

공공데이터 활용하기

공공데이터 활용하기

1. 공공데이터 분석 및 허가 받기

Chapter 1	공공데이터 개요 및 분석	3
Chapter 2	공공데이터 활용 신청	13
Chapter 3	Open API 인증키 확인	16

2. JSON, XML 기반 데이터 수집 실습

Chapter 1	JSON, XML 이해하기	18
Chapter 2	XML 데이터 파싱	20
Chapter 3	JSON 데이터 파싱	38
Chapter 4	REST API를 이용한 데이터 수집	50

공공데이터 활용하기

3. 공공데이터 기반 DB 분석 및 구축 실습

Chapter 1	공공데이터 DB 전달을 위한 DTO(VO) 설계	54
------------------	-----------------------------------	-----------

Chapter 2	응답 받은 공공데이터 분석	60
------------------	-----------------------	-----------

Chapter 3	공공데이터 저장용 DB 설계 및 구축	62
------------------	-----------------------------	-----------

Part 1. 공공데이터 분석 및 허가 받기

Chapter 1. 공공데이터 개요 및 분석

공공데이터 의미

공공 데이터(open data)

- 데이터베이스, 전자화된 파일 등 공공기관이 법령 등에서 정하는 목적을 위하여 생성 또는 취득하여 관리하고 있는 전자적 방식으로 처리된 자료 또는 정보

공공데이터 분류

- i. 「전자정부법」에 따른 행정정보
- ii. 「국가정보화 기본법」에 따른 정보 중 공공기관이 생산한 정보
- iii. 「공공기록물 관리에 관한 법률」에 따른 전자기록물 중 대통령령으로 정하는 전자기록물, 그 밖에 대통령령으로 정하는 자료 또는 정보

결론적으로 공공데이터는 개별 공공기관이 일상적 업무수행의 결과물로 생성 또는 수집·취득한 다양한 형태(텍스트, 수치, 이미지, 동영상, 오디오 등)의 모든 자료 또는 정보를 대상으로 하며, "기계 판독이 가능한 형태"로 "제공"되어야 함

- ▶ "기계 판독이 가능한 형태"란 소프트웨어로 데이터의 개별 내용 또는 내부 구조를 확인하거나 수정, 변환, 추출 등 가공할 수 있는 상태
- ▶ "제공"이란 공공기관이 이용자로 하여금 기계 판독이 가능한 형태의 공공데이터에 접근할 수 있게 하거나 이를 다양한 방식으로 전달하는 것

공공데이터 정책 추진

- 정부는 보유한 공공 정보를 적극적으로 개방하여 국민과 공유함으로써 소통과 협력을 확대하기 위해 공공데이터 정책을 추진
- 2013년 7월 공공데이터의 제공 및 이용활성화에 관한 법률 제정
- 약칭으로 공공데이터법을 제정, 2013년 10월부터 시행
- '행정안전부'가 총괄하며, 이 법률에 근거하여 2013년 11월, 정부나 공공기관이 보유한 공공데이터를 민간에서 쉽게 활용할 수 있도록 지원하기 위하여 [한국정보화진흥원에 공공데이터활용지원센터](#)를 설치
- 각 공공기관은 공공데이터 제공책임관 및 실무자를 지정하고 관련업무를 지원
- 정부는 공공데이터 관련 정책을 심의, 조정 및 그 추진사항을 점검하기 위하여 국무총리실 산하에 공공데이터전략위원회 설립
- 공공기관의 공공데이터 제공거부 또는 중단문제 발생시 해결을 위한 공공데이터제공분쟁조정위원회 설립

공공데이터 개방

- 공공데이터 개방이란 정부, 지방자치단체 및 공공기관이 보유·관리하는 공공데이터를 이용자가 자유롭게 활용할 수 있도록 다양한 형태로 개방 제공하는 것이고, 대한민국 정부, 지방자치단체 및 공공기관은 공공데이터를 이용자가 자유롭게 활용할 수 있도록 다양한 형태로 개방 제공하고 있음
- 단, 개인정보보호, 제3자의 권리(저작권 등)가 포함된 정보, 국가 안보와 질서에 관련한 정보 등 제외

- 공공 데이터는 누구나 자유롭게 이용할 수 있으며 공공데이터포털을 통해 국민이 쉽고 편리하게 공공데이터를 이용 가능
- 공공데이터는 파일데이터, 오픈API, 시각화 등 다양한 방식으로 제공

공공데이터 활용

- 정부는 공공데이터 이용에 대한 국민의 인식을 높이고 이용 활성화를 촉진하기 위하여, 공공데이터 활용 기업의 창업에서부터 성장까지 전과정을 지원

- 공공데이터 활용의 대표적인 예

1. 창업 공간인 오픈스퀘어-D

- * 오픈스퀘어-D는 공공데이터에 대한 창의적인 아이디어를 가진 사람들이 경험과 기술을 교류하고, 사업화/창업으로 성장할 수 있도록 종합적으로 지원하는 열린공간임
- * 창업자들을 위한 입주공간, 아이디어 교류 및 발전을 위한 협업공간, 서비스 개발에 필요한 테스트 및 데이터 분석을 위한 테스트 공간으로 구성
- * 창업단계별 교육, 맞춤형 컨설팅, 투자유치 및 네트워킹 등 다양한 프로그램이 제공됨

2. 창업경진대회

- * 공공데이터를 활용한 아이디어 또는 웹(앱) 제품 및 서비스 개발 아이템을 발굴하는 대회
- * 민간의 창의적인 아이디어를 공공데이터와 융합하여 새로운 비즈니스 모델을 발굴해 공공데이터 기반의 창업촉진 및 창업기업 육성 목적
- * 매년 1회 개최되며 국민 누구나 참여가 가능
- * 수상자에게는 시상금뿐만 아니라 제품 및 서비스 홍보, 사무공간 지원, 투자유치, 창업컨설팅 등 다양한 혜택 제공

공공데이터 포털 활용

- 공공데이터활용 관련 내용은 공공데이터 포털에서 확인할 수 있으며 어떤 공공데이터가 어떤 애플리케이션에 적용됐는지 확인할 수 있도록 아래와 같이 제공하고 있음

(<https://www.data.go.kr/tcs/puc/selectPublicUseCaseListView.do>)

공공데이터 활용사례

공공데이터를 활용하여 개발된 국내외 다양한 사례와 가공데이터를 공유하는 공간입니다.

전체 ▾





검색어를 입력하세요

검색

카테고리 열기 ▾

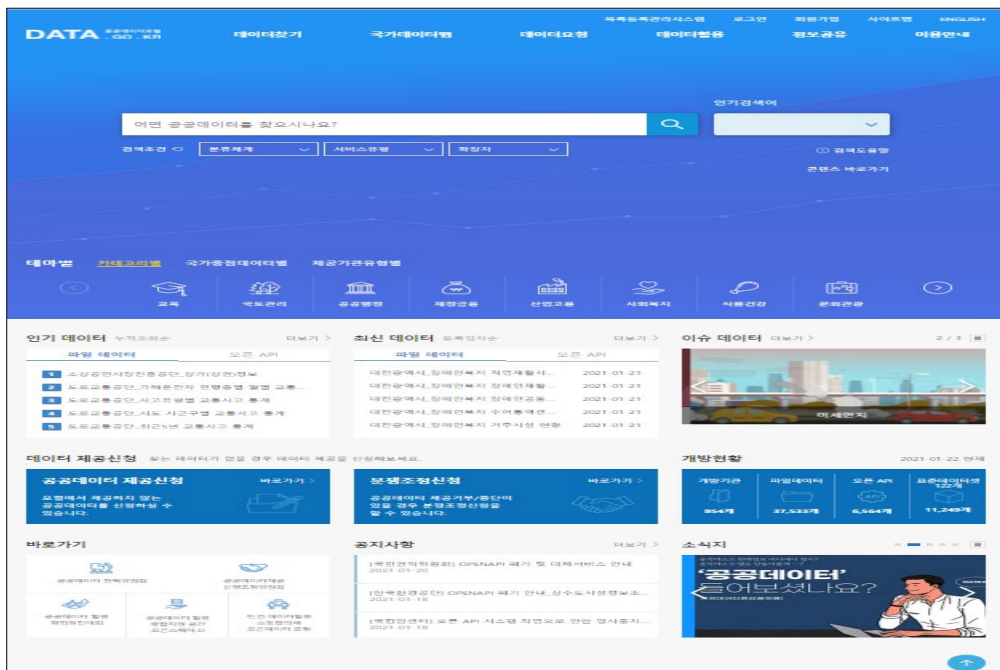
총 2,801건

내가 등록한 게시글만 보기 ☐ OFF [활용사례 등록하기](#)

 <div>환경기상 웹사이트</div> <p>나의 피부를 위한 오늘의 날씨 ...</p> <p>● 4차 산업혁명 시대, 기존 코스메틱 시장에 ICT 융복합 기술 도입을 통한 신</p> <p>2023-01-19</p>	 <div>재정금융 기타</div> <p>주식정보 수집 프로그램에서 주...</p> <p>특정 주식 정보를 수집하는 프로그램을 운영중입니다.</p> <p>2023-01-16</p>	 <div>교육 모바일앱</div> <p>전국평생교육학습찾기</p> <p>평생 학습 강좌에 관심이 있는 사람들이 전국 평생 교육 강좌에 대한 정보를</p> <p>2023-01-11</p>	 <div>보건의료 모바일앱</div> <p>시니어 헬스케어 서비스 링크</p> <p>'시니어 헬스케어 서비스 링크'은 자주 볼지 못하는 부모님께 즐거움과 건강을</p> <p>2023-01-09</p>
---	---	---	--

