



라이브 세션-2022.12.22(목)- Spring Data JDBC란

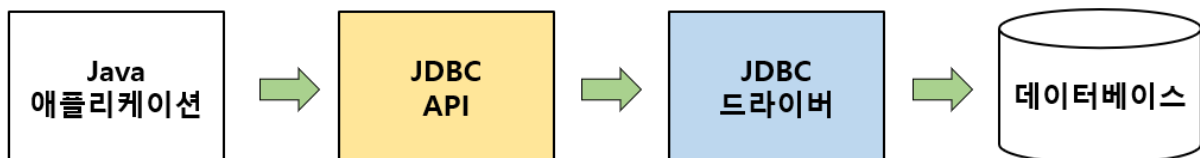
✓ 당부의 말씀

- 🌧️🌧️🌧️ Spring Data JDBC부터 **난이도가 갑자기 상승**하는 느낌이 듭니다 여러분!
- 복습할 때, Java의 처음부터 끝까지 학습할 생각보다는 섹션 3에서 사용되는 **Stream, Lambda, 인터페이스** 정도만 틈날때마다 학습을 해주세요
- 알고리즘 문제에 몰입하다보면 이 후 학습을 잘 따라오지 못할 수 도 있어요.
 - 알고리즘은 **구현(Implementation) 파트**를 중심으로 푸는게 좋습니다.
 - 진도를 따라가기 벅차고, 어떻게 타이핑 해야할지 전혀 감을 못잡는다면?
 - 알고리즘 학습 시간은 아침 시간 외에는 안하는게 나을 것 같습니다.

✓ 들어온 질문

1 JDBC의 동작 흐름의 순서가 잘못된 것 같습니다.

- 구체적인 흐름이라기 보다는 개념적인 흐름이라고 생각하면 될 것 같습니다.



Java 애플리케이션에서 JDBC API를 이용해 적절한 데이터베이스 드라이버를 로딩 한 후, 데이터베이스와 인터랙션 한다.

2 JDBC API, HTTP API, REST API의 차이점은 뭘까요?

- API(Application Programming Interface)
 - 우리 애플리케이션을 구현하는데 사용하기 위한 **다른 사람이 만든 기능**을 대표하는 무언가
 - 라이브러리
 - 프레임워크
 - Open API
 - RestAPI
- HTTP API
 - API의 한 유형
 - HTTP 프로토콜로 네트워크 통신을 할 수 있게 해주는 API
 - Open API
 - REST API
- JDBC API
 - API의 한 유형
 - 데이터베이스와 인터랙션 할 수 있게 해주는 API
 - JDBC를 구현하는 구현체
 - MSSQL, Oracle, MySQL 등의 벤더에서 제공하는 API



