

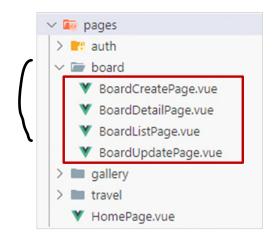
2025년 상반기 K-디지털 트레이닝

게시판 - 목록보기

[KB] IT's Your Life



💟 페이지 준비



○ 각 페이지를 만들고, 타이틀만 출력되도록 준비

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'),
```

☑ 게시판 목록 보기

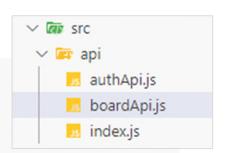
🛍 게시글 목록				
No	제목		작성자	작성일
127	페이징 테스트 110		user0	2024-07-31
126	페이징 테스트 109		user0	2024-07-31
125	페이징 테스트 108		user0	2024-07-31
124	페이징 테스트 107		user0	2024-07-31
123	페이징 테스트 106		user0	2024-07-31
122	페이징 테스트 105		user0	2024-07-31
121	페이징 테스트 104		user0	2024-07-31
120	페이징 테스트 103		user0	2024-07-31
119	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); V
const load = async () => {
 try {
   page.value = await api.getList();
   console.log(page.value);
 } catch {}
};
load();
</script>
```

Date객체는 json문자열로 표현할때 추가 동작이 필요 JACKSON라이브러리는 UTC값을 통해서 GSON라이브러리는 문자열을 통해서

pages/board/BoardListPage.vue

```
<template>
<div>
<h1 class="mb-3"><i class="fa-solid fa-paste"></i> 게시글 목록</h1>

<thead>

No
> M목
작성자
작성의
> 
작성일
> 
> 
< </th>
```

pages/board/BoardListPage.vue

```
{{ article.no }}
     <router-link :to="{ name: 'board/detail', params: { no: article.no }}"> //실제경로는 /board/detail/숫자
         {{ article.title }}
       </router-link>
     {{ article.writer }}
     {{ moment(article.regDate).format('YYYY-MM-DD') }}
    해석하고자하는 날짜 데이터 인자로
   <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

[KB] IT's Your Life

form 기반으로 바이너리 데이터를 보낼때 security filter config전에 multipart resolver가 먼저 반긴다.

이때 바이너리 데이터를 읽을 때 해석이 이미 다 끝난 상태즉 sercurity filter config의 문자셋 인코딩 필터 전에 해석이 이미 끝나서 깨져있다.

안깨지게 하려면 multipartresolver 앞에 문자셋 인코딩 필터를 위치시켜야 한다.

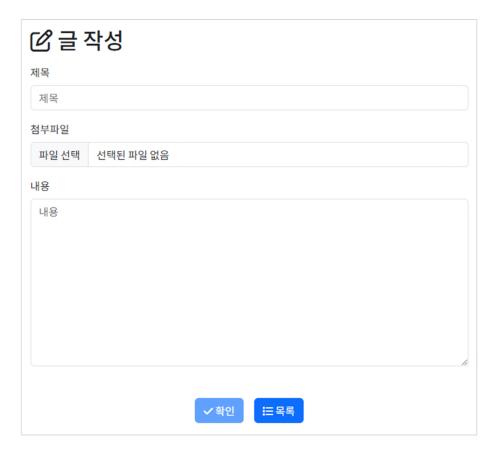


- ☑ Multipart Resolver를 Security Filter보다 앞에 설정해야 함.
 - o 문자 인코딩 필터를 Multipart Resolver 앞에 배치해야 함
 - WebConfig.java에 설정한 문자 인코딩 필터 제거
 - Filter[] getServletFilters() 메서드 제거

security.config.SecurityInitializer.java

☑ 글 작성

- ㅇ 주의 사항
 - 제목을 입력한 경우에만 확인 버튼 활성화



🗹 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(); \checkmark
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) {
    article.files = files.value.files;
  }

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

```
<template>
 <h1><i class="fa-regular fa-pen-to-square"></i> 글 작성</h1>
 <form @submit.prevent="submit">V
  <div class="mb-3 mt-3">
    <label for="title" class="form-label"> 제목 </label>
    <input type="text" class="form-control" placeholder="제목" id="title" 	extbf{v-model="article.title"} /> 	extbf{V}
  </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>
                                                                        자동으로 content 프러퍼티가
들어간다.
허용된다.
```



2025년 상반기 K-디지털 트레이닝

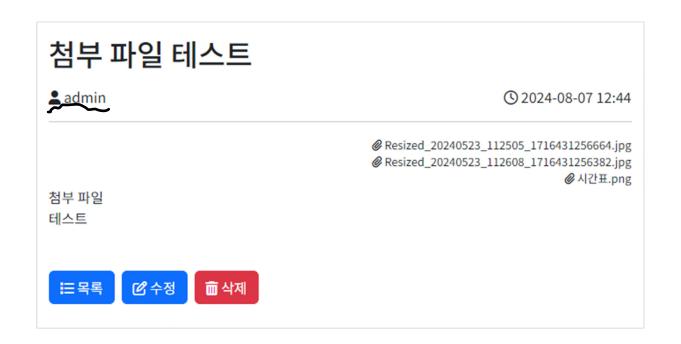
게시판 - 상세보기, 삭제

[KB] IT's Your Life



🕜 상세 보기

- 주의 사항
 - 수정, 삭제 버튼은 <u>로그</u>인 사용자와 작성자가 일치하는 경우에만 출력
 - <u>첨부파일에 다</u>운로드 처리



🗹 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 'uue';
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({});

?파라미터명=파라미터값&파라미터명2=파라미터값2...
```

```
const back = () = > {
 router.push({ name: 'board/list' });  v 이동
};
const update = () => {
 router.push({ name: 'board/update', params: { no: no } }); 이동시 쿼리스트링의 파라미터(라우터 파라미터)도 지정해주고
};
                                        아깐 템플릿에 썼지만 지금은 js로 쓴거죠?
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); V
load();
</script>
```

```
원래는 a태그의 href=다운로드경로 형식으로 다운로드를
                                                           유도했다.
<template>
 <h1>{{ article.title }}</h1>
                                                            하지만 spa에서는 a태그 사용 X.
 <div class="my-3 d-flex justify-content-between">
  <div>
                                                           is를 이용해야한다.
    <i class="fa-solid fa-user"></i>
                                                            2가지방법
   {{ article.writer }}
                                                           js가 ajax통신을 이용해서 직접 받는 방법. js가 바이너리데이터를
다루는 것은 까다롭다. blob처리와 buffer을 사용해야함
  </div>
  <div>
    <i class="fa-regular fa-clock"></i>
   {{ moment(article.updateDate).format('YYYY-MM-DD HH:mm') }}
  </div>
 </div>
                                                           다른 방법은 브라우저가 다운로드를 하게끔.
a태그를 사용하돼 js로 동적 실행시키게끔 하면
가능하다. a태그는 사용자한테 안보이게끔
 <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>
     {{ 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; css에서 세밀하게 공백을 지정할 수 있다.
엔터, 스페이스바 등등. textarea에 입력된 데이터를
</style>
                     다뤀때 활용함
```