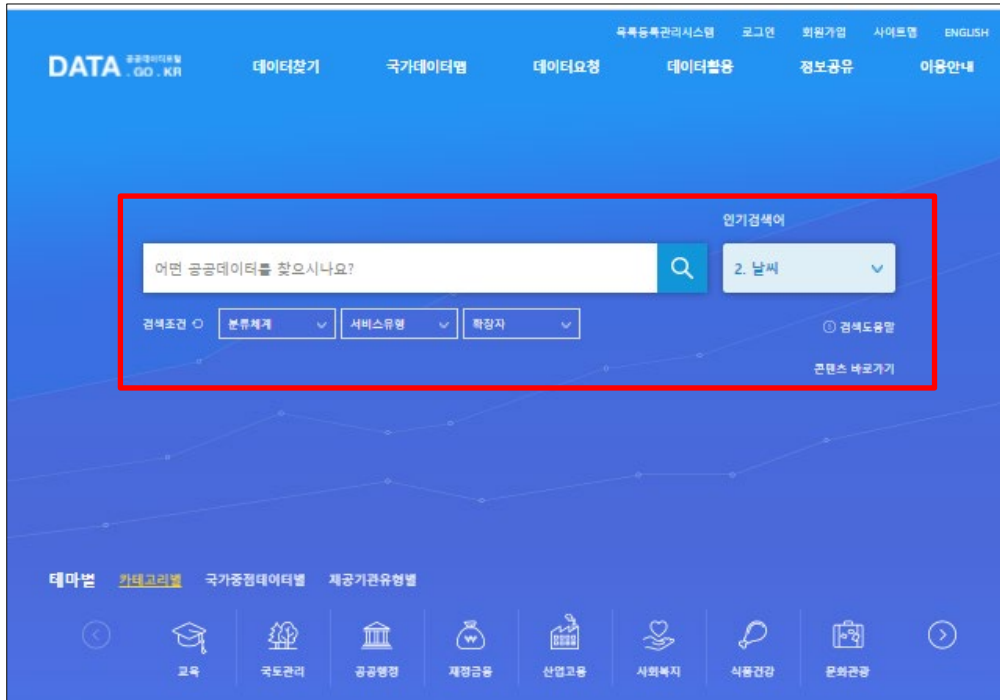
- 정부에서 운영하고 있는 공공데이터를 제공하는 웹애플리케이션으로 누구나 자유롭게 공공데이터를 확인하고 관련 내용을 활용하여 애플리케이션을 만들 수 있게 도움을 줌

<https://www.data.go.kr/> 접속



- 메인화면 구성은 상단에 통합조회를 하는 조회창이 있고, 각종 공공데이터를 간편하게 조회할 수 있도록 카테고리별, 인기데이터, 최신데이터 등을 리스트업 해놓았으며, 각종 정보를 확인요청 할 수 있게 구성되어 있음

- 공공데이터 포털에서는 다양한 방법으로 데이터 조회 가능하며, 일반적으로 통합검색을 많이 이용하며 검색창에서 Keyword를 입력하면 관련된 데이터가 조회되어 나옴



- 검색 창 '환경공단 에어코리아 대기오염' 입력

"환경공단 에어코리아 대기오염"에 대해 총 2,650건이 검색되었습니다.

조건검색 초기화

분류체계	서비스유형	제공기관유형	태그	확장자
국가중점데이터 분류 조건 추가하기 +				
조건열기 ▼				
<div style="display: flex; justify-content: space-between; border: 1px solid #ccc; padding: 5px;"> <div style="border: 1px solid #f00; padding: 2px 5px; background-color: #e0f0ff;">전체(2,650건)</div> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #fff;">파일데이터(2,067건)</div> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #fff;">오픈 API(454건)</div> <div style="border: 1px solid #ccc; padding: 2px 5px; background-color: #fff;">표준데이터셋1개(129건)</div> </div>				
<div style="display: flex; align-items: center;"> <div style="border: 1px solid #ccc; padding: 2px 5px;">정확도순 ▼</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">5개씩 ▼</div> <div style="background-color: #005596; color: white; padding: 2px 10px; margin-left: 10px;">정렬</div> </div>				

- 대기오염과 관련된 데이터들이 조회가 되며 제공하는 방식에 따라
파일 데이터, 오픈API, 표준데이터셋으로 구분
 - 파일데이터방식 : CSV, XML, xls형태의 파일로 데이터를 제공하는 것을 말하며 이 방식의 데이터는 사용자가 직접 파일을 다운로드하여 사용할 수 있음
 - 오픈API방식 : HttpRequest요청을 받으면 서버가 데이터를 보내주는 방식으로 요청주소, 파라미터, 반환데이터 등이 정해져있으며 이를 이용하기 위해서는 인증키를 발급받아야 한다. 웹 애플리케이션을 만들 때 주로 사용하는 방식임

- 추가적으로 조건열기를 선택하면 아래 그림과 같이 세부적인 항목이 나오고 조회도 가능

조건검색

초기화

분류체계	서비스유형	제공기관유형	태그	확장자
<input type="checkbox"/> 공공행정	<input type="checkbox"/> LINK	<input type="checkbox"/> 공공기관	<input type="checkbox"/> 고구려	<input type="checkbox"/> CSV
<input type="checkbox"/> 과학기술	<input type="checkbox"/> LOD	<input type="checkbox"/> 교육기관	<input type="checkbox"/> 고분벽화	<input type="checkbox"/> DOC
<input type="checkbox"/> 교육	<input type="checkbox"/> REST	<input type="checkbox"/> 교육행정기관	<input type="checkbox"/> 고양시	<input type="checkbox"/> DOCX
<input type="checkbox"/> 교통물류	<input type="checkbox"/> RSS/ATOM	<input type="checkbox"/> 국가행정기관	<input type="checkbox"/> 고전	<input type="checkbox"/> HWP
<input type="checkbox"/> 국토관리	<input type="checkbox"/> SOAP	<input type="checkbox"/> 위원회	<input type="checkbox"/> 관광	<input type="checkbox"/> JPG
<input type="checkbox"/> 농축수산	<input type="checkbox"/> 다운로드	<input type="checkbox"/> 입법기관	<input type="checkbox"/> 덕흥리	<input type="checkbox"/> JSON

국가중점데이터 분류 조건 추가하기 +

조건집기 ^

· 데이터 상세화면 보기

- 검색화면에 나온 데이터를 클릭하면 상세화면으로 들어가게 되는데 상세화면은 파일데이터방식과 오픈API방식이 약간 상이함

- 파일데이터방식 상세화면

- * 검색결과 탭에서 파일데이터를 선택 후 검색된 내용 중 임의의 항목 클릭

파일데이터 상세

경기도 부천시 대기오염측정결과

부천시 대기오염과 관련하여 일정한 측정값을 통하여 지속적인 환경관련 대기오염도 자료 제공

다운로드 | 오류신고 및 담당자 문의

파일데이터 정보

파일데이터명	경기도 부천시 대기오염측정결과_20200330		
분류체계	환경 - 대기	제공기관	경기도 부천시
관리부서명	스마트시티담당관	관리부서 전화번호	032-629-3882
보유근거		수집방법	
메타데이터 주기	연간	자기 등록 예정일	2021-03-31
메타데이터	엑셀	현재 행	14
확장자	CSV	다운로드(비로그인)	110
데이터 한계		키워드	대기오염도, 측정, 정보
등록	2017-06-19	수정	2020-09-25
제공형태	공공데이터포털에서 다운로드(원문파일등록)		
설명	부천시 대기오염과 관련하여 일정한 측정값을 통하여 지속적인 환경관련 대기오염도 자료 제공		
기타 유의사항			
비밀부과영향	무로	비밀부과기준 및 단위	단
이용허락범위	이동통신망의 전송, 보존		

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

미리보기

항목	SO2	NO2	O3	CO	PM
단위기준	0.02ppm/단	0.03ppm/단	0.06ppm/8시간	5ppm/8시간	50
평균	0.005	0.025	0.025	0.5	55
1월	0.005	0.036	0.013	0.7	76
2월	0.005	0.033	0.019	0.7	68
3월	0.005	0.031	0.028	0.6	63
4월	0.005	0.029	0.028	0.5	58
5월	0.005	0.029	0.042	0.5	60
6월	0.005	0.025	0.038	0.4	41
7월	0.004	0.02	0.03	0.4	38
8월	0.005	0.021	0.033	0.5	36
9월	0.005	0.025	0.022	0.4	34
10월	0.005	0.03	0.019	0.4	40
11월	0.006	0.037	0.013	0.6	54
12월	0.005	0.036	0.011	0.6	49

목록

- * 파일데이터방식은 페이지에서 바로 다운받을 수 있는 데이터도 있고 링크로 연결되어있는 데이터도 존재

- * 상세화면 상단 다운로드 버튼을 이용하면 파일을 다운로드할 수 있고 메타데이터로 파일의 확장자, 제공자 분류 등을 확인할 수 있음

* 하단에는 미리보기가 있어 어떤 데이터가 어떻게 들어가 있는지 한눈에 확인가능

- 오픈API 상세화면

* 오픈API도 파일을 제공하는 방식이 있는데 xml, json방식으로 제공

* 오픈API항목을 선택하고 '환경공단_에어코리아 대기오염' 클릭

오픈API 상세

XML

JSON

한국환경공단_에어코리아_대기오염정보

활용신청

각 측정소별 대기오염정보를 조회하기 위한 서비스로 기간별, 시도별 대기오염 정보와 통합대기환경지수 나쁨 이상 측정소 내역, 대기질(미세 먼지/오존) 예보 통보 내역 등을 조회할 수 있다.

※ 운영계정으로 사용하고자 할 경우 에어코리아 OpenAPI 사용자 관리시스템(<https://apiweb.airkorea.or.kr>)에 접속하여 인증키 및 개발보고서를 등록해주셔야 담당자 승인 후 사용이 가능합니다.