API \supseteq HTTP API \supseteq REST API

✓ 지난 시간 학습 리뷰

1 Controller, DTO, Service 클래스 빠르게 리뷰

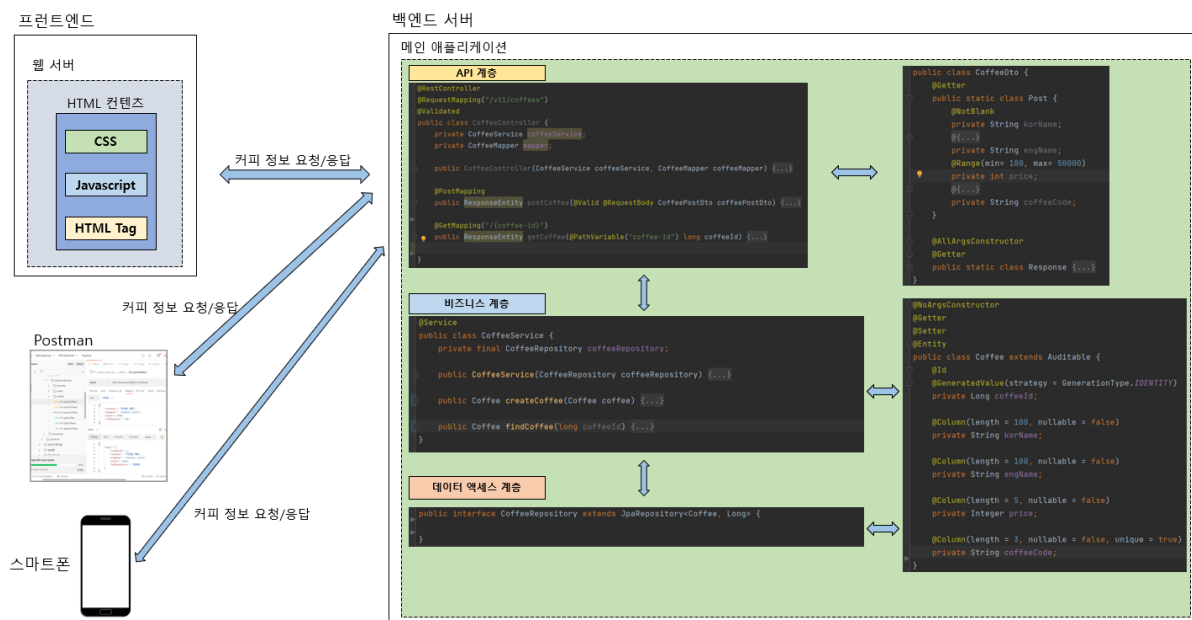
2 @ExceptionHandler 리뷰

3 @RestControllerAdvice 리뷰

4 비즈니스 로직 예외 처리 리뷰

5 Error 전용 Response 클래스 리뷰

✓ 우리가 배우는 샘플 애플리케이션의 아키텍처



✓ Spring Data JDBC에 대한 수준별 목표

1 Hello, World 조차 너무 어려워요

- ORM이 무엇인지 이해할 수 있도록 하자.
- Spring Data 패밀리의 기본적인 API는 사용할 수 있도록 하자.
- 나머지는 몰라도 그냥 넘어가자.
- JPA에 집중하자.

2 Hello, World는 이해되고, Spring Data JDBC의 설계 개념은 이해가 되요

