

IT 게임 영상

**1544-9970**
전국대표문의전화(연중무휴)IT·게임·영상
지점별 위치 찾기

국비지원과정

교육원소개

취업지원센터

프로젝트

커뮤니티

산학연계

마이페이지

상담센터

서동진

(디지털컨버전스)공공데이터 융합 자바개발자 양성과정A30

[강남 H] 2024. 11. 27 ~ 2025. 05. 20 09:00~18:00

김시연 강사 | 이은규 취업담임

남은 시간 11: 55: 11

수강생 평가

* 첨부파일의 확장자를 소문자로 등록하세요. - 예시 : test.jpg(O), test.JPG(X)
* 평가 답안에 윈도우 이모티콘 입력 불가

[비 NCS 교과(실기)] 프로젝트 기반 공공데이터 아키텍처 설계 (문제해결시나리오)

1회차

총점 : 100.0

1. OpenAPI의 이해
공공데이터 API 설계
데이터 배포를 위한 REST 설계
점수 (40.0점)

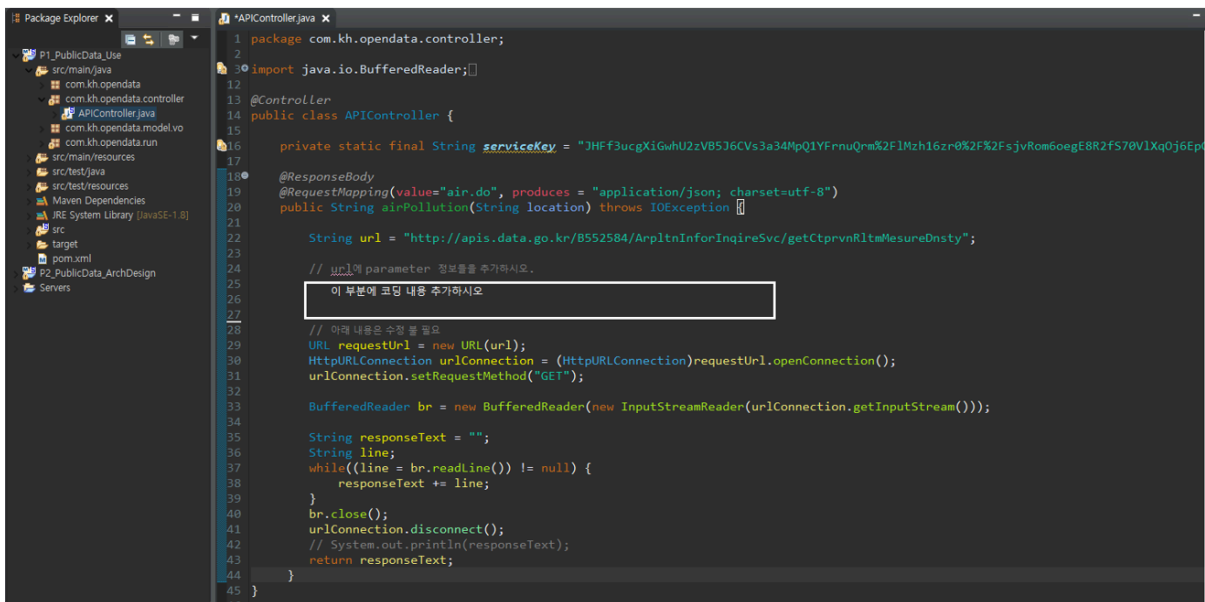
1. 공공데이터포털에서 제공하는 한국환경공단 에어코리아 대기오염정보 중 시도별 실시간 측정정보 조회 서비스 OpenAPI를 사용하기 위해 Controller class인 APIController.java를 구현하고자 한다.

Eclipse내에서 Project 이름 : PublicDataUse,
spring MVC project 생성할 때 기본 package : com.kh.opendata
APIController.java class의 package 이름 : com.kh.opendata.controller

공공 API Base URL : http://apis.data.go.kr/B552584/ArpltnInforInqireSvc/getCtprvnRltmMesureDnsty

이 때 OpenAPI에서 url에 코딩해야 하는 parameter들은 다음과 같다.
- serviceKey : 공공데이터포털에서 회원가입 개인이 승인받은 OpenAPI 인증키
- returnType (json으로 출력결과 받음)
- sidoName : 서울, 부산, 대전

Controller class인 APIController.java의 String url의 기능을 구현하는 내용을 [원인](20점)에, 전체 소스코드를 [조치내용](20점)에 작성하세요. (총 40점)