32

18

관심

OpenAPI 정보

메타데이터 다운로드

오픈API 예시코드

데이터 개선요청

오류신고 및 문의

분류체계	환경 - 대기	제공기관	한국환경공단
관리부서명	대기환경처 대기정책지원부	관리부서 전화번호	032-590-3508
API 유형	REST	데이터포맷	JSON+XML
활용신청	18471	키워드	미세먼지,코로나,오존
등록	2020-12-06	수정	2022-10-01
비용부과유무	무료	신청가능 트래픽	개발계정 : 500 / 운영계정 : 활용사레 등록시 신청하면 트래픽 증가 가능
심의유형	개발단계 : 자동승인 / 운영단계 : 심의승인		
이용허락범위	공공저작물_출처표시 + 변경금지		
참고문서	에어코리아 대기오염정보 조회 서비스 기술문서_v1.0.docx		

* 오픈API방식은 활용신청을 해서 인증키를 받아야 사용이 가능

* 하단 내용에 요청할 주소, 요청변수, 출력결과를 확인 할 수 있고, 데이터 이용 가이드라고 할 수 있는 참고 문서가 있으며 참고문서에는 상세한 정보들이 있기 때문에 반드시 다운받고 확인해야함

Chapter 2. 공공데이터 활용 신청

서비스 신청하기

- 오픈 API를 사용하려면 활용신청 후 인증키를 부여받은 후 사용 가능
- 상세화면에서 활용신청 버튼 클릭



- 활용신청을 하기 위해서는 회원가입 필요

DATA 공공데이터포털 . GO . KR

데이터찾기 국가데이터맵 데이터요청 데이터활용 정보공유 이용안내

국립등록관리시스템 로그인 **회원가입** 사이트맵 ENGLISH

회원가입

국민과 함께 하는 공공데이터포털에 오신 것을 환영합니다.

회원가입을 하시면 공공데이터포털을 통해 국가가 보유하고 있는 유용한 공공데이터와 오픈API 등 모든 정보를 자유롭게 이용하실 수 있습니다.

STEP1 가입확인 및 회원구분

STEP2 정보입력 및 약관동의

1. 회원가입여부 확인을 위해 이름, 이메일 주소를 입력해주세요.

이름

이메일

가입확인

2. 14세 이상은 일반회원으로 가입해주세요

일반인

만14세이상
내국인

가입하기

어린이

만14세미만
내국인

가입하기

- 이름과 이메일을 입력하고 가입 확인 후 가입 진행
- 이메일 인증과 핸드폰인증이 있으니 이메일은 반드시 본인이 쓰고 있는 메일주소로 작성, 세부정보를 입력하고 가입 완료

STEP1 가입확인 및 회원구분	STEP2 정보입력 및 약관동의
기본정보 입력 *표시는 필수 입력항목입니다.	
* 아이디 <input type="text"/> <small>4~20자 이내 영문 소문자, 숫자만 가능합니다.</small>	<input type="button" value="중복확인"/>
* 비밀번호 <input type="password"/> <small>한글, 영문문자, 개연정보 관련은 사용불가 숫자, 영문자, 특수문자(!, @, #, \$, %, ^, &, *, ~, ~, ~) 조합 9자이상 ~ 20자 이하</small>	
* 비밀번호 확인 <input type="password"/>	
개인정보 입력	
* 이름 <input type="text"/>	
* 이메일 <input type="text"/>	<input type="button" value="이메일 인증"/> <input type="checkbox"/> 이메일 수신 동의
<small>수신거부 시 일부 서비스에 제한이 있을 수 있습니다. 비밀번호 초기화 메일 수신 등에 반드시 필요한 정보이므로 정확히 이메일 주소를 입력해주세요.</small>	
전화번호 <input type="text"/>	
* 휴대전화번호 <input type="text"/>	
<input type="checkbox"/> SMS 수신 동의 <small>수신거부 시 일부 서비스에 제한이 있을 수 있습니다.</small>	
* 일반회원 인증	휴대폰 인증 <input type="button" value="본인 인증하기"/>
* 자동등록방지	<input type="text"/> 새로고침 <input type="button" value="보안문자를 입력하세요."/>
<input type="checkbox"/> 공공데이터포털 서비스 이용약관에 동의합니다.	
<div style="border: 1px solid #ccc; padding: 5px;"> 제1조 목적 본 약관은 공공데이터 포털(이하 '포털'이라 합니다)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. </div>	
<input type="checkbox"/> (필수) 개인정보 수집 및 이용에 동의합니다.	<input type="checkbox"/> (선택) 개인정보 수집 및 이용에 동의합니다.
<div style="border: 1px solid #ccc; padding: 5px;"> 가. 수집하는 개인정보의 항목 공공데이터포털은 다음의 개인정보 항목을 수집하고 있습니다. ○. 공공데이터포털 회원정보 * 회원가입 시 입력해야 하는 필수항목과 선택항목이 있으며, 선택항목은 입력하지 않으셔도라도 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 </div>	
<input type="checkbox"/> 전체 약관에 동의합니다.	
<div style="display: flex; justify-content: center; gap: 10px;"> <input type="button" value="취소"/> <input type="button" value="작성 완료"/> </div>	

- 로그인 후 활용신청 다시 진행
- 활용신청을 하면 활용 목적을 작성하는데 참고자료선택 후 "직업훈련교육
참고자료"입력 -> 활용신청

Chapter 3. Open API 인증키 확인

- 활용신청 후 마이페이지에서 신청한 데이터에 대한 정보 확인 가능

[illegible]

- 신청후 발급된 인증키로 데이터를 이용하려면 약간의 시간이 소요됨

· 마이페이지에서 다시 확인

<p>신청 0건 ></p> <p>신청중인 단계</p> <ul style="list-style-type: none"> · 보류 0건 · 반려 0건 	<p>활용 1건 ></p> <p>승인되어 활용중인 단계</p> <ul style="list-style-type: none"> · 변경신청 0건 	<p>중지0건 ></p> <p>중지신청하여 운영이 중지된 단계</p>
<p>상세검색 열기 ▾</p>		
<p>총1건 폐기된 목록 포함 보기 <input type="checkbox"/> OFF</p>		
<p>환경기상 한국환경공단</p> <p>활용신청 [승인] 한국환경공단_에어코리아_대기오염정보</p> <p>신청일 2022-11-29 만료예정일 2024-11-29</p>		

· 승인된 내용 클릭하여 상세보기 진행, 서비스정보 항목에서 인증키 확인 가능

서비스정보	
참고문서	에어코리아 대기오염정보 조회 서비스 기술문서 v1.0.docx
데이터포맷	JSON+XML
End Point	http://apis.data.go.kr/B552584/ArpltnInforInquireSvc
<p>API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다.</p> <p>포털에서 제공되는 Encoding/Decoding 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다.</p> <p>* 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.</p>	
일반 인증키 (Encoding)	ey51rbqUG4nLxm%2FIMXybyk%2FsuSNe95v4aIgn4Ydn0UhcJl91gToKXy7htiiqkWXyGCVxm6xyZpakkO4hk%2FLs3w%3D%3D
일반 인증키 (Decoding)	ey51rbqUG4nLxm/IMXybyk/suSNe95v4aIgn4Ydn0UhcJl91gToKXy7htiiqkWXyGCVxm6xyZpakkO4hk/Ls3w==

Part 2. JSON, XML 기반 데이터 수집 실습

Chapter 1. JSON, XML 이해하기

- OpenAPI 방식 사용 시 주요 내용은 요청변수와 출력결과 2가지가 중요
 - data.go.kr 메인 -> 검색 -> "환경공단 에어코리아 대기오염"
 - 오픈API 탭 -> "한국환경공단_에어코리아_대기오염정보" 클릭 후 확인
- 요청변수(Request Parameter)
OpenAPI 서버에 요청 시 필요한 속성들을 나열한 것으로,
항목구분이 필수인 속성들은 반드시 key=value형식으로
전달해야만 함

요청변수(Request Parameter)					
항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
서비스키	serviceKey	4	필	-	공공데이터포털에서 받은 인증키
데이터표출방식	returnType	4	옵	xml	xml 또는 json
한 페이지 결과 수	numOfRows	4	옵	100	한 페이지 결과 수(조회 날짜로 검색 시 사용 안함)
페이지 번호	pageNo	4	옵	1	페이지번호(조회 날짜로 검색 시 사용 안함)
조회 날짜	searchDate	10	옵	2020-11-14	통보시간 검색(조회 날짜 입력이 없을 경우 한달동안 예보통보 발령 날짜의 리스트 정보를 확인)
통보코드	InformCode	20	옵	PM10	통보코드검색(PM10, PM25, O3)

- 출력결과
OpenAPI 서버에 요청 결과 응답메시지에 포함된 데이터형식을
의미하며, 기본적으로 결과코드, 결과 메시지 등을 포함하고 있으며
실제 필요로 하는 값들이 담겨 있는 형식이므로 API문서를
잘 파악해서 구현하고자 하는 Application과 연동해야 함

응답결과(Response Element)					
항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
결과코드	resultCode	2	필	00	결과코드
결과메시지	resultMsg	50	필	NORMAL SERVICE	결과메시지
한 페이지 결과 수	numOfRows	4	옵	10	한 페이지 결과 수(조회 날짜 입력 시 결과 수 없음)
페이지 번호	pageNo	4	필	1	페이지번호(조회 날짜 입력 시 페이지 번호 없음)
전체 결과 수	totalCount	4	필	8	전체 결과 수
목록	items	0	옵	-	목록
통보시간	dateTime	20	필	2020-11-13 11시 발표	통보시간
통보코드	informCode	10	필	PM10	통보코드
예보개황	informOverall	500	필	○[미세먼지] 수도권·강원영서...	예보개황
발생원인	informCause	2000	필	○[미세먼지] 황사의 영향으로...	발생원인
예보등급	informGrade	100	필	서울: 나쁨, 제주: 나쁨, 전남: 나쁨...	예보등급
행동요령	actionKnack	2000	필	-	행동요령
첨부파일명	imageUrl1	600	필	https://www.airkorea.or.kr/dustimage/	시간대별 예측모형 결과사진(6:00, 12:00, 18:00, 24:00 KST)