- Spring Data JDBC를 이용해 직접 구현까지 해보자
- 구현 연습에 시간이 너무 오래 걸리면 시간이 남을때만 하자
- 나머지는 JPA에 집중하자

3 Spring Data JDBC를 프로젝트에서도 사용할 수 있을 것 같아요

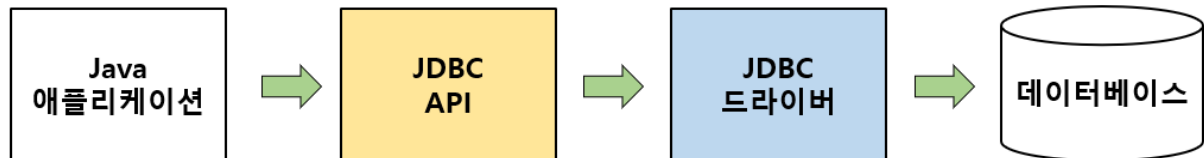
- DDD 방식으로 추가 기능을 적용해보자.
- Spring Data JPA를 DDD 방식으로 개선할 수 있는지에 대한 학습을 해보자.

✓ In-Memory DB란?

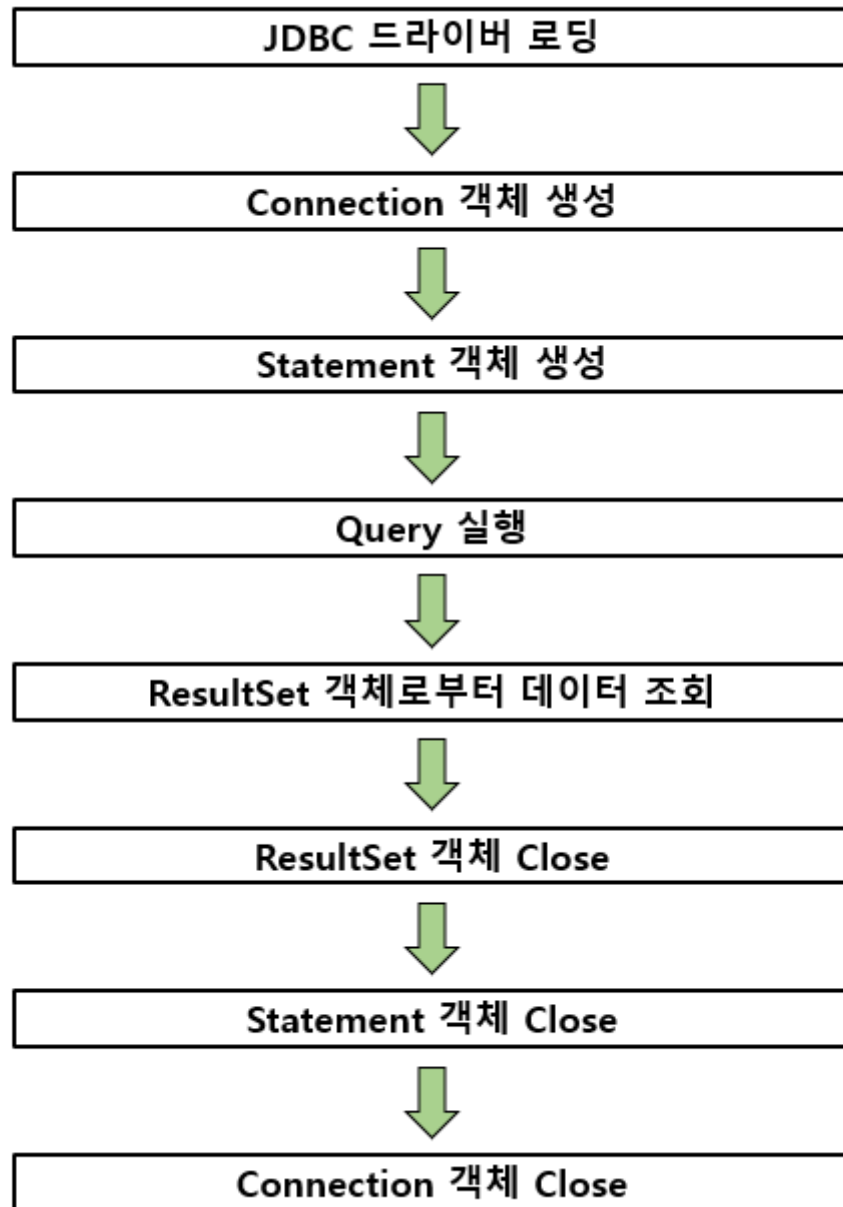
- 메모리 DB
- 애플리케이션이 종료되면 데이터는 사라짐
- ★ 로컬 개발 환경에서 테스트 용도로 사용
 - 테스트 용도로 잘 사용되는 이유를 생각해 봅시다.
- 우리가 사용하는 In-Memory DB는 H2
 - 설정 방법

