

공공데이터 아키텍처 설계

공공데이터 아키텍처 설계

1. Open API의 이해

Chapter 1	Open API 정의 이해	4
Chapter 2	Open API 이용 허락 범위 및 라이선스 확인	6

2. 공공 데이터 API 설계

Chapter 1	공공데이터 Open API 활용 신청	10
Chapter 2	공공데이터 요청하기	17
Chapter 3	요청한 데이터 응답 받기	20
Chapter 4	응답 데이터 파싱(XML/JSON)	24
Chapter 5	파싱 데이터를 Java 객체로 변환	34
Chapter 6	DB에 데이터 저장하기	45

공공데이터 아키텍처 설계

3. 데이터 배포를 위한 REST 설계

Chapter 1	RestController 구축(비동기 통신)	59
Chapter 2	요청 데이터 형식에 맞는 SQL 작성(Mybatis)	66
Chapter 3	JSON 형태로 응답하기	72

Part 1. Open API의 이해

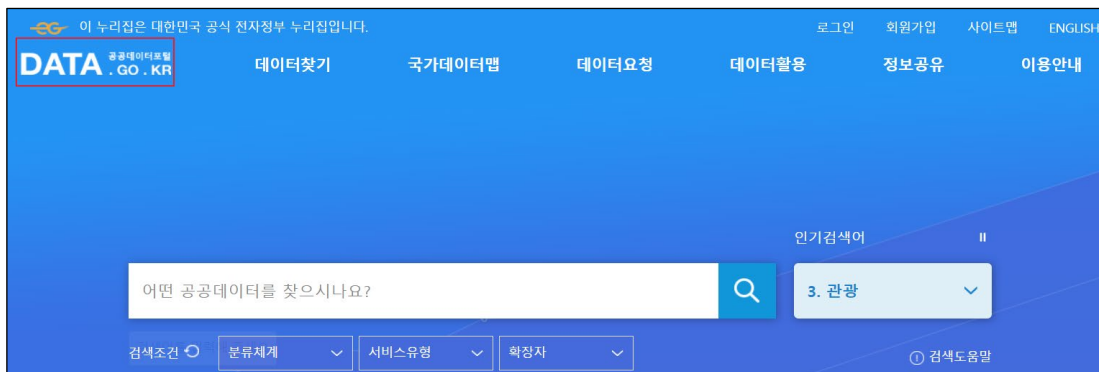
Chapter 1. Open API 정의 이해

· Open API(Application Programming Interface) 정의

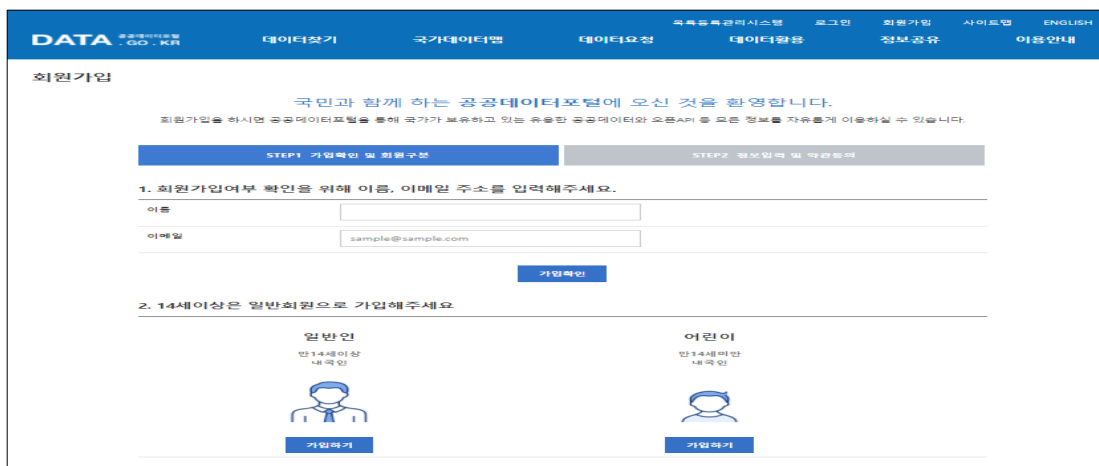
- 누구나 쉽게 사용할 수 있게 공개된 API
- 데이터를 표준화하고 프로그래밍해서 개발자들이 바로 개발에 활용할 수 있는 형태의 개방 형식
- 대표적으로 정부에서 제공하는 공공데이터포털(data.go.kr)을 예로 들 수 있음

- 공공데이터를 사용하기 위한 사이트 접속 및 회원 가입

* data.go.kr 사이트 접속



* 우측 상단 회원 가입 클릭하여 가입 진행



기본정보 입력

*표시는 필수 입력항목입니다.

* 아이디

중복확인

4~20자 이내 영문 소문자, 숫자만 가능합니다.

* 비밀번호

반복, 연속문자, 개인정보 관련은 사용불가
 숫자, 영문자, 특수문자(!, @, #, \$, %, ^, &, +, -, ~, *) 조합 9자 이상 ~ 20자 이하

* 비밀번호 확인

개인정보입력

* 이름

윤영승

* 이메일

prince215312@nate.com

이메일 인증

☐ 이메일 수신 동의

수신 거부 시 일부 서비스에 제한이 있을 수 있습니다.
 비밀번호 초기화 메일 수신 등에 반드시 필요한 정보이므로 정확히 이메일 주소를 입력해주세요.

전화번호

* 휴대전화번호

☐ SMS 수신 동의
 수신 거부 시 일부 서비스에 제한이 있을 수 있습니다.

* 일반회원 인증

휴대폰 인증

본인 인증하기

* 자동등록방지

yrew7

새로고침

보안문자를 입력하세요.

☐ 공공데이터포털 서비스 이용약관에 동의합니다.

제1조 목적

본 약관은 공공데이터포털(이하 '포털'이라 함니다)에서 제공되는 서비스의 이용과 관련하여, 이용조건 및 절차, 회원과 운영기관의 권리·의무 및 책임, 그 외 필요한 사항을 규정함을 목적으로 합니다.

제2조 용어의 정의

본 약관에서 사용하는 용어의 정의는 다음과 같습니다.

1. "회원"이란 포털에서 제공하는 소정의 회원가입절차에 따라 이용약관에 동의하고 가입을 완료한 본인 또는 개인을 말합니다.
2. "이용자"란 포털에 회원으로 가입하고 포털이 제공하는 서비스를 이용하는 자를 말하며, 공공기관을 제외합니다.

☐ (필수) 개인정보 수집 및 이용에 동의합니다.☐ (선택) 개인정보 수집 및 이용에 동의합니다.

가. 수집하는 개인정보의 항목

공공데이터포털은 다음의 개인정보 항목을 수집하고 있습니다.

- 공공데이터포털 회원정보
 - 회원가입 시 입력해야 하는 필수항목과 선택항목이 있으며, 선택항목은 입력하지 않으셔도 서비스 이용에는 제한이 없습니다.
- 일반회원가입
 - 필수항목 : 이름, 아이디, 비밀번호, 이메일, 휴대전화번호
 - 선택항목 : 전화번호
- 서비스 이용과정에서 아래와 같은 정보들이 자동으로 생성되어 수집될 수 있습니다.
 - 항목 : 방문 일시, 서비스 이용기록, IP 주소

☐ 전체 약관에 동의합니다.

취소

작성 완료

Chapter 2. Open API 이용 허락 범위 및 라이선스 확인

공공데이터 이용 정책

- 공공데이터 포털에서 제공하는 공공데이터는 공공데이터법에 따라 누구나 이용이 가능
- 영리목적의 이용을 포함한 자유로운 활용이 보장됨
- 위 사항은 공공데이터법 제1조, 제3조에 해당

공공데이터 제공

- 공공기관이 이용자로 하여금 공공데이터에 접근할 수 있게 하거나 이를 다양한 방식으로 전달하는 것을 의미

공공데이터 이용 방법

- 공공데이터포털을 통해 제공 중인 공공데이터는 별도의 신청절차 없이 이용 가능
- 제공되는 공공데이터의 목록은 각 공공기관의 홈페이지에서도 확인 가능
- 공공데이터포털에서 제공하고 있지 않은 데이터의 경우 제공 신청을 통해 이용할 수 있으나, 공공데이터법 제17조 상의 제외대상 정보가 포함된 경우 제공이 거부될 수 있으며, 이 경우 공공데이터제공분쟁조정위원회에 조정을 신청할 수 있음

공공데이터 이용허락범위

- 공공데이터 이용허락범위 관련하여 "이용허락범위 제한 없음"일 경우 자유로운 이용 가능
- 공공데이터 관리지침 제8조
 - 저작물이 포함된 공공데이터의 관리
 - ① 공공기관의 장은 저작물이 포함된 공공데이터를 관리하는 경우 제3자의 권리가 침해되지 않도록 권리 소재 및 권리자의 이용허락 범위를 명확히 파악해야

한다. 이 경우 권리자의 이용허락은 서면 등 명시적으로 의사를 확인할 수 있는 방법으로 받아야 한다.

② 공공기관의 장은 저작물이 포함된 공공데이터를 제공할 때에는 「저작권법」 등 관련 법령에 따른 공공누리 유형을 부착하여 공공데이터의 이용허락 범위를 표시해야 한다. 이 경우 소관 공공데이터에 제3자의 권리가 포함된 경우라도 권리자가 허락하는 경우에는 이용자가 저작물을 자유롭게 이용할 수 있도록 제공해야 한다.

③ 공공기관의 장은 소관 공공데이터에 포함된 저작물이 업무상 작성하여 공표된 것이거나 계약에 따라 해당 저작물의 저작권 전부를 보유한 경우에는 저작권법 제24조의2제1항 각 호의 어느 하나에 해당하는 경우를 제외하고 이용자가 별도의 이용허락 없이 자유롭게 이용할 수 있도록 제공해야 한다.

공공데이터 이용허락범위 유형

- 저작자표시

(가능한 이용)

복사/배포 가능, 상업적 이용 가능, 변경 후 2차 저작물 제작 가능

(제한된 이용)

저장자 및 출처만 표시한다면 제한없이 자유롭게 이용 가능

- 저작자표시-비영리

(가능한 이용)

복사/배포 가능, 변경 후 2차 저작물 제작 가능

(제한된 이용)

상업적 이용 불가

- 저작자표시-변경금지

(가능한 이용)

복사/배포 가능, 상업적 이용 가능

(제한된 이용)

저작물을 변경하거나, 이용해 2차 저작물을 제작하면 안 됨

- 저작자표시-비영리-변경금지

(가능한 이용)

복사/배포 가능, 단 반드시 저작자 및 출처를 표시해야 함

(제한된 이용)

상업적 이용 불가, 저작자 및 출처만 표시한다면 제한없이 자유롭게 이용 가능

- 저작자표시-동일조건변경허락

(가능한 이용)

복사/배포 가능, 상업적 이용 가능

저작물을 변경하거나, 이용해 2차 저작물을 만들어도 되지만, 반드시 원저작자 및 출처를 표시해야 함

(제한된 이용)

2차 저작물에 원저작물과 동일한 라이선스를 적용해야 함

- 저작자표시-비영리-동일조건변경허락

(가능한 이용)

복사/배포 가능, 반드시 저작자 및 출처를 표시해야 함

저작물을 변경하거나, 이용해 2차 저작물 생성 가능, 반드시 원저작자 및 출처를 표시해야 함

(제한된 이용)

상업적 이용 불가

2차 저작물에 원저작물과 동일한 라이선스를 적용해야 함

Part 2. 공공 데이터 API 설계

Chapter 1. 공공데이터 Open API 활용 신청

- 오픈 API를 사용하려면 활용신청 후 인증키를 부여받은 후 사용 가능
- 로그인 -> 검색창에서 "기상청 관광지" 입력 후
➔ 상단 오픈 API 선택, "기상청_관광코스별 관광지 상세 날씨 조회서비스" 클릭

전체(2,753건) 파일데이터(1,730건) **오픈 API(360건)** 표준데이터셋5개(663건)

정확도순 ▼

오픈 API (360건)

과학기술 국가행정기관 국가중점

XML JSON 기상청_관광코스별 관광지 상세 날씨 조회서비스

관광코스별 관광지의 동네예보, 기상지수예보, 시군구별 관광기후지수 정보를 조회하는 서비스

제공기관 기상청 수정일 2021-08-04 조회수 18404 활용신청 2037 키워드 관광코스별 동네예보, 기상지수예보, 시군구별관광기후지수

- OpenAPI 상세화면에서 활용신청 버튼 클릭

오픈API 상세

f t y URL 복사

XML JSON 기상청_관광코스별 관광지 상세 날씨 조회서비스

활용신청

관광코스별 관광지의 동네예보, 기상지수예보, 시군구별 관광기후지수 정보를 조회하는 서비스

6 0 관심

· 활용목적 : 참고자료 선택 후 “직업훈련교육 참고자료” 입력

OpenAPI 개발계정 신청

JSON+XML 기상청_관광코스별 관광지 상세 날씨 조회서비스

제공기관	기상청	서비스유형	REST
심의여부	자동승인	신청유형	개발계정 활용신청
처리상태	신청	활용기간	승인일로부터 24개월 간 활용가능

공공데이터 제공제도

- * 공공데이터중 위치정보를 포함한 서비스를 사용하고자 하는 사업자는 '위치정보의 보호 및 이용 등에 관한 법률'에 따라 방송통신위원회에 '위치정보서비스 허가'를 받거나 '위치기반 서비스사업 신고'를 하여야 합니다.
- * 이에 해당하는 사업자인 경우에는 정부파일에 '위치기반서비스사업신고필증'을 첨부해 주시기 바랍니다.
- * 활용신청 시 '위치기반서비스사업신고필증'이 등록되지 않으면 반려가 될 수 있으니 참고하시기 바랍니다.

