

2025년 상반기 K-디지털 트레이닝

vue-router를 이용한 라우팅 2

[KB] IT's Your Life



동적 라우트(Dynamic Route)

- 일정한 패턴의 URI 경로를 하나의 라우트에 연결하는 방법
- URI 경로의 일부에 실행에 필요한 파라미터 값을 포함하여 컴포넌트에서 이용

```
/경로/:파라미터명
/members/1001
/members/2
```

동적 라우트(Dynamic Route)

```
[ vue-router 객체 ]
const router = createRouter({
    history: createWebHistory(),
    routes : [
        { path: '/members/:id', component: MemberInfo },
        . . . . . .
                            [ 컴포넌트에서 ]
                           const currentRoute = useRoute()
})
                           const id = currentRoute.params.id
```

예) 요청된 경로: /members/6

{ path: '/members/:id', component: MemberInfo }

MemberInfo 컴포넌트

const currentRoute = useRoute() const id = currentRoute.parms.id

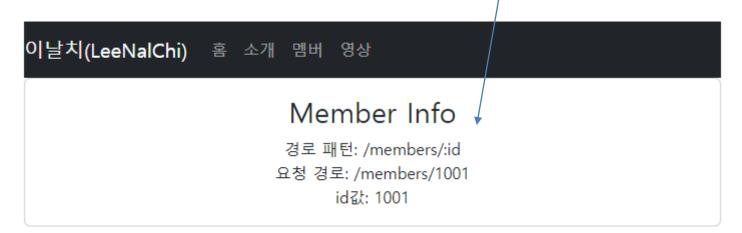
src/pages/MemberInfo.vue

```
<template>
  <div class="card card-body">
    <h2>Member Info</h2>
    <div>
      경로 패턴: /members/:id<br />
     요청 경로: {{ currentRoute.path }}<br />
     id값: {{ currentRoute.params.id }}<br />
    </div>
  </div>
</template>
<script>
import { useRoute } from 'vue-router';
export default {
 name: 'MemberInfo',
 setup() {
    const currentRoute = useRoute();
    return { currentRoute };
 },
</script>
```

src/router/index.js

```
import { createRouter, createWebHistory } from 'vue-router';
import Home from '@/pages/Home.vue';
import About from '@/pages/About.vue';
import Members from '@/pages/Members.vue';
import Videos from '@/pages/Videos.vue';
import MemberInfo from '@/pages/MemberInfo.vue';
const router = createRouter({
 history: createWebHistory(),
  routes: [
    { path: '/', component: Home },
    { path: '/about', component: About },
    { path: '/members', component: Members },
   { path: '/members/:id', component: MemberInfo },
    { path: '/videos', component: Videos },
});
export default router;
```

주소창에 http://localhost:5173/members/1001 직접 입력



src/members.json

```
"id": 1,
  "name": "장영규",
  "role": "베이스",
  "photo": "http://sstatic.naver.net/people/90/200810291643098681.jpg",
  "desc": "음악감독,베이스연주가,가수",
  "insta": "",
  "facebook": "",
  "youtube": ""
},
  "id": 2,
  "name": "이철희",
  "role": "드럼",
  "photo": "http://sstatic.naver.net/people/25/202004271203269741.jpg",
  "desc": "드럼연주가",
  "insta": "https://www.instagram.com/chulheedrum/",
  "facebook": "",
  "youtube": "https://www.youtube.com/channel/UCt4_srP6q_aPsmuZ3W3R-Uw"
},
```

동적 라우트

src/App.vue

```
<style>
@import url('https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css');
.fa {
 padding: 5px;
 width: 30px;
 text-align: center;
 text-decoration: none;
 margin: 5px 2px;
.fa-facebook {
  background: #3b5998;
 color: white;
.fa-youtube {
  background: #bb0000;
  color: white;
.fa-instagram {
  background: #125688;
  color: white;
</style>
```

동적 라우트

src/pages/MemberInfo.vue

```
<template>
 <div className="mt-5">
   <img :src="member.photo" class="img" />
    <h4 class="mt-2">{{ member.name }}({{ member.role }})</h4>
    {{ member.desc }}
   <a
     v-if="member.insta && member.insta !== ''"
     class="fa fa-instagram m-1"
      :href="member.insta"
   ></a>
    <a
     v-if="member.facebook && member.facebook !== ''"
     class="fa fa-facebook m-1"
      :href="member.facebook"
   ></a>
   <a
     v-if="member.youtube && member.youtube !== ''"
     class="fa fa-youtube m-1"
      :href="member.youtube"
   ></a>
   <br /><br />
   <router-link to="/members">멤버 목록으로</router-link>
 </div>
</template>
```



