

# 스프링 부트 3 백엔드 개발자 되기 (2판)

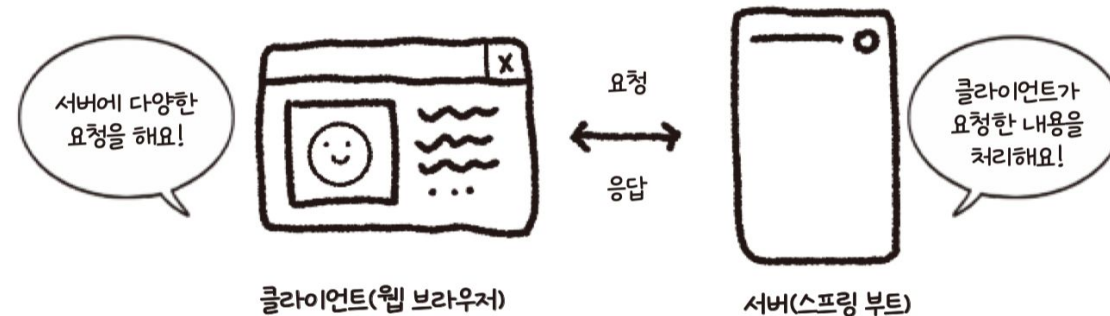
JPA+OAuth2+JWT+AWS와 배우는 스프링 부트 3  
Java 백엔드 입문자를 위한 풀 패키지

## 01 스프링 부트3 개발자가 알아 두면 좋은 지식

# 1.1 서버와 클라이언트

- 서버와 클라이언트

- 클라이언트 : 서버로 요청하는 프로그램
- 서버 : 클라이언트의 요청을 받아 처리하는 주체



# 1.2 데이터베이스

- 데이터베이스

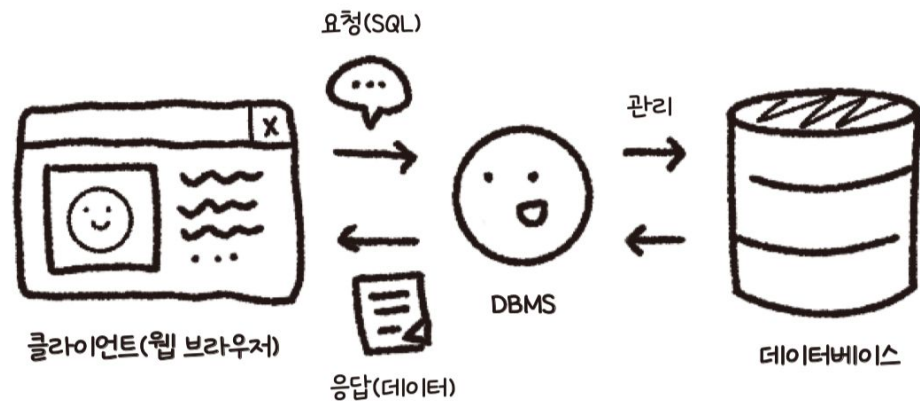
- 데이터를 모아 놓고 여러 사람이 사용할 목적으로 관리하는 데이터 저장소

- 데이터베이스 관리 시스템

- 데이터베이스를 관리하기 위한 시스템, 줄여서 **DBMS(database management system)**
  - MySQL, PostgreSQL, Oracle 등 다양한 데이터베이스 관리 시스템이 있음

- SQL

- 데이터베이스 관리 시스템에 명령을 내리기 위한 언어



# 1.3 아이피와 포트

- 아이피

- 인터넷에서 컴퓨터들이 서로를 식별하고 통신하기 위한 주소
- 웹 브라우저에 입력하는 주소([www.naver.com](http://www.naver.com))는 사실 아이피를 문자열로 대치한 것

- 포트

- 특정 아이피에서 운용하는 서비스를 구분하기 위한 것
- 아이피가 백화점이면 포트는 매장이라고 생각하면 됨
- <https://>는 443번 포트를 사용하기 위한 입력
- 톱캣 서버의 기본 포트값은 8080



