

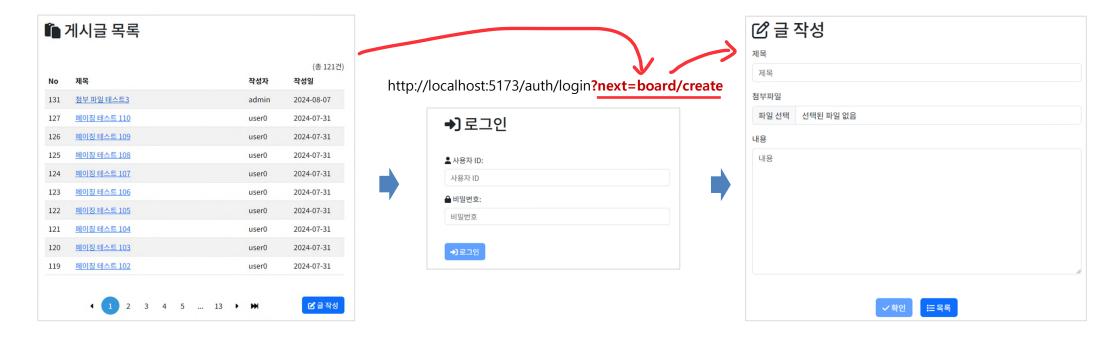
2025년 상반기 K-디지털 트레이닝

인증 가드

[KB] IT's Your Life



☑ 로그인하지 않은 상태에서 글쓰기를 누른 경우

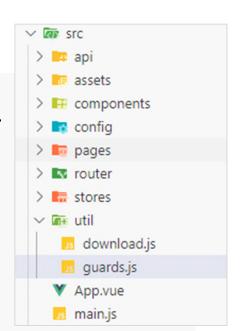


○ 라우팅 가드 처리

util/guards.js

```
import { useAuthStore } from '@/stores/auth';
* useAuthStore()를 바로 할 수 없다 => Store는 탑 레벨에서 호출 불가! => 아직 Store가 생성되기 전이기 때문이다.
export const isAuthenticated = (to, from) => {
    const auth = useAuthStore();

    if (!auth.isLogin) {
        console.log('로그인 필요.....'); => cr을 구성
        return { name: 'loqin', query: { next: to.name } };=> ?next=xxx를 구성하는 부분
    }
    console.log('로그인 인증');
};
```



router/board.js

```
import { isAuthenticated } from '@/util/guards';
export default [
   path: '/board/create',
   name: 'board/create',
   component: () => import('../pages/board/BoardCreatePage.vue'),
   beforeEnter: isAuthenticated,
   path: '/board/update/:no',
   name: 'board/update',
   component: () => import('../pages/board/BoardUpdatePage.vue'),
   beforeEnter: isAuthenticated,
```

=> 인증 객체가 아무것도 리턴 안하면 통과 뭔가를 리턴하면 중단하고 리턴으로 이동

pages/auth/LoginPage.vue

```
<script setup>
import { computed, reactive, ref } from 'vue';
import { useAuthStore } from '@/stores/auth';
import { useRoute, useRouter } from 'vue-router';
        => next라는 쿼리 파라미터가 있는지 확인하기 위함
const cr = useRoute();
const router = useRouter();
const auth = useAuthStore();
const login = async () => {
 try {
   await auth.login(member);
   if (cr.query.next) { // 로그인 후 이동할 페이지가 있는 경우
    router.push({ name: cr.query.next });
  } else { // 일반 로그인
    router.push('/');
 } catch (e) {
   error.value = e.response.data;
</script>
```