음악 스트리밍 서비스

Spring MVC with gradle build

2016.11.02 - 12.02

Contents of Our Slide

Project Goal

프로젝트 목표

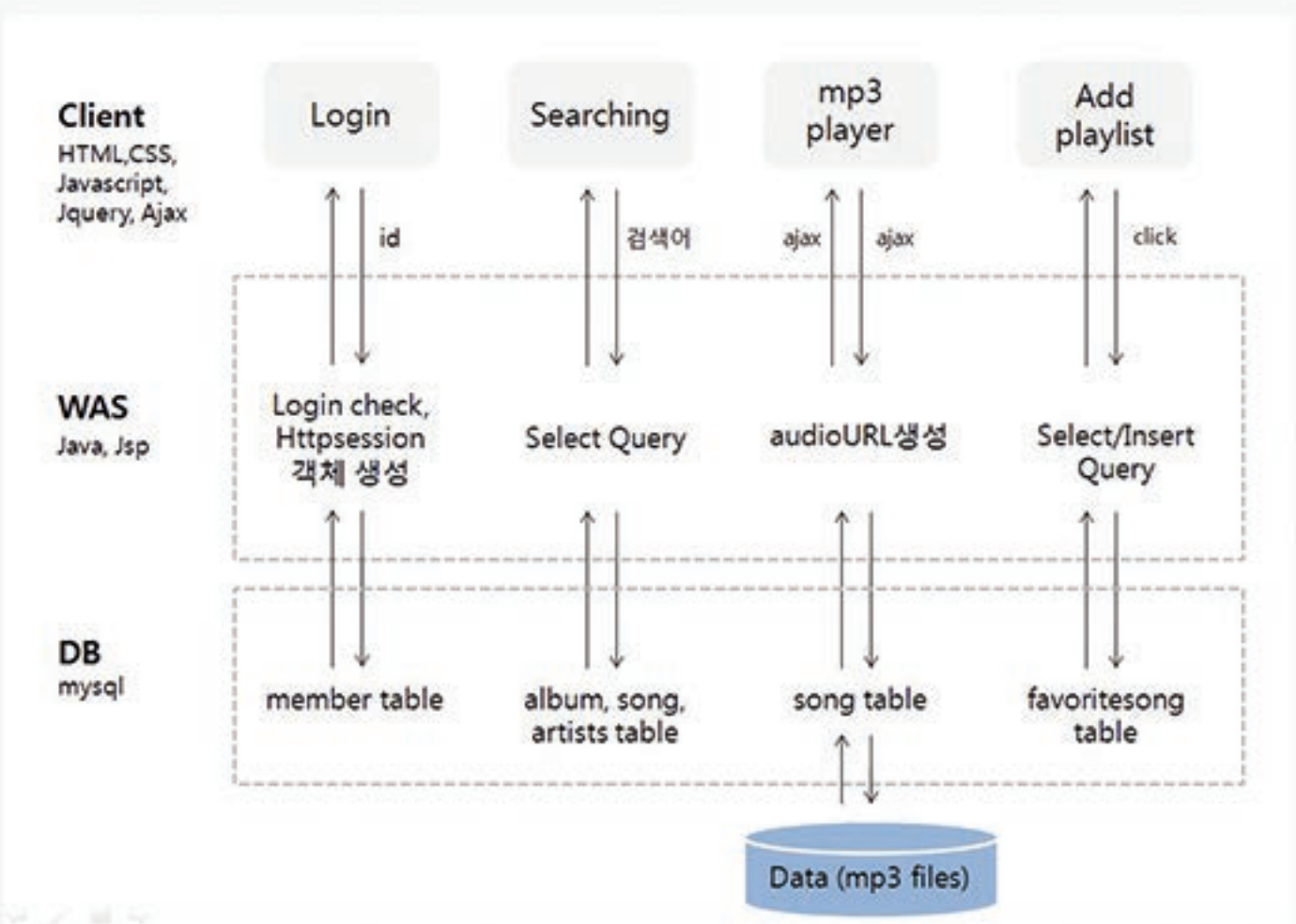
01 Spring framework의 구조와 DI개념을 숙지하고,
Annotation기반의 코딩 방식을 이해하고 활용한다.

02 MVC패턴을 통한 웹 어플리케이션 개발 방법과
과정을 이해하고 활용한다.

03 자바스크립트와 Ajax를 적극 활용함으로써
비동기 데이터 처리를 이해한다.