· 응답메시지에 포함된 데이터형식 종류

i. XML 형식

HTML과 같은 마크업언어(Markup Language)로 웹에서 구조화된 문서를 전송 가능하도록 설계되어 각 요소들의 독립성을 보장함으로써 문서의 호환성, 내용의 독립성, 요소 변경의 용이성 등의 특성을 제공하는 사용자정의 태그 문서

ii. JSON 형식

자바스크립트의 구문 형식으로 프로그래밍 언어나 플랫폼에 독립적이기 때문에 C, C++, java, javaScript 등에서 사용 가능

iii. CSV 형식

각 항목이나 내용마다 쉼표(,)로 구분되어 저장되며 CSV형식의 파일은 텍스트 파일로 보존되고, 문서처리기나 편집기에서 열람 및 편집이 가능하며 엑셀에서 '다른이름으로 저장' 시 파일형식을 CSV 포맷으로 변환도 가능함

iv. XLS 형식

마이크로소프트사의 엑셀 문서 형식으로 스프레드시트를 연결하고 통합하여 다양한 도형과 차트 등 자료를 작성하는 기능을 제공

Chapter 2. XML 데이터 파싱

· 이전 실습 내용을 참조(P13~17)하여 data.go.kr 사이트 회원가입 및
"한국환경공단_에어코리아_대기오염정보" 승인과정까지 진행

- 마이페이지 -> 승인된 "한국환경공단_에어코리아_대기오염정보" 클릭
- 활용신청 상세기능정보 내용 중 "5. 시도별 실시간 측정정보 조회" 확인

활용신청 상세기능정보				
NO	상세기능	설명	일일 트래픽	미리보기
1	대기질 예보통보 조회	통보코드와 통보시간으로 예보정보와 발생 원인 정보를 조회하는 대기질(미세먼지/오존) 예보통보 조회	500	확인
2	초미세먼지 주간예보 조회	통보코드와 통보시간으로 대기질 전망과 주간예보 정보를 조회하는 초미세먼지 주간예보통보 조회	500	확인
3	측정소별 실시간 측정정보 조회	측정소명과 측정데이터 기간(일,한달,3개월)으로 해당 측정소의 일반항목 측정정보를 제공하는 측정소별 실시간 측정정보조회	500	확인
4	통합대기환경지수 나쁨 이상 측정소 목록조회	통합대기환경지수가 나쁨 등급 이상인 측정소명과 주소 목록 정보를 제공하는 통합대기환경지수 나쁨 이상 측정소 목록조회	500	확인
5	시도별 실시간 측정정보 조회	시도명을 검색조건으로 하여 시도별 측정소목록에 대한 일반 항목과 CAI최종 실시간 측정값과 지수 정보 조회 기능을 제공하는 시도별 실시간 측정정보 조회	500	확인

- 요청변수(Request Parameter) 확인

요청변수(Request Parameter)			닫기
항목명	샘플데이터	설명	
serviceKey	-	공공데이터포털에서 받은 인증키	
returnType	xml	xml 또는 json	
numOfRows	100	한 페이지 결과 수	
pageNo	1	페이지번호	
sidoName	서울	시도 이름(전국, 서울, 부산, 대구, 인천, 광주, 대전, 울산, 경기, 강원, 충북, 충남, 전북, 전남, 경북, 경남, 제주, 세종)	
ver	1.0	버전별 상세 결과 참고	

미리보기

- returnType, xml확인 후 -> 미리보기 클릭 -> 응답데이터 확인

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL_CODE</resultMsg>
  </header>
  <body>
    <items>
      <item>
        ...
      </item>
      <item>
        <so2Grade>1</so2Grade>
        <coFlag/>
        <khaiValue>50</khaiValue>
        <so2Value>0.003</so2Value>
        <coValue>0.5</coValue>
        <pm25Flag/>
        <pm10Flag/>
        <o3Grade>1</o3Grade>
        <pm10Value>33</pm10Value>
        <khaiGrade>1</khaiGrade>
        <pm25Value>10</pm25Value>
        <sidoName>서울</sidoName>
        <no2Flag/>
        <no2Grade>1</no2Grade>
        <o3Flag/>
        <pm25Grade>1</pm25Grade>
        <so2Flag/>
        <dateTime>2023-01-25 13:00</dateTime>
        <coGrade>1</coGrade>
        <no2Value>0.023</no2Value>
        <stationName>한강대로</stationName>
      </item>
    </items>
  </body>
</response>

```


- 상단 URL확인

* URL정보 중 ? 기준으로 좌측이 요청 URL정보이며,

우측의 값들이 &를 중심으로 요청 내용들이 나열되어 있음

apis.data.go.kr/B552584/ArpltnInforInquireSvc/getCtprvnRltmMesureDnsty?

serviceKey=ey51rbqUG4nL.....&returnType=xml&.....&.....

* 해당 URL 내용 중 sidoName=서울 항목을 sidoName=부산으로 변경

* 부산지역에 대한 내용을 가져옴

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL_CODE</resultMsg>
  </header>
  <body>
    <items>
      <item>
        <so2Grade>1</so2Grade>
        <coFlag/>
        <khaiValue>40</khaiValue>
        <so2Value>0.002</so2Value>
        <coValue>0.3</coValue>
        <pm25Flag/>
        <pm10Flag/>
        <o3Grade>1</o3Grade>
        <pm10Value>10</pm10Value>
        <khaiGrade>1</khaiGrade>
        <pm25Value>2</pm25Value>
        <sidoName>부산</sidoName>
        <no2Flag/>
        <no2Grade>1</no2Grade>
        <o3Flag/>
        <pm25Grade>1</pm25Grade>
        <so2Flag/>
        <dateTime>2023-01-25 14:00</dateTime>
        <coGrade>1</coGrade>
        <no2Value>0.012</no2Value>
        <stationName>달리동</stationName>
      </item>
    </items>
  </body>
</response>
```

* 각각의 item항목 내용들이 하나의 측정소에 대한 정보들임

- 각 항목이 의미하는 값 확인은, 서비스정보항목의 참고문서를
다운로드 받은 후 “응답메시지 명세”에서 최종 확인 가능

항목명(영문) [↗]	항목명(국문) [↗]	항목크기 [↗]	항목구분 [↗]	샘플데이터 [↗]	항목설명 [↗]
items [↗]	목록 [↗]	- [↗]	0..n [↗]	- [↗]	목록 [↗]
dateTime [↗]	측정일 [↗]	20 [↗]	1 [↗]	2020-11-25 13:00 [↗]	오염도측정 연·월·일 [↗] 시간: 분 [↗]
mangName [↗]	측정망 정보 [↗]	10 [↗]	1 [↗]	도시대기 [↗]	측정망 정보 [↗] (도시 대기, 도로변 대기, 국가배경농도, 교외대기, 항만) [↗]
so2Value [↗]	아황산가스 농도 [↗]	10 [↗]	1 [↗]	0.007 [↗]	아황산가스 농도 [↗] (단위 : ppm) [↗]
coValue [↗]	일산화탄소 농도 [↗]	10 [↗]	1 [↗]	0.4 [↗]	일산화탄소 농도 [↗] (단위 : ppm) [↗]
o3Value [↗]	오존 농도 [↗]	10 [↗]	1 [↗]	0.043 [↗]	오존 농도 [↗] (단위 : ppm) [↗]
no2Value [↗]	이산화질소 농도 [↗]	10 [↗]	1 [↗]	0.024 [↗]	이산화질소 농도 [↗] (단위 : ppm) [↗]
pm10Value [↗]	미세먼지(PM ₁₀) 농도 [↗]	10 [↗]	1 [↗]	73 [↗]	미세먼지(PM ₁₀) 농도 [↗] (단위 : µg/m³) [↗]

- STS(Spring Tool Suite) 실행

* New -> Spring Legacy Project -> Project Name : openDataProject ->

Spring MVC Project 선택 -> com.kh.opendata -> Finish

* openDataProject > src > main > java > com > kh >.opendata

HomeController.java 삭제

* openDataProject 우 클릭 -> Properties -> Project Facets 에서 자바버전 체크

- data.go.kr 접속 후 return Type -> xml 설정

요청변수(Request Parameter) 닫기		
항목명	샘플데이터	설명
serviceKey	-	공공데이터포털에서 받은 인증키
returnType	xml	xml 또는 json
numOfRows	100	한 페이지 결과 수
pageNo	1	페이지번호
sidoName	서울	시도 이름(전국, 서울, 부산, 대구, 인천, 광주, 대전, 울산, 경기, 강원, 충북, 충남, 전북, 전남, 경북, 경남, 제주, 세종)
ver	1.0	버전별 상세 결과 참고

미리보기

* 미리보기 -> xml 형식의 응답내용 확인 가능

```
▼<response>
  ▼<header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL_CODE</resultMsg>
  </header>
  ▼<body>
    ▼<items>
      ►<item>
        ...
      </item>
      ▼<item>
        <so2Grade>1</so2Grade>
        <coFlag/>
        <khaiValue>50</khaiValue>
        <so2Value>0.003</so2Value>
        <coValue>0.5</coValue>
        <pm25Flag/>
        <pm10Flag/>
        <o3Grade>1</o3Grade>
        <pm10Value>33</pm10Value>
        <khaiGrade>1</khaiGrade>
        <pm25Value>10</pm25Value>
        <sidoname>서울</sidoname>
        <no2Flag/>
        <no2Grade>1</no2Grade>
        <o3Flag/>
        <pm25Grade>1</pm25Grade>
        <so2Flag/>
        <dateTime>2023-01-25 13:00</dateTime>
        <coGrade>1</coGrade>
        <no2Value>0.023</no2Value>
        <stationName>한강대로</stationName>
      </item>
    </items>
  </body>
</response>
```

- STS(Spring Tool Suite)

* opendata 우 클릭 -> New -> Package -> Name : com.kh.opendata.model.vo

Java Package	
Create a new Java package.	
Creates folders corresponding to packages.	
Source folder:	<input type="text" value="openDataProject/src/main/java"/>
Name:	<input type="text" value="com.kh.opendata.model.vo"/>

* vo 우 클릭 -> New -> Class -> Name : AirVO

- AirVO.java 구성

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

AirVO.java	
01	package com.kh.opendata.model.vo;
02	
03	public class AirVO {
04	
05	private String stationName; //측정소명
06	private String datetime; // 측정일시
07	private String khaiValue; //통합대기환경 수치
08	
09	private String pm10Value; //미세먼지 농도
10	private String so2Value; //아황산가스 농도
11	private String coValue; //일산화탄소 농도
12	private String no2Value; //이산화질소 농도