이철희(드럼) 드럼연주가





동적 라우트

src/pages/MemberInfo.vue

```
<script>
import { useRoute } from 'vue-router'
import members from '@/members.json'
export default {
    name : "MemberInfo",
    setup() {
        const currentRoute = useRoute()
        const id = parseInt(currentRoute.params.id, 10);
        const member = members.find((m)=>m.id === id)
        return { member }
</script>
```

src/pages/Members.vue

```
<template>
 <div>
    <h2 class="m-4">이날치 멤버</h2>
    <div class="container">
     <div class="row">
        <div
          v-for="m in members"
          :key="m.id"
          class="col-6 col-xs-6 col-sm-4 col-md-3 col-lg-2"
          <router-link :to="'/members/' + m.id">
            <img
              class="img-thumbnail"
              style="width: 90px; height: 110px"
              :src="m.photo"
              :title="m.name"
            /><br />
            <h6 class="display-7">{{ m.name }}</h6>
          </router-link>
        </div>
     </div>
    </div>
 </div>
</template>
```

export default { name: 'Members', setup() { return { members }; **}**; </script> 이날치 멤버 장영규

import members from '@/members.json';

<script>

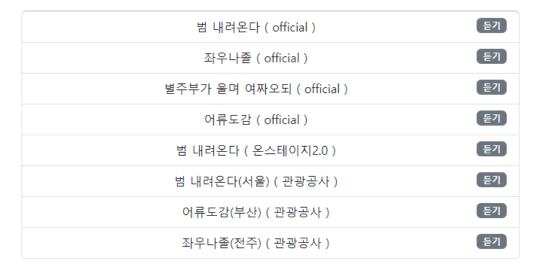
권송희

중첩 라우트

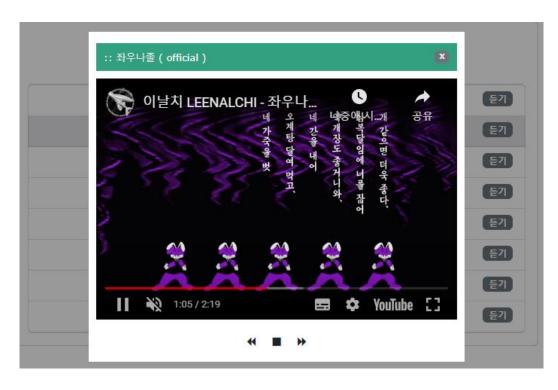
- o /members/1로 요청했을 때 화면을 완전히 전환
- 화면 전환없이 화면을 중첩시켜서 보아야 하는 경우 → 중첩 라우트(Nested Route)
- 팝업 화면 구성 시 주로 사용

/videos

영상 리스트

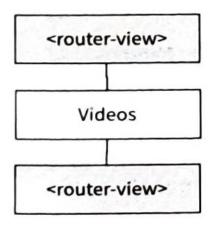


/videos/:id

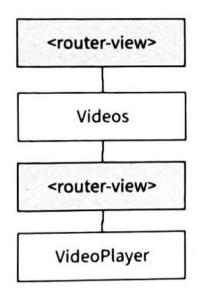


♡ 중첩 라우트

요청 경로 : /videos



요청 경로:/videos/:id



router 객체

명명된 라우트(Named Route)

- 라우트 정보에 이름을 부여하는 방법
- URI 경로가 아닌 이름을 이용해 경로를 전환

라이브러리 추가

- o youtube-vue3
 - 유튜브 영상 재생 라이브러리
 - npm install youtube-vue3

비디오 재생 목록

- o src/videos.json
- App에서 해당 데이터를 provide
- Videos.vue에서 inject 해서 비디오 목록 사용

src/router/index.js

```
const router = createRouter({
    history: createWebHistory(),
    routes : [
        { path: '/', name: 'home', component: Home },
        { path: '/about', name: 'about', component: About },
        { path: '/members', name:'members', component: Members },
        { path: '/members/:id', name: 'members/id', component: MemberInfo },
            path: '/songs', name:'videos', component: Videos,
            children : [
                { path: ':id', name:'videos/id', component: VideoPlayer }
       },
})
export default router;
```

3

src/components/Header.vue

```
<template>
  <div :class="navClass">
   <router-link class="nav-link" :to="{ name: 'home' }">喜</router-link>
    <router-link class="nav-link" :to="{ name: 'members' }">멤버</router-link>
    <router-link class="nav-link" :to="{ name: 'videos' }">영상</router-link>
    </div>
 </nav>
</template>
```

query.params

```
URI 패턴 〈router-link〉의 예

/members/:id 〈router-link :to="{ name:'members/id', params: { id: 1 } }">

......

〈/router-link〉

ex) /members/1

/members?a=1&b=2 〈router-link :to="{ name:'members', query: { a: 1, b: 2 } }">

......

〈/router-link〉
```

src/pages/Members.vue

```
<template>
    <div>
      <h2 class="m-4">이날치 멤버</h2>
      <div class="container">
        <div class="row">
          <div v-for="m in members" :key="m.id"</pre>
            class="col-6 col-xs-6 col-sm-4 col-md-3 col-lg-2">
            <router-link :to="{ name: 'members/id', params: { id: m.id } }">
              <img class="img-thumbnail"</pre>
                style="width:90px; height:110px;"
                :src="m.photo"
                :title="m.name" /><br/>
              <h6 class="display-7">{{m.name}}</h6>
            </router-link>
          </div>
        </div>
      </div>
    </div>
</template>
```

src/pages/MemberInfo.vue

```
<template>
   <div className="mt-5">
     <router-link :to="{ name:'members' }">멤버 목록으로</router-link>
   </div>
</template>
```