System Architecture

전체 아키텍처



Process of Development

개발과정 및 내용

01

음원의 메타데이터 추출 및 데이터 생성 과정

- -개발 환경
- -데이터 구조 및 생성
- -아키텍처
- -구현 내용

02

웹 어플리케이션 개발 과정

- -개발환경
- -아키텍처
- -시퀀스 다이어그램
- -구현 내용

01

메타데이터 추출 및 데이터 생성 과정

1-1.개발환경

· IDE : Intellij Community Edition (ver.2016.2.4)

· OS : Window7 (64 bit)

· Platform : Java SE(Standard Edition)

· Language : Java 1.8

· DB : MySQL 5.1.60

· Using External Libraries

dom4j-1.6

mybatis-3.4.1

mysql-connector-java-5.1.40-bin

poi-3.15

tika-core-1.8

01

메타데이터 추출 및 데이터 생성 과정

1-2.데이터 구조 및 생성

총 5개의 테이블로 구성

: album, artists, song, member, favoritesong

```
1 CREATE TABLE `album` (
2    `albumid` INT(11) NOT NULL AUTO_INCREMENT,
3    `album` VARCHAR(120) NOT NULL,
4    `artist` VARCHAR(120) NOT NULL,
5    `genre` VARCHAR(120) NOT NULL,
6    `releasedate` VARCHAR(40) NOT NULL,
7    PRIMARY KEY (`albumid`),
8    UNIQUE INDEX `albumid` (`albumid`),
9    INDEX `fk_artist` (`artist`)
10 )
11 COLLATE='utf8_general_ci'
12 ENGINE=InnoDB
13 AUTO_INCREMENT=2093
14 ;
15
```

```
1 CREATE TABLE `artists` (
2    `artist` VARCHAR(250) NOT NULL,
3    PRIMARY KEY (`artist`)
4 )
5 COLLATE='utf8_general_ci'
6 ENGINE=InnoDB
7 ;
8
```

01

메타데이터 추출 및 데이터 생성 과정

1-2.데이터 구조 및 생성

검색기준(album, artists, song테이블)과
회원가입/로그인을 위한 member테이블,
좋아하는 곡만을 모아서 재생리스트를 생성하는
favoritesong테이블로 총 5개의 테이블을 구성

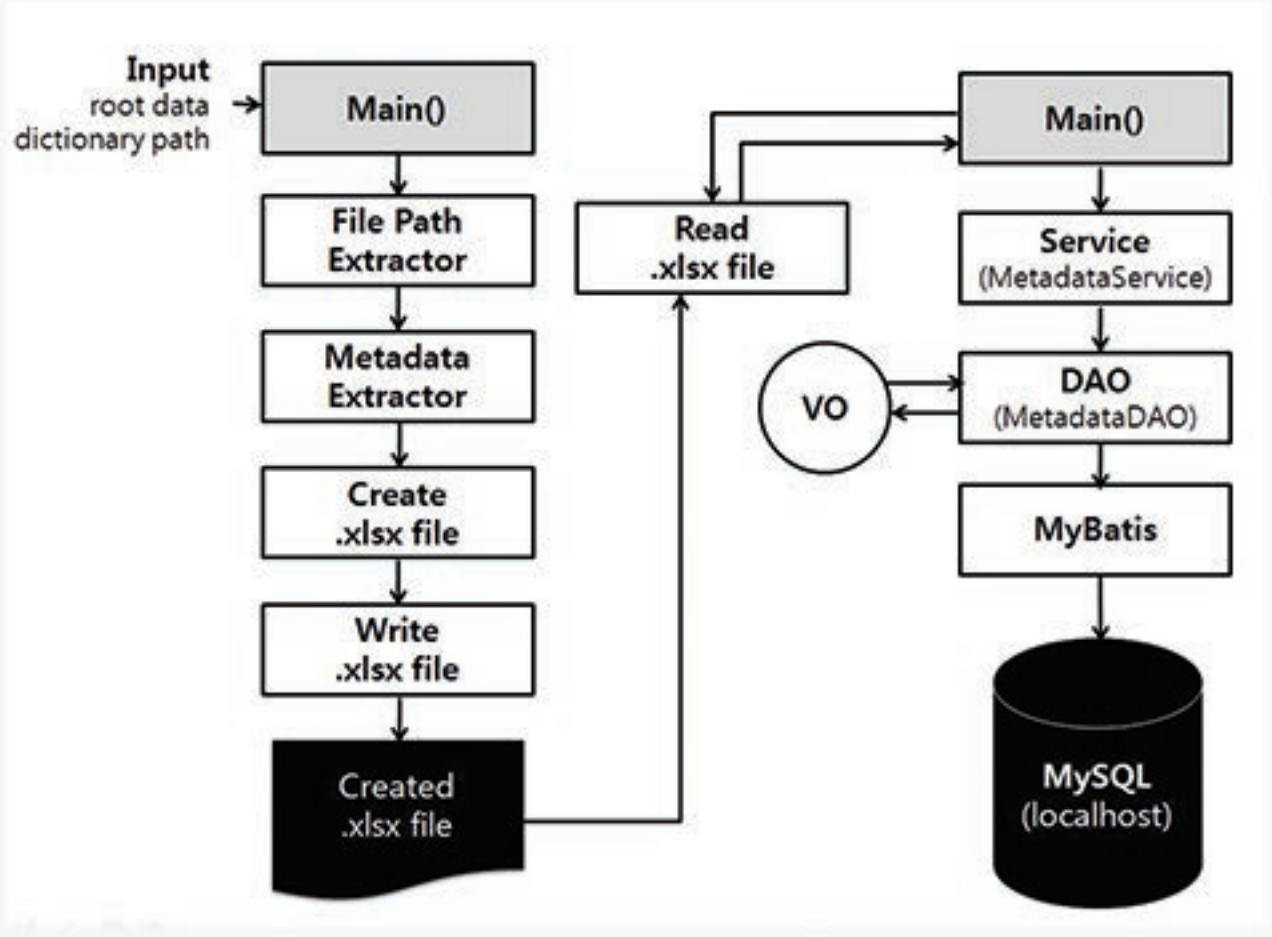
```
CREATE TABLE `member` (
    `id` VARCHAR(16) NOT NULL,
    `password` VARCHAR(25) NOT NULL,
    `name` VARCHAR(25) NOT NULL,
    `email` VARCHAR(120) NOT NULL,
    `phone` VARCHAR(50) NOT NULL,
    `address` VARCHAR(150) NULL DEFAULT NULL,
    `infoEmail` TINYINT(2) NULL DEFAULT '0',
    `infoSMS` TINYINT(2) NULL DEFAULT '0',
    PRIMARY KEY (`id`),
    UNIQUE INDEX `id` (`id`),
    UNIQUE INDEX `email` (`email`),
    INDEX `idx` (`id`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

