

01-2. Web ApplicationOverview

Overview

프론트 웹 애플리케이션의 사상은 브라우저를 OS 로 보는 것이다.

- 애플리케이션을 실행시켜 주는 것을 흔히 OS 혹은 플랫폼이라고 부릅니다. 프런트 웹 애플리케이션은 브라우저에 의해 실행됩니다.
- 그럼으로 브라우저가 웹 애플리케이션 입장에서 보면 OS 이기도 하고 플랫폼이기도 합니다.
- 웹 애플리케이션의 기본 사상은 브라우저를 애플리케이션이 실행되는 OS 로 보고 이 브라우저 OS 에서 동작하는 애플리케이션을 만들겠다는 것입니다.
- 그럼으로 웹 애플리케이션은 브라우저가 이해할 수 있는 HTML, CSS, 자바스크립트로 만들어 지는 것입니다.

Standalone Application

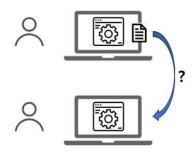
- Standalone Application : 네트워킹을 하지 않는 애플리케이션
- Client Server Application : 네트워킹을 하는 애플리케이션
- 애플리케이션의 업무를 진행할 때 다른 어떤 컴퓨터(혹은 서버)와 네트워킹을 하지 않는 애플리케이션을 Standalone Application 이라고 합니다.
- 애플리케이션 혼자 그 애플리케이션의 업무 처리가 가능한 애플리케이션입니다.



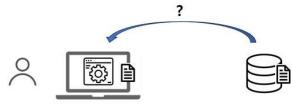
Client – Server Application

• 업무 처리를 위해 사용자 애플리케이션과 서버 애플리케이션이 네트워킹을 하는 애플리케이션입니다.

• Case 1 : 사용자끼리 데이터를 공유해야 하는 경우

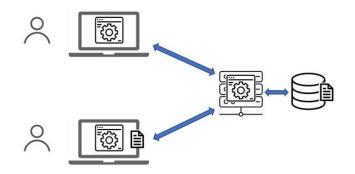


• Case 2: 서버 데이터베이스 데이터를 사용자가 이용해야 하는 경우



Client – Server Application

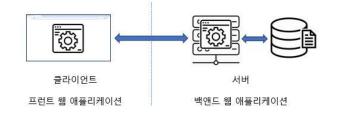
• 사용자끼리 데이터를 공유해야 하거나, 서버 데이터베이스 데이터를 이용해야 하는 경우 클라이언트-서버 구조로 애플리케이션이 개발되어야 합니다.



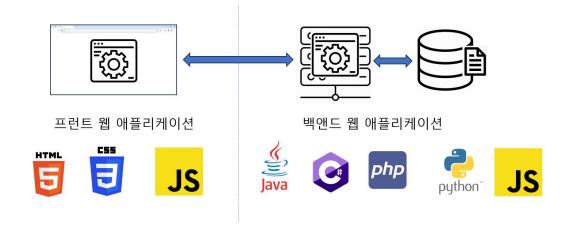
- 클라이언트 서버 구조란 클라이언트 애플리케이션과 서버 애플리케이션이 상호 네트워킹을 통해 업무를 진행하는 구조입니다.
- 클라이언트 서버 구조가 되려면 클라이언트 애플리케이션과 서버 애플리케이션이 각각 만들어 져야 합니다. 클라이언트 애플리케이션이란 사용자의 컴퓨터에 설치되는 애플리케이션이며 서버 애플리케이션은 서버에 설치되어 동작하는 애플리케이션입니다.

Front Web Application – Backend Web Application

- 클라이언트 서버 애플리케이션은 범용적인 용어
- 클라이언트 서버 구조에서 애플리케이션들이 웹의 프로토콜인 HTTP 프로토콜을 이용해 네트워킹을 한다면 이를 웹 애플리케이션이라고 부릅니다.
- 웹 애플리케이션 중 서버에서 실행되는 애플리케이션을 백앤드 웹 애플리케이션, 사용자 브라우저에서 실행되는 애플리케이션을 프런트 웹 애플리케이션이라고 부릅니다.



• 프런트 웹 애플리케이션과 백앤드 웹 애플리케이션은 실행되는 곳도 다르고 목적도 다름으로 각각 개발하는 기술이 다릅니다.