활용목적 선택

*표시는 필수 입력항목입니다.

* 활용목적

☐ 웹 사이트 개발 ☐ 앱개발 (모바일,솔루션등) ☐ 기타 ☒ 참고자료 ☐ 연구(논문 등)

직업훈련교육 참고자료

. 하단 활용신청 버튼 클릭하여 신청 완료

첨부파일

파일 선택

Drag & Drop으로 파일을 선택 가능합니다.

상세기능정보 선택

<input checked="" type="checkbox"/>	상세기능	설명	일일 트래픽
<input checked="" type="checkbox"/>	시군구별관광기후지수조회	조회조건(현재시각, 예보기간, 시군구 아이디)으로 시군구별 관광기후지수 데이터(시군구이름, 관광기후지수, 관광기후등급 등)를 조회하는 기능	10000
<input checked="" type="checkbox"/>	기상지수예보조회	조회조건(현재시각, 예보기간, 코스ID)으로 기상지수예보 데이터 (코스명, 관광지명, 테마, 식중독지수, 체감온도, 자외선지수 등)를 조회하는 기능	10000
<input checked="" type="checkbox"/>	동네예보조회	조회조건(현재시각, 예보기간, 코스ID)으로 동네예보 데이터(코스명, 관광지명, 테마, 기온, 습도 등)를 조회하는 기능	10000

라이선스 표시

*이용허락범위

저작자표시
☒ 동의합니다.

취소

활용신청

. 마이페이지 하단 신청한 내용에 대한 “승인” 확인

과학기술

기상청

활용신청 [승인] 기상청_관광코스별 관광지 상세 날씨 조회서비스

신청일 2023-02-02 만료예정일 2025-02-02

- 활용신청을 하고 나면 마이페이지에서 신청한 데이터에 대한 정보 확인이 가능하며 해당 내용 클릭하면 개발계정 상세보기 항목이 화면에 출력됨
- 기본정보 및 서비스정보

기본정보

데이터명	기상청_관광코스별 관광지 상세 날씨 조회서비스			상세설명
서비스유형	REST	심의여부	자동승인	
신청유형	개발계정 활용신청	처리상태	승인	
활용기간	2023-02-02 ~ 2025-02-02			

서비스정보

참고문서	기상청27_관광코스별_관광지_상세_날씨_조회서비스_오픈API활용가이드.docx
데이터포맷	JSON+XML
End Point	http://apis.data.go.kr/1360000/TourStnInfoService

* 서비스 정보부분을 보면 참고 문서가 있는데 참고문서에는 데이터활용에 대한 더 상세한 정보가 있기 때문에 공공데이터를 이용할때는 이 문서를 이용하는것이 유용함

- 인증키 확인

일반 인증키 (Encoding)	ey51rbqUG4nLxm%2FIMXybyk%2FsuSNe95v4algn4Ydn0UhcJl91gToKXy7htiikWXYGCVxm6xyZpakkO4hk%2FLs3w%3D%3D
----------------------	---

- 활용신청 상세기능 정보

활용신청 상세기능정보				
NO	상세기능	설명	일일 트래픽	미리보기
1	시군구별관광기후지수조회	조회조건(현재시각, 예보기간, 시군구 아이디)으로 시군구별 관광기후지수 데이터(시군구이름, 관광기후지수, 관광기후등급 등)를 조회하는 기능	10000	<input type="button" value="확인"/>
2	기상지수예보조회	조회조건(현재시각, 예보기간, 코스ID)으로 기상지수예보 데이터 (코스명, 관광지명, 테마, 식중독지수, 체감온도, 자외선지수 등)를 조회하는 기능	10000	<input type="button" value="확인"/>
3	동네예보조회	조회조건(현재시각, 예보기간, 코스ID)으로 동네예보 데이터(코스명, 관광지명, 테마, 기온, 습도 등)를 조회하는 기능	10000	<input type="button" value="확인"/>

* 실습은 동네예보조회 항목을 이용해 진행할 예정이므로 동네예보조회 -> 확인

* 요청 변수(Request Parameter) 항목 확인

요청변수(Request Parameter) 닫기		
항목명	샘플데이터	설명
ServiceKey	-	공공데이터포털에서 받은 인증키
pageNo	1	페이지번호
numOfRows	10	한 페이지 결과 수
dataType	XML	요청자료형식(XML/JSON)
CURRENT_DATE	2019122010	2016-12-01 01시부터 조회
HOURL	24	CURRENT_DATE부터 24시간 후까지의 자료 호출
COURSE_ID	1	관광 코스ID
<div>미리보기</div>		

- 미리보기를 클릭하면 요청 후 응답데이터에 대한 내용 확인이 가능

```
▼<response>
  ▼<header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL_SERVICE</resultMsg>
  </header>
  ▼<body>
    <dataType>XML</dataType>
    ▼<items>
      ▼<item>
        <tm>2019-12-22 09:00</tm>
        <thema>문화/예술</thema>
        <courseId>1</courseId>
        <courseAreaId>47000000000</courseAreaId>
        <courseAreaName>경상북도</courseAreaName>
        <courseName>남호고택에서의 하룻밤</courseName>
        <spotAreaId>102</spotAreaId>
        <spotAreaName>봉화</spotAreaName>
        <spotName>(봉화)춘양목 송이마을</spotName>
        <th3>-2</th3>
        <wd>297</wd>
        <ws>0</ws>
        <sky>3</sky>
        <rh>90</rh>
        <pop>20</pop>
      </item>
      ▼<item>
        <tm>2019-12-22 09:00</tm>
        <thema>종교/역사/전통</thema>
        <courseId>1</courseId>
```

* 데이터 형식이 xml 로 되어있어 xml 파일 형식으로 반환

Chapter 2. 공공데이터 요청하기

- 요청 및 응답되는 내용에 대한 세부 내용 확인은 서비스정보 항목의 참고문서 참조

- 실습은 Chapter 3 ~ Chapter 4에 걸쳐

“기상청 관광코스별 관광지 상세 날씨 조회 서비스” 공공데이터의

“동네예보조회” 내용을 활용하여 실습 진행 예정

* 상세기능정보

1) [동네예보조회] 상세기능명세 [↗]			
a) 상세기능정보 [↗]			
상세기능 번호 [↗]	1 [↗]	상세기능 유형 [↗]	조회 (목록) [↗]
상세기능명(국문) [↗]	동네예보조회 [↗]		
상세기능 설명 [↗]	조회조건(현재시각, 예보기간, 코스 ID)으로 동네예보 데이터(코스명, 관광지명, 테마, 기온, 습도 등)를 조회하는 기능 [↗]		
Call Back URL [↗]	http://apis.data.go.kr/1360000/TourStnInfoService/getTourStnVilageFcst [↗]		
최대 메시지 사이즈 [↗]	[910] byte [↗]		
평균 응답 시간 [↗]	[300] ms [↗]	초당 최대 트랙잭션 [↗]	[30] tps [↗]

* 요청 메시지 명세 내용 확인

b) 요청 메시지 명세					
항목명(영문)	항목명(국문)	항목크기	항목구분	샘플데이터	항목설명
serviceKey	인증키	100	1	인증키 (URL Encode)	공공데이터포털에서 발급받은 인증키
numOfRows	한 페이지 결과 수	4	1	0	한 페이지 결과 수 Default: 10
pageNo	페이지 번호	4	1	0	페이지 번호 Default: 1
dataType	응답자료형식	4	0	XML	요청자료형식(XML/JSON) Default: XML
CURRENT_DATE	현재시각	4	1	2016120101	2016-12-01 01시부터 조회
HOUR	예보기간	4	1	24	CURRENT_DATE 부터 24 시간 후까지의 자료 호출 (#별첨 2 참고)
COURSE_ID	코스 ID	20	1	1	관광 코스 ID

※ 항목구분 : 필수(1), 옵션(0), 1 건 이상 복수건(1..n), 0 건 또는 복수건(0..n)

- 응답데이터 상세명세 확인

c) 응답 메시지 명세					
항목명(영문)	항목명(국문)	항목크기	항목구분	샘플데이터	항목설명
numOfRows	한 페이지 결과 수	4	1	1	한 페이지당 표출 데이터 수
pageNo	페이지 번호	4	1	1	페이지 수
totalCount	데이터 총 개수	10	1	1	데이터 총 개수
resultCode	응답메시지 코드	2	1	00	응답 메시지코드
resultMsg	응답메시지 내용	100	1	NORMAL SERVICE	응답 메시지 설명
dataType	데이터 타입	4	1	XML	응답자료형식 (XML/JSON)
tm	예보시각	4	DATE	201612010100	동네예보 예보 시각
courseld	코스 ID	20	VARCHAR	1	관광 코스 ID
courseAreald	코스- 지역아이디	10	NUMBER	47000000000	코스 지역아이디
courseAreaName	코스-지역이름	20	VARCHAR	경북	코스 지역이름

spotAreald	관광지- 지역아이디	10	NUMBER	4792031001	관광지점 지역아이디
spotAreaName	관광지-지역 이름	20	VARCHAR	봉화	관광지점 지역이름
courseName	코스 명	100	VARCHAR	남호고택에서의 하룻 밤	관광코스 명
spotName	관광지명	50	VARCHAR	계서당(성이성 생가)	관광지 명
thema	테마	20	VARCHAR	생태&힐링	관광지 명
th3	일 3 시간 기온	5	NUMBER	21	3 시간 기온
maxTa	최고기온	5	NUMBER	25.4	최고 기온
minTa	최저기온	5	NUMBER	17.4	최저 기온
wd	풍향	5	NUMBER	237	풍향
ws	풍속	5	NUMBER	2.8	풍속
sky	하늘상태	8	VARCHAR	W2	하늘상태

rh	습도	5	NUMBER	90	습도
pop	강수확률	5	NUMBER	60	강수확률
rn	강수량	5	NUMBER	10	강수량

※ 항목구분 : 필수(1), 옵션(0), 1 건 이상 복수건(1..n), 0 건 또는 복수건(0..n), 코드표별첨

Chapter 3. 요청한 데이터 응답 받기

- STS(Spring Tool Suite) 실행

- * New -> Spring Legacy Project -> Project Name : TourWeatherProject ->
Spring MVC Project 선택 -> com.kh.TourWeather -> Finish
- * TourWeatherProject > src > main > java > com > kh > TourWeather
HomeController.java 삭제
- * TourWeatherProject 우 클릭 -> Properties -> Project Facets에서 자바버전 체크
- * TourWeather 우 클릭 -> New -> Package -> Name : com.kh. TourWeather.model.vo
- * vo 우 클릭 -> New -> Class -> Name : TourWeatherVO

- TourWeatherVO.java 구성

* 구성 내용 : 필드 / 기본생성자 / getter_setter / toString / 매개변수 생성자

TourWeatherVO.java	
01	package com.kh.TourWeather.model.vo;
02	
03	public class TourWeatherVO {
04	private String spotAreaName; // 관광지-지역 이름
05	private String spotName; // 관광지명
06	private String th3; // 일 3시간 기온
07	private String rhm; // 습도
08	private String pop; // 강수확률
09	
10	public TourWeatherVO() {}
11	
12	public TourWeatherVO(String spotAreaName, String spotName, String th3,
13	String rhm,String pop) {
14	super ();
15	this .spotAreaName = spotAreaName;
16	this .spotName = spotName;
17	this .th3 = th3;
18	this .rhm = rhm;
19	this .pop = pop;
20	}
21	public String getSpotAreaName() {
22	return spotAreaName;
23	}
24	
25	public void setSpotAreaName(String spotAreaName) {
26	this .spotAreaName = spotAreaName;
27	}
28	
29	public String getSpotName() {
30	return spotName;


```

31     }
32
33     public void setSpotName(String spotName) {
34         this.spotName = spotName;
35     }
36
37     public String getTh3() {
38         return th3;
39     }
40
41     public void setTh3(String th3) {
42         this.th3 = th3;
43     }
44
45     public String getRhm() {
46         return rhm;
47     }
48
49     public void setRhm(String rhm) {
50         this.rhm = rhm;
51     }
52
53     public String getPop() {
54         return pop;
55     }
56
57     public void setPop(String pop) {
58         this.pop = pop;
59     }
60
61     @Override
62     public String toString() {
63         return "TourWeatherVO [spotAreaName=" + spotAreaName + ",
64 spotName=" + spotName + ", th3=" + th3 + ", rhm="

