

3) library-app

3-1) build-gradle : gradle 이 build 전체의 설정을 다룸

- build : 작성자가 아닌 컴퓨터 기준으로 실행가능한 파일로 변환해주는 것
- build tool
 - maven : JAVA 전용 //pom.xml , project of model
 - gradle : spring, android에 최적화, java, c++, python 가능
 - groovy : JVM 상에서 스크립트처리
 - node는 자바스크립트 처리 전용?
 - gradle 과 groovy 는 set

3-2) [maven vs gradle](#)

- 가독성은 gradle 이 좋다. 캐시메모리사용, 재작업빠름
- 의존성이 늘어날거면 gradle이 좋다
- maven 은 상속, apache open source
- gradle은 import

```
plugins { //dependency를 사용할 주체를 정의
    id 'org.springframework.boot' version '2.7.6'
    id 'io.spring.dependency-management' version '1.0.12.RELEASE'
    id 'java'
}

group = 'com.example'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11' //java 버전, 17로 바뀌면 적용, 위 2.7.6도 3.0.1 이상필요

repositories {
    mavenCentral() //maven 설정을 100% 사용하기 위해
}

dependencies { //앞서 의존성 선택한 내역, 직접 입력시 build 해도 가능한
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    runtimeOnly 'com.h2database:h2'
    runtimeOnly 'mysql:mysql-connector-java'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
}

tasks.named('test') {
    useJUnitPlatform()
}
```

3-3) application.yml //도서관리 애플리케이션

```
spring:          #spring 의 db 관련
config:
  activate:
    on-profile: local
datasource:
  url: "jdbc:h2:mem:library;MODE=MYSQL;NON_KEYWORDS=USER"
  username: "sa"
  password: ""
  driver-class-name: org.h2.Driver
jpa:
  hibernate: #없으면 만들라고 하는 의미로 create
  ddl-auto: create
  properties:
    hibernate: #db 연동
    show_sql: true
    format_sql: true
    dialect: org.hibernate.dialect.H2Dialect
h2: #db 서버 안갖출때 사용
  console:
    enabled: true
    path: /h2-console
---
spring:          #mysql 했을때
config:
  activate:
    on-profile: dev    #배포할 위치에 사용할 db명
datasource:
  url: "jdbc:mysql://localhost/library" #db가 위치할곳, 타url에 올렸으면 그걸 작성
  username: "root"
  password: "Abcd1234!"
  driver-class-name: com.mysql.cj.jdbc.Driver
jpa:
  hibernate: #있으면 none
  ddl-auto: none
  properties:
    hibernate:
    show_sql: true
    format_sql: true
    dialect: org.hibernate.dialect.MySQL8Dialect
```

- 실행시 터미널에서 확인되는 메시지는 다음과 같음

```
Hibernate:
  select
    u1_0.id,
    u1_0.age,
    u1_0.name
  from
    user u1_0
```

- 개발시에는 drop the table 이지만, 배포시에는 해당 내용은 삭제해서 배포해야 깔끔

3-4) UserServiceV1 (src > main > java > com > group > libraryapp > service > user)

```
package com.group.libraryapp.service.user;

import com.group.libraryapp.dto.user.request.UserCreateRequest;
import com.group.libraryapp.dto.user.request.UserUpdateRequest;
import com.group.libraryapp.dto.user.response.UserResponse;
import com.group.libraryapp.repository.user.UserJdbcRepository;
import org.springframework.stereotype.Service;
import java.util.List;

@Service
public class UserServiceV1 {
    private final UserJdbcRepository userJdbcRepository;
    public UserServiceV1(UserJdbcRepository userJdbcRepository) {
        this.userJdbcRepository = userJdbcRepository;
    }
    public void saveUser(UserCreateRequest request) {
        userJdbcRepository.saveUser(request.getName(), request.getAge());
    }
    public List<UserResponse> getUsers() {
        return userJdbcRepository.getUsers();
    }
    public void updateUser(UserUpdateRequest request) {
        if (userJdbcRepository.isUserNotExist(request.getId())) {
            throw new IllegalArgumentException();
        }
        userJdbcRepository.updateUserName(request.getName(), request.getId());
    }
    public void deleteUser(String name) {
        if (userJdbcRepository.isUserNotExist(name)) {
            throw new IllegalArgumentException();
        }
        userJdbcRepository.deleteUser(name);
    }
}
```

- DAO¹ : 실제로 DB의 data에 접근하기 위한 객체
 - mybatis 나 orm 이면 DAO없어도 접근 가능
 - CRUD 기능을 수행, Repository package가 바로 DAO, Service와 DB 연결고리
- DTO : 계층 간 데이터 교환, 로직을 가지지 않는 순수한 데이터 객체(Java Beans), 가변적값
 - 엔티티를 DTO 형태로 변환한 후 사용
 - DTO는 즉, getter/setter 메서드만 가진 클래스를 의미
 - DB에서 데이터를 얻어서 Service나 Controller 등으로 보낼 때 사용
- VO : DTO와 달리 Read-Only속성을 지닌 값 오브젝트로 수정불가, 불변값
- 순서 : repository(db) > DTO > controller > service (화면)

¹ [DAO\(Data Transfer Object\) 와 DTO\(Data Access Object\)의 차이 그리고 VO\(Value Object\)](#)