

2025년 상반기 K-디지털 트레이닝

게시판 – 페이지네이션 (프론트엔드) (심화 1)

[KB] IT's Your Life

- ✓ **페이지네이션 라이브러리를 설치하고, 초기화 하세요.**
 - `npm i vue-awesome-paginate`

main.js

```
import './assets/main.css';
import 'bootstrap/dist/css/bootstrap.css';
import 'vue-awesome-paginate/dist/style.css';

import { createApp } from 'vue';
import { createPinia } from 'pinia';
import VueAwesomePaginate from 'vue-awesome-paginate';

import App from './App.vue';
import router from './router';

const app = createApp(App);

app.use(VueAwesomePaginate);
app.use(createPinia());
app.use(router);

app.mount('#app');
```

- ✓ main.css에 페이지네이션 스타일을 다음과 같이 정의하세요.

⏮ ◀ 1 ... 3 4 5 6 7 ... 12 ▶ ⏭

assets/main.css

```
...
/* 페이지 네이션 */
.pagination-container {
  display: flex;
  /* column-gap: 10px; */
}

.paginate-buttons {
  height: 40px;
  width: 40px;
  /* border-radius: 20px; */
  cursor: pointer;
  background-color: white; /*rgb(242, 242, 242);*/
  border: 1px solid white; /* rgb(217, 217, 217);*/
  color: black;
}

.paginate-buttons:hover {
  border-radius: 20px;
  background-color: #d8d8d8;
}
```

assets/main.css

```
.active-page {  
  border-radius: 20px;  
  background-color: #3498db;  
  border: 1px solid #3498db;  
  color: white;  
}  
  
.active-page:hover {  
  background-color: #2988c8;  
}
```

2025년 상반기 K-디지털 트레이닝


게시판 – 페이지네이션 (프론트엔드) (심화 2)

[KB] IT's Your Life

✓ 다음 그림과 같이 페이지네이션을 적용한 목록보기 화면을 작성하세요.



- 다른 페이지로 이동할 때, 페이지네이션 쿼리 파라미터를 같이 전달

<http://localhost:5173/board/list?page=4&amount=10>

 **게시글 목록**

(총 120건)

No	제목	작성자	작성일
97	페이징 테스트 80	user0	2024-07-31
96	페이징 테스트 79	user0	2024-07-31
95	페이징 테스트 78	user0	2024-07-31
94	페이징 테스트 77	user0	2024-07-31
93	페이징 테스트 76	user0	2024-07-31
92	페이징 테스트 75	user0	2024-07-31
91	페이징 테스트 74	user0	2024-07-31
90	페이징 테스트 73	user0	2024-07-31
89	페이징 테스트 72	user0	2024-07-31
88	페이징 테스트 71	user0	2024-07-31

  1 2 3 **4** 5 6 ... 12  

 글 작성

BoardListPage.vue

```
<script setup>
import api from '@api/boardApi';
import { ref, reactive, computed, watch } 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.list);

const pageRequest = reactive({
  page: parseInt(cr.query.page) || 1,
  amount: parseInt(cr.query.amount) || 10,
});
```

BoardListPage.vue

```
// 페이지네이션 페이지 변경
const handlePageChange = async (pageNum) => {
  // url 변경 --> query 파트만 변경되므로 컴포넌트가 다시 마운트되지 않음
  // watch를 통해 cr이 변경됨을 감지하여 페이지 로드해야 함
  router.push({
    query: { page: pageNum, amount: pageRequest.amount },
  });
};

// pageRequest의 값 변경된 경우 호출
watch(cr, async (newValue) => {
  console.log('WATCH', cr.query.page);
  pageRequest.page = parseInt(cr.query.page);
  pageRequest.amount = parseInt(cr.query.amount);
  await load(pageRequest);
});
...

</script>
```

BoardListPage.vue

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

    <div class="mt-5 text-end">(총 {{ page.totalCount }}건)</div>
    <table class="table table-striped">
      <thead>
        <tr>...</tr>
      </thead>
      <tbody>
        <tr v-for="article in articles" :key="article.no">
          <td>{{ article.no }}</td>
          <td>
            <router-link :to="{ name: 'board/detail', params: { no: article.no }, query: cr.query }">
              {{ article.title }}
            </router-link>
          </td>
          <td>{{ article.writer }}</td>
          <td>{{ moment(article.regDate).format('YYYY-MM-DD') }}</td>
        </tr>
      </tbody>
    </table>
```

BoardListPage.vue

```
<div class="my-5 d-flex">
  <div class="flex-grow-1 text-center">
    <!-- 페이지 네이션 -->
    <vue-awesome-paginate
      :total-items="page.totalCount"
      :items-per-page="pageRequest.amount"
      :max-pages-shown="5"
      :show-ending-buttons="true"
      v-model="pageRequest.page"
      @click="handlePageChange"
    >
      <template #first-page-button><i class="fa-solid fa-backward-fast"></i></template>
      <template #prev-button><i class="fa-solid fa-caret-left"></i></template>
      <template #next-button><i class="fa-solid fa-caret-right"></i></template>
      <template #last-page-button><i class="fa-solid fa-forward-fast"></i></template>
    </vue-awesome-paginate>
  </div>
```

BoardListPage.vue

```
<div>
  <router-link :to="{ name: 'board/create', query: cr.query }" class="btn btn-primary">
    <i class="fa-solid fa-pen-to-square"></i> 글 작성</router-link>
  </div>
</div>
</div>
</template>
```

BoardCreatePage.vue

```
<script setup>
import boardApi from '@api/boardApi';
import { reactive } from 'vue';
import { useAuthStore } from '@stores/auth';
import { useRoute, useRouter } from 'vue-router';

const auth = useAuthStore();
const router = useRouter();
const cr = useRoute();

const article = reactive({
  writer: auth.username,
});

const submit = async () => {
  if (!confirm('등록할까요?')) return;
  await boardApi.create(article);
  router.push('/board/list');
};
</script>
```

BoardCreatePage.vue

```
<template>
  <h1><i class="fa-regular fa-pen-to-square"></i> 글 작성</h1>

  <form class="was-validated" @submit.prevent="submit">
    ...

    <div class="my-5 text-center">
      <button type="submit" class="btn btn-primary me-3">
        <i class="fa-solid fa-check"></i>
        확인
      </button>

      <router-link class="btn btn-primary" :to="{ name: 'board/list', query: cr.query }">
        <i class="fa-solid fa-list"></i>
        목록
      </router-link>
    </div>
  </form>
</template>
```

BoardDetailPage.vue

```
<script setup>
import { useRoute, useRouter } from 'vue-router';
import api from '@api/boardApi';
...
const auth = useAuthStore();
const cr = useRoute();
const router = useRouter();
...
const back = () => {
  router.push({ name: 'board/list', query: cr.query });
};

const update = () => {
  router.push({ name: 'board/update', params: { no: no }, query: cr.query });
};

const remove = async () => {
  if (!confirm('삭제할까요?')) return;
  await api.delete(no);
  router.push({ name: 'board/list', query: cr.query });
};
...
```


BoardUpdatePage.vue

```
<script setup>
...
const cr = useRoute();
const router = useRouter();
...

const back = () => {
  router.push({ name: 'board/detail', params: { no }, query: cr.query });
};
...

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 });
};
...
```