```
CREATE TABLE `favoritesong` (
    `userid` VARCHAR(120) NOT NULL,
    `album` VARCHAR(120) NOT NULL,
    `title` VARCHAR(200) NOT NULL,
    `artist` VARCHAR(120) NOT NULL,
    `duration` FLOAT NOT NULL,
    `filepath` VARCHAR(250) NOT NULL
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;

```

01 메타데이터 추출 및 데이터 생성 과정

1-3.아키텍처



01 메타데이터 추출 및 데이터 생성 과정

1-4.구현내용

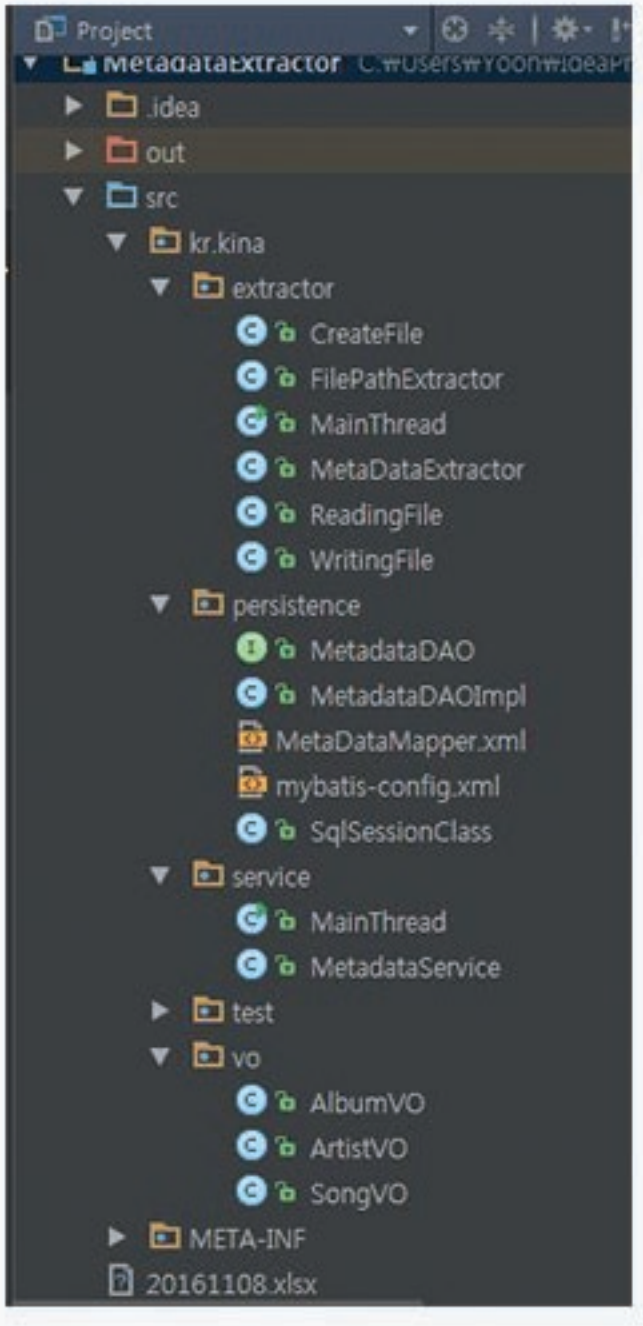
주요 기능	mp3파일에서 메타데이터를 파싱하여 엑셀파일을 만들고, 엑셀파일에 저장된 데이터를 DB에 저장한다.
구현 내용	모든 데이터를 담고 있는 최상위 디렉토리에서 재귀함수를 사용하여 모든 파일루트를 찾은 후, 각 mp3파일에 담긴 메타데이터를 파싱한다. 파싱한 데이터는 엑셀파일로 만들고, 엑셀파일에 저장된 데이터를 다시 MyBatis를 통해 MySQL에 저장한다.
스프링 프레임워크와 분리	웹 어플리케이션을 개발하기 위해 필요한 데이터를 생성하는 준비작업이므로, 스프링 프레임워크를 사용하지 않고, JavaSE버전으로 개발하여 별도로 분리시켰다.

01

메타데이터 추출 및 데이터 생성 과정

1-4.구현내용

전체 파일 구조



01

메타데이터 추출 및 데이터 생성 과정

1-4.구현내용

소스 코드 일부분