```

65	+ rhm + ", pop=" + pop + "]" ;
66	}
67	}

Chapter4. 응답 데이터 파싱(XML/JSON)

- 응답데이터를 이용한 자바 애플리케이션 개발 실습

* TourWeather 우 클릭 -> New -> Package -> Name : com.kh.TourWeather.run

* run 우 클릭 -> New -> Class -> Name : TourWeatherJavaApp -> Finish

TourWeatherJavaApp.java	
01	package com.kh.TourWeather.run;
02	
03	import java.io.BufferedReader;
04	import java.io.IOException;
05	import java.io.InputStreamReader;
06	import java.net.HttpURLConnection;
07	import java.net.URL;
08	import java.time.LocalDateTime;
09	import java.time.format.DateTimeFormatter;
10	import java.util.ArrayList;
11	
12	import com.google.gson.JsonArray;
13	import com.google.gson.JsonObject;
14	import com.google.gson.JsonParser;
15	import com.kh.TourWeather.model.vo.TourWeatherVO;
16	
17	
18	public class TourWeatherJavaApp {
19	
20	// 발급받은 인증키(반드시 본인의 키 입력) 정보 변수에 담아두기
21	public static final String serviceKey =
22	"eyJ51rbqUG4nLXm%2FIMXybyk%2FsuSNe95v4algn4Ydn0UhcJI91gToKXy7htiiqkWXyG
23	CVxm6xyZpakkO4hk%2FLs3w%3D%3D";
24	
25	
26	public static void main(String[] args) throws IOException {

27	
28	// OpenAPI서버로 요청하고자 하는 URL만들기
29	// 응답받은 xml URL내용 중 ?전까지의 내용이 요청 URL,
30	따라서 ?전까지 복사, = 우측에 붙여넣기
31	String url =
32	"http://apis.data.go.kr/1360000/TourStnInfoService/getTourStnVilageFcst";
33	// url += "?serviceKey=서비스키"; // 서비스키가 제대로 부여되지
34	않았을 경우 SERVICE_KEY_IS_NOT_REGISTERED_ERROR 오류 발생
35	url += "?serviceKey=" + serviceKey;
36	url += "&numOfRows=10";
37	url += "&pageNo=1";
38	url += "&dataType=xml";
39	LocalDateTime datetime = LocalDateTime.now();
40	DateTimeFormatter formatter =
41	DateTimeFormatter.ofPattern("yyyyMMddHH");
42	url += "&CURRENT_DATE=" + datetime.format(formatter);
43	url += "&HOUR=24";
44	url += "&COURSE_ID=" + "1";
45	//System.out.println(url); 확인 후 잘 되면 주석처리
46	
47	// ** HttpURLConnection 객체를 활용해서 OpenAPI 요청절차 **
48	// 1. 요청하고자하는 url 전달하면서 java.net.URL 객체 생성
49	URL requestUrl = new URL(url);
50	
51	// 2. 1번과정으로 생성된 URL객체를 가지고, HttpURLConnection
52	객체 생성
53	HttpURLConnection urlConnection =
54	(HttpURLConnection)requestUrl.openConnection();
55	
56	// 3. 요청에 필요한 Header설정하기
57	urlConnection.setRequestMethod("GET");
58	
59	// 4. 해당 OpenAPI 서버로 요청 보낸 후 입력 스트림을 통해
60	응답데이터 읽어들이기

```

61         // xml형식으로 반환되는 내용에서 한 줄씩 읽어들이려 함
62         BufferedReader br = new BufferedReader(new
63     InputStreamReader(urlConnection.getInputStream()));
64             // 보조스트림
65             // 기반스트림(바이트 스트림, 1바이트)
66
67         String responseText = "";
68         String line;
69         while((line = br.readLine()) != null){
70             // System.out.println(line); 확인 후 주석처리
71             responseText += line;
72         }
73
74
75         System.out.println(responseText);
76
77         // 5. 사용한 스트림 반납
78         br.close();
79         urlConnection.disconnect();
80     }
81 }

```

* console 창 -> xml형식의 데이터 확인

- STS(Spring Tool Suite) 실행

- * xml 실습 시 진행했던 내용을 활용하여 json실습 진행
- * TourWeatherJavaApp.java 최종 파일 내용 38행 내용 변경
dataType = xml ➔ **dataType = json**
- * 저장 후 실행, 확인 -> 콘솔창 결과가 json형식 확인
- * JSONObject / JSONArray / JsonElement를 이용해 원하는 데이터 추출 실습을 진행할 예정이며, 세부적으로는 각각의 정보를 TourWeatherVO객체에 담고, ArrayList에 저장하는 형태로 진행
- * JSONObject / JSONArray / JsonElement 3개 항목은 gson라이브러리에서 제공
- * pom.xml 파일에 gson라이브러리 추가
 1. <https://mvnrepository.com/> -> gson검색 -> 2.8.6버전 클릭
 2. <dependency> ~~~ </dependency> 복사
 3. pom.xml 파일, </dependencies> 바로 윗줄에 붙여넣기 후 저장

```
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.6</version>
</dependency>
</dependencies>
```

4. 저장, JsonObject / JsonArray / JsonElement 3개 항목 사용 가능

* TourWeatherJavaApp.java 내용 중 현재,
responseText에 있는 내용(단순문자열)들을 파싱하기 위해 문자열을
JsonObject객체로 변경해야 추출이 가능

* TourWeatherJavaApp.java의 내용에서
75행 주석 처리 후 -> // System.out.println(responseText);
77행부터 내용부터 추가

TourWeatherJavaApp.java	
01	package com.kh.TourWeather.run;
02	
03	import java.io.BufferedReader;
04	import java.io.IOException;
05	import java.io.InputStreamReader;
06	import java.net.HttpURLConnection;
07	import java.net.URL;
08	import java.time.LocalDateTime;
09	import java.time.format.DateTimeFormatter;
10	import java.util.ArrayList;
11	
12	import com.google.gson.JsonArray;
13	import com.google.gson.JsonObject;
14	import com.google.gson.JsonParser;
15	import com.kh.TourWeather.model.vo.TourWeatherVO;
16	
17	
18	public class TourWeatherJavaApp {
19	

20	// 발급받은 인증키(반드시 본인의 키 입력) 정보 변수에 담아두기
21	public static final String serviceKey =
22	"ey51rbqUG4nLxm%2FIMXybyk%2FsuSNe95v4algn4Ydn0UhcJI91gToKXy7htiikWXYGCVxm6xyZ
23	pakkO4hk%2FLs3w%3D%3D";
24	
25	
26	public static void main(String[] args) throws IOException {
27	
28	
29	// OpenAPI서버로 요청하고자 하는 URL만들기
30	String url =
31	"http://apis.data.go.kr/1360000/TourStnInfoService/getTourStnVilageFcst";
32	// url += "?serviceKey=서비스키"; // 서비스키가 제대로 부여되지 않았을
33	경우 SERVICE_KEY_IS_NOT_REGISTERD_ERROR 오류 발생
34	
35	url += "?serviceKey=" + serviceKey;
36	url += "&numOfRows=10";
37	url += "&pageNo=1";
38	url += "&dataType=json";
39	LocalDateTime datetime = LocalDateTime.now();
40	DateTimeFormatter formatter =
41	DateTimeFormatter.ofPattern("yyyyMMddHH");
42	url += "&CURRENT_DATE=" + datetime.format(formatter);
43	url += "&HOUR=24";
44	url += "&COURSE_ID=" + "1";
45	// System.out.println(url); // 정상 요청되는지 확인 후 잘 되면 주석처리
46	
47	
48	// ** HttpURLConnection 객체를 활용해서 OpenAPI 요청절차 **
49	// 1. 요청하고자하는 url 전달하면서 java.net.URL 객체 생성
50	URL requestUrl = new URL(url);
51	
52	// 2. 1번과정으로 생성된 URL객체를 가지고, HttpURLConnection 객체
53	생성


```

54         HttpURLConnection urlConnection =
55 (HttpURLConnection)requestUrl.openConnection();
56
57         // 3. 요청에 필요한 Header설정하기
58         urlConnection.setRequestMethod("GET");
59
60         // 4. 해당 OpenAPI 서버로 요청 보낸 후 입력 스트림을 통해
61 응답데이터 읽어들이기
62         // xml형식으로 반환되는 내용에서 한 줄씩 읽어들이려 함
63         BufferedReader br = new BufferedReader(new
64 InputStreamReader(urlConnection.getInputStream()));
65                                     // 보조스트림
66 // 기반스트림(바이트 스트림, 1바이트)
67
68         String responseText = "";
69         String line;
70         while((line = br.readLine()) != null){
71             // System.out.println(line); 확인 후 주석처리
72             responseText += line;
73         }
74
75         // System.out.println(responseText);
76
77         // JsonObject / JSONArray, JsonElement를 이용해서 파싱할 수
78 있음(gson라이브러리에서 제공) => 내가 원하는 데이터만을 추출할 수 있음
79         // 각각의 정보를 => TourWeather객체에 담고 => ArrayList에 쌓기
80
81         // responseText에 담긴 문자열을 JsonObject객체로 변경
82         JsonObject totalObj =
83 JsonParser.parseString(responseText).getAsJsonObject();
84         // totalObj에 Json객체가 담겨 있음
85         JsonObject responseObj = totalObj.getAsJsonObject("response");
86         // response속성 접근 => 응답데이터 {}에 있는 내용들을 JsonObject화
87         // System.out.println(responseObj); 출력 확인 후 주석

```

88	JsonObject bodyObj =
89	responseObj.getAsJsonObject("body").getAsJsonObject("items");
90	// body속성접근 => 응답데이터 {}에 있는 내용들을 JsonObject화
91	
92	// System.out.println(bodyObj); // 출력확인 후 주석
93	
94	JsonArray itemArr = bodyObj.getAsJsonArray("item");
95	// items속성 접근 => [] JsonArray
96	
97	//System.out.println(itemArr); // 중간결과 확인 후 주석
98	
99	ArrayList<TourWeatherVO> list = new ArrayList<>();
100	for(int i=0; i<itemArr.size();i++) {
101	JsonObject item = itemArr.get(i).getAsJsonObject();
102	// System.out.println(item); // 출력확인 후 주석
103	
104	TourWeatherVO tw = new TourWeatherVO();
105	tw.setSpotAreaName(item.get("spotAreaName").getString());
106	tw.setSpotName(item.get("spotName").getString());
107	tw.setTh3(item.get("th3").getString());
108	tw.setRhm(item.get("rhm").getString());
109	tw.setPop(item.get("pop").getString());
110	
111	list.add(tw);
112	}
113	
114	// System.out.println(list); 확인 후 주석
115	for(TourWeatherVO a : list) {
116	System.out.println(a);
117	}
118	
119	
120	// 5. 사용한 스트림 반납
121	br.close();

122	urlConnection.disconnect();
123	
124	}
125	
126	}

* TourWeatherJavaApp.java 소스코드 작성 완료 후 저장/실행 -> console 확인

- 확인 결과 하단의 내용과 같이 출력되는지 최종 확인

TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)준양목 송이마을, th3=6, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)계서당 (성이성 쟁가), th3=7, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)계서당 (성이성 쟁가), th3=7, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)계서당 (성이성 쟁가), th3=7, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)지림사, th3=7, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)지림사, th3=7, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)지림사, th3=7, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)지림사, th3=7, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)계서당 (성이성 쟁가), th3=7, rhm=40, pop=0]
TourWeatherVO	[spotAreaName=봉화, spotName=(봉화)준양목 송이마을, th3=6, rhm=40, pop=0]

➔ 웹 페이지를 통한 요청에 대해 응답되어 온 데이터를 웹 브라우저에 출력하는
실습 내용은 Part 3에서 이어서 진행 예정

Chapter 5. 파싱 데이터를 Java 객체로 변환

- 코로나 19관련 공공데이터를 이용한 실습
- 공공데이터 구조 분석 및 클래스 설계
 - data.go.kr로그인 후 "보건복지부 코로나19 시도 발생현황" 검색

데이터목록

보건복지부_코로나19 시도발생_현황

본 콘텐츠에 대한 검색어를 입력하면 검색어를 반드시 포함하는 결과값을 표시합니다. (ex. "미세먼지 현황정보")

연관 제공기관별 검색 상세검색

"보건복지부_코로나19 시도발생_현황"에 대해 총 22,509건이 검색되었습니다.