먼저 PublicDataUse라는 project 생성
 Spring MVC Project 생성 패키지 : com.kh.opendata
 ApiController.java : com.kh.opendata.Controller

수강생 답
 (원인)

```

v PublicDataUse [08_spring-workspace main]
  > .settings
  v > src
    v > main
      v > java
        v > com
          v > kh
            v > opendata
              v > controller
                APIController.java

```

전체 url에 대해 계속 중첩으로 더해주는 구문을 실행시켜야 한다.

url = 메인주소
 1. url += 서비스키
 2. url += 시도이름(여기서는 location이라는 변수에 다음 jsp에서)
 3. url += 리턴타입

이렇게 하면 된다.

답안첨삭

모범답안
 OpenAPI에 서비스 인증키인 serviceKey parameter 정보를 url에 추가한다.
 OpenAPI에 parameter인 sidoName에 airPollution의 argument인 location 정보를 encoding하여 url에 추가한다.
 OpenAPI parameter returnType에 json을 설정하여 url에 추가한다.

수강생 답
 (조치내용)
 1. url += "?serviceKey=" + serviceKey;
 2. url += "&sidoName=" + URLEncoder.encode(location,"utf-8");
 3. url += "&returnType=json";

```

PublicDataUse ▶ src/main/java ▶ com.kh.opendata.controller ▶ APIController ▶ airPollution(String) : String
private static final String serviceKey = "VSgXc34Vs94ijfKVtpeJr85n70S9Xq9bY8PLyw9wCrGAzraqE4Gj6ecSZ88ZSiuSvN3.

@ResponseBody
@RequestMapping(value = "air.do", produces = "application/json; charset=utf-8")
public String airPollution(String location) throws IOException {
    String url = "https://apis.data.go.kr/B552584/ArpltnInforInquireSvc/getCtprvnRltmMesureDnsty";
    url += "?serviceKey=" + serviceKey;
    url += "&sidName=" + URLEncoder.encode(location, "utf-8");
    url += "&returnType=json";

    URL requestUrl = new URL(url);
    HttpURLConnection urlConnection = (HttpURLConnection)requestUrl.openConnection();
    urlConnection.setRequestMethod("GET");

    BufferedReader br = new BufferedReader(new InputStreamReader( urlConnection.getInputStream()));

    String responseText = "";
    String line;
    while((line=br.readLine()) != null) {
        responseText += line;
    }

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

    return responseText;
}

```

그리고 컨트롤러에서 @controller 어노테이션을 절대 빼먹지 않도록 하자..

답안첨삭

APIController.java에서 url에 OpenAPI parameter들을 반영한 전체 source code는 다음과 같다.

모범답안

```

package com.kh.opendata.controller;

import java.io.BufferedReader;

@Controller
public class APIController {

    private static final String serviceKey = "JHfF3ucgXiGwhU2zVB536CVs3a34MpQ1YfrnuQrm%2F1Mzh16zr0%2F%2FsJvRom6oegE8R2fS70V1XqQj6EpD...";

    @ResponseBody
    @RequestMapping(value="air.do", produces = "application/json; charset=utf-8")
    public String airPollution(String location) throws IOException {

        String url = "http://apis.data.go.kr/B552584/ArpltnInforInquireSvc/getCtprvnRltmMesureDnsty";

        // url에 parameter 정보들을 추가함
        url += "?serviceKey=" + serviceKey;
        url += "&sidName=" + URLEncoder.encode(location, "UTF-8");
        url += "&returnType=json";

        // 아래 내용은 수정 불필요
        URL requestUrl = new URL(url);
        HttpURLConnection urlConnection = (HttpURLConnection)requestUrl.openConnection();
        urlConnection.setRequestMethod("GET");

        BufferedReader br = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));

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

```

	성취기준	채점기준
평가항목	31~40	공공데이터 활용간 절차를 아주 정확하게 이해하고, 어플리케이션에 아주 정확하게 적용/구현함.
	11~30	공공데이터 활용간 절차를 거의 이해하지 못하고, 어플리케이션에 어느정도 적용/구현함.
	0~10	공공데이터 활용간 절차를 이해하지 못하고, 어플리케이션에 적용/구현하지 못함.