```java
private void extract(String rootPath) throws IOException {

    File[] file = new File(rootPath).listFiles();  // 1depth 폴더 추출

    for(File name : file){
        if(name.isDirectory()) {
            extract(name.getCanonicalPath());  // N depth 폴더 안에서 재귀 호출

        }else if(name.isFile()) { // 1depth 안에서 파일만 추출
            if( name.getName().endsWith("mp3") || name.getName().endsWith(".mp4")) {
                totalFilePath.add(name.getCanonicalFile());
            }
        }
    }
}
```

```
mapper  insert
        (#{album}, #{artist}, #{genre}, #{releasedate})
    </insert>

    <insert id="song" parameterType="kr.kina.vo.SongVO">
        insert into song
        values
        (#{albumid}, #{album}, #{title}, #{artist}, #{duration}, #{tracknum}, #{releaseDate}, #{filepath})
    </insert>

    <select id="select" parameterType="java.util.Map" resultType="int" >
        select albumid from
        album
        where album = #{album}
        and
        artist= #{artist}
        and
        genre = #{genre}
    </select>

    <select id="selectLastId" resultType="int">
        select * from
        album
        order by
        albumid desc limit 1;
    </select>

    <insert id="addAlbum" parameterType="kr.kina.vo.AlbumVO">
        insert into album
        (album, artist, genre, releasedate)
        values
        (#{album}, #{artist}, #{genre}, #{releasedate})
    </insert>
```

Process of Development

개발과정 및 내용

01

