label>  
 </div>  
  
 <div>  
 <label for="priority">Priority</label>  
 <select id="priority" v-model="selectedPriority">  
 <option v-for="item in priorities" :key="item.id">{{item}}</option>  
 </select>  
 </div>  
  
 <div>  
 <!--v-model을 걸면 자식한테 자동으로 prop으로 넘어가는구나 ㅜㅜ 신기하당-->  
 <my-switch v-model="dataSwitch"></my-switch>  
 </div>  
  
 <br>  
 <p> Mail : {{userdata.email}}</p>  
 <p> PW : {{userdata.password}}</p>  
 <p> Age : {{userdata.age}}</p>  
 <p style="white-space: pre"> Message : {{msg}}</p>  
 <ul>  
 <!-- v-for로 li 만들기-->  
 <li v-for="item in sendMail" :key="item.id">{{item}}</li>  
 </ul>  
 <p> Gender : {{gender}}</p>  
 <p> Priority : {{selectedPriority}}</p>  
 <p> On?Off? : {{dataSwitch}}</p>  
  
 </section>  
 </div>  
</template>  
  
<script>  
**import** MySwitch **from './MySwitch.vue'  
  
export default** {  
 name: **'rs-import-section'**,  
 components: {MySwitch},  
 data () {  
 **return** {  
 userdata: {  
 email: **''**,  
 password: **''**,  
 age: 27  
 },  
 msg: **''**,  
 sendMail: [],  
 gender: **'male'**,  
 priorities: [**'High'**, **'Medium'**, **'Low'**],  
 selectedPriority: **'High'**,  
 dataSwitch: **true** }  
 }  
}  
</script>  
  
<style scoped>  
 @import url(**'https://fonts.googleapis.com/css?family=Raleway'**);  
  
 section {  
 position: absolute;  
 left: 20%;  
 width: 100%;  
 height: 100%;  
 background-color: #fbd4f6;  
 font-family: **'Raleway'**, sans-serif;  
 }  
</style>

제출까지 누르면 이렇게

<template>  
 <div class="rs-import-section">  
 <section>  
 <h2>Import from Swagger</h2>  
 <p>Swagger의 Base URL을 copy & paste해주세</p>  
  
 <div>  
 <input type="email" id="email" v-model="userdata.email">  
 <input type="password" id="password" v-model.lazy="userdata.password">  
 <input type="text" id="age" v-model.number="userdata.age">  
 </div>  
  
 <div>  
 <textarea id="message" rows=5 v-model="msg"></textarea>  
 </div>  
  
 <div>  
 <!-- 체크박스 v-model에 똑같은거 2개 걸면 자동으로 array로 값 들어간다! 엄청엄청 신기하다-->  
 <label for="sendemail"><input type="checkbox" id="sendemail" value="sendMail" v-model="sendMail">SendMail</label>  
 <label for="sendInfoemail"><input type="checkbox" id="sendInfoemail" value="sendInfoMail" v-model="sendMail">Send Info Mail</label>  
 </div>  
  
 <div>  
 <!-- 라디오 v-model에 똑같은거 2개 걸면 하나만 선택해서 값 들어간다! 엄청엄청 신기하다-->  
 <label for="male"><input type="radio" id="male" value="male" v-model="gender">Male</label>  
 <label for="female"><input type="radio" id="female" value="female" v-model="gender">Female</label>  
 </div>  
  
 <div>  
 <label for="priority">Priority</label>  
 <select id="priority" v-model="selectedPriority">  
 <option v-for="item in priorities" :key="item.id">{{item}}</option>  
 </select>  
 </div>  
  
 <div>  
 <!--v-model을 걸면 자식한테 자동으로 prop으로 넘어가는구나 ㅜㅜ 신기하당-->  
 <my-switch v-model="dataSwitch"></my-switch>  
 </div>  
  
 <div>  
 <button @click="submitted">Submit</button>  
 </div>  
  
 <br>  
 <div v-if="isSubmittted">  
 <p> Mail : {{userdata.email}}</p>  
 <p> PW : {{userdata.password}}</p>  
 <p> Age : {{userdata.age}}</p>  
 <p style="white-space: pre"> Message : {{msg}}</p>  
 <ul>  
 <!-- v-for로 li 만들기-->  
 <li v-for="item in sendMail" :key="item.id">{{item}}</li>  
 </ul>  
 <p> Gender : {{gender}}</p>  
 <p> Priority : {{selectedPriority}}</p>  
 <p> On?Off? : {{dataSwitch}}</p>  
 </div>  
 </section>  
 </div>  
</template>  
  
<script>  
**import** MySwitch **from './MySwitch.vue'  
  
export default** {  
 name: **'rs-import-section'**,  
 components: {MySwitch},  
 data () {  
 **return** {  
 userdata: {  
 email: **''**,  
 password: **''**,  
 age: 27  
 },  
 msg: **''**,  
 sendMail: [],  
 gender: **'male'**,  
 priorities: [**'High'**, **'Medium'**, **'Low'**],  
 selectedPriority: **'High'**,  
 dataSwitch: **true**,  
 isSubmittted: **false** }  
 },  
 methods: {  
 submitted () {  
 **this**.isSubmittted = **true** }  
 }  
}  
</script>  
  
<style scoped>  
 @import url(**'https://fonts.googleapis.com/css?family=Raleway'**);  
  
 section {  
 position: absolute;  
 left: 20%;  
 width: 100%;  
 height: 100%;  
 background-color: #fbd4f6;  
 font-family: **'Raleway'**, sans-serif;  
 }  
</style>

**vue-router**

**1. 기본 동작 원리**

1) main.js에 vue-router 등록

2) routes.js파일 만들어서 내 라우터 만들기 (“routes배열”) => path와 component들 등록

3) main.js에서 new Router로 객체 생성(이때, 내 router로 객체 생성)

4)App.vue에서 맨 마지막에 <router-view>추가

2. History모드

기본모드는 hash 모드. url에 # 붙는거

mode: history 로 하면 #안붙이는 모드 할 수 있음

3.

뷰에서는 <a> 앵커 태그를 쓰지 않는 방법을 권장한다!

<!--<li role="presentation"><router-link to="/">Home</router-link></li>-->  
<!--<li role="presentation"><router-link to="/user">User</router-link></li>-->  
<router-link to="/" tag="li" active-class="active" exact><a>Home</a></router-link>  
<router-link to="/user" tag="li" active-class="active"><a>User</a></router-link>

4. dynamic parameter 주기

\*\* vue router에서는 이미 있는 component면 router가 다시 load되지 않는다!!

* Watch를 써야 함
* $route를 감지해서, $route가 변하면, 어떤 동작을 하도록 해야 한다(즉, 함수 형태여야 함)

watch: {  
 **'$route'**: **function** (to) {  
 **this**.id = to.params.id  
 }  
},  
data () {  
 **return** {  
 id: **this**.$route.params.id  
 }  
}

5. Child Route

6. router의 name 옵션으로 부르기

7 . query string

<router-link tag="button" :to="{ name: 'UserEdit', params: { id: $route.params.id }, query: {locale: 'en', q:100 } }">Edit!</router-link>

8. Redirect

{  
 path: **'/redirect-me'**,  
 redirect: **'/user'**}

다른 거 입력했을 때 \*로 처리하면 redirect 할 수 있음

9. Animiation

그냥 넘어가작

beforeEnter

beforeRouterEnter

beforeRouterLeave

이런 함수 적용시켜서

컴포넌트 빠져나가기 전에 confirm 주는것도 가능하고 그렇다