

React

With Next.js 

0. 프론트엔드 입문하기

연사 & 멘토 소개

연사 : 박승범 (@killerwhale)

멘토 : 박승범 (@killerwhale)

멘토 : 박승범 (@killerwhale)

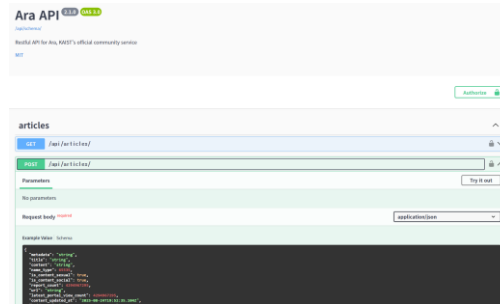
프론트엔드

프론트엔드는 사용자에게 보여지는 화면을 개발하는 것



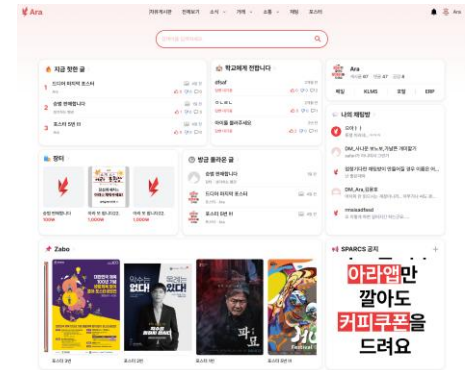
DB

BE



API

FE

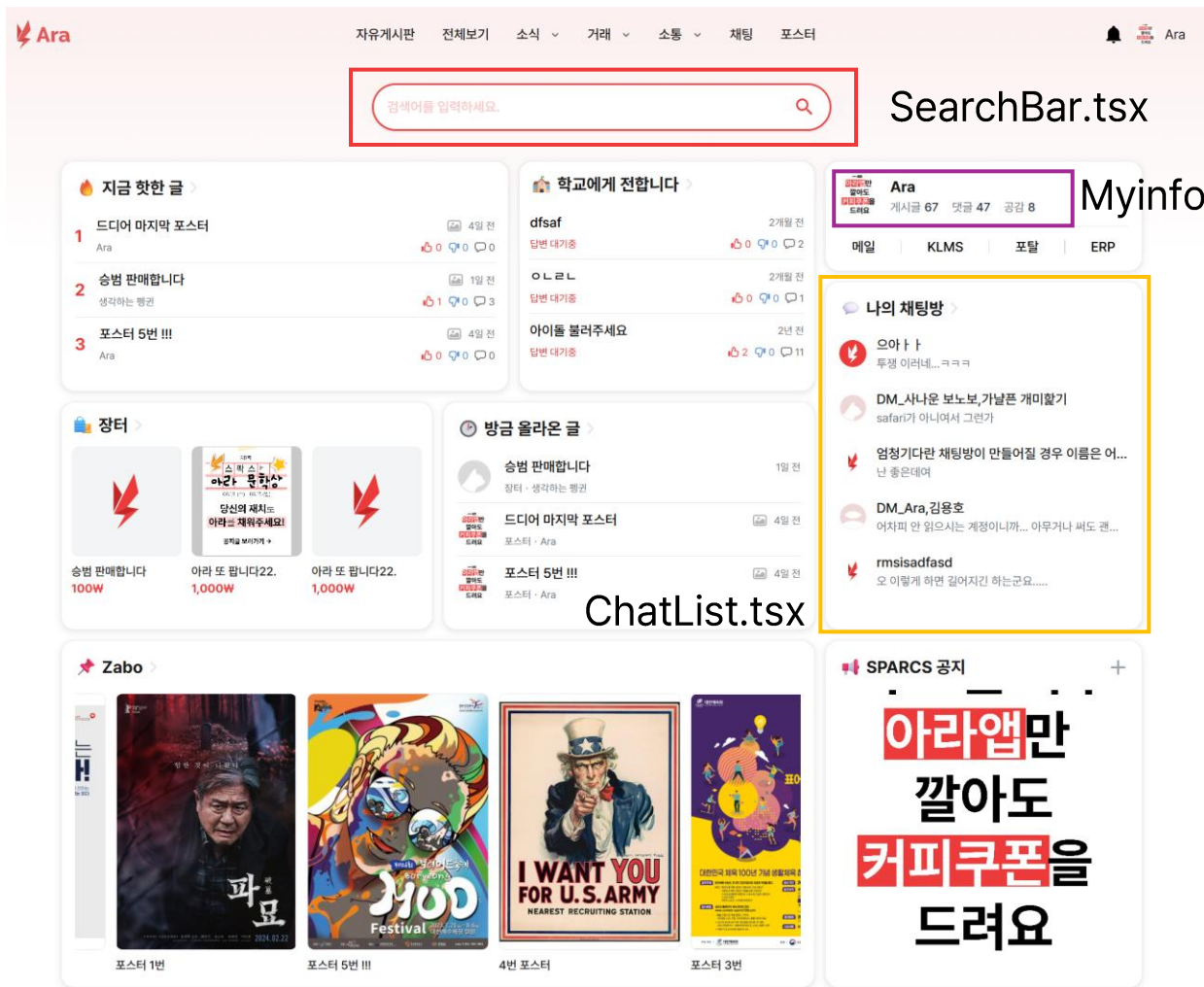


UI

백엔드가 DB로 부터 API를 만든다면,
프론트는 API로 부터 UI를 만들고, 로직을 구현한다.

컴포넌트

프론트엔드 개발에서의 UI 혹은 기능의 단위



컴포넌트들을 모아 더 큰 하나의 컴포넌트를 만들 수 있다.

여러 컴포넌트들을 모아 하나의 페이지가 만들어진다.

컴포넌트, 어떤 기준으로 나눠야 하는가?

컴포넌트를 나누는 기준의 정답은 없다. 컴포넌트를 나누는 방식은 프로젝트 아키텍처의 차이이다.

Ara : Container-Presentational Pattern

로직을 담당하는 컴포넌트와 UI를 담당하는 컴포넌트를 분리

+) 재사용성 기준으로 컴포넌트 나누기

OTL : Design System

디자인 일관성을 위해 버튼, 카드와 같은 요소들을 컴포넌트화

Clubs : Atomic Architecture

가능한 최소의 단위로 컴포넌트 나누기

컴포넌트를 만들기 위한 단계

컴포넌트 어떤 것들을 작업해 주어야 할까?

1. UI (User Interface)
2. 컴포넌트 내부의 Logic & Event
3. 컴포넌트간의 상호작용
4. API와의 상호작용

Next.js 시작하기 - npm

1. 아래의 링크를 통해 Node.js 설치하기 (LTS)

<https://nodejs.org/ko/>

2. Next 프로젝트 설정하기

-> 물어보는 질문에는 권장 설정 사용하기

3. Next 프로젝트 설정하기

-> localhost:3000 으로 접속하기



```
> node -v  
> npm -v  
> npx create-next-app@latest (프로젝트 폴더 이름)  
> cd (프로젝트 폴더 이름)  
> npm run dev
```

Next.js 의 라우팅

폴더 구조 기반 라우팅