프런트 웹 애플리케이션 개발 기술 – HTML, CSS, JavaScript

- 프런트 웹 애플리케이션은 브라우저에서 실행됩니다.
- 브라우저는 크롬, 엣지, 파이어폭스, 사파리등 다양한 브라우저가 있습니다.
- 하지만 웹에 대한 표준을 W3C 에서 정의하고 있고 대부분의 브라우저가 이 표준을 준수하기 때문에 어느 브라우저에서 프런트 웹 애플리케이션이 실행되든 상관없이 동일한 기술로 개발이 됩니다.
- 브라우저에서 실행되는 웹 애플리케이션은 HTML, CSS, 자바스크립트 조합으로 개발됩니다.







프런트 웹 애플리케이션 개발 기술 – 자바스크립트 프레임워크





• 구글에서 만든 프레임워크로 현재의 자바스크립트 프레임워크의 기본 사상인 SPA 개념을 만든 프레임워크 입니다.



- React.js
- 페이스북에서 프런트 웹 애플리케이션을 개발하기 위해 만든 프레임워크
- 프런트 애플리케이션 개발에 적용되는 많은 개념 및 구조를 만들어낸 프레임워크로 선언형 UI 프로그래밍, 가상 DOM, 상태 관리등의 많은 개념이 React.js 부터 시작되었습니다.



- Vue.js
- Evan You 에 의해 만들어진 프레임워크로 Angular, React.js 와거의 유사한 개념 및 구조를 지원하는 프레임 워크입니다.

백앤드 웹 애플리케이션 개발 기술

- 백앤드 웹 애플리케이션은 개발팀에서 개발 환경을 어떻게 선택하냐에 따라 다양한 기술이 이용될 수 있습니다.
- 자바 기반의 백앤드 웹 애플리케이션 개발



• 자바스크립트 기반의 백앤드 웹 애플리케이션 개발



백앤드 웹 애플리케이션 개발 기술

• C# 기반의 백앤드 웹 애플리케이션 개발



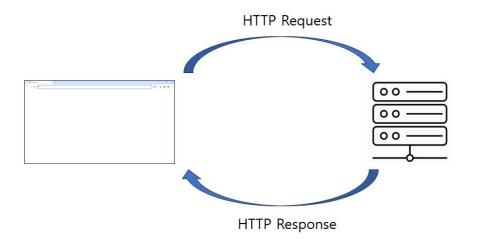
• 파이썬 기반의 백앤드 웹 애플리케이션 개발



• PHP 기반의 백앤드 웹 애플리케이션 개발

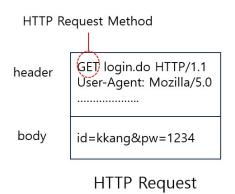


- 클라이언트 서버 구조에서 상호 네트워킹을 하며 데이터를 주고 받는데 이때 HTTP 프로토콜을 이용하는 애플리케이션을 웹 애플리케이션이라고 합니다
- HTTP 프로토콜은 HTTP Request 규약과 HTTP Response 규약으로 나누어 집니다.



HTTP Request

- HTTP Request 는 프런트 애플리케이션이 백앤드에 전달하는 데이터에 대한 규약입니다.
- HTTP 통신에서 주고 받는 데이터는 header 와 body 로 구분됩니다. header 는 부가 정보이며 body 는 실제 전달하는 데이터입니다.

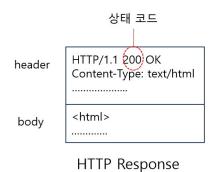


HTTP Request

- User-Agent 는 요청하는 프런트에 대한 정보로 브라우저명, 브라우저 버전등에 대한 정보가 담겨 있습니다.
- HTTP Request Method 는 흔히 요청 방식이라고 하며 GET, POST, PUT, DELETE, HEAD, PATCH 가 있습니다.
- GET : 서버의 데이터를 얻기 위한 요청
- POST: 서버에 데이터를 전달하여 저장하기 위한 요청
- PUT : 서버에 데이터를 전달하여 수정하기 위한 요청
- DELETE : 서버의 데이터를 삭제하기 위한 요청
- HEAD : 서버와 각종 정보를 확인하기 위한 요청
- PATCH: 서버의 데이터를 수정하기 위한 요청
- 백앤드에 요청할 때 요청 방식을 따로 지정하지 않으면 기본으로 GET 방식으로 요청되게 됩니다.

HTTP Response

• 백앤드에서 프런트에 전달되는 데이터를 HTTP Response 라 하며 이 HTTP Response 도 header 와 body 로 구성됩니다.



