

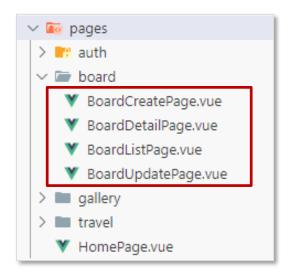
2025년 상반기 K-디지털 트레이닝

게시판 - 목록보기 (심화 1)

[KB] IT's Your Life



♡ 다음과 같이 게시판 페이지를 생성하세요.



- 각 페이지를 만들고, 타이틀만 출력되도록 준비
- ▽ router/board.js에 위에서 만든 페이지에 대한 라우팅을 정의하세요.

router/board.js

```
export default [
    path: '/board/list',
    name: 'board/list',
    component: () => import('../pages/board/BoardListPage.vue'),
 },
    path: '/board/detail/:no',
    name: 'board/detail',
    component: () => import('../pages/board/BoardDetailPage.vue'),
 },
    path: '/board/create',
    name: 'board/create',
    component: () => import('../pages/board/BoardCreatePage.vue'),
 },
    path: '/board/update/:no',
    name: 'board/update',
    component: () => import('../pages/board/BoardUpdatePage.vue'),
 },
```

- ♡ 다음과 같이 게시판 목록 보기 페이지를 정의하세요.
 - o boardApi에 목록 얻기 함수 정의

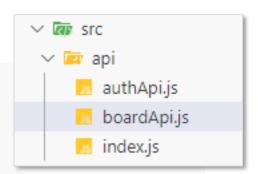
🛍 게시글 목록				
No	제목		작성자	작성일
127	페이징 테스트 110		user0	2024-07-31
126	페이징 테스트 109		user0	2024-07-31
125	페이징 테스트 108		user0	2024-07-31
124	페이징 테스트 107		user0	2024-07-31
123	페이징 테스트 106		user0	2024-07-31
122	페이징 테스트 105		user0	2024-07-31
121	페이징 테스트 104		user0	2024-07-31
120	페이징 테스트 103		user0	2024-07-31
119	페이징 테스트 102		user0	2024-07-31
118	페이징 테스트 101		user0	2024-07-31
		페이지 네이션		௴ 글 작성

✓ api/boardApi.js

```
import api from '@/api';

const BASE_URL = '/api/board';

export default {
  async getList(params) {
    const { data } = await api.get(BASE_URL, { params });
    console.log('BOARD GET LIST: ', data);
    return data;
  },
}
```



pages/board/BoardListPage.vue

```
<script setup>
import api from '@/api/boardApi';
import { ref, reactive, computed } from 'vue';
import moment from 'moment';
import { useRoute, useRouter } from 'vue-router';
const cr = useRoute();
const router = useRouter();
const page = ref({ });
const articles = computed(() => page.value);
const load = async () => {
 try {
    page.value = await api.getList();
    console.log(page.value);
  } catch {}
};
load();
</script>
```

pages/board/BoardListPage.vue

pages/board/BoardListPage.vue

```
{{ article.no }}
       <router-link :to="{ name: 'board/detail', params: { no: article.no }}">
             {{ article.title }}
          </router-link>
       {{ article.writer }}
       {{ moment(article.regDate).format('YYYY-MM-DD') }}
      <div class="my-5 d-flex">
    <div class="flex-grow-1 text-center">페이지 네이션</div>
    <div>
      <router-link :to="{ name: 'board/create' }" class="btn btn-primary">
         <i class="fa-solid fa-pen-to-square"></i> 글 작성</router-link>
    </div>
   </div>
 </div>
</template>
```



2025년 상반기 K-디지털 트레이닝

게시판 - 글 작성 (심화 1)

[KB] IT's Your Life



게시판 - 글작성

- ♡ 다음과 같이 글 작성 페이지를 작성하세요.
 - ㅇ 주의 사항
 - 제목을 입력한 경우에만 확인 버튼 활성화
 - 필요한 경우 백엔드 시큐리티 설정 추가



