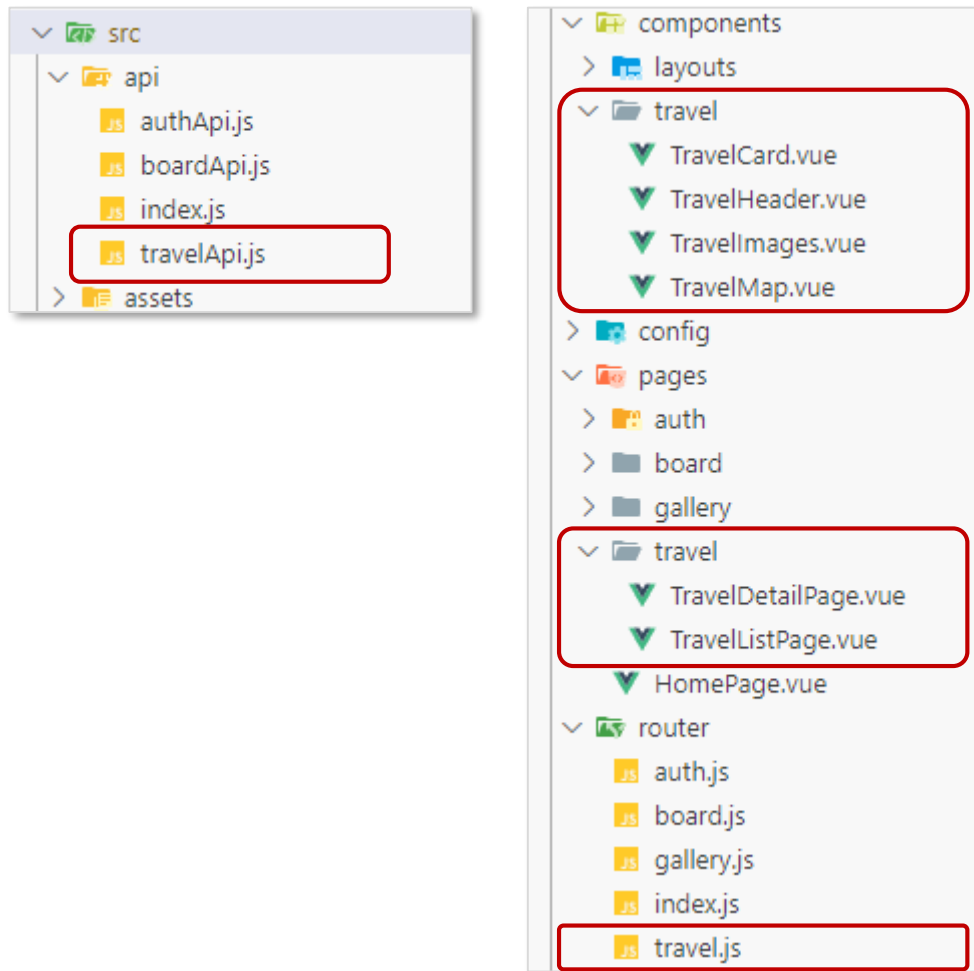


2025년 상반기 K-디지털 트레이닝

Travel – 프론트엔드 (심화 1)

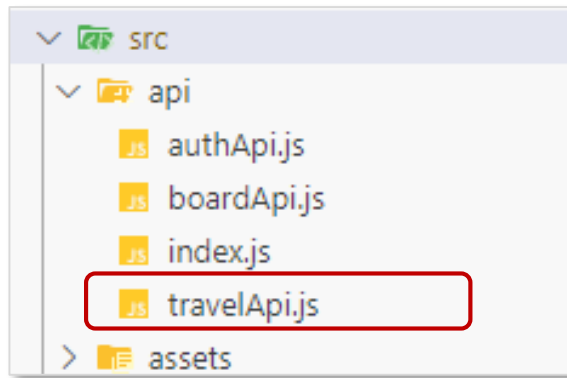
[KB] IT's Your Life

- ✓ 프론트엔드 프로젝트에 Travel을 위한 구성을 추가하세요.



✓ Travel API를 위한 travelApi.js를 정의하세요.