- header 는 부가 정보이며 body 가 실제 백앤드에서 프런트에 전달하는 데이터입니다.
- 웹임으로 HTML 문서가 body 에 담겨서 전달 될 수도 있고 JSON 같은 데이터 일 수도 있습니다.

HTTP Response

• 상태 코드는 숫자로 표현되며 프런트의 요청을 어떻게 처리한 것인지에 대한 상태를 표현하는 숫자입니다.

• 2xx : 요청을 정상적으로 처리 했음

• 3xx : 요청 완료를 위해 추가적인 작업이 필요함

• 4xx : 요청 잘못으로 요청을 처리하지 못함

• 5xx: 서버 잘못으로 요청을 처리하지 못함

웹 기본 지식 – URL 분석

- 프런트가 백앤드에 요청하는 업무 혹은 데이터를 통칭해서 리소스라고 합니다.
- 프런트에서 원하는 리소스가 식별될 수 있게 URL 을 구성해서 백앤드에 요청을 하고, 백앤드는 요청 정보의 URL 을 분석해서 어떤 리소스를 프런트에게 서비스 할것인지를 판단하게 됩니다.
- URL 은 Uniform Resource Locator 의 약어이며 아래의 구조로 만들어진 문자열이어야 합니다.

<프로토콜>://<호스트>:<포트>/<경로>?<쿼리>#<해시>

front 만을 위한 것
(상태 표시만 한 것)

https://search.naver.com/search.naver?where=image&query=javascript#imgld=image_sas
프로토콜
호스트
경로
레리 문자열
백엔드가 받아서 해석하는 데이터

웹 기본 지식 – URL 분석

• 프로토콜

s가 붙으면 보안이 적용된 것

• 프로토콜은 네트워킹에 사용할 통신 규약으로 웹에서는 일반적으로 HTTP 혹은 HTTPS 가 사용됩니다.

• 호스트

• 호스트는 흔히 도메인이라 불리우는 부분으로 요청하는 서버의 네트워크 상에서의 주소입니다. 서버의 IP 주소일 수도 있고 도메인일 수도 있습니다. 이 호스트에 의해 네트워크 상의 어느 서버와 통신할 것인지가 결정되게 됩니다. = IP 주소를 문자열로 표현한 것

• 포트

- 포트는 하나의 서버에서 네트워킹하는 애플리케이션을 식별하기 위한 숫자 입니다.
- 웹의 HTTP 로 통신하는 경우 포트를 생략하게 되면 자동으로 80 포트가 지정되게 됩니다.

• 경로 회원가입, 로그인, 상품정보 등의 리소스

- 경로는 서버 애플리케이션의 리소스 식별입니다. 어느 리소스를 원하는지를 명시하기 위해 사용됩니다.
- 이 경로는 프런트 애플리케이션 개발자와 백앤드 애플리케이션 개발자가 상호 규칙에 의해 정의하고 그 규칙대로 경로를 설정하면 됩니다.

웹 기본 지식 – URL 분석

• 쿼리 문자열

• 경로 뒤에 ?가 추가되고 그 뒤에 추가되는 문자열을 흔히 쿼리 문자열(Query String) 혹은 검색 문자열 (Search String) 이라고 합니다. 이 쿼리 문자열은 프런트에서 백앤드에 요청을 하면서 전달하는 데이터입니다.



• 해시문자열

- 해시문자열은 # 뒤에 추가되는 문자열입니다. 특별한 작성규칙이 있지는 않으며 #10 혹은 #top 등 원하는 형태로 추가해 주면 됩니다. 해시 문자열은 URL 에 포함되기는 하지만 백앤드와는 관련이 없습니다. 프런트 웹 애플리케이션에서 현재의 애플리케이션의 상태를 URL 에 추가하기 위해서 사용됩니다.
- 화면이 길게 스크롤 될 때 어느 특정 위치를 지정하고 그 위치까지 스크롤 되게 하고자 할 때 URL 의 해시 문자열에 그 위치에 해당되는 문자열이 추가되어 자동으로 스크롤 되게 할 수 있습니다.
- 하나의 HTML 문서내에서 여러 화면을 제공하여 유저에게 화면 전환이 되는 것처럼 효과를 내고 싶은 경우



감사합니다

단단히 마음먹고 떠난 사람은 산꼭대기에 도착할 수 있다. 산은 올라가는 사람에게만 정복된다.

> 윌리엄 셰익스피어 William Shakespeare