게시판 - 글 작성(백엔드)

security.config.SecurityInitializer.java

```
public class SecurityInitializer extends AbstractSecurityWebApplicationInitializer {
   // 문자셋 필터
   private CharacterEncodingFilter encodingFilter() {
       CharacterEncodingFilter encodingFilter = new CharacterEncodingFilter();
       encodingFilter.setEncoding("UTF-8");
       encodingFilter.setForceEncoding(true);
       return encodingFilter;
   @override
   protected void beforeSpringSecurityFilterChain(ServletContext servletContext) {
       insertFilters(servletContext, encodingFilter(), new MultipartFilter());
```

✓ api/boardApi.js

```
const headers = { 'Content-Type': 'multipart/form-data' };
export default {
 async create(article) {
   const formData = new FormData();
   formData.append('title', article.title);
   formData.append('writer', article.writer);
   formData.append('content', article.content);
   if (article.files) {
     for (let i = 0; i < article.files.length; i++) {
        formData.append('files', article.files[i]);
   const { data } = await api.post(BASE_URL, formData, { headers });
   console.log('BOARD POST: ', data);
   return data;
 },
```

```
<script setup>
import boardApi from '@/api/boardApi';
import { computed, reactive, ref } from 'vue';
import { useAuthStore } from '@/stores/auth';
import { useRoute, useRouter } from 'vue-router';
const auth = useAuthStore();
const router = useRouter();
const cr = useRoute();
const files = ref(null);
const article = reactive({
 writer: auth.username,
 title: '',
 files: null,
});
const disableSubmit = computed(() => !article.title);
```

```
const submit = async () => {
  if (!confirm('등록할까요?')) return;

  if (files.value.files.length > 0) { // 첨부파일 선택이 있는 경우
    article.files = files.value.files;
  }

  await boardApi.create(article);
  router.push('/board/list');
};
</script>
```

게시판 - 글작성

```
<template>
 <h1><i class="fa-regular fa-pen-to-square"></i> 글 작성</h1>
 <form @submit.prevent="submit">
   <div class="mb-3 mt-3">
      <label for="title" class="form-label"> 제목 </label>
     <input type="text" class="form-control" placeholder="제목" id="title" v-model="article.title" />
   </div>
   <div class="mb-3 mt-3">
      <label for="files" class="form-label"> 첨부파일 </label>
     <input type="file" class="form-control" placeholder="첨부파일" id="files" ref="files" multiple />
   </div>
   <div class="mb-3 mt-3">
      <label for="content" class="form-label"> 내용 </label>
      <textarea class="form-control" placeholder="내용" id="content" v-model="article.content" rows="10"></textarea>
   </div>
```



2025년 상반기 K-디지털 트레이닝

게시판 - 상세보기, 삭제 (심화 2)

[KB] IT's Your Life



- 💟 다음과 같이 상세 보기 페이지를 작성하세요.
 - o boardApi.js에 상세 보기, 삭제 함수 작성
 - ㅇ 주의 사항
 - 수정, 삭제 버튼은 로그인 사용자와 작성자가 일치하는 경우에만 출력
 - 첨부파일에 다운로드 처리
 - 삭제 버튼 클릭 시 삭제 기능



