

# Boiler Plate

개발을 시작할 수 있는 기초가 되는 구조들

정년 기술에서 막히는 보일러 플레이트 (초급 저급 부름에서 유래) 는  
변경 없이 계속해서 재 사용되고 있는 저작품의 의미.

초보자 입장에서 JS를 모든 브라우저들이 이해 가능하도록 있다 → Webpack 이라는 Configuration Library 사용  
해야하는 데 이를 해결해준다.

명령어 : `npm create-react-app` 파싱

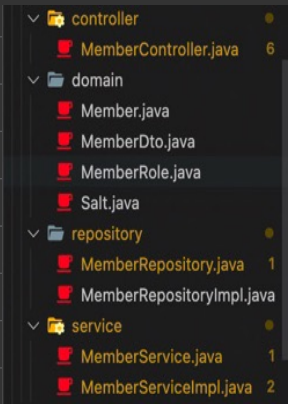
민중의 웹개발 시작을 위한 프로젝트를 제공해준다.

\* Protocol : 공통의 데이터 교환 방법 및 순서에 대해 정의한 규칙, 약속...

BE 쪽에서는 Spring Boot 가 이를 해결해준다.

↳ Spring Initializer

## \* Spring Boot Boiler Plate



① Controller : 모든 요청의 Servlet 역할은 여기가 담당한다.

- View or Server 와의 연결
- Response body 객체 생성
- Service 호출

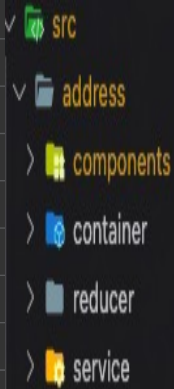
domain : Entity or Component (DTO) 를 가지고 있다  
· Database 관련 코드.

② Repository : DB or Server 간 연결  
마이크로 서비스에서는 JPA 를 사용한다.

③ Service : Repository Controller 사이를 연결  
비즈니스 로직 담당.

작업 순서 : Domain → Repo → Service → Controller

## \* React Boiler Plate



\*가독성↑ 위해 boiler 같은 이름의 dir mk!

① 컨테이너: 자바 (Servlet)의 Controller에 연결

JS의 컨트롤러 역할 → **axios** 가 들어가야 함!

: 컴포넌트의 일종. (**Smart Component**)

리덕스의 소동하면서 앱의 State 를 제어하는 컴포넌트

액티비트 (State 변화)가 자주 일어나는 곳을 나눔.

② 서비스: 자바의 서비스와 같은 기능/비즈니스로직 담당.

③ 리듀서: 리덕스 파트.

④ 컴포넌트 : 뷰에 가까운 파트