src/pages/Videos.vue

```
<template>
 <div class="card card-body">
   <h2 class="m-3">영상 리스트</h2>
   <li
       v-for="v in videos"
       :key="v.id"
       class="list-group-item text-left"
       :class="playingVideo(v.id)"
       {{ v.title }} ( {{ v.category }} )
       <router-link :to="{ name: 'videos/id', params: { id: v.id } }">
         <span class="float-end badge bg-secondary">듣기</span>
       </router-link>
     <router-view></router-view>
 </div>
</template>
```

src/pages/Videos.vue

```
<script>
export default {
 name: 'VideoPlayer',
 components: { YoutubeVue3 },
 setup() {
    const stopVideo = () => {
     playerRef.value.player.stopVideo();
     router.push({ name: 'videos' });
    };
    const playNext = () => {
     const index = videos.findIndex((v) => v.id === videoInfo.video.id);
     const nextVideo = videos[index + 1];
     if (nextVideo) {
        videoInfo.video = nextVideo;
        router.push({ name: 'videos/id', params: { id: nextVideo.id } });
     } else {
        videoInfo.video = videos[0];
        router.push({ name: 'videos/id', params: { id: videos[0].id } });
    };
```

src/pages/Videos.vue

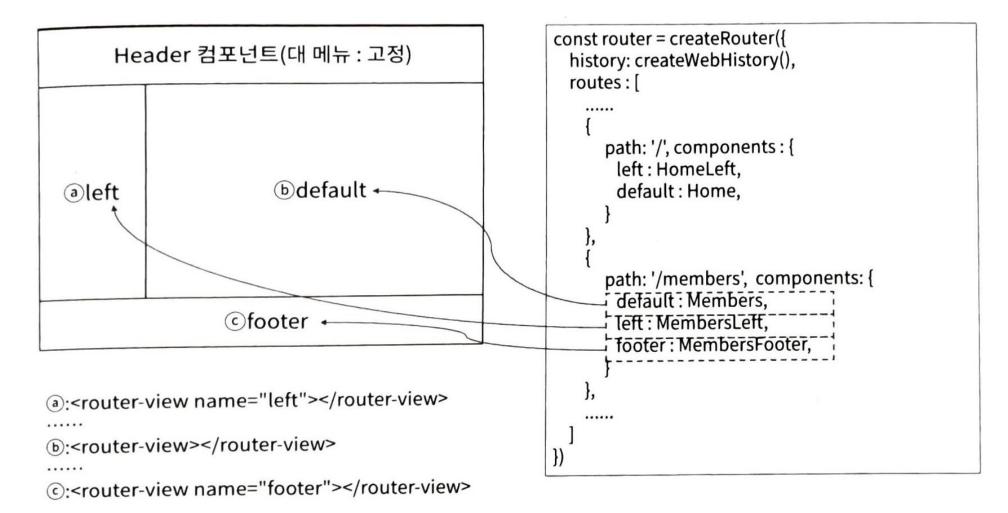
```
const playPrev = () => {
     const index = videos.findIndex((v) => v.id === videoInfo.video.id);
     const prevVideo = videos[index - 1];
     if (prevVideo) {
       videoInfo.video = prevVideo;
       router.push({ name: 'videos/id', params: { id: prevVideo.id } });
   };
   return { videoInfo, playerRef, playNext, stopVideo, playPrev };
 },
</script>
```

src/router/index.js

```
const router = createRouter({
 history: createWebHistory(),
  routes: [
    { path: '/', name: 'home', component: Home },
    { path: '/about', name: 'about', component: About },
    { path: '/members', name: 'members', component: Members },
    { path: '/members/:id', name: 'members/id', component: MemberInfo },
     path: '/songs',
      name: 'videos',
      component: Videos,
      children: [{ path: ':id', name: 'videos/id', component: VideoPlayer }],
});
export default router;
```

명명된 뷰(Named View)

- o <router-view>를 중첩시키지 않고 한꺼번에 화면에 나타내야 하는 경우
- <router-view>에 이름을 부여, 여러 컴포넌트를 지정된 이름의 뷰로 나타냄



- 명명된 뷰(Named View)
 - 이름이 지정되지 않은 뷰 → 기본 뷰(default view)

src/router/index.js

```
const router = createRouter({
    history: createWebHistory(),
    routes : [
            path: '/',
            components: {
                default:Home,
                left:HomeLeft,
        },
{
            path: '/members',
            components: {
                default: Members,
                left: MembersLeft,
                footer: MembersFooter,
        },
})
export default router;
```

src/App.vue

```
<template>
 <div class="container">
    <Header />
    <div class="card card-body border-0 pl-3 pr-3 pt-0">
     <div class="row">
        <div class="col-md-3 border">
          <router-view name="left"></router-view>
        </div>
        <div class="col-md-9 border">
          <router-view></router-view> <!-- default -->
        </div>
     </div>
     <div class="row">
        <div class="col-md-12 border">
          <router-view name="footer"></router-view>
       </div>
     </div>
    </div>
 </div>
</template>
```