✓ api/boardApi.js

```
export default {
  async get(no) {
    const { data } = await api.get(`${BASE_URL}/${no}`);
    console.log('BOARD GET', data);
    return data;
 },
  async delete(no) {
    const { data } = await api.delete(`${BASE_URL}/${no}`);
    console.log('BOARD DELETE: ', data);
    return data;
 },
```

```
<script setup>
import { useRoute, useRouter } from 'vue-router';
import api from '@/api/boardApi';
import { ref } from 'vue';
import moment from 'moment';
import { useAuthStore } from '@/stores/auth';
import { downloadFile } from '@/util/download';
const auth = useAuthStore();
const cr = useRoute();
const router = useRouter();
const no = cr.params.no;
const article = ref({});
```

```
const back = () \Rightarrow \{
  router.push({ name: 'board/list' });
};
const update = () => {
  router.push({ name: 'board/update', params: { no: no } });
};
const remove = async () => {
  if (!confirm('삭제할까요?')) return;
  await api.delete(no);
  router.push({ name: 'board/list'});
};
const load = async () => {
  article.value = await api.get(no);
};
load();
</script>
```

```
<template>
 <h1>{{ article.title }}</h1>
  <div class="my-3 d-flex justify-content-between">
    <div>
     <i class="fa-solid fa-user"></i>
     {{ article.writer }}
    </div>
    <div>
      <i class="fa-regular fa-clock"></i>
      {{ moment(article.updateDate).format('YYYY-MM-DD HH:mm') }}
    </div>
  </div>
 <hr />
 <!-- 첨부 파일 목록 -->
  <div class="text-end">
    <div v-for="file in article.attaches" :key="file.no" class="attach">
      <span>
        <i class="fa-solid fa-paperclip"></i></i>
        {{ file.filename }}
      </span>
    </div>
  </div>
```

```
<div class="content">{{ article.content }}</div>
 <div class="my-5">
   <button class="btn btn-primary me-2" @click="back"><i class="fa-solid fa-list"></i> 목록</button>
   <!-- 로그인 사용자와 작성자가 같은 경우 -->
   <template v-if="auth.username == article.writer">
      <button class="btn btn-primary me-2" @click="update"><i class="fa-regular fa-pen-to-square"></i> 수정</button>
      <button class="btn btn-danger" @click="remove"><i class="fa-solid fa-trash-can"></i> 삭제</button>
   </template>
 </div>
</template>
<style scoped>
.attach {
 font-size: 0.8rem;
 cursor: pointer;
.content {
 white-space: pre-line;
</style>
```

♡ 첨부파일 다운로드 기능을 추가하세요.

util/download.js

```
import axios from 'axios';
export const downloadFile = async (fileUrl) => {
 try {
   const link = document.createElement('a'); // a 노드 생성
   link.href = fileUrl;
                                         // a노드의 href에 다운로드 파일 url 설정
   document.body.appendChild(link); // dom에 추가
   link.click();
                // click 이벤트 강제 발생
   document.body.removeChild(link); // dom에서 제거
 } catch (error) {
   console.error(error);
};
```

```
🗦 📴 api
 assets
 > III components
 > s config
 > lip pages
 > I router
 > 🔚 stores

✓ Image util

    download.js
   App.vue
   nain.js
```

```
<script setup>
import { downloadFile } from '@/util/download';
const download = async (no) => {
  const URL = '/api/board/download/' + no;
  await downloadFile(URL);
};
const load = async () \Rightarrow {
  article.value = await api.get(no);
  console.log('DETAIL', article.value);
};
load();
</script>
```

```
<template>
 <div class="text-end">
   <div v-for="file in article.attaches" :key="file.no" class="attach">
      <span @click="download(file.no)">
        <i class="fa-solid fa-paperclip"></i>
        {{ file.filename }}
      </span>
   </div>
 </div>
 <div class="content">{{ article.content }}</div>
</template>
```



2025년 상반기 K-디지털 트레이닝

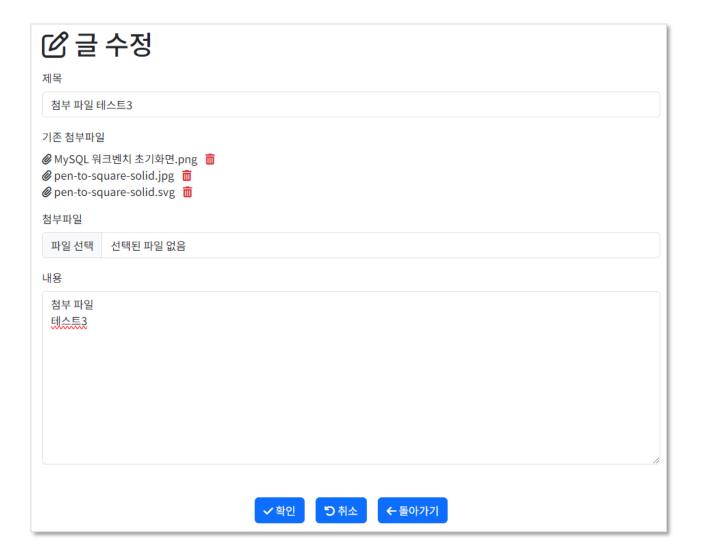
게시판 - 수정 (심화 2)