2.OpenAPI의 이해

공공데이터 API 설계

데이터 배포를 위한 REST 설계

점수 (60.0점)

2. 공공데이터포털에서 제공하는 한국환경공단 에어코리아 대기오염정보 중 시도별 실시간 측정정보 조회 결과를 json으로 받아 index.jsp 파일을 생성하여 index.jsp 파일의 빈칸 부분(for문 본문 내 코딩)을 완성하여 출력결과가 화면에 보이도록 구현하시오.

[고려사항]

- 1번 문제에서 생성한 PublicDataUse 프로젝트와 APIController.java controller를 사용한다.

```
- 공공데이터 포털 사이트에서 측정소명, 일산화탄소농도 등의 영문 항목명은 "에어코리아_대기오염정보 조회 서비스_기술문서_v1.0.docx"를 참조하시오
구현할 내용을 파악해 [원인](30점)에 작성하고, 올바른 구현결과를 [조치내용](30점)에 작성하시오. (총 60점)
작성할 index.jsp 전체코드 :
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script src ="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
</head>
<body>
<h2>실시간 대기오염 정보</h2>
지역 :
<select id = "location">
<option>서울</option>
<option>부산</option>
<option>대전</option>
</select> <!-- 내용이 많아 3개만 option 지정 --%>
<button id = "btn1">해당 지역 대기오염 정보</button>
<br><br>
<table id = "result1" border = "1">
<thead>
<tr>
<th>측정소명</th>
<th>측정일시</th>
<th>통합대기환경수치</th>
<th>미세먼지농도</th>
<th>일산화탄소농도</th>
<th>이산화질소농도</th>
<th>아황산가스농도</th>
<th>오존농도</th>
</tr>
</thead>
<tbody>
</tbody>
</table>
<script>
$(function(){
$("#btn1").click(function(){
$.ajax({
url:"air.do",
data:{location:$("#location").val()},
success:function(data){
const itemArr = data.response.body.items;
let value = "";
for(let i in itemArr){
// for문 안에 있는 내용을 코딩하시오.
}
$("#result1 tbody").html(value);
}
})
})
}
</script>
</body>
</html>
```

사용자가 요청한 값들을
<tr>과 <td> 태그를 동적으로 생성시키면서
<table id = "result1" border = "1"> html 테이블에 심어주는 구문이다.

수강생 답
(원인)

측정소명 : stationName
측정일시 : dateTime
통합대기환경수치: khaiValue24
미세먼지농도: pm25Value
일산화탄소농도: coValue
이산화질소농도:no2Value
아황산가스농도: so2Grade
오존농도 : o3Value

답안첨삭

모범답안

json 문서객체의 복수개의 row태그를 처리할 수 없다. 반복문을 통해 여러 row태그에 접근하고, 그 자식태그의 값을 가져올 수 있어야 한다.
이를 테이블태그의 자식 tr태그로 구성해 추가해야 한다.

수강생 답
(조치내용)

삽입내용은 위의 값들과 일치하는 값을 item에서 꺼내오면 된다.

```
let item = itemArr[i];
value += "<tr>"
      + "<td>" + item.stationName + "</td>"
      + "<td>" + item.dateTime + "</td>"
      + "<td>" + item.khaiValue + "</td>"
      + "<td>" + item.pm10Value + "</td>"
      + "<td>" + item.coValue + "</td>"
      + "<td>" + item.no2Value + "</td>"
      + "<td>" + item.so2Value + "</td>"
      + "<td>" + item.o3Value + "</td>"
      + "</tr>"
```

=====서울=====

실시간 대기오염 정보

지역 : 서울 ▾ 해당 지역 대기오염 정보

측정소명	측정일시	통합대기환경수치	미세먼지농도	일산화탄소농도	이산화질소농도	아황산가스농도	오존농도
강남구	2025-05-02 10:00	60	18	0.3	0.012	0.003	0.041
서초구	2025-05-02 10:00	64	19	0.3	0.009	0.003	0.046
도산대로	2025-05-02 10:00	56	19	0.4	0.011	0.002	0.036
강남대로	2025-05-02 10:00	51	34	0.4	0.015	0.003	0.030
송파구	2025-05-02 10:00	57	28	0.3	0.013	0.003	0.037
강동구	2025-05-02 10:00	64	19	0.4	0.010	0.003	0.046
천호대로	2025-05-02 10:00	59	12	0.4	0.013	0.002	0.040
금천구	2025-05-02 10:00	63	19	0.2	0.011	0.003	0.044
시흥대로	2025-05-02 10:00	58	20	0.3	0.018	0.003	0.039
강북구	2025-05-02 10:00	65	15	0.3	0.006	0.002	0.048