13	private String o3Value; //오존 농도
14	
15	public AirVO() {}
16	
17	public AirVO(String stationName, String datetime, String khaiValue, String
18	pm10Value, String so2Value, String coValue, String no2Value, String o3Value) {
19	super();
20	this.stationName = stationName;
21	this.datetime = datetime;
22	this.khaiValue = khaiValue;
23	this.pm10Value = pm10Value;
24	this.so2Value = so2Value;
25	this.coValue = coValue;
26	this.no2Value = no2Value;
27	this.o3Value = o3Value;
28	}
29	
30	public String getStationName() {
31	return stationName;
32	}
33	
34	public void setStationName(String stationName) {
35	this.stationName = stationName;
36	}
37	
38	public String getDatetime() {
39	return datetime;
40	}
41	
42	public void setDatetime(String datetime) {
43	this.datetime = datetime;
44	}

45	
46	public String getKhaiValue() {
47	return khaiValue;
48	}
49	
50	public void setKhaiValue(String khaiValue) {
51	this.khaiValue = khaiValue;
52	}
53	
54	public String getPm10Value() {
55	return pm10Value;
56	}
57	
58	public void setPm10Value(String pm10Value) {
59	this.pm10Value = pm10Value;
60	}
61	
62	public String getSo2Value() {
63	return so2Value;
64	}
65	
66	public void setSo2Value(String so2Value) {
67	this.so2Value = so2Value;
68	}
69	
70	public String getCoValue() {
71	return coValue;
72	}
73	
74	public void setCoValue(String coValue) {
75	this.coValue = coValue;
76	}

```

77
78         public String getNo2Value() {
79             return no2Value;
80         }
81
82         public void setNo2Value(String no2Value) {
83             this.no2Value = no2Value;
84         }
85
86         public String getO3Value() {
87             return o3Value;
88         }
89
90         public void setO3Value(String o3Value) {
91             this.o3Value = o3Value;
92         }
93
94         @Override
95         public String toString() {
96             return "AirVO [stationName=" + stationName + ", datetime="
97 + datetime + ", khaiValue=" + khaiValue + ", pm10Value=" + pm10Value + ",
98 so2Value=" + so2Value + ", coValue=" + coValue + ", no2Value=" + no2Value + ",
99 o3Value=" + o3Value + "]\n";
100         }
101
102     }

```


- 응답데이터를 기반으로 자바 어플리케이션 개발 진행

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

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

Name: **AirPollutionJavaApp**

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

* AirPollutionJavaApp.java 코드 구성

== OpenAPI 서버로 요청하고자하는 URL 생성 ==

AirPollutionJavaApp.java	
01	package com.kh.opendata.run;
02	
03	import java.io.IOException;
04	import java.net.URLEncoder;
05	
06	public class AirPollutionJavaApp {
07	
08	// 발급받은 인증키(반드시 본인의 키 입력) 정보 변수에 담아두기

```

09      public static final String serviceKey =
10      "ey51rbqUG4nLxm%2FIMXybyk%2FsuSNe95v4algn4Ydn0UhcJl91gToKXy7
11      htiiqkWXyGCVxm6xyZpakkO4hk%2FLs3w%3D%3D";
12
13      public static void main(String[] args) throws IOException {
14
15      // OpenAPI서버로 요청하고자 하는 URL만들기
16      // 응답받은 xml URL내용 중 ? 전까지의 내용이 요청 URL, 따라서 ?
17      전까지 복사, = 우측에 붙여넣기
18      String url =
19      "http://apis.data.go.kr/B552584/ArpltnInforInquireSvc/getCtprvnRltmMesur
20      eDnsty";
21
22      // 이어서 요청할 전달값 추가(서비스키값과 시도명은 필수)
23      url += "?serviceKey=" + serviceKey;
24      url += "&sidoname=" + URLEncoder.encode("서울",
25      "UTF-8");
26      // 요청시 전달값 중 한글이 있을 경우 encoding해야 함
27      url += "&returnType=xml";
28      //영문이므로 encoding 필요 없음
29
30      System.out.println(url);
31      }
32  }

```

- 테스트

- * 실행 후 콘솔창에 나오는 내용 복사 -> 브라우저 URL 경로에 붙여넣기
 - ➔ xml 형식의 데이터 확인
 - ➔ 확인 후 30 행 주석처리 할 것 //System.out.println(url);

* AirPollutionJavaApp.java 코드 구성(이어서 작성)

== HttpURLConnection 객체 활용, OpenAPI 요청 절차 ==

1. 요청하고자하는 url 전달하면서 java.net.URL 객체 생성

(// System.out.println(url); 주석 처리 후 바로 하단에 추가)

URL requestUrl = new URL(url);

2. 1 번과정을 통해 생성된 URL 을 갖고, HttpURLConnection 객체 생성

HttpURLConnection urlConnection =

(HttpURLConnection)requestUrl.openConnection();

(한 줄로 작성할 것)

3. 요청에 필요한 Header 설정

urlConnection.setRequestMethod("GET");

4. OpenAPI 서버로 요청 후 입력 스트림을 통해 응답데이터 읽어들이기

BufferedReader br = new BufferedReader(new

InputStreamReader(urlConnection.getInputStream()));

(한 줄로 작성할 것)

String line;

while((line = br.readLine()) != null){

System.out.println(line);

}

5. 사용한 스트림 반납

```
br.close();  
urlConnection.disconnect();
```

- 테스트

코드 저장 후 실행 -> 하단 콘솔 창에 xml 형식 내용 확인

```
<?xml version="1.0" encoding="UTF-8"?>  
<response>  
  <header>  
    <resultCode>00</resultCode>  
    <resultMsg>NORMAL_CODE</resultMsg>  
  </header>  
  <body>  
    <items>  
      <item>  
        <so2Grade>1</so2Grade>  
        <coFlag/>  
        <khaiValue>-</khaiValue>  
        <so2Value>0.002</so2Value>  
        <coValue>0.7</coValue>  
        <pm10Flag/>  
        <o3Grade>1</o3Grade>  
        <pm10Value>26</pm10Value>  
        <khaiGrade/>  
        <sidName>서울</sidName>  
        <no2Flag/>  
        <no2Grade/>  
        <o3Flag/>  
        <so2Flag/>  
        <dateTime>2023-01-26 15:00</dateTime>  
        <coGrade>1</coGrade>  
        <no2Value>0.031</no2Value>  
        <stationName>도봉구</stationName>
```

- 테스트가 정상적으로 완료되었다면 한줄씩 출력되는 내용을 특정변수에 저장하기 위한 코드 작성 진행

* 4 번 항목 중

```
String line;  
while((line = br.readLine()) != null){  
    System.out.println(line);  
}
```

을 다음과 같이 추가 및 수정

```
String responseText = "";  
String line;  
while((line = br.readLine()) != null){  
    // System.out.println(line);  
    responseText += line;  
}  
System.out.println(responseText);
```

- AirPollutionJavaApp.java 최종 코드 내용

AirPollutionJavaApp.java	
01	package com.kh.opendata.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.net.URLEncoder;
09	
10	public class AirPollutionJavaApp {
11	
12	// 발급받은 인증키(반드시 본인의 키 입력) 정보 변수에 담아두기
13	public static final String <i>serviceKey</i> =
14	"eyJ51rbqUG4nLxm%2FIMXybyk%2FsuSNe95v4algn4Ydn0UhcJl91gToKXy7
15	htiiqkWXyGCVxm6xyZpakkO4hk%2FLs3w%3D%3D";
16	
17	public static void main(String[] args) throws IOException {
18	
19	// OpenAPI서버로 요청하고자 하는 URL만들기
20	// 응답받은 <u>xml</u> URL내용 중 ? 전까지의 내용이 요청 URL, 따라서 ?
21	전까지 복사, = 우측에 붙여넣기
22	String url =
23	"http://apis.data.go.kr/B552584/ArpltnInforInquireSvc/getCtprvnRltmMesur
24	eDnsty";
25	
26	// 이어서 요청할 전달값 추가(서비스키값과 시도명은 필수)
27	url += "?serviceKey=" + <i>serviceKey</i> ;
28	url += "&sidoname=" + URLEncoder.encode("서울", "UTF-8");

```

29 // 요청시 전달값 중 한글이 있을 경우 encoding해야 함
30
31     url += "&returnType=xml";
32 //영문이므로 encoding 필요 없음
33
34     // System.out.println(url);
35
36
37
38 // ** HttpURLConnection 객체를 활용해서 OpenAPI 요청절차 **
39 // 1. 요청하고자하는 url 전달하면서 java.net.URL 객체 생성
40     URL requestUrl = new URL(url);
41
42
43
44 // 2. 1번과정으로 생성된 URL객체를 가지고, HttpURLConnection 객체
45 생성
46     HttpURLConnection urlConnection =
47 (HttpURLConnection)requestUrl.openConnection();
48
49
50
51 // 3. 요청에 필요한 Header설정하기
52     urlConnection.setRequestMethod("GET");
53
54
55
56 // 4. 해당 OpenAPI 서버로 요청 보낸 후 입력 스트림을 통해
57 응답데이터 읽어들이기
58 // xml형식으로 반환되는 내용에서 한 줄씩 읽어들이려 함
59     BufferedReader br = new BufferedReader(new
60 InputStreamReader(urlConnection.getInputStream()));

```

```
61
62
63     String responseText = "";
64     String line;
65     while((line = br.readLine()) != null){
66         // System.out.println(line);
67         responseText += line;
68     }
69
70     System.out.println(responseText);
71
72
73     // 5. 사용한 스트림 반납
74     br.close();
75     urlConnection.disconnect();
76 }
77 }
```