[KB] IT's Your Life



게시판 - 수정

- ♡ 다음과 같이 수정 페이지를 작성하세요.
 - o boardApi에 필요한 함수 추가
 - 기존 첨부 파일은 삭제만 제공
 - 첨부 파일 추가 가능



✓ api/boardApi.js

```
const headers = { 'Content-Type': 'multipart/form-data' };
export default {
 // 게시글 수정
 async update(article) {
   const formData = new FormData();
   formData.append('no', article.no);
   formData.append('title', article.title);
   formData.append('writer', article.writer);
   formData.append('content', article.content);
   if (article.files) { // 첨부파일이 있는 경우
     for (let i = 0; i < article.files.length; i++) {
       formData.append('files', article.files[i]);
   }
   const { data } = await api.put(`${BASE_URL}/${article.no}`, article, { headers });
   console.log('BOARD PUT: ', data);
   return data;
 },
```

게시판 - 수정

✓ api/boardApi.js

```
// 첨부파일 삭제
async deleteAttachment(no) {
  const { data } = await api.delete(`${BASE_URL}/deleteAttachment/${no}`);
  console.log('ATTACHMENT DELETE: ', data);
  return data;
},
};
```

```
<script setup>
import { ref, reactive } from 'vue';
import { useRoute, useRouter } from 'vue-router';
import boardApi from '@/api/boardApi';
const cr = useRoute();
const router = useRouter();
const no = cr.params.no;
const article = reactive({});
const attachments = ref([]);
const orgArticle = ref({});
const files = ref(null);
const back = () \Rightarrow \{
  router.push({ name: 'board/detail', params: { no } });
};
```

```
const removeFile = async (no, filename) => {
 if (!confirm(filename + '을 삭제할까요?')) return;
  await boardApi.deleteAttachment(no);
  const ix = attachments.value.findIndex((f) => f.no === no);
  attachments.value.splice(ix, 1);
};
const submit = async () => {
 if (!confirm('수정할까요?')) return;
 if (files.value.files.length > 0) {
    article.files = files.value.files;
 await boardApi.update(article);
  router.push({ name: 'board/detail', params: { no }, query: cr.query });
};
```

```
const reset = () => {
  article.no = orgArticle.value.no;
  article.title = orgArticle.value.title;
  article.writer = orgArticle.value.writer;
  article.content = orgArticle.value.content;
  console.log(article);
};
const load = async () => {
  const data = await boardApi.get(no);
  orgArticle.value = { ...data };
  attachments.value = data.attaches;
  reset();
};
load();
</script>
```

```
<template>
 <h1><i class="fa-regular fa-pen-to-square"></i> 글 수정</h1>
 <form @submit.prevent="submit">
   <div class="mb-3 mt-3">
     <label for="title" class="form-label"> 제목 </label>
     <input type="text" class="form-control" placeholder="제목" id="title" v-model="article.title" />
     <div class="invalid-feedback">제목은 필수 요소입니다.</div>
   </div>
   <div class="mb-3 mt-3">
     <label class="form-label"> 기존 첨부파일 </label>
     <div v-for="file in attachments" :key="file.no" class="attach">
      <i class="fa-solid fa-paperclip"></i> {{ file.filename }}
      </div>
   </div>
   <div class="mb-3 mt-3">
     <label for="files" class="form-label"> 첨부파일 </label>
     <input type="file" class="form-control" placeholder="첨부파일" id="files" ref="files" multiple />
   </div>
```

```
<div class="mb-3 mt-3">
      <label for="content" class="form-label"> 내용 </label>
     <textarea class="form-control" placeholder="내용" id="content" v-model="article.content" rows="10"></textarea>
   </div>
   <div class="my-5 text-center">
      <button type="submit" class="btn btn-primary me-3">
       <i class="fa-solid fa-check"></i> 확인
     </button>
      <button type="button" class="btn btn-primary me-3" @click="reset">
       <i class="fa-solid fa-undo"></i> 취소
     </button>
      <button class="btn btn-primary" @click="back">
       <i class="fa-solid fa-arrow-left"></i> 돌아가기
     </button>
   </div>
 </form>
</template>
<style>
.fa-trash-can { cursor: pointer; }
</style>
```