1. 기본적으로 src/app/page.tsx 가 메인 페이지
2. src/app/{페이지 이름}/page.tsx = localhost:3000/{페이지 이름}
3. page.tsx 파일 기준으로 라우팅 (page.tsx 파일이 없는 경우 라우팅에 해당 X)
4. src/app/[id]/page.tsx 와 같이 동적 라우팅 가능

02. State & hook 에서 자세히 다뤄요!

Tailwind CSS

Next.js 는 기본적으로 tailwind css를 사용한다.

Normal CSS vs tailwind CSS

```
/* style.css */  
.btn {  
  background: blue;  
  color: white;  
  padding: 8px 16px;  
  border-radius: 8px;  
}
```

html

```
<button class="btn">버튼</button>
```



```
<button class="bg-blue-500 text-white px-4 py-2 rounded-lg">  
  버튼  
</button>
```

컴포넌트 만들어보기

page.tsx에 버튼 컴포넌트를 하나 만들어 봅시다!

```
"use client"; // 버튼은 클라이언트 컴포넌트여야 함
```

```
import { useState } from "react";
```

Import 문

```
export default function ToggleButton() {
```

컴포넌트 이름 + props

```
  const [active, setActive] = useState(false);
```

```
  return (  
    <button  
      onClick={() => setActive(!active)}  
      className={`px-4 py-2 rounded-lg text-white transition  
        ${active ? "bg-green-500" : "bg-gray-500"}  
        hover:opacity-80`}  
    >  
      {active ? "활성화됨" : "비활성화됨"}  
    </button>  
  );  
}
```

컴포넌트 이름 (함수 이름)은
반드시 파스칼 케이스 (대문자 시작)

Const & hook
컴포넌트 내부 상태 or 생명주기

컴포넌트 UI (return)