Chapter 3. JSON 데이터 파싱

· 이전 실습 내용을 참조(P13~17)하여 data.go.kr 사이트 회원가입 및
"한국환경공단_에어코리아_대기오염정보" 승인과정까지 진행

- 마이페이지 -> 승인된 "한국환경공단_에어코리아_대기오염정보" 클릭
- 활용신청 상세기능정보 내용 중 "5. 시도별 실시간 측정정보 조회" 확인

활용신청 상세기능정보				
NO	상세기능	설명	일일 트래픽	미리보기
1	대기질 예보통보 조회	통보코드와 통보시간으로 예보정보와 발생 원인 정보를 조회하는 대기질(미세먼지/오존) 예보통보 조회	500	확인
2	초미세먼지 주간예보 조회	통보코드와 통보시간으로 대기질 전망과 주간예보 정보를 조회하는 초미세먼지 주간예보통보 조회	500	확인
3	측정소별 실시간 측정정보 조회	측정소명과 측정데이터 기간(일,한달,3개월)으로 해당 측정소의 일반항목 측정정보를 제공하는 측정소별 실시간 측정정보조회	500	확인
4	통합대기환경지수 나쁨 이상 측정소 목록조회	통합대기환경지수가 나쁨 등급 이상인 측정소명과 주소 목록 정보를 제공하는 통합대기환경지수 나쁨 이상 측정소 목록조회	500	확인
5	시도별 실시간 측정정보 조회	시도명을 검색조건으로 하여 시도별 측정소목록에 대한 일반 항목과 CAI최종 실시간 측정값과 지수 정보 조회 기능을 제공하는 시도별 실시간 측정정보 조회	500	확인

- 요청변수(Request Parameter) 내용 확인 후 returnType를 json으로 변경

요청변수(Request Parameter)
닫기

항목명	샘플데이터	설명
serviceKey	-	공공데이터포털에서 받은 인증키
returnType	json	xml 또는 json
numOfRows	100	한 페이지 결과 수
pageNo	1	페이지번호
sidoName	서울	시도 이름(전국, 서울, 부산, 대구, 인천, 광주, 대전, 울산, 경기, 강원, 충북, 충남, 전북, 전남, 경북, 경남, 제주, 세종)
ver	1.0	버전별 상세 결과 참고

미리보기

- 요청 시 전달 값(Parameter)
 - * serviceKey = 발급받은 인증키
 - * returnType = xml 또는 json
 - * numOfRows = 한 페이지당 결과 값 개수
 - * pageNo = 요청하고자하는 페이지번호
 - * sidoName = 조회하고자하는 지역명
 - * ver = 버전

- 각 항목이 의미하는 값 확인은, 서비스정보항목의 참고문서를
다운로드 받은 후 “응답메시지 명세”에서 최종 확인 가능

항목명(영문) [↗]	항목명(국문) [↗]	항목크기 [↗]	항목구분 [↗]	샘플데이터 [↗]	항목설명 [↗]
items [↗]	목록 [↗]	- [↗]	0..n [↗]	- [↗]	목록 [↗]
dateTime [↗]	측정일 [↗]	20 [↗]	1 [↗]	2020-11-25 13:00 [↗]	오염도측정 연·월·일 [↗] 시간: 분 [↗]
mangName [↗]	측정망 정보 [↗]	10 [↗]	1 [↗]	도시대기 [↗]	측정망 정보 [↗] (도시 대기, 도로변 대기, 국가배경농도, 교외대기, 항만) [↗]
so2Value [↗]	아황산가스 농도 [↗]	10 [↗]	1 [↗]	0.007 [↗]	아황산가스 농도 [↗] (단위 : ppm) [↗]
coValue [↗]	일산화탄소 농도 [↗]	10 [↗]	1 [↗]	0.4 [↗]	일산화탄소 농도 [↗] (단위 : ppm) [↗]
o3Value [↗]	오존 농도 [↗]	10 [↗]	1 [↗]	0.043 [↗]	오존 농도 [↗] (단위 : ppm) [↗]
no2Value [↗]	이산화질소 농도 [↗]	10 [↗]	1 [↗]	0.024 [↗]	이산화질소 농도 [↗] (단위 : ppm) [↗]
pm10Value [↗]	미세먼지(PM ₁₀) 농도 [↗]	10 [↗]	1 [↗]	73 [↗]	미세먼지(PM ₁₀) 농도 [↗] (단위 : µg/m³) [↗]

- STS(Spring Tool Suite) 실행

- * xml 실습 시 진행했던 내용을 활용하여 json실습 진행
- * AirPollutionJavaApp.java 최종 파일 내용(P36) 중 31행 내용을 변경
returnType = xml → **returnType = json**
- * 저장 후 실행, 확인 -> 콘솔창 결과가 json형식임을 확인(P40 참조)
- * JsonObject / JsonArray / JsonElement를 이용해 원하는 데이터 추출 실습을
진행할 예정이며, 세부적으로는 각각의 정보를 AirVO객체에 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

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

* AirPollutionJavaApp.java의 내용에서
P37, 70행 주식 처리 후 -> // System.out.println(responseText);
하단 내용 추가(72행부터 내용부터 추가)

AirPollutionJavaApp.java	
01	기존 내용 동일
~	// System.out.println(responseText);
71	
72	// JsonObject / JsonArray, JsonElement를 이용해서 파싱할 수
73	있음(gson라이브러리에서 제공)
74	=> 내가 원하는 데이터만을 추출할 수 있음
75	// 각각의 item정보를 => AirVO객체에 담고 => ArrayList에 쌓기
76	
77	// responseText에 담긴 문자열을 JsonObject객체로 변경
78	JsonObject totalObj =
79	JsonParser.parseString(responseText).getAsJsonObject();
80	
81	JsonObject responseObj = totalObj.getAsJsonObject("response");
82	// response속성 접근 => 응답데이터 {}에 있는 내용들을 JsonObject화
83	// System.out.println(responseObj); 출력 확인 후 주식
84	
85	JsonObject bodyObj = responseObj.getAsJsonObject("body"); //
86	body속성접근 => 응답데이터 {}에 있는 내용들을 JsonObject화
87	// System.out.println(bodyObj);
88	

```

89  int totalCount = bodyObj.get("totalCount").getAsInt();
90  // totalCount 속성 접근 => 40이라는 숫자이므로 int
91
92  JSONArray itemArr = bodyObj.getAsJSONArray("items");
93  // items속성 접근 => [] JSONArray
94      //System.out.println(totalCount); // 출력 확인 후 주석
95      //System.out.println(itemArr); // 출력 확인 후 주석
96
97
98  ArrayList<AirVO> list = new ArrayList<>();
99      for(int i=0; i<itemArr.size();i++) {
100          JsonObject item = itemArr.get(i).getAsJsonObject();
101          // System.out.println(item); 출력확인 후 주석
102
103          AirVO air = new AirVO(); //AirVO에서 필드값 확인
104          air.setStationName(item.get("stationName").getString());
105          air.setDatetime(item.get("dateTime").getString());
106          air.setKhaiValue(item.get("khaiValue").getString());
107          air.setPm10Value(item.get("pm10Value").getString());
108          air.setSo2Value(item.get("so2Value").getString());
109          air.setCoValue(item.get("coValue").getString());
110          air.setNo2Value(item.get("no2Value").getString());
111          air.setO3Value(item.get("o3Value").getString());
112
113          list.add(air);
114      }
115          // System.out.println(list); 확인 후 주석
116      for(AirVO a : list) {
117          System.out.println(a);
118      }
119      // 5. 사용한 스트림 반납
120      br.close();

```

121	<code>urlConnection.disconnect();</code>
122	<code>}</code>
123	<code>}</code>

* AirPollutionJavaApp.java 소스코드 작성 완료 후 저장/실행

Console창에서 결과 값 확인

- 결과 값이 제대로 출력이 되었다면 다음 이어지는 실습은

웹 페이지를 통한 요청에 대해 응답되어 온 데이터를 화면에 출력시키는 실습

- 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>testOK!!</body>` => testOK!! 내용 추가

== Tomcat 설정 ==

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

나머지 항목이 있다면 remove -> Tomcat start

* 브라우저에서 확인

브라우저 URL에서 localhost:포트번호/opendata로 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"></scri
10	pt>
11	</head>
12	<body>
13	<h2>실시간 대기오염 정보</h2>
14	
15	지역 :
16	<select id = "location">
17	<option>서울</option>
18	<option>부산</option>
19	<option>대전</option>
20	</select> <!-- 내용이 많아 3개만 option 지정 --%>
21	
22	<button id = "btn1"> 해당 지역 대기오염 정보</button>
23	
24	<table id = "result1" border = "1">

```

25         <thead>
26             <tr>
27                 <th>측정소명</th>
28                 <th>측정일시</th>
29                 <th>통합대기환경수치</th>
30                 <th>미세먼지농도</th>
31                 <th>일산화탄소농도</th>
32                 <th>이산화질소농도</th>
33                 <th>아황산가스농도</th>
34                 <th>오존농도</th>
35             </tr>
36         </thead>
37         <tbody>
38         </tbody>
39     </table>
40     <script>
41         $(function(){
42             $("#btn1").click(function(){
43                 $.ajax({
44                     url:"air.do",
45                     data:{location:$("#location").val()},
46                     success:function(){
47
48                         }
49                 })
50             })
51         })
52     </script>
53 </body>
54 </html>

```

* 해당 내용 작성 후

성공 시 실행되는 success 관련 내용은 추 후 작성 예정이며, 먼저
air.do라는 url로 요청 시 요청을 받아 처리해주는 controller 구성 실습