조건검색 초기화

분류체계	서비스유형	제공기관유형	태그	확장자
<input type="checkbox"/> 공공행정 <input type="checkbox"/> 과학기술 <input type="checkbox"/> 교육 <input type="checkbox"/> 교통수송 <input type="checkbox"/> 국토관리 <input type="checkbox"/> 농축수산	<input type="checkbox"/> LINK <input type="checkbox"/> LOD <input type="checkbox"/> REST <input type="checkbox"/> RSS/ATOM <input type="checkbox"/> SOAP <input type="checkbox"/> 다운로드	<input type="checkbox"/> 공공기관 <input type="checkbox"/> 교육기관 <input type="checkbox"/> 교육행정기관 <input type="checkbox"/> 국가행정기관 <input type="checkbox"/> 위원회 <input type="checkbox"/> 입법기관	<input type="checkbox"/> 고유어 <input type="checkbox"/> 고분해화 <input type="checkbox"/> 교양시 <input type="checkbox"/> 고전 <input type="checkbox"/> 관광 <input type="checkbox"/> 덕후리	<input type="checkbox"/> CSV <input type="checkbox"/> DOC <input type="checkbox"/> DOCX <input type="checkbox"/> HWP <input type="checkbox"/> JPG <input type="checkbox"/> JSON

국가공공데이터 분류 조건 추가하기 + 조건열기

전체(22,509개) 파일데이터(19,654개) 오픈 API(2,619개) 표준데이터셋122개(236개)

정확도순 10개씩

오픈 API (2,619건)

보건복지부_코로나19 시도발생_현황 미리보기

코로나19감염증으로 인한 시·도별 신규확진자, 신규사망자, 격리중입원환자수, 격리해제환자수 등에 대한 현황자료 (이 제공자료는 관련 발생 상황에 대한 정보를 신속 투명하게 공개하기 위...

수정일 2020-12-10 조회수 18127 활용신청 2548 활용신청

XML 보건복지부_코로나19 감염_현황 미리보기

코로나19감염증으로 인한 일별 확진자, 완치자, 치료중입원환자, 사망자 등에 대한 현황자료(이 제공자료는 관련 발생 상황에 대한 정보를 신속 투명하게 공개하기 위한 것으로, 추가적인 역...

수정일 2020-12-10 조회수 40835 활용신청 4172 활용신청

XML 보건복지부_코로나19 해외발생_현황 미리보기

코로나19감염증으로 국가별 일일 확진자, 사망자, 사망률 등에 대한 현황자료 (이 제공자료는 관련 발생 상황에 대한 정보를 신속 투명하게 공개하기 위한 것으로, 추가적인 역학조사 결과...

수정일 2020-12-10 조회수 4144 활용신청 627 활용신청

XML 보건복지부_코로나19 연령별·성별감염_현황 미리보기

- 이 전 실습내용을 참조하여 오픈 API 상세 항목 확인 및 활용신청 진행

오픈API 상세

XML

보건복지부_코로나19 시·도발생_현황

7

0

관심

활용신청

오픈API 정보

메타데이터 다운로드

분류체계

보건 - 보건의료

관리부서명

공공데이터활용지원센터

API 유형

REST

활용신청

2556

등록

2020-04-15

심의유형

개발단계 : 허용 / 운영단계 : 허용

비용부과유무

무로

이용허락범위

이용허락범위 제한 없음

참고문서

03.보건복지부_OpenAPI_활용가이드_코로나바이러스감염증_시도발생_현황_조회_서비스_v0.3.2.docx

상세기능

활용사례

상세기능

목록

코로나19시·도발생_현황 조회 서비스

조회

코로나19감염증으로 인한 시·도별 신규확진자, 신규사망자, 격리중단환자수, 격리해제환자수 등에 대한 현황자료

· 활용유형 및 개발단계 : 허용 / 운영단계 : 허용

· 신청가능 트랙 1000000 / 운영계정용 활용사례 등록시 신청하면 트랙릭 증가 가능

· 요청주소

· 서비스URL http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19SidoInfStateJson

활용신청

마이페이지

오픈API

개발계정

활용현황

운영계정

연동키 발급현황

DATA

나의 문의

나의 관심

나의 제공신청

나의 분쟁조정

회원정보 수정

OpenAPI 개발계정 신청

XML

보건복지부_코로나19 시·도발생_현황

제공기관

공공데이터활용지원센터

서비스유형

REST

심의여부

자동승인

처리상태

신청

신청유형

개발계정 | 활용신청

활용기간

승인일로부터 24개월 간 활용가능

공공데이터 제공제도

· 공공데이터를 위차정보를 포함한 서비스를 사용하고자 하는 사업자는 '위차정보의 보호 및 이용 등에 관한 법률'에 따라 방송통신위원회에 '위차정보서

비스 허가'를 받거나 '위차기반 서비스사업 신고'를 하여야 합니다.

· 이에 해당하는 사업자인 경우에는 정부과정에 '위차기반서비스사업신고필증'을 첨부해 주시기 바랍니다.

· 활용신청 시 '위차기반서비스사업신고필증'이 등록되지 않으면 반려가 될 수 있으며 참고 하시기 바랍니다.

활용목적 선택

*표시는 필수 입력항목입니다.

*활용목적

○ 앱 사이트 개발

○ 연계발 (모바일, 송수신 등)

○ 기타

●

참고자료

○ 연구(논문 등)

작업종연 교육을 위한 참고자료

16/250

정부과일

파일 선택

Drag & Drop으로 파일을 선택 가능합니다.

시스템유형

시스템 유형

●

일반

상세기능정보 선택

☑

상세기능

☑

코로나19시·도발생_현황 조회 서비스

코로나19감염증으로 인한 시·도별 신규확진자, 신규사망자, 격리중단환자수, 격리해제환자수 등에 대한 현황자료

1000

라이선스 표시

*이용허락범위

이용허락범위 제한 없음

☐ 동의합니다.

취소

활용신청

↑

- 35 -

- 마이페이지에서 신청한 내용에 대한 목록 확인 후 상세 기능 확인

* 서비스 요청주소 및 요청변수 확인

상세기능

목록 코로나19시도발생_현황 조회 서비스

조회

코로나19감염증으로 인한 시·도별 신규확진자, 신규사망자, 격리중환자수, 격리해제환자수 등에 대한 현황자료

- 활용승인 절차 개발단계 : 허용 / 운영단계 : 허용
- 신청가능 트랙젝 10000000 / 운영계정은 활용사례 등록시 신청하면 트랙젝 증가 가능
- 요청주소
- 서비스URL <http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19SidoInfStateJson>

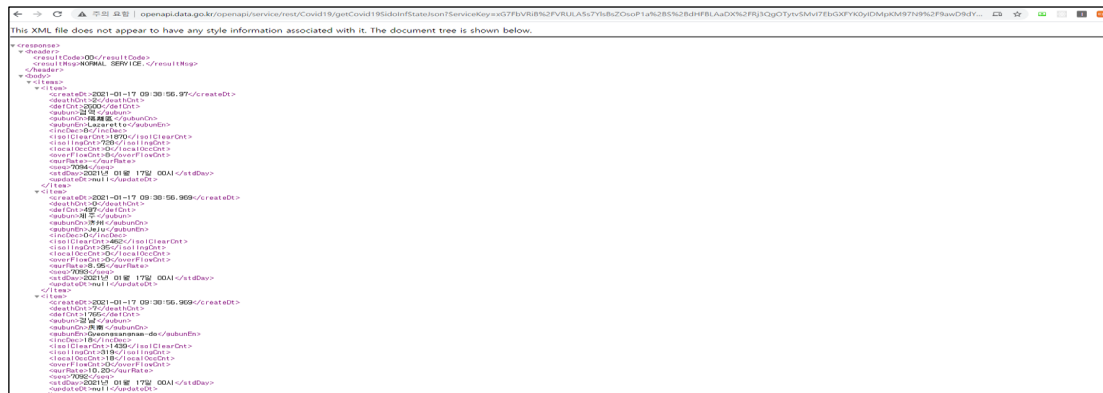
활용신청

요청변수(Request Parameter)

항목명(영문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
서비스키	ServiceKey	4	필수	-	응답 데이터포맷에서 받은 인증키
페이지 번호	pageNo	4	옵션	1	페이지번호
한 페이지 결과 수	numOfRows	4	옵션	10	한 페이지 결과 수
데이터 생성일 시작일	startCreateDt	30	옵션	20200410	검색할 생성일 범위의 시작
데이터 생성일 종료일	endCreateDt	15	옵션	20200410	검색할 생성일 범위의 종료

* 브라우저를 열고 주소창에 서비스요청 주소 입력 시 xml형식의 응답데이터 확인



































<http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19SidoInfStateJson?ServiceKey=xG7FbVRiB%2FVRULA5s7YlSbSZOsoP1a%2BS%2BdHFBLaADX%2FRj3QgQTYtvSMvI7EbGXfYK0yIDMpKM97N9%2F9awD9dYg%3D%3D>



* 응답데이터의 상세 내용은 출력결과에서 확인 가능

출력결과(Response Element)					
항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
결과코드	resultCode	2	필수	00	결과코드
결과메시지	resultMsg	50	필수	OK	결과메시지
한 페이지 결과 수	numOfRows	4	필수	10	한 페이지 결과 수
페이지 번호	pageNo	4	필수	1	페이지번호
전체 결과 수	totalCount	4	필수	3	전체 결과 수
계시글번호(국내 시도별 발생현황 고유값)	SEQ	30	필수	130	계시글번호(국내 시도별 발생현황 고유값)
등록일시분초	CREATE_DT	30		2020-04-10 11:15:59.026	등록일시분초
사망자 수	DEATH_CNT	15		204	사망자 수
시도명(한글)	GUBUN	30		부산	시도명(한글)
시도명(중국어)	GUBUN_CN	30		null	시도명(중국어)
시도명(영어)	gubunEn	30		null	시도명(영어)
전일대비 증감 수	INC_DEC	15		39	전일대비 증감 수
격리 해제 수	ISOL_CLEAR_CNT	15		6973	격리 해제 수
10만명당 발생률	QUR_RATE	30		20.10	10만명당 발생률
기준일시	STD_DAY	30		2020년 3월 13일 00시	기준일시
수정일시분초	UPDATE_DT	30		null	수정일시분초

- 해당 내용을 기반으로 클래스 설계

<<Java Class>>	
 Corona19Data kh.java.data.corona19.model.vo	
SoF	serialVersionUID: long seq: int createDt: String deathCnt: int defCnt: int gubun: String gubunCn: String gubunEn: String incDec: int isoClearCnt: int isollingCnt: int localOccCnt: int overflowCnt: int qurRate: String stdDay: String updateDt: String
	 Corona19Data()  Corona19Data(int,String,int,int,String,String,String,int,int,int,int,int,String,String,String)  getSeq():int  setSeq(int):void  getCreateDt():String  setCreateDt(String):void  getDeathCnt():int  setDeathCnt(int):void  getDefCnt():int  setDefCnt(int):void  getGubun():String  setGubun(String):void  getGubunCn():String  setGubunCn(String):void  getGubunEn():String  setGubunEn(String):void  getIncDec():int  setIncDec(int):void  getIsoClearCnt():int  setIsoClearCnt(int):void  getIsollingCnt():int  setIsollingCnt(int):void  getLocalOccCnt():int  setLocalOccCnt(int):void  getOverflowCnt():int  setOverflowCnt(int):void  getQurRate():String  setQurRate(String):void  getStdDay():String  setStdDay(String):void  getUpdateDt():String  setUpdateDt(String):void  toString():String

- 실습

* 2 개 패키지 생성 후 기능구현 클래스와 데이터저장용 클래스 생성

kh.java.data.corona19 > ApiExplorer.java

kh.java.data.corona19.model.vo > Coron19Data.java

* ApiExplorer.java 는 자바애플리케이션을 실행하는 클래스로 main 메소드, run(),
parseXML()메소드를 포함

* Corona19Data.java 는 응답 데이터를 저장하는 클래스

* Corona19Data.java 는 이전 실습 내용 참조 후 작성

* ApiExplorer.java 구현

kh.java.data.corona19.ApiExplorer	
01	package kh.java.data.corona19;
02	
03	import java.io.BufferedReader;
04	import java.io.IOException;
05	import java.io.InputStream;
06	import java.io.InputStreamReader;
07	import java.io.UnsupportedEncodingException;
08	import java.net.HttpURLConnection;
09	import java.net.MalformedURLException;
10	import java.net.URL;
11	import java.net.URLEncoder;
12	import java.util.ArrayList;
13	import java.util.List;
14	import javax.xml.parsers.DocumentBuilder;
15	import javax.xml.parsers.DocumentBuilderFactory;
16	import javax.xml.parsers.ParserConfigurationException;
17	import org.w3c.dom.Document;
18	import org.w3c.dom.Node;
19	import org.w3c.dom.NodeList;
20	import org.xml.sax.SAXException;
21	import kh.java.data.corona19.model.vo.Corona19Data;
22	

```

23 public class ApiExplorer {
24     public static void main(String[] args){
25         new ApiExplorer().run();
26     }
27
28     private void run() {
29
30         //1. 요청 url 생성
31         StringBuilder urlBuilder = new
32         StringBuilder("http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19Sid
33         olnfStateJson"); /*URL*/
34         try {
35             urlBuilder.append("? " +
36             URLEncoder.encode("ServiceKey","UTF-8") +
37             "=xG7FbVRiB%2FVRULA5s7YIsBsZOsoP1a%2BS%2BdHFBLaADX%2FRj3QgOTyvtSMvl
38             7EbGXfYk0yIDMpKM97N9%2F9awD9dYg%3D%3D");/*Service Key*/
39             urlBuilder.append("& " +
40             URLEncoder.encode("pageNo","UTF-8") + "=" + URLEncoder.encode("1", "UTF-8"));
41             urlBuilder.append("& " +
42             URLEncoder.encode("numOfRows","UTF-8") + "=" + URLEncoder.encode("10", "UTF-
43             8")); /*한 페이지 결과 수*/
44             urlBuilder.append("& " +
45             URLEncoder.encode("startCreateDt","UTF-8") + "=" +
46             URLEncoder.encode("20200410", "UTF-8")); /*검색할 생성일 범위의 시작*/
47             urlBuilder.append("& " +
48             URLEncoder.encode("endCreateDt","UTF-8") + "=" +
49             URLEncoder.encode("20200410", "UTF-8")); /*검색할 생성일 범위의 종료*/
50         } catch (UnsupportedEncodingException e) {
51             e.printStackTrace();
52         }
53         //2. connection객체 생성
54         URL url = null;
55         HttpURLConnection conn = null;
56         try {