☑ 첨부파일 다운로드

- o
 - a 태그를 직접 사용하면 SPA 아키텍처가 무너짐
- JS에서 프로그램적으로 처리해야 함

util/download.js

```
import axios from 'axios';
                                              아예 생성하고 실행하고 제거하는것을
동적으로 오오
export const downloadFile = async (fileUrl) => {
 try {
  const link = <u>document</u>.createElement('a');// a 노드 생성<sup>V</sup>V
  link.href = fileUrl; 🗸
                                         // a노드의 href에 다운로드 파일 url 설정
                                                        돔 끝부분에 a태그 추가됨
  document.body.appendChild(link); // dom에 추가
  link.click();
                               // click 이벤트 강제 발생Ⅴ
  document.body.removeChild(link); // dom에서 제거 V
 } catch (error) {
                                                          다운로드하는 동안
잠깐만 존재하게끔!!
  console.error(error);
};
```

```
**script setup>
import { downloadFile } from '@/util/download';
...

const download = async (no) => {
    const URL = '/api/board/download/' + no;
    await downloadFile(URL);
};

const load = async () => {
    article.value = await api.get(no);
    console.log('DETAIL', article.value);
};

load();
    </script>
```

pages/board/BoardDetailPage.vue

작업을 프론트에서 하면은 js로 구현하면 되고 백에서 하면은 java로 구현하고 그 결과를 json으로 보내면 된다!



2025년 상반기 K-디지털 트레이닝

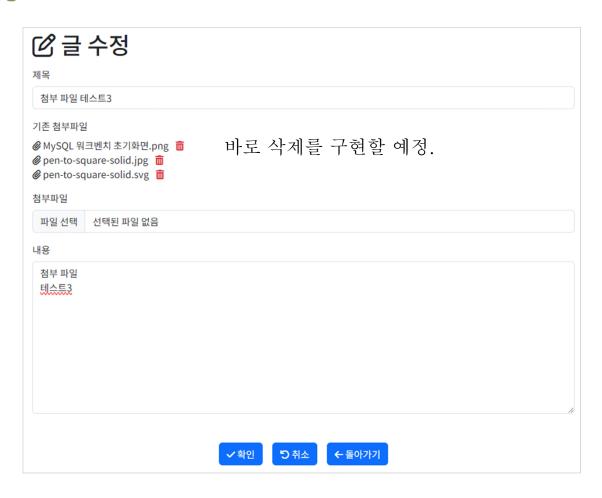
게시판 - 수정

[KB] IT's Your Life



🕜 수정 기능

- 기존 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; <math>i++) {
      formData.append('files', article.files[i]);
                                                                         formData
   const { data } = await api.put(`${BASE_URL}/${article.no}`, article, { headers });
   console.log('BOARD PUT: ', data);
   return data;
```

🗹 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({}); 원문
const attachments = ref([]); 첨부파일목록
const orgArticle = ref({});위규
const files = ref(null);
const back = () = > {
 router.push({ name: 'board/detail', params: { no } });상세보기로 돌아갈수있는
};
```

```
const removeFile = async (no, filename) => { 첨부파일의 번호와 이름 인자로
  if (!confirm(filename + '을 삭제할까요?')) return;
√await boardApi.deleteAttachment(no);
┍ const ix = attachments.value.findIndex((f) => f.no === no); 조건에 참을 나타내느 첫번째 인덱스 반환
 Lattachments.value.splice(ix, 1); 인덱스/삭제개수
 };
 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 }); 상세보기로 돌아가는.
                                 /경로/{no},?page=1&ref=2 근데 페이지 네이션에 쓰일
쿼리파라미터 정보와
 };
                                                                라우트 파라미터 정보 와 함께
```

```
const reset = () => {
    article.no = orgArticle.value.no;
    article.title = orgArticle.value.writer;
    article.writer = orgArticle.value.writer;
    article.content = orgArticle.value.content;
    console.log(article);
};

const load = async () => {
    const data = await boardApi.get(no);
    P본데이터 얻고
    orgArticle.value = { ...data }; original value
    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 }}
     <i class="fa-solid fa-trash-can text-danger ms-2" @click="removeFile(file.no, file.filename)"></i>
    </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>
```