- 페이지별 목록 얻기
- 한 개의 Travel 얻기



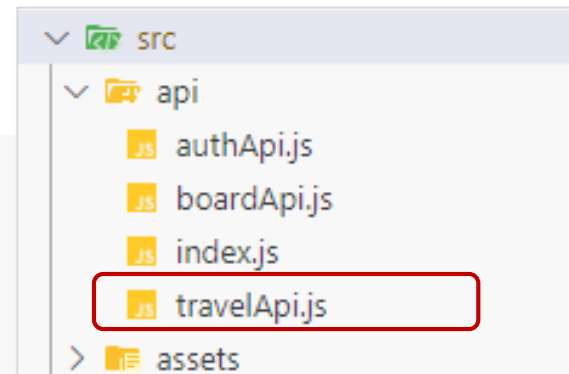
travelApi.js

```
import api from '@api';

const BASE_URL = '/api/travel';

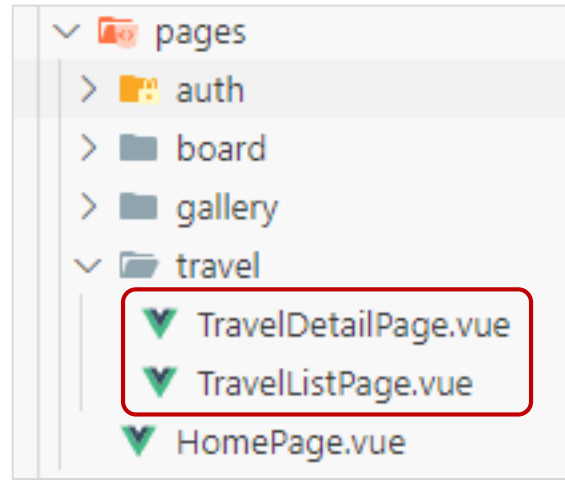
export default {
  async getList(params) {
    const { data } = await api.get(BASE_URL, { params });
    console.log('TRAVEL GET LIST: ', data);
    return data;
  },

  async get(no) {
    const { data } = await api.get(`${BASE_URL}/${no}`);
    console.log('TRAVEL GET', data);
    return data;
  },
};
```

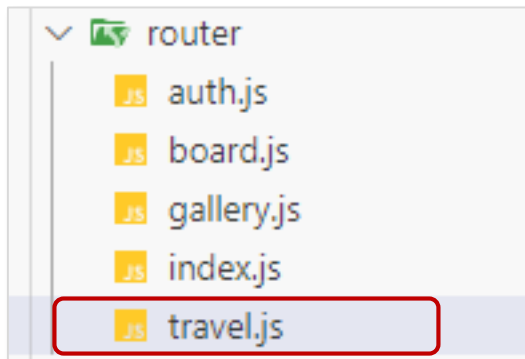


✓ 다음과 페이지 컴포넌트를 정의하세요.

- TravelListPage.vue
- TravelDetailPage.vue
 - 기본 골격(타이틀)만 준비

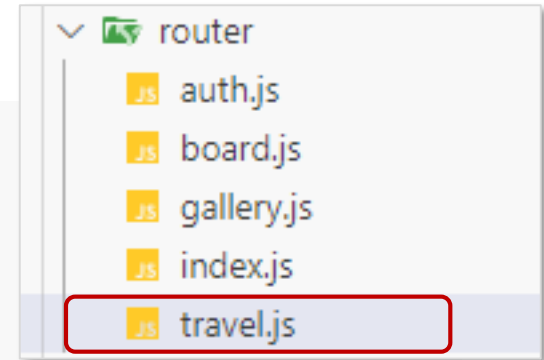


✓ 해당 페이지를 위한 라우트를 정의하세요.



travel.js

```
export default [  
  {  
    path: '/travel/list',  
    name: 'travel/list',  
    component: () => import('../pages/travel/TravelListPage.vue'),  
  },  
  {  
    path: '/travel/detail/:no',  
    name: 'travel/detail',  
    component: () => import('../pages/travel/TravelDetailPage.vue'),  
  },  
];
```



2025년 상반기 K-디지털 트레이닝

Travel – 프론트엔드 (심화 2)

[KB] IT's Your Life

✓ 다음 기능을 가지는 Travel 목록 보기 페이지 TravelListPage.vue를 정의하세요.

○ 카드로 1개 여행지 출력



강릉 경포 해변

동해안 최대 해변으로 유명하며 강문동, 안현동에 있고 시내에서 북으로 6km, 경포대에서 1km되는 곳에 동해의 ...