```

57	url = new URL(urlBuilder.toString());
58	conn = (HttpURLConnection) url.openConnection();
59	conn.setRequestMethod("GET");
60	conn.setRequestProperty("Content-type",
61	"application/json");
62	System.out.println("Response code: " +
63	conn.getResponseCode());
64	} catch (MalformedURLException e) {
65	e.printStackTrace();
66	} catch (IOException e) {
67	e.printStackTrace();
68	}
69	
70	//3. 요청전송 및 응답 처리
71	BufferedReader br = null ;
72	try {
73	int statusCode = conn.getResponseCode();
74	if (statusCode >= 200 && statusCode <= 300)
75	br = new BufferedReader(new
76	InputStreamReader(conn.getInputStream()));
77	else
78	br = new BufferedReader(new
79	InputStreamReader(conn.getErrorStream()));
80	
81	Document doc = parseXML(conn.getInputStream());
82	
83	//a. item 태그객체 목록으로 가져온다.
84	NodeList descNodes =
85	doc.getElementsByTagName("item");
86	
87	//b. Corona19Data List객체 생성
88	List<Corona19Data> list = new ArrayList<>();
89	
90	//c. 각 item태그의 자식태그에서 정보 가져오기

```

91         for(int i = 0; i < descNodes.getLength(); i++){
92             //item
93             Node item = descNodes.item(i);
94             Corona19Data data = new Corona19Data();
95
96             //item 자식태그에 순차적으로 접근
97             for(Node node = item.getFirstChild(); node != null; node
98 = node.getNextSibling()){ //첫번째 자식을 시작으로 마지막까지 다음 형제를 실행
99                 //System.out.println(node.getNodeName() + " : "
100 + node.getTextContent());
101                 switch(node.getNodeName()) {
102                     case "seq":
103 data.setSeq(Integer.parseInt(node.getTextContent())); break;
104                     case "createDt":
105 data.setCreateDt(node.getTextContent()); break;
106                     case "deatCnt":
107 data.setDeathCnt(Integer.parseInt(node.getTextContent())); break;
108                     case "defCnt":
109 data.setDefCnt(Integer.parseInt(node.getTextContent())); break;
110                     case "gubun":
111 data.setGubun(node.getTextContent()); break;
112                     case "gubunCn":
113 data.setGubunCn(node.getTextContent()); break;
114                     case "gubunEn":
115 data.setGubunEn(node.getTextContent()); break;
116                     case "incDec":
117 data.setIncDec(Integer.parseInt(node.getTextContent())); break;
118                     case "isolClearCnt":
119 data.setIsolClearCnt(Integer.parseInt(node.getTextContent())); break;
120                     case "isolIngCnt":
121 data.setIsolIngCnt(Integer.parseInt(node.getTextContent())); break;
122                     case "localOccCnt":
123 data.setLocalOccCnt(Integer.parseInt(node.getTextContent())); break;
124                     case "overFlowCnt":

```

```

125 data.setOverFlowCnt(Integer.parseInt(node.getTextContent())); break;
126             case "qurRate":
127 data.setQurRate(node.getTextContent()); break;
128             case "stdDay":
129 data.setStdDay(node.getTextContent()); break;
130             case "updateDt":
131 data.setUpdateDt(node.getTextContent()); break;
132         }
133     }
134     //d. List객체에 추가
135     list.add(data);
136
137 }
138 //e.최종확인
139 for(Corona19Data d : list)
140     System.out.println(d);
141
142 } catch (IOException e) {
143     e.printStackTrace();
144 } finally {
145     try {
146         br.close();
147     } catch (Exception e) {
148         e.printStackTrace();
149     }
150 }
151 //4. 자원반환
152 conn.disconnect();
153 }
154
155 private Document parseXML(InputStream stream) {
156
157     DocumentBuilderFactory objDocumentBuilderFactory =
158     DocumentBuilderFactory.newInstance();

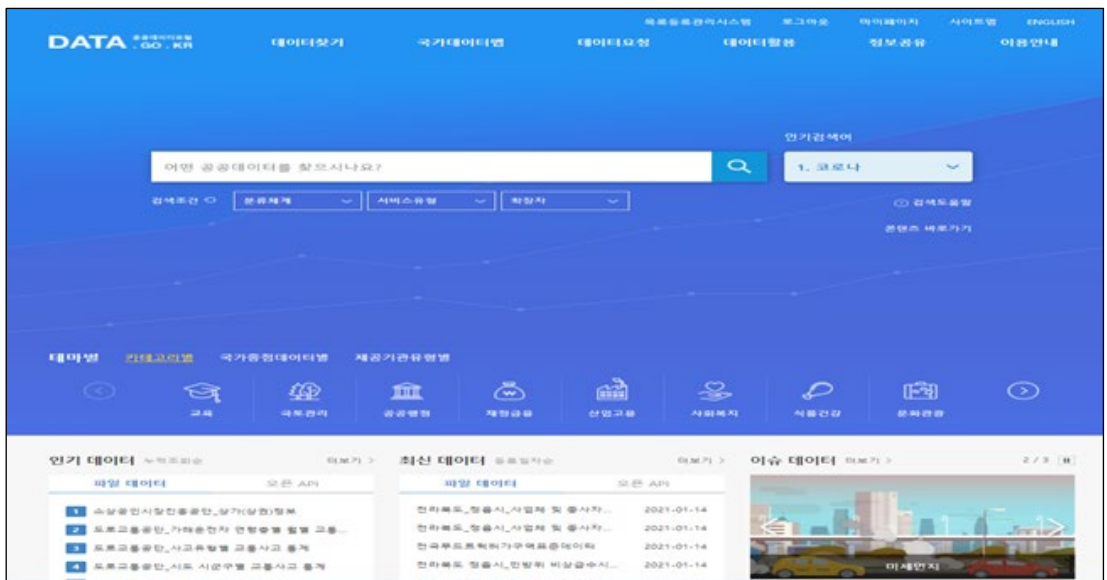
```


Chapter 6. DB 에 데이터 저장하기

· 이전 내용을 참조하여 data.go.kr사이트 접속



· 회원가입(필요 시) 및 로그인 후 조회창에서 “전기충전소” 검색



· 전기충전소 관련 데이터 목록 출력

데이터목록

전기충전소

※ 공·공급료 안에 검색어를 입력하면 검색어를 반드시 포함하는 결과만을 표시합니다. (ex. "미세먼지 현황정보")

연관

제공기관별 검색

상세검색

"전기충전소"에 대해 총 815건이 검색되었습니다.

조건검색

분류체계

☐ 공공행정
☐ 과학기술
☐ 교육
☐ 교통물류
☐ 국토관리
☐ 농축수산업

서비스유형

☐ LINK
☐ LOD
☐ REST
☐ RSS/ATOM
☐ SOAP
☐ 다운로드

제공기관유형

☐ 공공기관
☐ 교육기관
☐ 교육행정기관
☐ 국가행정기관
☐ 위원회
☐ 입법기관

태그

☐ 고구려
☐ 고분벽화
☐ 고당시
☐ 고전
☐ 관공
☐ 악동리

확장자

☐ CSV
☐ DOC
☐ DOCX
☐ HWP
☐ JPG
☐ JSON

국가공급데이터 분류 조건 추가하기 +

조건합기 ^

전체(815개)

파일데이터(394개)

오픈 API(52개)

요즘데이터셋122개(369개)

정확도순

10개씩

파일데이터 (394건)

상단목록

지식탐색기

미리보기

CSV

경기도 구리시_전기차충전소

구리시 전기차충전소 정보

수정일 2020-11-09 조회수 180 다운로드 242

다운로드

확보기관

공공기관

미리보기

CSV

한국환경공단_전기차 충전기 현황_환경부 공공급속충전기

2015년~2020년 지역별 전기차 충전기 현황(공공급속충전기)

수정일 2020-07-02 조회수 441 다운로드 319

다운로드

상단목록

공공기관

미리보기

CSV

한국전력공사_지역별 전기차 충전소 현황정보

한전 지역별 전기차 충전소 현황 정보입니다.

수정일 2020-10-14 조회수 1190 다운로드 624

다운로드

상단목록

지식탐색기

미리보기

CSV

대구광역시_전기차 충전소(일반, 충전기별) 사용현황

이전수업

계시판

· 파일데이터 상세보기

- 출력 내용 중 파일데이터(csv형식) 선택 -> "경기도 구리시_전기차충전소" 클릭
- 선택 내용에 대한 상세 정보를 확인할 수 있음

파일데이터 상세

f

URL 복사

CSV

경기도 구리시_전기차충전소

다운로드

구리시 전기차충전소 정보

0

0

관심

오류신고 및

담당자 문의

파일데이터 정보

메타데이터 다운로드

파일데이터명	경기도 구리시_전기차충전소_20200731		
분류체계	산업·통상·중소기업 - 에너지및자원개발	제공기관	경기도 구리시
관리부서명	정보통신과	관리부서 전화번호	031-550-2087
보유근거		수집방법	
업데이트 주기	연간	자기 등록 예정일	2021-07-31
매체유형	텍스트	전체 행	44
확장자	CSV	다운로드(바로그가)	244
데이터 한계		키워드	전기차,전기차충전,구리시
등록	2020-11-09	수정	2020-11-09
제공형태	공공데이터포털에서 다운로드(원문파일등록)		
설명	구리시 전기차충전소 정보		
기타 유의사항			
비용부과유무	무료	비용부과기준 및 단위	건
이용허락범위	이용허락범위 제한 없음		

- 하단에 미리보기를 확인하여 데이터 구조와 내용을 확인한 후
데이터 저장 방식 및 Class 설계 가능

미리보기 ※ 파일 데이터의 일부 내용을 제공하고 있으며, 전체 내용이 필요한 경우 해당 파일을 다운로드 받으시기 바랍니다.

중전소명	중전소위치상세	설치시도명	휴점일	이용가능시작시각	이용가능종료시각	완속중전가능여부	급속중전가능여부	급속중전타입구분	완속중전기대수
경기도 구리시청	경기도 구리시 아차산로 439 본관 홍연실 옆	경기도	24시간 이용가능	00:00	00:00	Y	Y	DC자대모+AC3상 +DC롬보	3
구리휴게소(외곽 방향)	경기도 구리시 서 울회곡순환고속도 로 32 (사노동)	경기도	24시간 이용가능	00:00	00:00	Y	Y	DC자대모+AC3상 +DC롬보	3
경기도 구리시 청 정복지센터	경기도 구리시 청 육관로 74	경기도	24시간 이용가능	00:00	00:00	Y	Y	DC롬보	1
경기도 구리시 청 정복지센터	경기도 구리시 청 육관로 74	경기도	24시간 이용가능	00:00	00:00	Y	Y	DC롬보	1
구리한강시민공원 (주차장)	경기도 구리시 코 스모스길14번길 2 49	경기도	24시간 이용가능	00:00	00:00	Y	Y	DC롬보	1
구리한강시민공원 (주차장)	경기도 구리시 코 스모스길14번길 2 49	경기도	24시간 이용가능	00:00	00:00	Y	Y	DC롬보	1
구리경찰서	경기도 구리시 아 차산로 359	경기도	24시간 이용가능	00:00	00:00	Y	Y	DC자대모+AC3상 +DC롬보	3

- csv 파일을 다운로드하게 되면 다음 이미지와 같은 엑셀로 데이터 내용 확인 가능

The screenshot shows an Excel spreadsheet with the following columns (A-T):

- A: 중전소명 (Station Name)
- B: 중전소위치상세 (Detailed Station Location)
- C: 설치시도명 (Installation Province Name)
- D: 휴점일 (Status)
- E: 이용가능시작시각 (Available Start Time)
- F: 이용가능종료시각 (Available End Time)
- G: 완속중전가능여부 (Slow Charging Available)
- H: 급속중전가능여부 (Fast Charging Available)
- I: 급속중전타입구분 (Fast Charging Type)
- J: 완속중전기대수 (Slow Charging Capacity)
- K: 급속중전기대수 (Fast Charging Capacity)
- L: 완속중전기대수 (Slow Charging Capacity)
- M: 완속중전기대수 (Slow Charging Capacity)
- N: 완속중전기대수 (Slow Charging Capacity)
- O: 완속중전기대수 (Slow Charging Capacity)
- P: 완속중전기대수 (Slow Charging Capacity)
- Q: 완속중전기대수 (Slow Charging Capacity)
- R: 완속중전기대수 (Slow Charging Capacity)
- S: 완속중전기대수 (Slow Charging Capacity)
- T: 완속중전기대수 (Slow Charging Capacity)

The data rows contain station names, locations, and various numerical and categorical values representing station specifications and status.