음원의 메타데이터 추출 및
데이터 생성 과정

-개발 환경

-데이터 구조 및 생성

-아키텍처

-구현 내용

02

웹 어플리케이션 개발 과정

-개발환경

-아키텍처

-시퀀스 다이어그램

-구현 내용

02

웹 어플리케이션 개발 과정

1-1.개발환경

· IDE : STS (Spring Tool Suite) 3.8.1.RELEASE

· WAS : Tomcat 7.0

· OS : Window7 (64 bit)

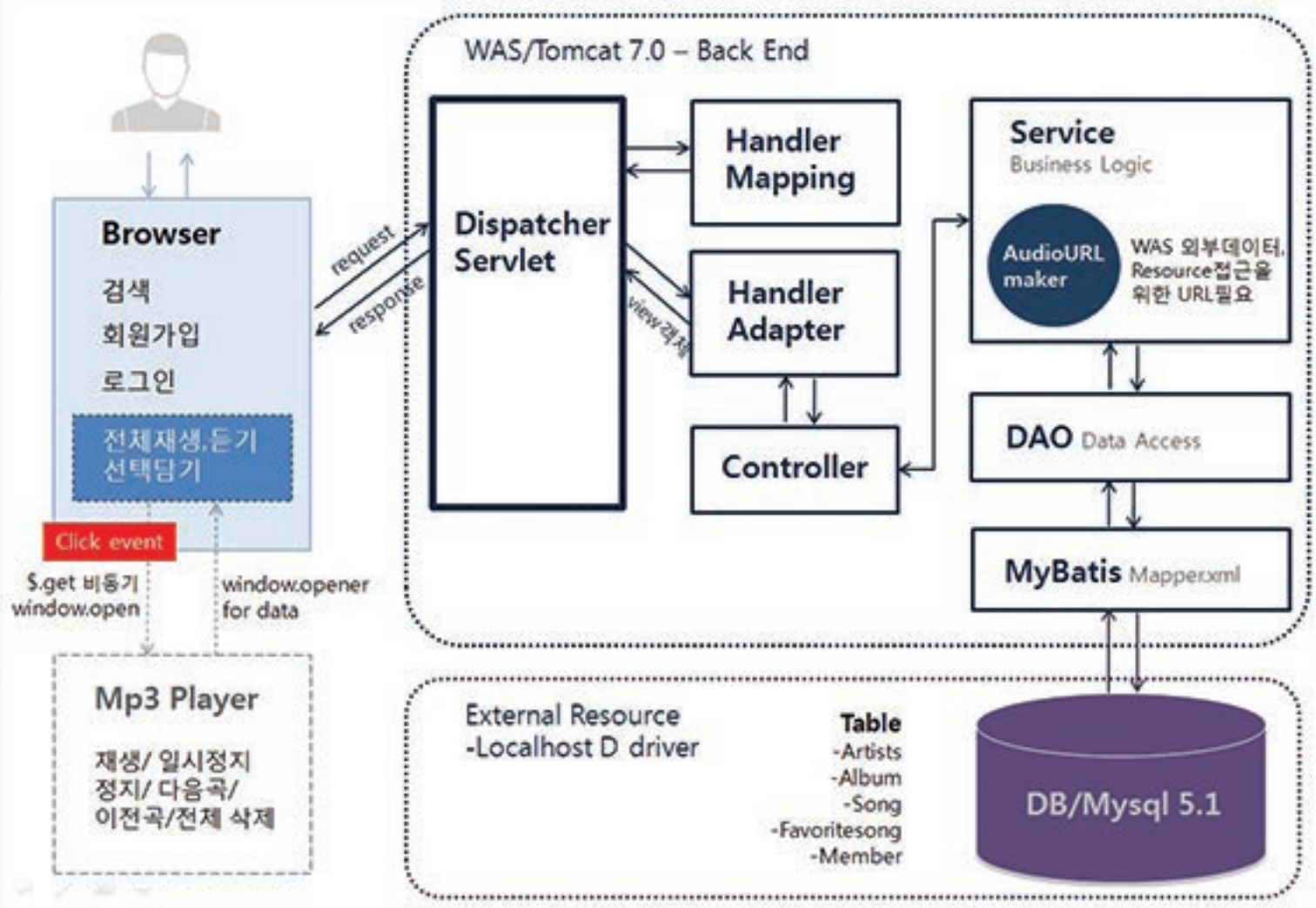
· Language : Java, HTML, CSS, Javascript, JQuery, Ajax

· DB : MySQL 5.1.60

· Persistence Framework : MyBatis

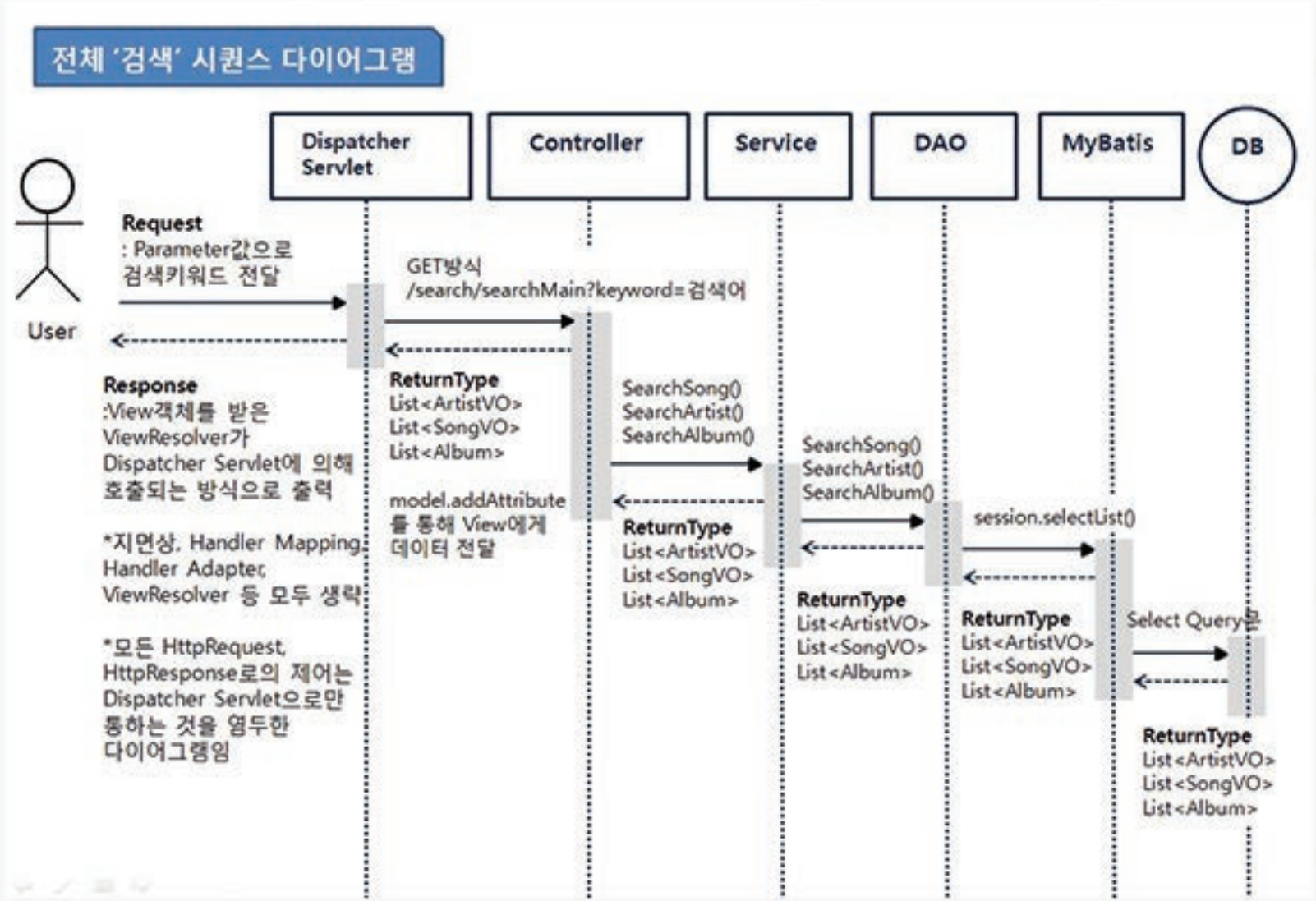
02 웹 어플리케이션 개발 과정

1-2.아키텍처



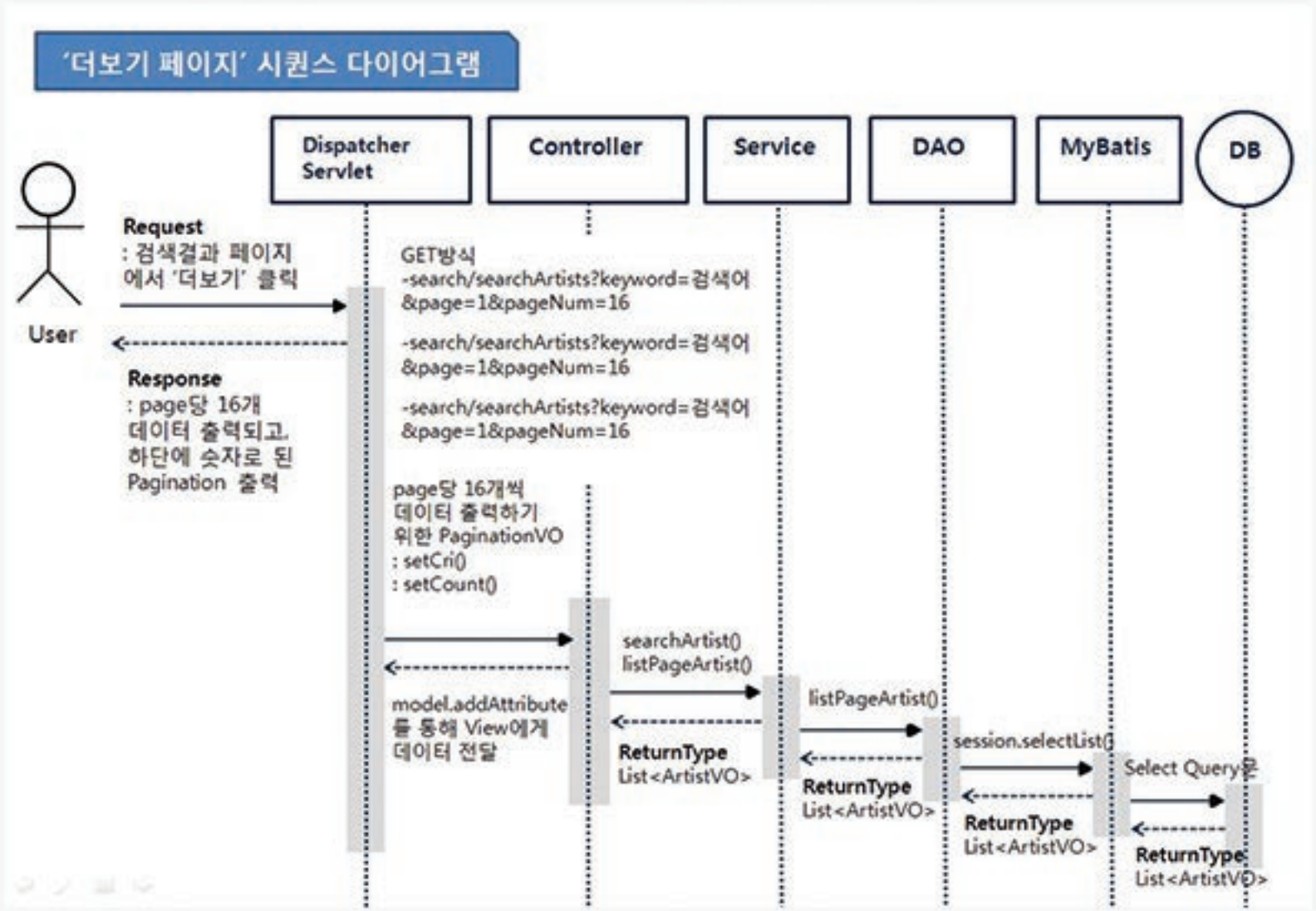
02 웹 어플리케이션 개발 과정

1-3.시퀀스 다이어그램 - 전체 검색



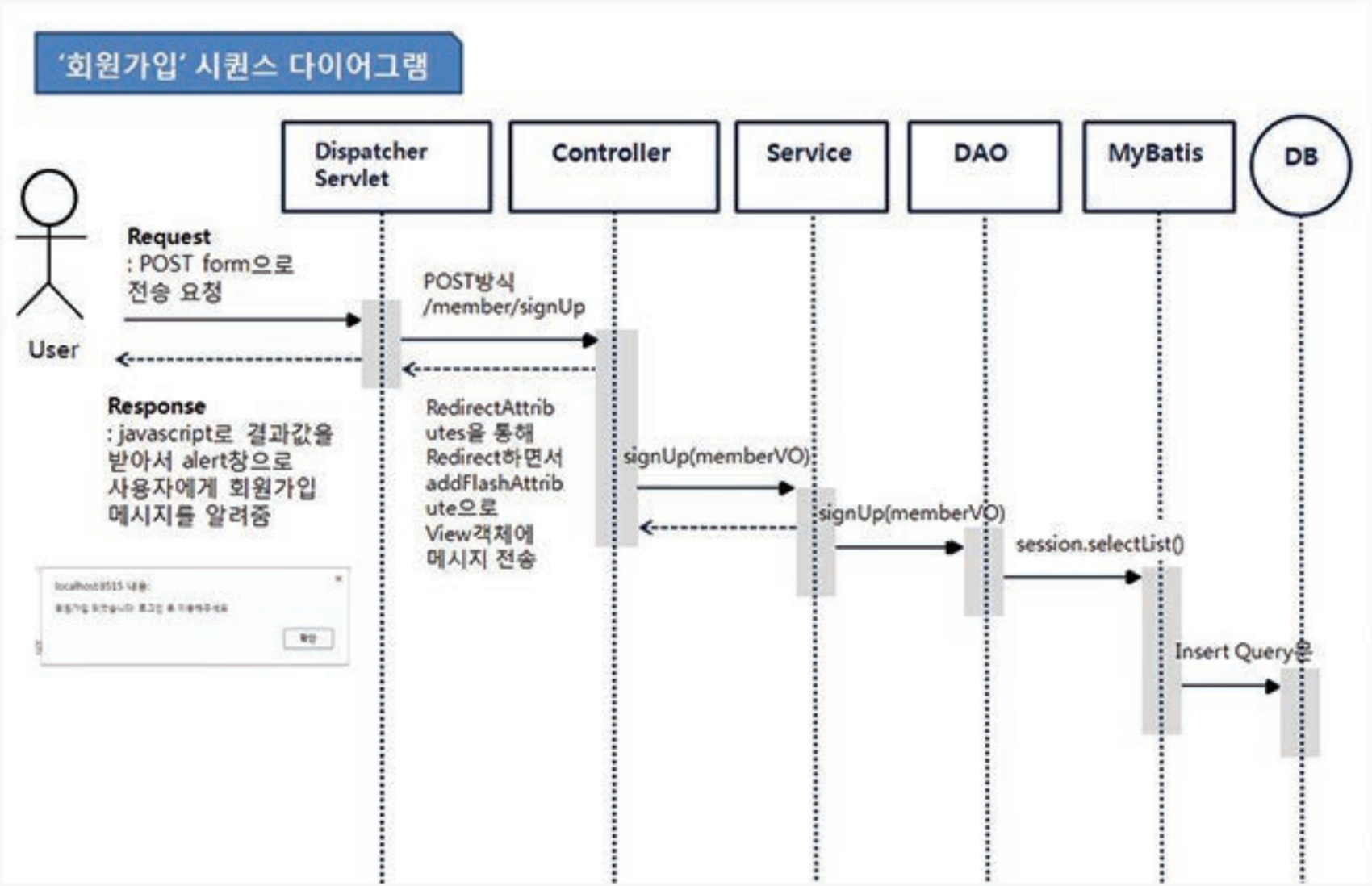
02 웹 어플리케이션 개발 과정

1-3.시퀀스 다이어그램 - 더보기 클릭시



02 웹 어플리케이션 개발 과정

1-3.시퀀스 다이어그램 - 회원가입



02 웹 어플리케이션 개발 과정

1-4.구현내용

주요 기능

1) 검색

· 아티스트명/곡명/앨범명으로 모두 검색 가능