* opendata 우클릭 -> New -> Class -> Package : com.kh.opendata.controller,
Name : ApiController -> Finish

```
ApiController.java
01  package com.kh.opendata.controller;
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.net.URLEncoder;
09  import org.springframework.stereotype.Controller;
10  import org.springframework.web.bind.annotation.RequestMapping;
11  import org.springframework.web.bind.annotation.ResponseBody;
12
13  @Controller
14  public class ApiController {
15
16      private static final String serviceKey =
17      "ey51rbqUG4nLxm%2FIMXybyk%2FsuSNe95v4aI91gToKXy7
18      htiiqkWXyGCVxm6xyZpakkO4hk%2FLs3w%3D%3D";
19
20      @ResponseBody
21      @RequestMapping(value="air.do", produces = "application/json;
22      charset=utf-8")
23      public String airPollution(String location) throws IOException {
```

```

24
25         String url =
26         "http://apis.data.go.kr/B552584/ArpltnInforInquireSvc/getCtprvnRltmMesur
27         eDnsty";
28         url += "?serviceKey=" + serviceKey;
29         url += "&sidoname=" + URLEncoder.encode(location, "UTF-8");
30         url += "&returnType=json";
31         // System.out.println(url);
32
33         URL requestUrl = new URL(url);
34         HttpURLConnection urlConnection =
35         (HttpURLConnection)requestUrl.openConnection();
36         urlConnection.setRequestMethod("GET");
37
38         BufferedReader br = new BufferedReader(new
39         InputStreamReader(urlConnection.getInputStream()));
40
41         String responseText = "";
42         String line;
43         while((line = br.readLine()) != null) {
44             responseText += line;
45         }
46         br.close();
47         urlConnection.disconnect();
48         // System.out.println(responseText);
49         return responseText;
50     }
51 }

```

Chapter 4. REST API를 이용한 데이터 수집

REST API

- REST(Representational State Transfer) 정의
 - * Resource 를 HTTP Method 를 통해 처리하도록 설계된 아키텍처
- REST 구성요소
 - * 자원(URI) / 행위(HTTP Method) / 표현(Message)
- REST 사용 이유
 - * HTTP 표준을 기반으로 구현되며 메시지에 대한 의도 파악이 용이
 - * 다양한 브라우저와 디바이스에서 통신을 해야할 경우 어플리케이션을 분리 및 통합하기 위해 사용
- REST API
 - * REST 기반으로 서비스 API 를 구현한 것을 의미

- Page 49 내용 이어서 실습 진행

* 마지막으로 index.jsp내용 추가

기존 index.jsp내용의 1행 ~ 39행까지 수정내용 없으며,
40행부터 하단 내용 추가

Index.jsp

01	
~	이전 실습 내용과 동일
~	
40	<script>
41	\$(function(){
42	\$("#btn1").click(function(){
43	\$.ajax({
44	url:"air.do",
45	data:{location:\$("#location").val()},
46	success:function(data){
47	// console.log(data); 확인 후 주석

```

48 // console.log(data.response.body.items); 확인 후 주석
49     const itemArr = data.response.body.items;
50
51     let value = "";
52     for(let i in itemArr){
53 // console.log(itemArr[i]); 확인 후 주석
54         let item = itemArr[i];
55
56         value += "<tr>"
57             + "<td>" + item.stationName + "</td>"
58             + "<td>" + item.dataTime + "</td>"
59             + "<td>" + item.khaiValue + "</td>"
60             + "<td>" + item.pm10Value + "</td>"
61             + "<td>" + item.coValue + "</td>"
62             + "<td>" + item.no2Value + "</td>"
63             + "<td>" + item.so2Value + "</td>"
64             + "<td>" + item.o3Value + "</td>"
65             + "</tr>";
66     }
67
68     $("#result1 tbody").html(value);
69
70     }
71 })
72 })
73 })
74 </script>
75
76 </body>
77 </html>

```

- 코드 작성 완료 후

Tomcat restart / 브라우저에서 최종 확인

-> localhost:포트번호/opendata

-> 특정 지역 선택 후 해당 지역 대기오염 정보 버튼 클릭

실시간 대기오염 정보							
지역 : 서울		해당 지역 대기오염 정보					
측정소명	측정일시	통합대기환경수치	미세먼지농도	일산화탄소농도	이산화질소농도	아황산가스농도	오존농도
중구	2023-01-31 18:00	-	-	-	-	-	-
한강대로	2023-01-31 18:00	70	52	0.6	0.038	0.004	0.026
종로구	2023-01-31 18:00	62	44	0.4	0.020	0.004	0.040
청계천로	2023-01-31 18:00	64	47	0.5	0.024	0.004	0.031
종로	2023-01-31 18:00	64	42	0.5	0.029	0.004	0.031
용산구	2023-01-31 18:00	66	49	0.3	0.013	0.003	0.046
광진구	2023-01-31 18:00	58	42	0.3	0.019	0.003	0.035
성동구	2023-01-31 18:00	62	43	0.4	0.025	0.003	0.037
강변북로	2023-01-31 18:00	66	43	0.4	0.035	0.003	0.028
중랑구	2023-01-31 18:00	61	43	0.4	0.032	0.004	0.033

- 브라우저 상에서 위와 같은 결과가 출력되는지 최종확인

- 지역 내용을 바꾸어 시도했을 경우 해당 지역의 정보가 잘 출력되는지 확인

- 해당 내용보다 더 많은 양의 데이터를 출력(기본 10개 항목)하고 싶다면

요청변수(Request Parameter)		
항목명	샘플데이터	
serviceKey	-	공공데이터포털에서 받은 인증키
returnType	xml	xml 또는 json
numOfRows	100	한 페이지 결과 수
pageNo	1	페이지번호
sidoName	서울	시도 이름(전국, 서울, 부산, 대구, 전남, 경북, 경남, 제주, 세종)
ver	1.0	버전별 상세 결과 참고

요청변수 항목 중 numOfRows 내용을 ApiController.java에서 코드 추가 필요

* P49, 30행 url += "&returnType=json"; 다음 줄에 다음 내용 추가

-> url += "&numOfRows=15"; // 15개 항목 출력 시

-> Tomcat restart / 브라우저에서 최종 확인

실시간 대기오염 정보							
지역 : 부산 ▾ 해당 지역 대기오염 정보							
측정소명	측정일시	통합대기환경수치	미세먼지농도	일산화탄소농도	이산화질소농도	아황산가스농도	오존농도
당리동	2023-01-31 18:00	42	13	0.3	0.018	0.002	0.022
부산항	2023-01-31 18:00	64	19	0.2	0.012	0.003	0.047
부산신항	2023-01-31 18:00	54	18	0.2	0.033	0.003	0.024
부산북항	2023-01-31 18:00	51	27	0.2	0.025	0.005	0.025
광복동	2023-01-31 18:00	56	12	0.3	0.014	0.003	0.037
초량동	2023-01-31 18:00	48	15	0.3	0.021	0.002	0.029
태종대	2023-01-31 18:00	63	18	0.2	0.007	0.002	0.045
청학동	2023-01-31 18:00	60	11	0.2	0.009	0.003	0.042
전포동	2023-01-31 18:00	-	19	0.7	0.013	0.003	0.029
온천동	2023-01-31 18:00	66	16	0.4	0.040	0.003	0.016
명장동	2023-01-31 18:00	47	17	-	0.020	-	0.027
대연동	2023-01-31 18:00	50	24	0.4	0.019	0.002	0.030
용호동	2023-01-31 18:00	52	17	0.3	0.016	0.004	0.032
학장동	2023-01-31 18:00	61	31	0.2	0.025	0.002	0.020
덕천동	2023-01-31 18:00	61	25	0.4	0.037	0.003	0.018

- 데이터목록 출력

파일데이터 (37,349건)

보건의료

공공기관

국가중점

미리보기

CSV, XLSX

건강보험심사평가원_전국 병의원 및 약국 현황

요양기관 기본정보, 세부정보, 시설정보, 의료장비정보 등 요양기관별 상세 현황 정보 / 건강보험심사평가원 병원정보서비스+약국정보서비스+의료기관별상세정보서비스 오픈API 데이...

제공기관 건강보험심사평가원 수정일 2022-11-22 조회수 47896 다운로드 7287 주기성 데이터 3 키워드 요양기관, 병원, 약국

다운로드

- 해당 내용 클릭하여 파일데이터 상세 항목으로 이동 -> 우측 상단 다운로드

DATA

공공데이터포털

GO . KR

데이터찾기

국가데이터맵

데이터요청

데이터활용

정보공유

이용안내

목록등록관리시스템

로그아웃

마이페이지

사이트맵

ENGLISH

파일데이터 상세

ZIP

건강보험심사평가원_전국 병의원 및 약국 현황

다운로드

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

0

0

관심

오류신고 및

담당자 문의

파일데이터 정보 메타데이터 다운로드

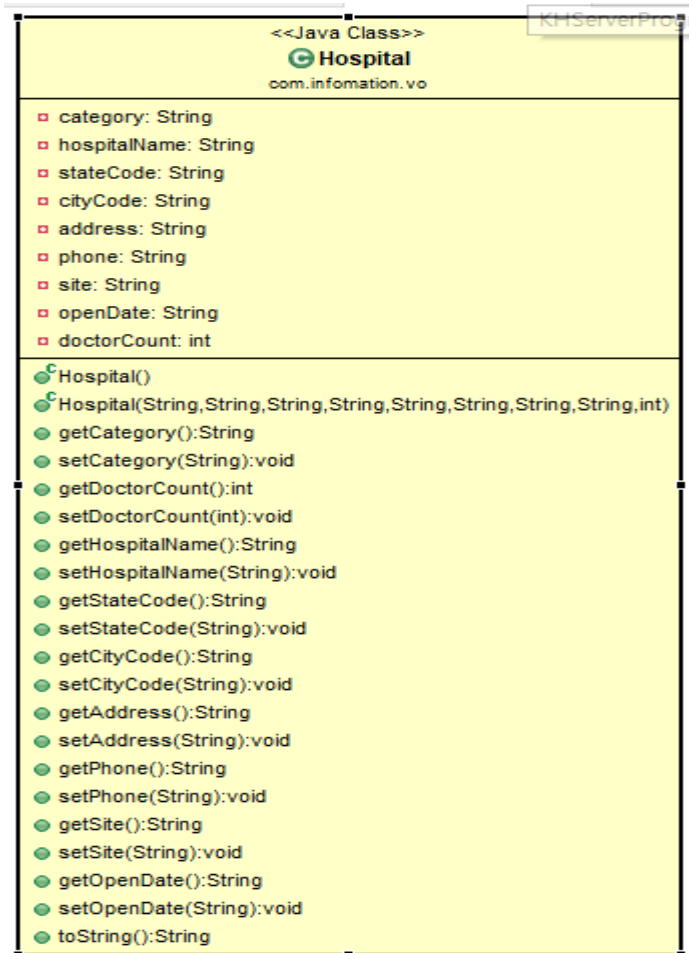
파일데이터명	건강보험심사평가원_전국 병의원 및 약국 현황_20200930		
분류체계	보건 - 건강보험	제공기관	건강보험심사평가원
관리부서명	빅데이터실 빅데이터전략부	관리부서 전화번호	033-739-1047
보유근거	국민건강보험법	수집방법	
업데이트 주기	연간	자기 등록 예정일	2021-10-16
매체유형	텍스트	전체 행	1
확장자	ZIP	다운로드(바로그기)	1745
데이터 한계		키워드	요양기관, 병원, 약국
등록	2020-10-16	수정	2020-10-29
제공형태	공공데이터포털에서 다운로드(원문파일등록)		
설명	요양기관 기본정보, 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 원 말 기은 데이터와 집계상 차이가 있으나 수치를 비교하지 말 것		
비용부과유무	무로	비용부과기준 및 단위	건
이용허락범위	<div>CC BY NC ND</div> <div>OPEN</div> <div>공공저작물_출처표시</div>		