· 공공데이터 파일을 이용한 테이블 생성

- csv 방식 공공데이터 선정

* 조회 창 : “건강보험 심사평가원_전국 병의원 및 약국현황”

* 조회한 내용에 대한 데이터 목록이 출력되며, 그 중 파일데이터 항목의
“건강보험 심사평가원_전국 병의원 및 약국현황” 클릭하여 상세항목으로 이동

DATA 공공데이터포털
GO . KR

데이터찾기
 국가데이터맵
 데이터요청
 데이터활용
 정보공유
 이용약관

목록등록관리시스템
 로그인아웃
 마이페이지
 사이트맵
 ENGLISH

파일데이터 상세

[URL 복사](#)

ZIP
건강보험심사평가원_전국 병의원 및 약국 현황

요양기관 기본정보, 세부정보, 시설정보, 의료장비정보 등 요양기관별 상세 현황 정보 / 건강보험심사평가원 병원정보서비스+약국정보서비스+의료기관별상세정보서비스 오픈API 데이터셋 / 분기별 업데이트

0
 0
 [다운로드](#)
[요류선고 및
담당자 문의](#)

파일데이터 정보
[메타데이터 다운로드](#)

파일데이터명	건강보험심사평가원_전국 병의원 및 약국 현황_20200930		
분류체계	보건 - 건강보험	제공기관	건강보험심사평가원
관리부서명	빅데이터실 빅데이터전략부	관리부서 전화번호	033-799-1047
보유근거	국민건강보험법	수집방법	
업데이트 주기	연간	자기 등록 예정일	2021-10-16
매체유형	엑셀	전체 행	1
확장자	ZIP	다운로드(바로그가기)	1745
데이터 한계		키워드	요양기관, 병원, 약국
등록	2020-10-16	수정	2020-10-29
제공형태	공공데이터포털에서 다운로드(원본파일등록)		
설명	요양기관 기본정보, 세부정보, 시설정보, 의료장비정보 등 요양기관별 상세 현황 정보 / 건강보험심사평가원 병원정보서비스+약국정보서비스+의료기관별상세정보서비스 오픈API 데이터셋 / 분기별 업데이트		
기타 유의사항	해당 자료는 오픈API 서비스의 데이터셋을 추출한 자료로 원본 자료의 데이터와 집계상 차이가 있으나 수치를 비교하지 말 것		
비용부과유무	무료	비용부과기준 및 단위	건
이용허락범위	공공저작물_출자표시		

주기성 과거 데이터 (1건)
 [여보기 >](#)

[목록](#)

- 우측 상단 다운로드 클릭

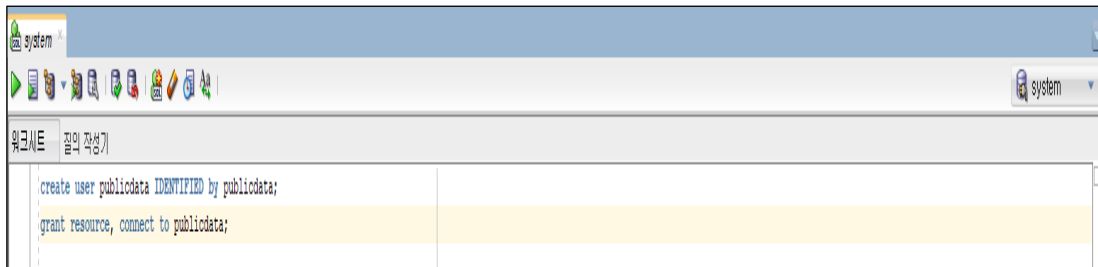
* 해당 파일은 압축파일로 제공이 되며, 압축을 풀면 다음과 같은 내용 확인 가능



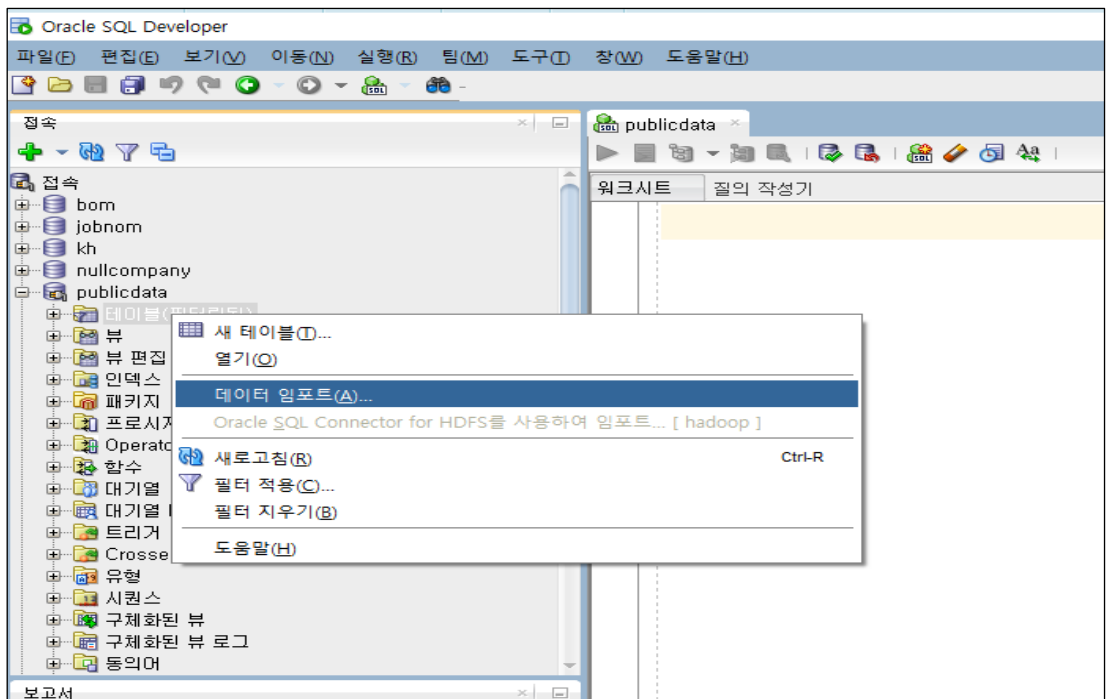
- sql developer를 이용한 테이블 구성

* system계정으로 사용자 계정 생성 및 권한 부여

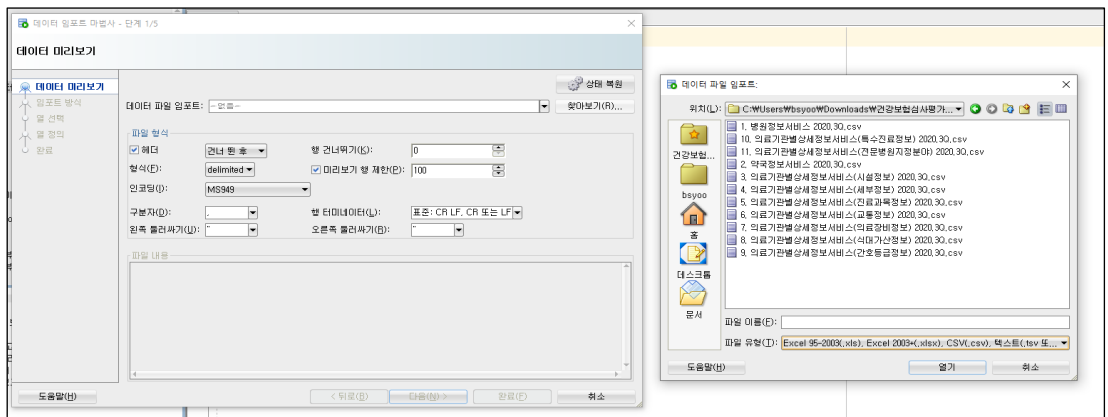
계정명 : publicdata, 권한 : resource, connect



* publicdata 계정으로 접속 후 테이블 폴더에서 데이터 임포트 선택



* 공공데이터에서 받아온 csv 중 병원정보서비스 항목을 테이블로 생성



* "병원정보서비스" 파일을 선택 -> DB의 테이블 형식 확인 -> 다음

데이터 임포트 마법사 - 단계 1/5

데이터 미리보기

데이터 파일 임포트: [강북삼성병원_건국 병의원 및 약국 현황 2020.9W1, 병원정보서비스 2020.3Q.csv]

파일 형식

☒ 헤더 건너뛰기(K): 0 행 건너뛰기(L): 100

형식(E): CSV ☒ 미리보기 행 제한(P): 100

인코딩(I): MS949 구분자(D): , 행 터미네이터(L): 표준: CR LF, CR 또는 LF

왼쪽 둘러싸기(W): " 오른쪽 둘러싸기(B): "

파일 내용

암호화YK...	요양기관명	중별코드	중별코드명	시도코드	시도코드명	시군구코드	시군구코드...	읍면동	우
JDQ4MTY...	가톨릭대학...	1	상급종합	220000	인천	220003	인천부평구		21
JDQ4MTg...	강북삼성병원	1	상급종합	110000	서울	110016	종로구		03
JDQ4MTg...	건국대학교...	1	상급종합	110000	서울	110023	광진구		05
JDQ4MTY...	경북대학교...	1	상급종합	230000	대구	230006	대구중구		41
JDQ4MTg...	경상대학교...	1	상급종합	380000	경남	380500	진주시		52
JDQ4MTg...	경희대학교...	1	상급종합	110000	서울	110007	동대문구		02
JDQ4MTY...	계명대학교...	1	상급종합	230000	대구	230007	대구달서구	신당동	42
JDQ4MTg...	고려대학교...	1	상급종합	110000	서울	110005	구로구		08
JDQ4MTY...	고려대학교...	1	상급종합	310000	경기	311101	안산단원구		15

도움말(H) < 뒤로(B) 다음(N) > 완료(F) 취소

* 테이블 이름 : hospital -> 다음

데이터 임포트 마법사 - 단계 2/4

임포트 방식

데이터 임포트 방식을 선택하십시오. 외부 테이블 방식의 경우 파일의 데이터를 읽을 수 있도록 외부 테이블이 생성됩니다. 스테이징 외부 테이블 방식의 경우 대상 테이블을 임포트할 수 있도록 외부 테이블이 스테이징 테이블로 생성됩니다. 기타 방식의 경우 새 테이블이 생성되고 데이터가 임포트됩니다.

임포트 방식: **상업**

테이블 이름:

☐ SQL 워크시트로 생성 스크립트 전송

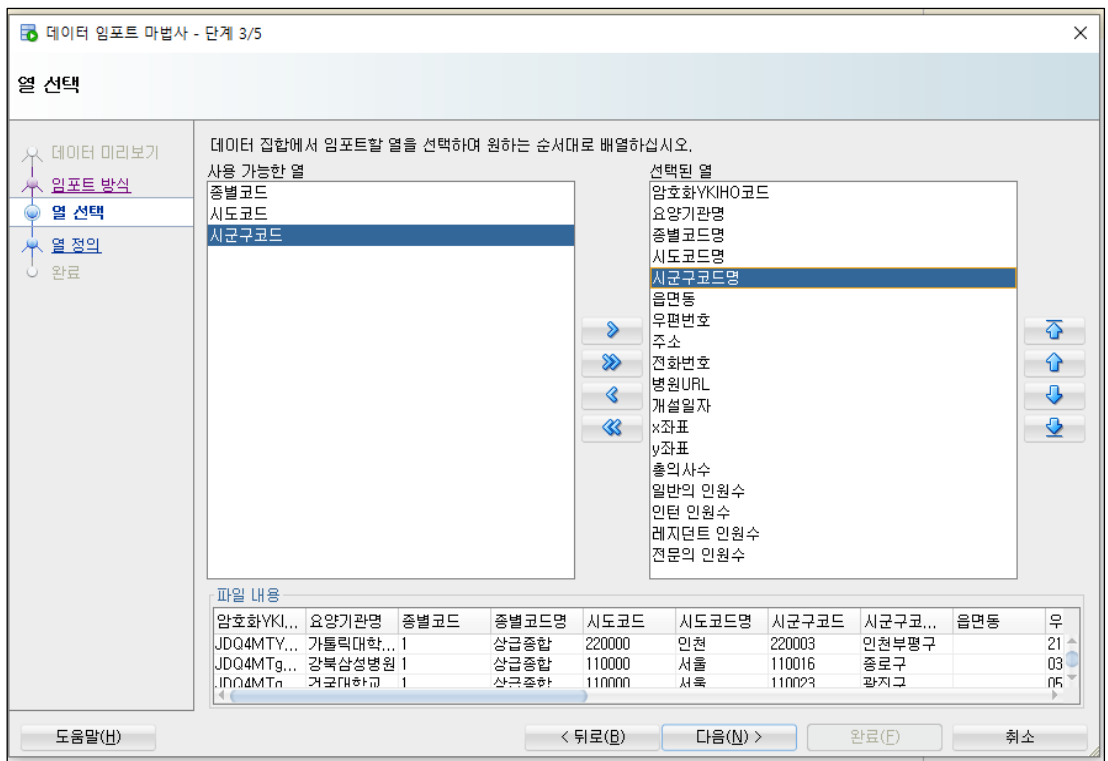
☐ 임포트 행 제한(I): 100

파일 내용

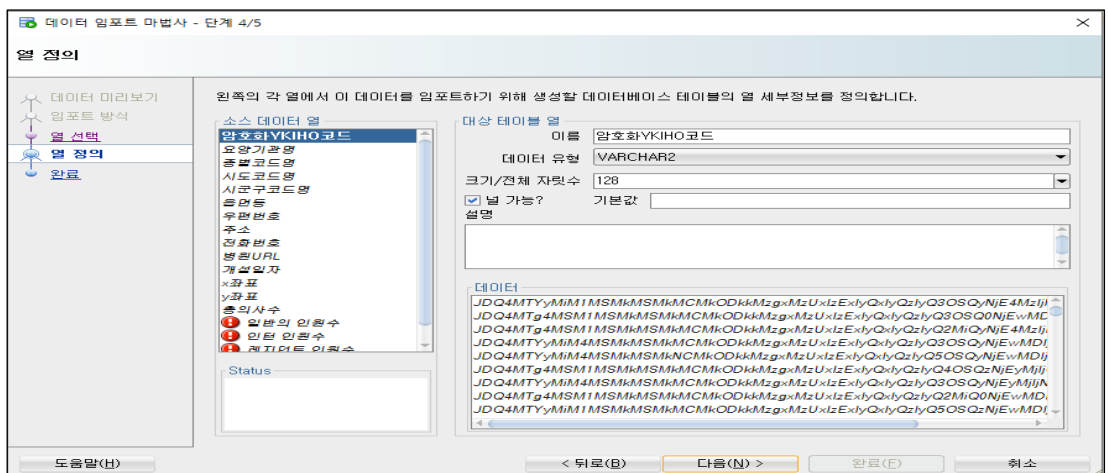
암호화YK...	요양기관명	중별코드	중별코드명	시도코드	시도코드명	시군구코드	시군구코드...	읍면동	우
JDQ4MTY...	가톨릭대학...	1	상급종합	220000	인천	220003	인천부평구		21
JDQ4MTg...	강북삼성병원	1	상급종합	110000	서울	110016	종로구		03
JDQ4MTg...	건국대학교...	1	상급종합	110000	서울	110023	광진구		05
JDQ4MTY...	경북대학교...	1	상급종합	230000	대구	230006	대구중구		41
JDQ4MTg...	경상대학교...	1	상급종합	380000	경남	380500	진주시		52
JDQ4MTY...	경희대학교...	1	상급종합	110000	서울	110007	동대문구		02
JDQ4MTg...	계명대학교...	1	상급종합	230000	대구	230007	대구달서구	신당동	42
JDQ4MTg...	고려대학교...	1	상급종합	110000	서울	110005	구로구		08
JDQ4MTY...	고려대학교...	1	상급종합	310000	경기	311101	안산단원구		15
JDQ4MTA...	고신대학교...	1	상급종합	210000	부산	210006	부산서구		49
JDQ4MTY...	단국대학교...	1	상급종합	340000	충남	340202	천안남구		31
JDQ4MTY...	대구가톨릭...	1	상급종합	230000	대구	230001	대구남구		42
JDQ4MTA...	독마대학교...	1	상급종합	210000	부산	210006	부산서구	독대신동3가	49

도움말(H) < 뒤로(B) 다음(N) > 완료(F) **취소**

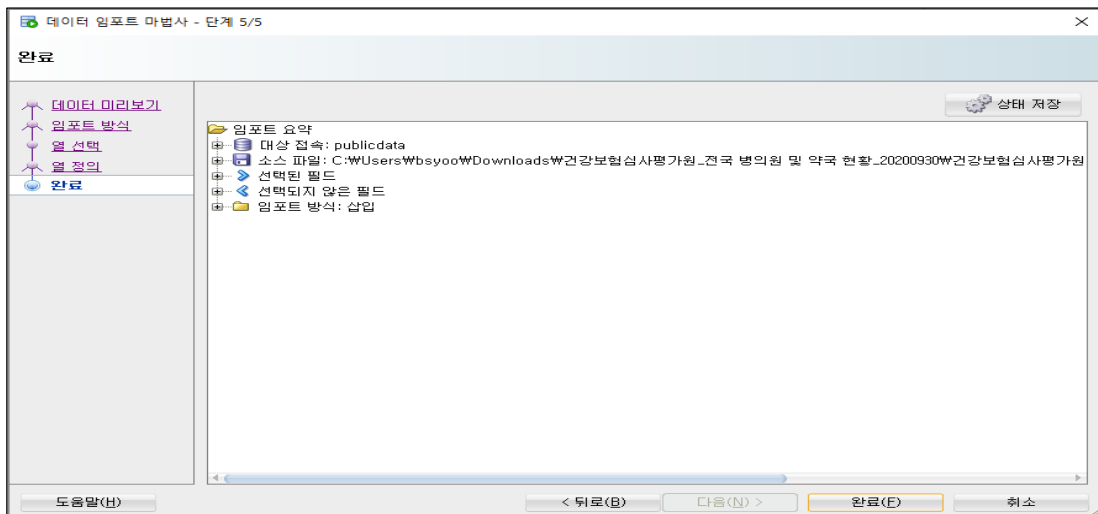
* csv에서 필요한 항목만 선택하여 테이블을 생성할 수 있으며, 필요없다고 생각되는 코드는 제거 가능



* 항목별 데이터의 자료형, 컬럼명을 설정할 수 있으며, 영어로 변환하여 저장



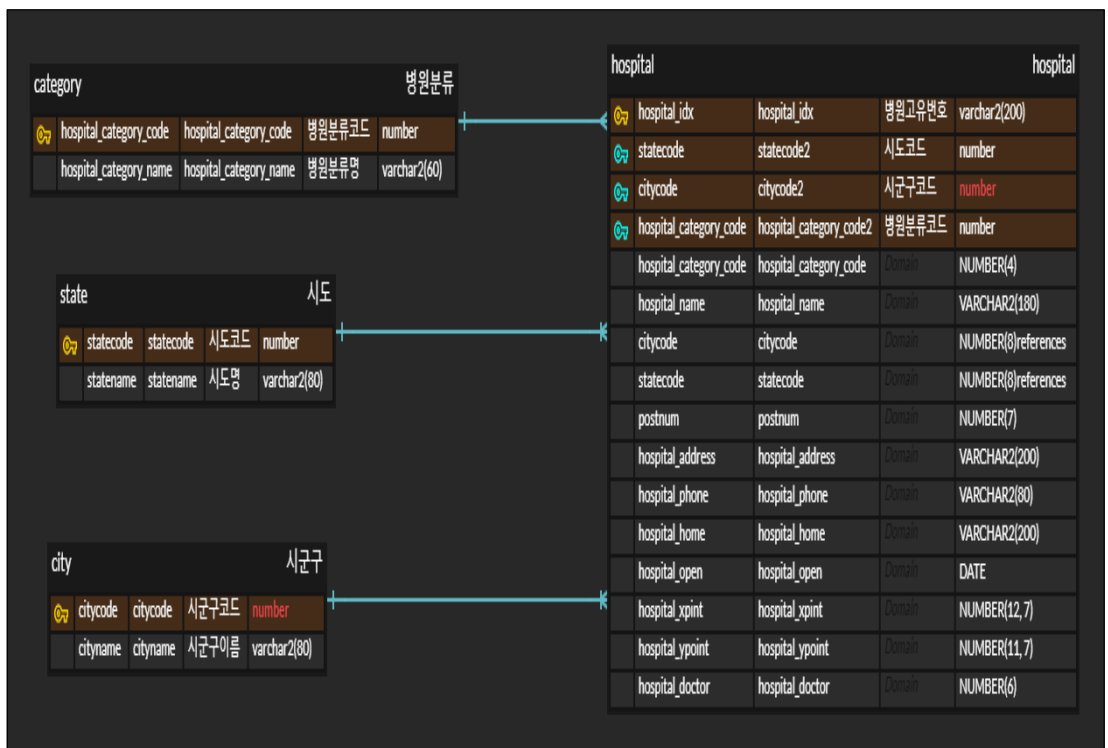
* 최종 완료



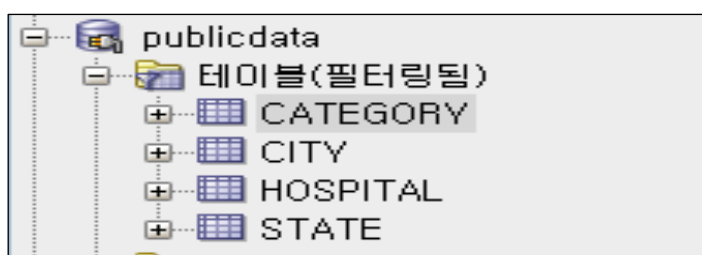
- 공공데이터 사이트에서 받아온 csv형태의 자료를 데이터베이스에서 활용 가능
- 테이블로 변환 뒤 웹 애플리케이션에서 사용할 테이블과 제약조건을 설정하여 사용할 수 있음

- 공공데이터 분석, 모델링

- * csv 데이터를 참조해 간단하게 정규화를 진행해보면 병원정보데이터에서 아래와 같이 모델링하여 테이블을 나누어 저장할 수 있음



* 테이블 구성



* 임시 테이블을 이용한

CATEGORY 테이블에 들어갈 값의 중복 값 제거

```
create table temp(  
    code number,  
    name varchar2(100)  
);  
insert into temp  
select distinct HOSPITAL_CATEGORY_CODE, HOSPITAL_CATEGORY_NAME from category;  
delete from category;  
insert into category  
select * from temp;  
select * from category;  
delete from temp;  
insert into temp  
select distinct stateCODE, stateNAME from state;  
select * from temp;  
delete from state;  
insert into state  
select * from temp;  
select * from state;  
delete from temp;  
insert into temp  
select distinct citycode,cityname from city;  
delete from city;  
insert into city  
select * from temp;
```

* CATEGORY 테이블과 hospital 테이블 연결을 위한 제약 조건 설정

```
alter table hospital add constraint pk_idx primary key(hospital_idx);  
alter table category add constraint pk_category primary key(hospital_category_code);  
alter table state add constraint pk_state primary key(statecode);  
alter table city add constraint pk_city primary key(citycode);  
alter table hospital add constraint fk_category foreign key(hospital_category_code) references category(hospital_category_code);  
alter table hospital add constraint fk_state foreign key(statecode) references state(statecode);  
alter table hospital add constraint fk_city foreign key(citycode) references city(citycode);
```

Part 3. 데이터 배포를 위한 REST 설계

Chapter 1. RestController 구축(비동기 통신)

비동기 통신

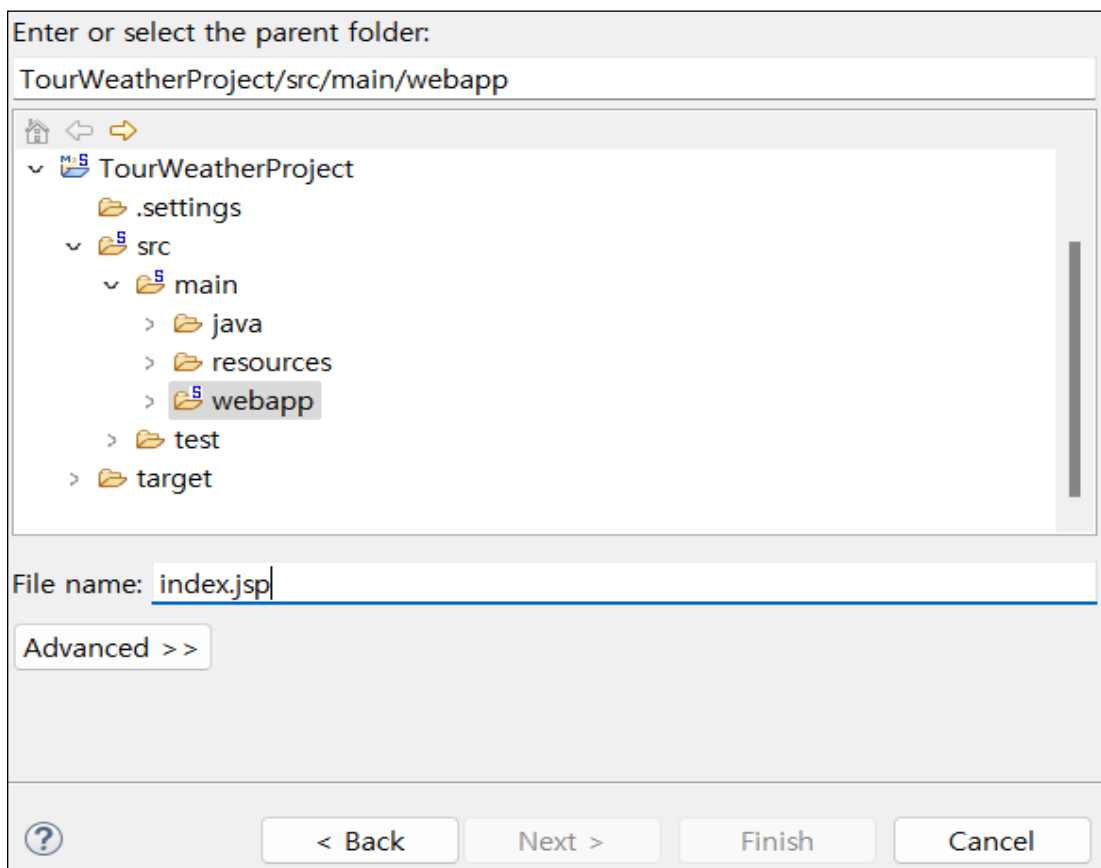
- 서버와 브라우저가 비동기방식으로 데이터를 교환할 수 있는 통신 방식
- 웹페이지를 리로드하지 않고, 데이터를 불러오는 방식
- 장점
 - * 필요한 부분만 리로드해서 사용하므로 웹페이지 속도 향상
 - * 서버에서의 처리가 완료될때까지 기다리지 않고 처리

