

Frontend 필수개념

3회차 recap

1. Lighthouse
2. Performance
3. Profiler

Server-Side Rendering vs Client-Side Rendering

1. 웹개발의 역사를 이해하면 조금 도움됨

2. Static

- a. 이미 만들어진 html파일들을 불러오는 것
- b. /home, /list, /mypage 등등에 접근할 때마다 해당 html파일을 호출해서 가져옴

3. AJAX

- a. json으로 데이터를 주고받을 수 있게 됨
- b. HTML파일을 가져오고,
- c. JavaScript를 활용해서 서버에 데이터를 호출
- d. 데이터를 json으로 받아와서 화면에 보여줌
- e. SPA 등장

Client Side Rendering

1. 서버에서 `index.html`을 클라이언트에게 전송
 - a. `index.html`에는 아무것도 없음
 - a. `main.tsx` (원래는 app.js) 에서 application에서 필요한 소스코드를 불러옴
2. 장점
 - a. 요즘 트렌드
 - b. 개발 편함
3. 단점
 - a. 소스코드가 커서 첫 로딩이 오래걸림
 - b. SEO에 안좋은 왜냐면 `index.html`이 비어있기 때문에

```
<body>
  <div id="root"></div>
  <script type="module" src="/src/main.tsx"></script>
</body>
```

Server Side Rendering

1. Static이 사실 Server Side Rendering
2. 서버에 접근하면 `index.html`을 만들어서 전송함
 - a. 비어있지 않아서 CSR보다 렌더링 빠름
 - b. 자바스크립트 코드도 같이 보내줌
3. 장점
 - a. 초기 로딩이 빠르다는 장점
 - b. index.html에 이것저것 들어있어서 SEO에 좋음
4. 단점
 - a. `index.html`을 서버에서 다시 불러오는 것
 - b. 서버에 지속적인 요청을 보내기 때문에 서버에 부하가 걸릴 수 있음
 - c. 동적인 JavaScript가 붙을때까지 interactive하지 않음

Static Site Generation

1. React + Gatsby
2. `index.html`들을 생성해서 서버에 미리 배포하는 것
3. JavaScript 파일을 얹어서 주면 동적으로도 작동할 수 있음

Frontend MSA

1. 하나의 프로젝트를 작은 단위로 나누는 것
2. Monolithic vs MSA
 - a. 하나가 다 처리하느냐
 - b. 여러개로 나눠서 처리하느냐
3. 어느정도 규모(?)가 있는 서비스가 아니라면 MSA는 무의미함

Backend for Frontend

1. Backend를 Frontend에 맞추는 것
2. MSA(Micro-service Architecture)가 주목 받으면서 등장함
3. Web, Mobile, Desktop 등 프론트엔드의 니즈에 맞게 서버를 따로 구축함
4. 읽을거리
 - a. <https://fe-developers.kakaoent.com/2022/220310-kakaopage-bff/>

프론트엔드 디자인 패턴

1. 요즘은 거의 atomic + react-query
2. 디자인 패턴 - <https://yozm.wishket.com/magazine/detail/1663/>
3. Atomic - <https://yozm.wishket.com/magazine/detail/1531/>
4. 최선의 디자인 패턴은 없음
 - a. 필요에 맞게
 - b. 컨벤션에 맞게
 - c. `요즘 이게 힙하니까 이걸 따른다`는 하지 않아도 됨

테스트코드

1. Unit Test

- a. 함수, 컴포넌트가 독립적으로 잘 작동하는지 확인하는 것
- b. 디펜던시 신경 안쓰고 이게 잘 되나요??

2. Integration Test

- a. 여러 함수들 또는 여러 컴포넌트들이 같이 잘 동작하는지 확인하는 것

3. E2E test

- a. 사용자인것마냥 테스트 하는 것

[아하!모먼트] 취업/이직을 준비하는 나만의 방법

1. 분기별 이력서 업데이트

- a. 이직을 하지 않더라도 평가시즌/연봉협상 할 때 좋음
- b. 내가 어떤 업무를 했는지 실적 위주로 기록
 - i. 단순히 뭘 했다 보다는 이걸 해서 뭐가 어떻게 좋아졌다

2. 이력서 커스텀

- a. 지원하는 회사 공고를 보고 회사에 맞게 내용을 수정
 - i. 회사별로 중요하게 보는게 다르기 때문
 - ii. 해당 회사에서 중요하게 생각하는 역량을 어필하기 위함

3. 이력서 공유

- a. 원티드 등 플랫폼에 이력서 공유
- b. '좋은 제안이 오면 이직할 생각이 있다'