

2025년 상반기 K-디지털 트레이닝

# 히스토리 모드와 404 라우트

[KB] IT's Your Life



#### 🧿 히스토리 모드

○ 라우팅 모드라고도 하며 URI 경로와 동기화하여 UI를 나타내기 위해 어떤 정보를 이용할 것인지를 지정하는 방법을 제공

```
import { createRouter, createWebHistory } from 'vue-router'
...

const router = createRouter({
   history: createWebHistory(), // HTML5 모드, 브라우저의 HTML5 history API 이용
...
});
```

○ Vue에서는 해시모드(Hash Mode)와 HTML5 모드, 메모리 모드를 제공

#### ○ 해시 히스토리 모드

- URI 경로에서 # 이후의 문자열 정보를 이용해 라우팅
  - localhost:8000/#/about

```
import { createRouter, createWebHistory } from 'vue-router'
...

const router = createRouter({
  history: createWebHashHistory(), // 해시모드
  ...
});
```

#### ㅇ 메모리 모드

■ 메모리에 저장된 경로 정보를 이용

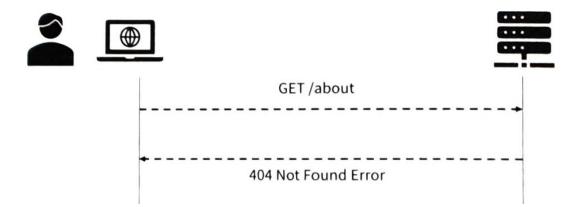
#### 💟 배포시 히스토리 모드 운영

- 웹 서버에 폴백 페이지(Fallback UI)가 올바르게 설정되어야 함
  - 개발 서버에는 설정되어 있음

- 💟 serve 패키지로 패포판 운영하기
  - 빌드 결과를 c:₩www에 복사
    - npm run build
  - o serve 패키지로 배포판을 웹 서버에서 운영

    - npx serve
  - 브라우저에서 localhost:3000/about으로 직접 요청
    - → 404 에러 발생

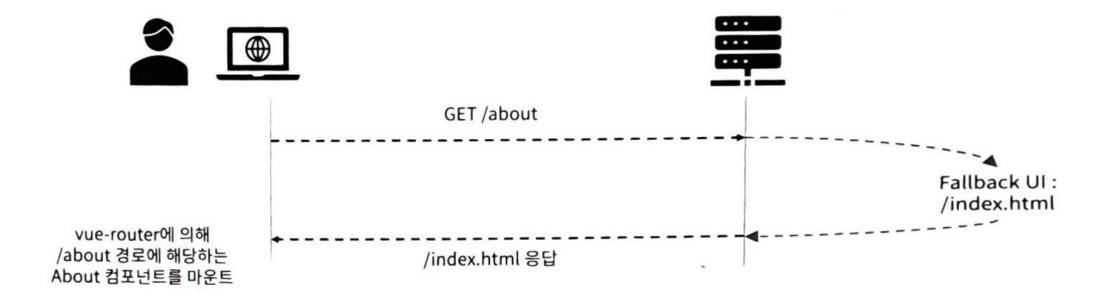
Vue 애플리케이션을 호스팅하는 웹서버에 Fallback UI가 설정되지 않은 경우



#### 💟 serve 패키지로 패포판 운영하기

- o SPA(Single Page Application)은 단 하나의 index.html 문서만 존재
- 그 외의 페이지는 존재하지 않음 → 다른 url 요청은 404 에러 발생
- o 해결방법은 Fallback UI를 지정

# Vue 애플리케이션을 호스팅하는 웹서버에 Fallback UI가 된 경우



#### 😕 serve 패키지로 패포판 운영하기

- o npx serve -s
  - -s: single page를 의미 → Fallback UI 설정
- o 서버별 fallback ui 지정 방법
  - https://router.vuejs-korea.org/ko/guide/essentials/history-mode.html

#### 🧿 404 라우트

- 라우트로 설정되 않은 경로로 요청하면 빈화면이 나타남
  - → 매칭된 라우트가 존재하지 않으므로 아무런 컴포넌트가 마운트 되지 않음
- 404 라우트 추가
  - 경로가 매칭되는 것이 없을 때 보여줄 라우트
  - 마지막 라우트로 등록해야 함

# src/pages/NotFound.vue

# src/router/index.js

```
import NotFound from '@/pages/NotFound.vue'
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,
            beforeEnter:membersIdGuard
        },
            path: '/videos', name:'videos', component: Videos,
            children : [
                { path: ':id', name:'videos/id', component: VideoPlayer }
        },
        { path: '/:paths(.*)*', name: 'NotFound', component: NotFound },
})
export default router;
```