=====부산=====

실시간 대기오염 정보

지역 : 부산 ▾ 해당 지역 대기오염 정보

측정소명	측정일시	통합대기환경수치	미세먼지농도	일산화탄소농도	이산화질소농도	아황산가스농도	오존농도
광복동	2025-05-02 10:00	64	23	0.3	0.008	0.003	0.047
초량동	2025-05-02 10:00	68	19	0.2	0.009	0.003	0.051
태종대	2025-05-02 10:00	55	41	0.2	0.009	0.002	0.035
청학동	2025-05-02 10:00	57	23	0.2	0.008	0.002	0.037
전포동	2025-05-02 10:00	64	17	0.2	0.003	0.002	0.046
운천동	2025-05-02 10:00	56	35	0.2	0.018	0.002	0.035
명장동	2025-05-02 10:00	65	15	0.3	0.008	0.002	0.047
대연동	2025-05-02 10:00	62	34	0.4	0.009	0.002	0.043
용호동	2025-05-02 10:00	66	22	0.3	0.009	0.003	0.049
학장동	2025-05-02 10:00	54	27	0.3	0.014	0.005	0.034

=====대전=====

실시간 대기오염 정보

지역 :

대전

 해당 지역 대기오염 정보

측정소명	측정일시	통합대기환경수치	미세먼지농도	일산화탄소농도	이산화질소농도	아황산가스농도	오존농도
읍내동	2025-05-02 10:00	57	21	0.4	0.012	0.002	0.038
문평동	2025-05-02 10:00	64	20	0.3	0.007	0.003	0.047
문창동	2025-05-02 10:00	65	23	0.4	0.010	0.003	0.047
구성동	2025-05-02 10:00	66	19	0.4	0.006	0.005	0.048
노은동	2025-05-02 10:00	67	21	0.3	0.005	0.003	0.050
상대동(대전)	2025-05-02 10:00	65	18	0.3	0.006	0.002	0.048
관평동	2025-05-02 10:00	61	17	0.4	0.004	0.002	0.043
지족동	2025-05-02 10:00	64	29	0.4	0.008	0.003	0.046
대흥동1	2025-05-02 10:00	59	37	0.3	0.018	0.003	0.030
성남동1	2025-05-02 10:00	65	26	0.3	0.014	0.002	0.047

답안철삭

모범답안

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script src ="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
</head>
<body>
<h2>실시간 대기오염 정보</h2>
지역 :
<select id = "location">
<option>서울</option>
<option>부산</option>
<option>대전</option>
</select> <!-- 내용이 많아 3개만 option 지정 --%>
<button id = "btn1">해당 지역 대기오염 정보</button>
<br><br>
<table id = "result1" border = "1">
<thead>
<tr>
<th>측정소명</th>
<th>측정일시</th>
<th>통합대기환경수치</th>
<th>미세먼지농도</th>
<th>일산화탄소농도</th>
<th>이산화질소농도</th>
<th>아황산가스농도</th>
<th>오존농도</th>
</tr>
</thead>
<tbody>
</tbody>
</table>
<script>
$(function(){
$("#btn1").click(function(){
$.ajax({
url:"air.do",
data:{location:$("#location").val()},
success:function(data){
const itemArr = data.response.body.items;
let value = "";
for(let i in itemArr){
// console.log(itemArr[i]); 확인 후 주석
let item = itemArr[i];
value += "<tr>"
+ "<td>" + item.stationName + "</td>"
+ "<td>" + item.dateTime + "</td>"
+ "<td>" + item.khaiValue + "</td>"
+ "<td>" + item.pm10Value + "</td>"
```

```
+ "<td>" + item.coValue + "</td>"
+ "<td>" + item.no2Value + "</td>"
+ "<td>" + item.so2Value + "</td>"
+ "<td>" + item.o3Value + "</td>"
+ "</tr>";
}
$("#result1 tbody").html(value);
}
})
})
})
</script>
</body>
</html>
```

	성취기준	채점기준
평가항목	51~60	공공데이터 활용간 절차를 아주 정확하게 이해하고, 어플리케이션에 아주 정확하게 적용/구현함.
	11~50	공공데이터 활용간 절차를 거의 이해하지 못하고, 어플리케이션에 어느정도 적용/구현함.
	0~10	공공데이터 활용간 절차를 이해하지 못하고, 어플리케이션에 적용/구현하지 못함.

평가 메인으로