· 메인화면 검색 > 검색결과 화면에서 데이터의 일부분 출력

· 검색결과 화면에서 '더보기'링크 클릭 시, 검색어에 따른 모든 데이터 출력 (pagination)

· 검색결과 화면에서 아티스트명/앨범명 클릭 시, 해당되는 모든 곡 출력

2) 회원가입 및 로그인

· '마이뮤직'페이지 회원전용 (좋아하는 곡을 따로 담아 재생리스트를 만들 수 있음)

· 해당 곡의 '좋아요' 버튼 클릭 시, '마이뮤직' 페이지에서 곡 추가 생성

3) 음악 스트리밍

· 전체 재생 및 전체 삭제, 한 곡 듣기, 선택 담기를 통해 현재 재생리스트에 곡 추가 가능

02 웹 어플리케이션 개발 과정

1-4.구현내용

주요 기능

1) 검색

· 아티스트명/곡명/앨범명으로 모두 검색 가능

· 메인화면 검색 > 검색결과 화면에서 데이터의 일부분 출력

· 검색결과 화면에서 '더보기'링크 클릭 시, 검색어에 따른 모든 데이터 출력 (pagination)

· 검색결과 화면에서 아티스트명/앨범명 클릭 시, 해당되는 모든 곡 출력

2) 회원가입 및 로그인

· '마이뮤직'페이지 회원전용 (좋아하는 곡을 따로 담아 재생리스트를 만들 수 있음)

· 해당 곡의 '좋아요' 버튼 클릭 시, '마이뮤직' 페이지에서 곡 추가 생성

3) 음악 스트리밍

· 전체 재생 및 전체 삭제, 한 곡 듣기, 선택 담기를 통해 현재 재생리스트에 곡 추가 가능

02 웹 어플리케이션 개발 과정

1-4.구현내용