✓ JDBC API 리뷰

1 JDBC의 동작 흐름

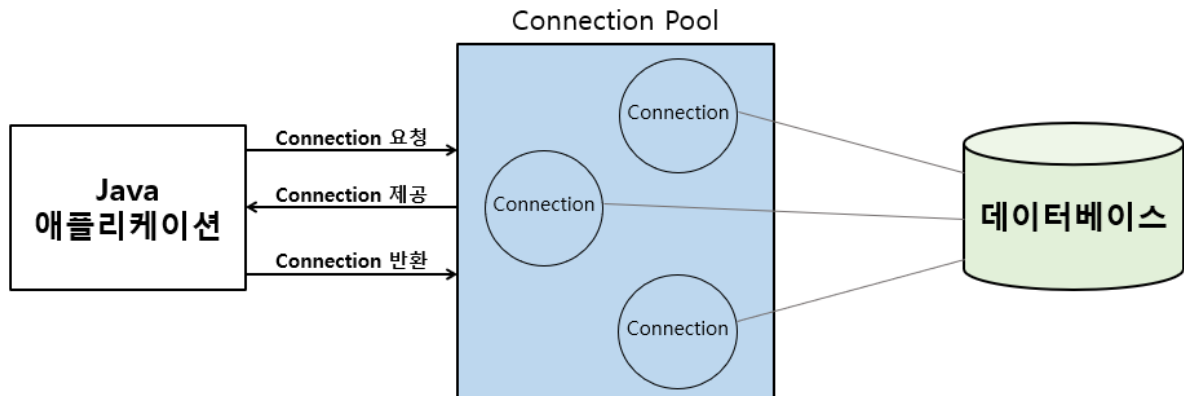


2 JDBC API 사용 흐름



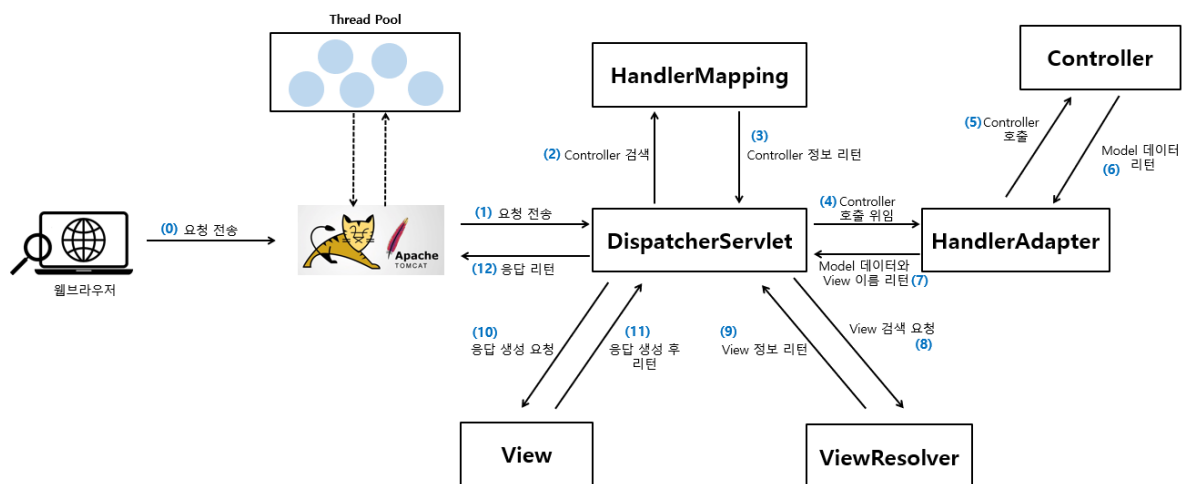
3 DB Connection Pool

- Apache commons DBCP
- HikariCP



* 참고

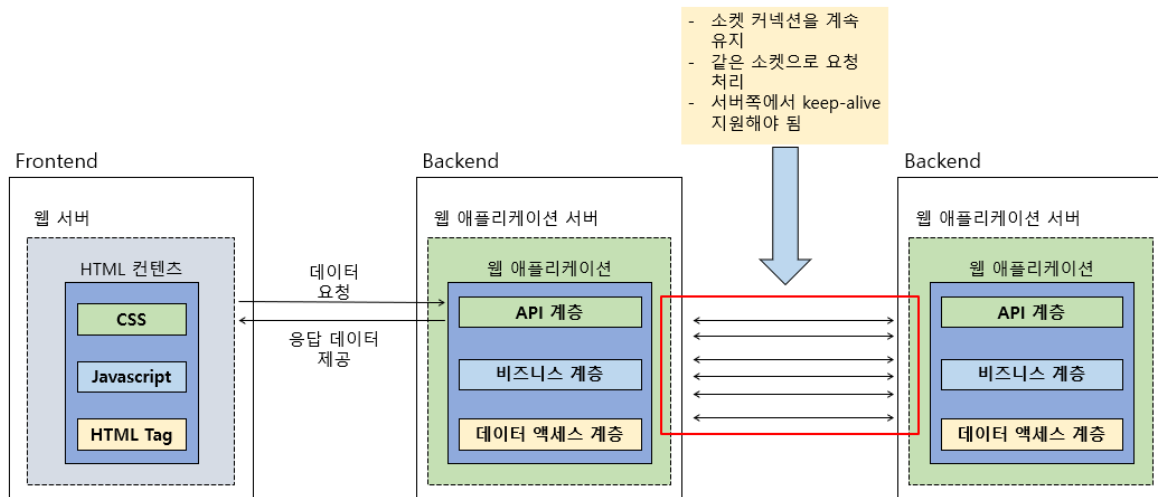
1 Thread Pool



2 HTTP Connection Pool

- Server가 Keep-Alive 지원 필요
- HttpClient

- HttpClientHttpRequestFactory



✓ Java의 데이터 액세스 기술 종류

1 SQL 중심 기술

- 순수 JDBC API 이용
- iBatis 또는 myBatis
- Spring JDBC
 - JdbcTemplate이라는 기술을 이용한다.

2 ORM(Object-Relational Mapping)중심 기술

- Spring Data JDBC
- JPA
- Spring Data JPA

3 순수 JDBC API 기술 들여다 보기

- 구현 코드 살짝 들여다 보기

4 Spring JDBC 들여다 보기

- 구현 코드 살짝 들여다 보기

✓ Hello, Spring Data JDBC 코드 리뷰

1 Spring Data JDBC 적용 순서

1. `build.gradle` 에 사용할 데이터베이스를 위한 의존 라이브러리를 추가합니다.
2. `application.yml` 파일에 사용할 데이터베이스에 대한 설정을 합니다.
3. '`schema.sql`' 파일에 필요한 테이블 스크립트를 작성합니다.
4. `application.yml` 파일에서 '`schema.sql`' 파일을 읽어서 `테이블을 생성` 할 수 있도록 초기화 설정을 추가합니다.
5. 데이터베이스의 테이블과 매핑할 `엔티티(Entity) 클래스를 작성` 합니다.
6. 작성한 엔티티 클래스를 기반으로 데이터베이스의 작업을 처리할 `Repository 인터페이스를 작성` 합니다.
7. 작성된 Repository 인터페이스를 `서비스 클래스에서 사용할 수 있도록 DI` 합니다.
8. DI 된 Repository의 메서드를 사용해서 `서비스 클래스에서 데이터베이스에 CRUD 작업을 수행` 합니다.