# 1.4 라이브러리와 프레임워크

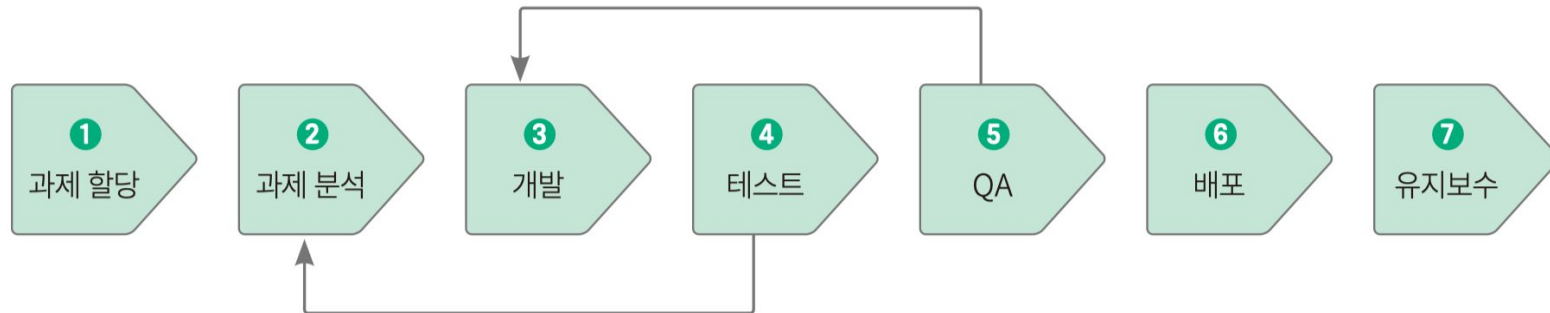
- 라이브러리

- 앱 개발에 필요한 기능인 클래스, 함수 등을 모아놓은 코드의 모음
- 라이브러리 간 영향을 주지 않음

- 프레임워크

- 틀(frame) + 일하다(work)의 합성어
- 개발을 위한 틀을 제공
- 개발자는 틀에 맞춰 개발해야 함

# 1.5 백엔드 개발자의 업무



# 1.5 백엔드 개발자의 업무(cont.)

- 1. 과제 할당
  - 대부분의 부서에 과제를 할당하며 업무 시작
- 2. 과제 분석
  - 소셜 서비스에서 ‘친구 초대 이벤트를 만들어 보자’라고 한다면?
  - 친구 정의, 초대 정의, 이벤트 정의 등 과제 연관 분석을 진행
- 3. 개발
  - 개발 진행, 버그 수정의 반복
  - 동료의 리뷰를 받아 더 나은 코드로 완성도를 높임
- 4. 테스트
  - 로직 문제가 없는지?
  - 성능 한계는 어느 정도인지?
  - 코드가 읽기 편한지?
  - 코드를 다른 개발자가 쓰기 편한한지?



# 1.5 백엔드 개발자의 업무(cont.)

- 5. QA

- 품질 보증 단계 = quality assurance

- 6. 배포

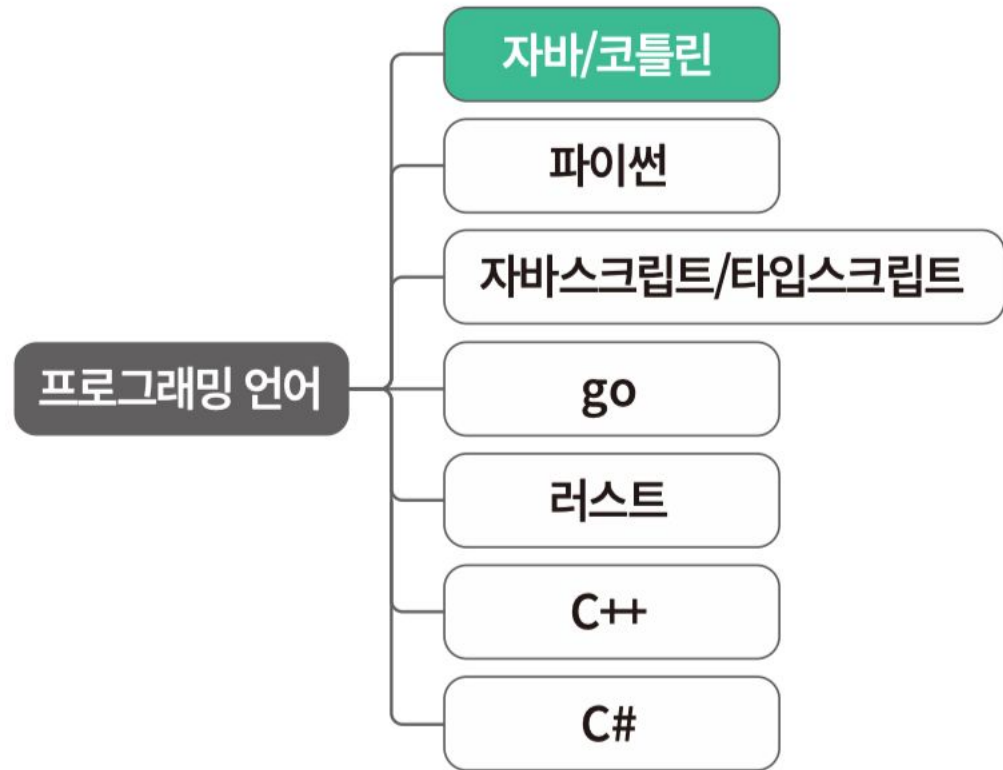
- 서버에 개발한 앱 배포
- 새로운 기능이 추가되면 추가 배포
- 쉽고, 안전하고, 롤백이 쉬운 방법으로 배포
- 롤링 배포, 블루 그린 배포, 카나리 배포 등 방법 다양

- 7. 유지보수

- 개발부터 배포까지 반복하는 과정을 고도화하여 앱 품질 유지 및 보수
- CI / CD = continuous integration / continuous delivery & continuous deployment

# 1.6 백엔드 프로그래밍 언어

- 백엔드에서 쓰이는 언어는 매우 다양
  - 파이썬 : 장고, 플라스크, FastAPI
  - 자바 : 스프링, 스프링 부트
  - C# : 닷넷 프레임워크
  - 자바스크립트 : 익스프레스, NestJS



# 1.7 자바 애너테이션

- 자바로 작성한 코드에 추가하는 표식
- @ 기호를 사용하여 표시
- JDK 1.5 버전부터 사용할 수 있음

## ▼ 자바의 대표적인 애너테이션

애너테이션 이름	설명
@Override	선언된 메서드가 오버라이드 되었음
@Deprecated	더 이상 사용되지 않음
@SuppressWarnings	컴파일 경고를 무시함

## ▼ 오버라이드의 사용 예

```
public class A extends B {  
    @Override // 선언된 메서드가 오버라이드 되었음을 나타내는 애너테이션  
    public void print() {  
        System.out.println("Hello, World!");  
    }  
}
```