○ 반응형 웹 구성(그리드)

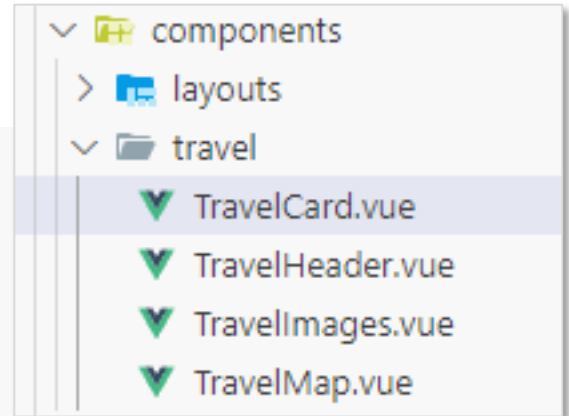
▪ 화면 폭에 따라서 1, 2, 3, 4 컬럼 개 수 조정

○ 이미지나 타이틀을 클릭하면 상세 보기 페이지로 이동

TravelCard.vue

```
<script setup>
import { useRoute } from 'vue-router';
const cr = useRoute();

const props = defineProps({
  travel: { Type: Object, required: true },
});
</script>
```



TravelCard.vue

```
<template>
  <div class="card" style="width: 100%">
    <router-link :to="{ name: 'travel/detail', params: { no: travel.no }, query: cr.query }">
      
    </router-link>

    <div class="card-body">
      <h5 class="card-title">
        <router-link :to="{ name: 'travel/detail', params: { no: travel.no }, query: cr.query }">
          {{ travel.title }}
        </router-link>
      </h5>
      <p class="card-text">{{ travel.description.substring(0, 250) }}</p>
    </div>
  </div>
</template>
```



강릉 경포 해변

동해안 최대 해변으로 유명하며 강문동, 안현동에 있고 시내에서 북으로 6km, 경포대에서 1km되는 곳에 동해의 ...

TravelCard.vue

```
<style>
.card-img-top {
  height: 200px;
  object-fit: cover;
}

.card-text {
  overflow: hidden;
  text-overflow: ellipsis;
  display: -webkit-box;
  -webkit-line-clamp: 2;
  -webkit-box-orient: vertical;
}

a {
  text-decoration: none;
}
</style>
```

TravelListPage.vue

```
<script setup>
import api from '@api/travelApi';
import { ref, reactive, computed, watch } from 'vue';
import { useRoute, useRouter } from 'vue-router';

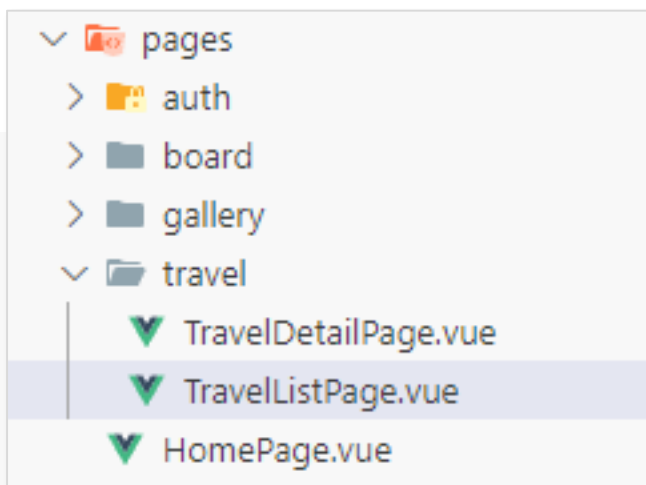
import TravelCard from '@components/travel/TravelCard.vue';

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

const page = ref({});

const travels = computed(() => page.value.list);

const pageRequest = reactive({
  page: parseInt(cr.query.page) || 1,
  amount: parseInt(cr.query.amount) || 12,
});
```



TravelListPage.vue

```
// 페이지네이션 페이지 변경
const handlePageChange = async (pageNum) => {
  router.push({
    query: { page: pageNum, amount: pageRequest.amount },
  });
};

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

const load = async (query) => {
  try {
    page.value = await api.getList(query);
    console.log(page.value);
  } catch {}
};

load(pageRequest);
</script>
```

TravelListPage.vue

```
<template>
  <div>
    <h1 class="mb-3"><i class="fa-solid fa-person-walking-luggage"></i> 여행지 목록</h1>

    <div class="">총 {{ page.totalCount }}건</div>
    <div class="row">
      <div class="col-sm-12 col-md-6 col-lg-4 col-xl-3 mb-3" v-for="travel in travels" :key="travel.no">
        <travel-card :travel="travel" />
      </div>
    </div>

    <div class="mt-5 flex-grow-1 text-center">
      <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>
  </div>
</template>
```