주기성 과거 데이터 (1건)

목록

더보기 >

- 다운로드 시 zip파일 형태이고, 압축해제
 - * 실습은 여러 내용 중 병원정보서비스 이용
- 병원정보서비스 파일을 참조, 데이터에 맞게 자료형 및 변수명을 설정하여 클래스 설계(P60~61, chapter 2 내용 참조)



- 먼저 병원정보를 저장할 클래스인 Hospital.java 구성부터 진행

com.information.vo.Hospital	
01	package com.information.vo;

```

02
03 public class Hospital {
04     private String category;//종별
05     private String hospitalName;//요양기관명
06     private String stateCode;//시도코드명
07     private String cityCode;//시군구코드명
08     private String address;//주소
09     private String phone;//전화번호
10     private String site;//홈페이지주소
11     private String openDate;//개설일
12     private int doctorCount;//총의사수
13
14     public Hospital() {
15     }
16     public Hospital(String category, String hospitalName, String
17 stateCode, String cityCode, String address,
18                     String phone, String site, String openDate, int
19 doctorCount) {
20         super();
21         this.category = category;
22         this.hospitalName = hospitalName;
23         this.stateCode = stateCode;
24         this.cityCode = cityCode;
25         this.address = address;
26         this.phone = phone;
27         this.site = site;
28         this.openDate = openDate;
29         this.doctorCount = doctorCount;
30     }
31     public String getCategory() {
32         return category;
33     }

```

34	public void setCategory(String category) {
35	this.category = category;
36	}
37	public int getDoctorCount() {
38	return doctorCount;
39	}
40	public void setDoctorCount(int doctorCount) {
41	this.doctorCount = doctorCount;
42	}
43	public String getHospitalName() {
44	return hospitalName;
45	}
46	public void setHospitalName(String hospitalName) {
47	this.hospitalName = hospitalName;
48	}
49	public String getStateCode() {
50	return stateCode;
51	}
52	public void setStateCode(String stateCode) {
53	this.stateCode = stateCode;
54	}
55	public String getCityCode() {
56	return cityCode;
57	}
58	public void setCityCode(String cityCode) {
59	this.cityCode = cityCode;
60	}
61	public String getAddress() {
62	return address;
63	}
64	public void setAddress(String address) {
65	this.address = address;

```

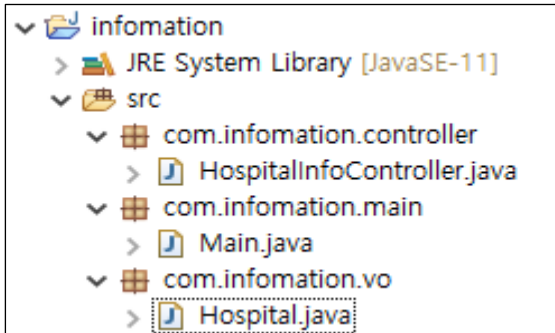
66     }
67     public String getPhone() {
68         return phone;
69     }
70     public void setPhone(String phone) {
71         this.phone = phone;
72     }
73     public String getSite() {
74         return site;
75     }
76     public void setSite(String site) {
77         this.site = site;
78     }
79     public String getOpenDate() {
80         return openDate;
81     }
82     public void setOpenDate(String openDate) {
83         this.openDate = openDate;
84     }
85     @Override
86     public String toString() {
87         return "Hospital [hospitalName=" + hospitalName + ",
88 stateCode=" + stateCode + ", cityCode=" + cityCode
89             + ", address=" + address + ",
90 phone=" + phone + ", site=" + site + ", openDate=" + openDate
91             + ", docktorCount=" + doctorCount
92 + "];"
93     }

```


- * 메모장을 이용해 열기를 하면 각 라인별로 데이터가 들어가고,
데이터를 구분하는 것은 \n이고, 항목별 구분은 ",", 이것을 확인할 수 있음
- * 총 21개의 항목이 존재하지만 실습에서는
종별코드명 / 기관명 / 시도코드명 / 시군구코드명 / 주소 / 전화번호
병원URL / 개설일 / 총의사수 항목 활용 예정

Chapter 3. 공공데이터 저장용 DB 설계 및 구축

- 프로젝트 구조



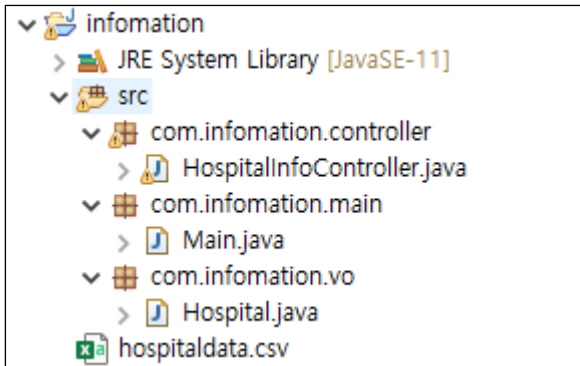
* 3개 패키지 작성

Main.java : 프로젝트 실행 클래스

HospitalInfoController.java : Hospital객체를 관리하는 클래스

Hospital.java : 병원정보 저장 클래스(구성 완료)

* csv파일의 데이터를 편하게 불러오기 위해 프로젝트 최상단에 파일을 저장
파일명은 hospitalData.csv



- Main.java 구성

com.information.main	
01	package com.infomation.main;
02	
03	import com.infomation.controller.HospitalInfoController;
04	
05	public class Main {
06	
07	public static void main(String[] args) {
08	HospitalInfoController controller= new
09	HospitalInfoController();
10	//저장된 list값을 유지하기 위해 변수로 주소값을 가지고 있음
11	controller.inputData();
12	//프로젝트에 저장된 병원정보 <u>csv</u> 파일을 가져와 java클래스에 대입하는
13	매소드(기능)
14	controller.printHospitalData();
15	//대입된 내용을 출력해주는 매소드(기능)
16	}
17	}

- HospitalInfoController.java 구성

com.information.controller.HospitalInfoController	
01	package com.infomation.controller;
02	
03	import java.io.BufferedReader;
04	import java.io.File;
05	import java.io.FileReader;
06	import java.io.IOException;
07	import java.util.ArrayList;
08	import java.util.List;
09	import java.util.StringTokenizer;

```

10  import com.infomation.vo.Hospital;
11
12  public class HospitalInfoController {
13
14      private List<Hospital> list=new ArrayList();
15
16      public void inputData() {
17          //csv파일 불러오기
18          String
19  path=HospitalInfoController.class.getResource("/").getPath();
20          File data=new File(path+"/hospitaldata.csv");
21          StringBuffer sb=new StringBuffer();
22          //문자열로 저장되거있어 Reader를 이용하고 받아오는
23  속도를 향상하기 위해 BufferedReader보조 스트림을 사용.
24          //각데이터를 라인별로 받아서 StringBuffer에 저장하여
25  관리함.
26          try(BufferedReader breader=new BufferedReader(new
27  FileReader(data));){
28              String ch="";
29              while((ch=breader.readLine())!=null) {
30                  sb.append(ch+"\n");
31                  //csv에서 받아온파일을 StringBuffer에 저장
32              }
33
34              }catch(IOException e) {
35                  e.printStackTrace();
36              }
37
38              String[] datas=sb.toString().split("\n");
39              //버퍼에 저장된 내용을 라인별로 나눠거 배열로 저장
40
41              for(String d : datas) {

```

42	
43	String[]
44	detail=d.replace("W","@").replace("W","").split("@");
45	//항목별로 다시 배열로 저장하여 각 데이터 불러오기
46	//csv에서 사용한 ,가 주소를 표시하는 부분에도 있어
47	항목을 구분한 ,를 @로 변경하여 파싱하고, 데이터 부분은 모두 ""이
48	불어있어 이를 제거하는 구문
49	
50	//원하는 항목만 hospital객체에 대입하여 저장하기.
51	int count=0;
52	//csv의 최초라인이 항목명을 가져오고있어 그부분에
53	대한 예외처리 구문
54	try {
55	count=Integer.parseInt(detail[16]);
56	}catch(NumberFormatException e) {
57	
58	}
59	//모든 항목데이터가 배열로 대입이 되어있어,
60	배열의 인덱스로 해당하는 자료를 대입하는 구문
61	list.add(new
62	Hospital(detail[3],detail[1],detail[5],detail[7],detail[10],detail[11],detail[12],d
63	etail[13],count));
64	
65	}
66	list.remove(0);//항목명 부분 삭제
67	}
68	
69	public void printHospitalData() {
70	System.out.printf("%-10s %-20s %-10s %-10s %-30s %-
71	10s %-30s %-10s %-4s\n",
72	"종류","병원명","시도명","시군구명
73	","주소","전화번호","홈페이지주소","개설일","총의사수");