- 33 페이지에서 1 차완료된 기상청 관광지 공공데이터 실습을 이어서 진행할 예정
 - 응답데이터에 대한 처리를 자바애플리케이션에서 실행되는 실습 완료
 - 지금부터의 실습은
웹 페이지를 통한 요청에 대해 응답되어 온 데이터를 웹 브라우저에 출력

- STS(Spring Tool Suite) 실행

* index.jsp파일 생성

* src -> main -> webapp 우클릭 -> New -> JSP file -> Name : index.jsp



== index.jsp기본 설정 ==

상단 <%@ taglib prefix="c" ~~~~ %> => 내용 삭제

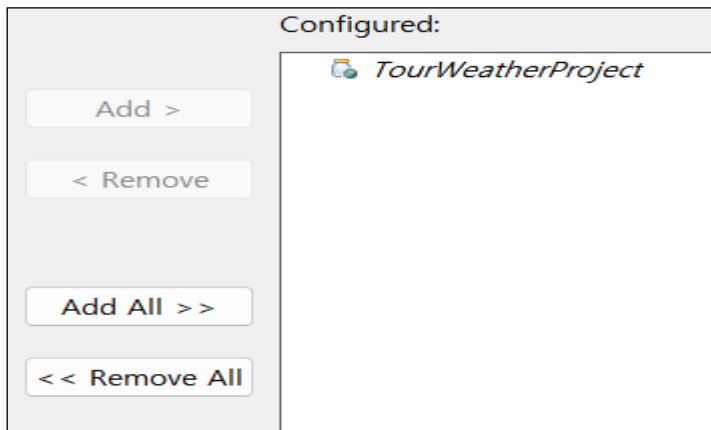
<body>TourWeather</body> == TourWeather 내용 추가

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="UTF-8">
7   <title>Insert title here</title>
8 </head>
9 <body>
10   TourWeather
11 </body>
12 </html>
```

== Tomcat 설정 ==

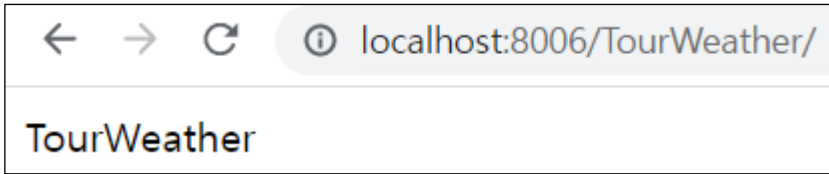
Tomcat 우클릭 -> Add & Remove -> TourWeatherProject만 Add,

나머지 항목이 있다면 remove -> Tomcat start



* 브라우저에서 확인

브라우저 URL에서 localhost:포트번호/TourWeather로 확인



- 비동기식으로 Web Application에 적용하고자 할 경우, 절차

1. jsp에서 현재 Web Application 서버로 ajax 요청

2. Controller에서 요청 받기

3. HttpURLConnection객체 활용, OpenAPI서버에 요청 후 응답데이터 받기

4. 3번 과정에서의 응답데이터를 Client에게 다시 응답

5. Client측에서는 응답받은 데이터를 기반으로 파싱 작업 후 웹페이지에
결과 화면 출력

* index.jsp파일 수정 및 추가

Index.jsp	
01	<%@ page language="java" contentType="text/html; charset=UTF-8"
02	pageEncoding="UTF-8"%>
03	<!DOCTYPE html>
04	<html>
05	<head>
06	<meta charset="UTF-8">
07	<title>Insert title here</title>
08	<script src
09	= "https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"> </script>
10	</head>
11	<body>
12	<h2>관광지 날씨</h2>
13	
14	관광지명 :
15	<select id = "location">
16	<option value=52>서울홍대</option>
17	<option value=55>서울명동</option>
18	<option value=58>서울한강</option>
19	</select> <!-- 내용이 많아 3개만 option 지정 -->
20	
21	<button id = "btn1">관광지 날씨</button>
22	
23	
24	<table id = "result1" border = "1">
25	<thead>
26	<tr>
27	<th>관광지역</th>
28	<th>관광지명</th>
29	<th>기온</th>
30	<th>습도</th>
31	<th>강수확률</th>
32	</tr>

```

33         </thead>
34     </tbody>
35 </tbody>
36 </table>
37
38 <script>
39     $(function(){
40         $("#btn1").click(function(){
41             $.ajax({
42                 url:"weather.do",
43                 data:{location:$("#location").val()},
44                 success:function(data){
45                     // console.log(itemAr.log(data); 확인 후 주석
46                     // console.log(data.response.body.items); 확인 후 주석
47                     const itemArr =
48 data.response.body.items.item;
49
50                     let value = "";
51                     for(let i in itemArr){
52                         let item = itemArr[i];
53
54                         value += "<tr>"
55                             +
56 "<td>" + item.spotAreaName + "</td>"
57                             +
58 "<td>" + item.spotName + "</td>"
59                             +
60 "<td>" + item.th3 + "</td>"
61                             +
62 "<td>" + item.rhm + "</td>"
63                             +
64 "<td>" + item.pop + "</td>"
65                             + "</tr>";
66                     }

```

67	
68	<code>\$("#result1 tbody").html(value);</code>
69	
70	<code>}</code>
71	<code>}}</code>
72	<code>}}</code>
73	<code>}}</code>
74	<code></script></code>
75	
76	<code></body></code>
77	<code></html></code>

Chapter 2. 요청 데이터 형식에 맞는 SQL 작성(Mybatis)

* controller 구성 실습

* TourWeather 우클릭 -> New -> Class -> Package : com.kh.TourWeather.controller,
Name : APITourController -> Finish

New Java Class

Java Class
Create a new Java class.

Source folder: TourWeatherProject/src/main/java Browse...

Package: com.kh.TourWeather.controller Browse...

☐ Enclosing type: Browse...

Name: **APITourWeather**

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...
Remove

Which method stubs would you like to create?
☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Finish Cancel

* APITourController.java 구성

APITourController.java	
01	package com.kh.TourWeather.controller;
02	
03	
04	import java.io.BufferedReader;
05	import java.io.IOException;
06	import java.io.InputStreamReader;
07	import java.net.HttpURLConnection;
08	import java.net.URL;
09	import java.net.URLEncoder;
10	import java.time.LocalDateTime;
11	import java.time.format.DateTimeFormatter;
12	
13	import org.springframework.stereotype.Controller;
14	import org.springframework.web.bind.annotation.RequestMapping;
15	import org.springframework.web.bind.annotation.ResponseBody;
16	
17	
18	
19	@Controller
20	public class APITourController {
21	
22	private static final String serviceKey =
23	"ey51rbqUG4nLxm%2FIMXybyk%2FsuSNe95v4algn4Ydn0UhcJl91gToKXy7htiikWXyGCVxm6xy
24	ZpakkO4hk%2FLs3w%3D%3D";
25	
26	
27	@ResponseBody
28	@RequestMapping(value="weather.do", produces = "application/json; charset=utf-8")
29	public String TourWeather(String location) throws IOException {
30	
31	String url =

```

32 "http://apis.data.go.kr/1360000/TourStnInfoService/getTourStnVilageFcst";
33     url += "?serviceKey=" + serviceKey;
34     url += "&numOfRows=10";
35     url += "&pageNo=1";
36     url += "&dataType=json";
37     LocalDateTime datetime = LocalDateTime.now();
38     DateTimeFormatter formatter =
39     DateTimeFormatter.ofPattern("yyyyMMddHH");
40     url += "&CURRENT_DATE=" + datetime.format(formatter);
41     url += "&HOUR=24";
42     url += "&COURSE_ID=" + location;
43     // System.out.println(url); 확인 후 주석처리
44
45     // System.out.println(url);
46
47
48     URL requestUrl = new URL(url);
49     HttpURLConnection urlConnection =
50     (HttpURLConnection)requestUrl.openConnection();
51     urlConnection.setRequestMethod("GET");
52
53     BufferedReader br = new BufferedReader(new
54     InputStreamReader(urlConnection.getInputStream()));
55
56     String responseText = "";
57     String line;
58     while((line = br.readLine()) != null) {
59         responseText += line;
60     }
61
62
63     br.close();
64     urlConnection.disconnect();
65

```

66	//System.out.println(responseText);
67	return responseText;
68	
69	}
70	
71	}

- Tomcat 재시작 후

* 브라우저에서 localhost:포트번호/TourWeather 접속

[←](#) [→](#) [↻](#) [🔍](#) localhost:8006/TourWeather/

관광지 날씨

관광지명 : 서울홍대 ▾ 관광지 날씨

관광지역 관광지명 기온 습도 강수확률

* 해당 내용 출력 확인 후 관광지 날씨 버튼을 클릭하여 각 각의 결과 확인

관광지 날씨					
관광지명 : 서울홍대 ▾ 관광지 날씨					
관광지역	관광지명	기온	습도	강수확률	
서울	(서울)KT&G 상상마당	0	50	20	
서울	(서울)공연장	0	50	20	
서울	(서울)홍대벽화거리&포시즌하우스	0	50	20	
서울	(서울)홍대벽화거리&포시즌하우스	0	50	20	
서울	(서울)공연장	0	50	20	
서울	(서울)KT&G 상상마당	0	50	20	
서울	(서울)트릭아이미술관	0	50	20	
서울	(서울)트릭아이미술관	0	50	20	
서울	(서울)KT&G 상상마당	-2	55	20	
서울	(서울)KT&G 상상마당	-2	55	20	

- 출력되는 내용보다 적은 양의 데이터를 출력(기본 10개 항목)하고 싶다면

요청변수(Request Parameter)		
항목명	샘플데이터	설명
ServiceKey	-	공공데이터포털에서 받은 인증키
pageNo	1	페이지번호
numOfRows	10	한 페이지 결과 수
dataType	XML	요청자료형식(XML/JSON)
CURRENT_DATE	2019122010	2016-12-01 01시부터 조회
HOUR	24	CURRENT_DATE부터 24시간 후까지의 자료 호출
COURSE_ID	1	관광 코스ID

요청변수 항목 중 numOfRows 내용을 ApiController.java에서 코드 추가 필요

* P68, 34행 내용을 -> `url += "&numOfRows=5";` 로 변경 후 저장

* Tomcat restart / 브라우저에서 확인 -> 5개 항목만 출력

관광지 날씨				
관광지명 : 서울홍대 ▾ 관광지 날씨				
관광지역	관광지명	기온	습도	강수확률
서울	(서울)KT&G 상상마당	0	50	20
서울	(서울)공연장	0	50	20
서울	(서울)홍대벽화거리&포시즌하우스	0	50	20
서울	(서울)홍대벽화거리&포시즌하우스	0	50	20
서울	(서울)공연장	0	50	20

Chapter 3. JSON 형태로 응답하기

- 출력되는 내용에 새로운 항목 추가

* 실습 상황이므로 풍속항목 1개만 추가 예정

* 응답 메시지 상세 확인

WS	풍속
----	----

* index.jsp내용 추가(하단 이미지 행번호 참조 후 내용 추가)

```
25<tr>
26    <th>관광지역</th>
27    <th>관광지명</th>
28    <th>기온</th>
29    <th>풍속</th>
30    <th>습도</th>
31    <th>강수확률</th>
32</tr>
```

```
54         value += "<tr>"
55             + "<td>" + item.spotAreaName + "</td>"
56             + "<td>" + item.spotName + "</td>"
57             + "<td>" + item.th3 + "</td>"
58             + "<td>" + item.ws + "</td>"
59             + "<td>" + item.rhm + "</td>"
60             + "<td>" + item.pop + "</td>"
61             + "</tr>";
```

* 해당 내용 추가 후 저장 → Tomcat restart → 브라우저에서 확인

관광지 날씨

관광지명 :

관광지역	관광지명	기온	풍속	습도	강수확률
서울	(서울)KT&G 상상마당	5	2	30	20
서울	(서울)공연장	5	2	30	20
서울	(서울)홍대벽화거리&포시즌하우스	5	2	30	20
서울	(서울)홍대벽화거리&포시즌하우스	5	2	30	20
서울	(서울)공연장	5	2	30	20