```
@Controller
@RequestMapping("/player")
public class PlayerController {

    static final Logger log = LoggerFactory.getLogger(PlayerController.class);

    @Inject
    AudioURLMaker url;

    /** make audioURL
     *  @return List<> all data include audioURL
     * */
    @RequestMapping(value="/playList", method=RequestMethod.POST, consumes="application/json; charset=UTF-8")
    public @ResponseBody List<Map<String,String>> playListPOST(@RequestBody List<Map<String,String>> songData) throws Exception {
        log.debug("playList : One or All");

        for(Map<String,String> song : songData){
            song.put("audioSrc", url.urlMaker(song.get("filePath")));
            song.remove("filePath");
        }
        return songData;
    }

    /** make audioURL
     *  @resutn List<> added data & newer add data
     * */
    @RequestMapping(value="/addList", method=RequestMethod.POST, consumes="application/json; charset=UTF-8")
    public void addListPOST(@RequestBody List<Map<String,String>> beforeData,
                            @RequestBody List<Map<String,String>> addData, Model model) throws Exception {
        log.debug("addList ..");

        List<Map<String,String>> result = new ArrayList<>();
```

02 웹 어플리케이션 개발 과정

1-4.구현내용

```
//CheckBox total check event
$(document).ready(function(){
    $("#totalListCheck").click(function(){
        if($("#totalListCheck").prop("checked")){
            $("input[name='addsong']").prop("checked", true);
        }else{
            $("input[name='addsong']").prop("checked", false);
        }
    });
});


//Like it favorite song
function likeit(no){

    var likeitSong = songWrapper(doc.paramValue[no-1]);
    likeitSong.userid = sessionId;

    $.ajax({
            type : "POST",
            url  : "/favorite/savedSong",
            data : JSON.stringify(likeitSong),
            success : function(result) {
                alert(result);
            },
            error:function(request,status,error){
                console.log("code:"+request.status+"\n"+"message:"+request.responseText+"\n"+"error:"+error);
              },
            headers : {
                'Accept' : 'application/json; charset=UTF-8',
                'Content-Type' : 'application/json; charset=UTF-8'
            }
        });
}
```

감사합니다.

깃허브
https://github.com/kinayoon/MetadataExtractor
https://github.com/kinayoon/Music-Searching-and-Streaming

유튜브
https://www.youtube.com/watch?v=PJmvVzF63wY&feature=youtu.be

이메일
kina2018@gmail.com