Next.js 의 파일들

1. layout.tsx : 전체 페이지에 적용될 레이아웃 설정 (ex : navbar, footer, font 등)

```
// WebView 페이지 : 기본 NavBar와 Footer를 사용하지 않음.
if (pathname.startsWith("/web_view")) {
  return (
    <html lang="ko">
      <head>
        <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
      </head>
      <body className="h-screen">
        <main className="h-full">{children}</main>
      </body>
    </html>
  );
}

// 일반 페이지 : NavBar와 Footer포함해서 렌더링
return (
  <html lang="ko">
    <head>
      <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet" />
    </head>
    <body>
      <NavBar />
      <main>{children}</main>
      <Footer />
    </body>
  </html>
);
}
```

Next.js 의 파일들

2. global.css : 전역으로 사용될 css 클래스를 정의

```
/* 스크롤바 숨김(스크롤은 가능) */
.no-scrollbar {
  -ms-overflow-style: none;
  /* IE/Edge */
  scrollbar-width: none;
  /* Firefox */
}

.no-scrollbar::-webkit-scrollbar {
  /* Chrome/Safari */
  display: none;
}

/* 메인 페이지 컴포넌트들의 그림자 */
.main-page-block-shadow {
  border-radius: 16px;
  box-shadow: 0 2px 8px 0 rgba(0, 0, 0, 0.10);
}
```

스크롤바 숨김과 같은 css의 경우
다양한 컴포넌트에서 사용되므로
Global.css 에 별도의 클래스로 정의해 두기도 한다.

정의된 스타일은 별도의 import 없이 간편하게 사용할 수 있어요!

```
<div className="max-h-64 min-h-64 overflow-y-auto no-scrollbar">
  {loading && <div className="text-gray-400 text-sm text-center py-4">검색 중...</div>}
  {!loading && users.length === 0 && (
    <div className="text-gray-400 text-sm text-center py-4">검색 결과가 없습니다.</div>
  )}
  {users.map(user => (
    <div key={user.user} className="relative group">
      <div className="w-full flex items-center text-left px-2 py-2 rounded">
        <Image
          src={user.picture}
          alt={user.nickname}
          width={32}
          height={32}
          className="rounded-full object-cover aspect-square mr-2"
        />

```

Next.js 의 파일들

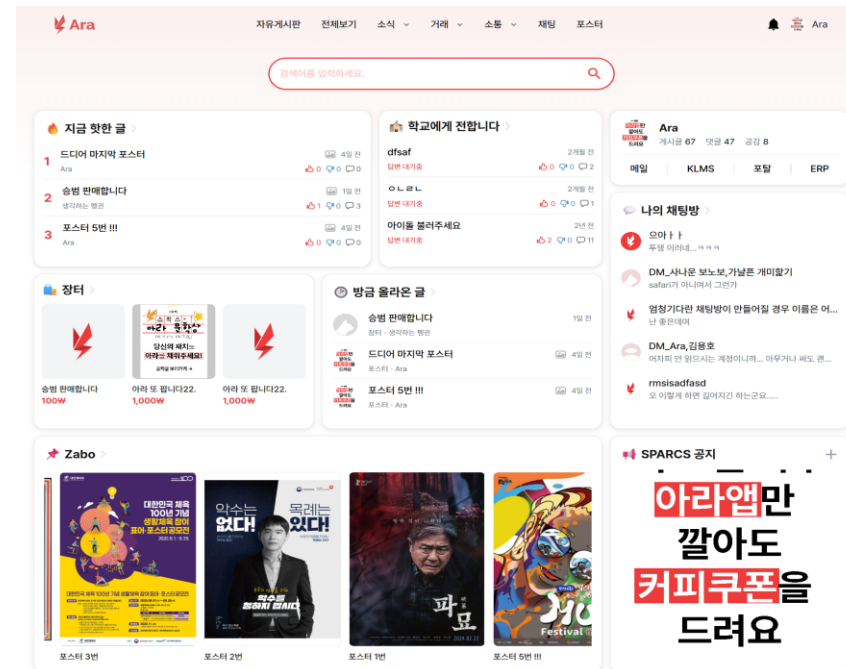
3. /public 폴더 : 전역으로 사용되는 이미지들 (ex. 아이콘) 등을 넣는 폴더

4. next.config.js : Next Js 설정파일 / tailwind.config.ts : 테일윈드 설정파일

플러그인 or 커스텀 설정을 정의한 파일

프론트엔드 개발자의 장점!

1. **삭막하지 않다!** (아래 2개는 동일한 서비스 입니다.)

[illegible]

백엔드 개발자들이 만드는 것.

프론트엔드 개발자들이 만드는 것.

2. Vibe 코딩이 상대적으로 쉽다.

~~3. 막강한 권력을 갖는다.~~

Homework

Tailwind CSS와 친해지기 (Optional)

The End

감사합니다