

김재영

(디지털컨버전스)공공데이터 융합 자바개발자 양성과정A30

[강남 H] 2023. 11. 20 ~ 2024. 05. 02 09:00~18:00

김시연 강사 | 이은정 취업담당

남은 시간 00: 45: 26

수강생 평가

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

다음과 같은 상황이 발생하는 원인과 조치내용에 대해서 보고서 작성 후 제출하시오.

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

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/ArpltnInforInquireSvc/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)

```
1 package com.kh.opendata.controller;
2
3 import java.io.BufferedReader;
4
5 @Controller
6 public class ApiController {
7
8     private static final String serviceKey = "3Hf3ucgXiGwHJ2rVB536CVs3a34PpQ1VfrnuQrW2F1Mh16zr0N2T%2FsjvRw6oegE8R2f570V1Xq0j6Epl";
9
10    @ResponseBody
11    @RequestMapping(value="air.do", produces="application/json; charset=utf-8")
12    public String airPollution(String location) throws IOException {
13
14        String url = "http://apis.data.go.kr/B552584/ArpltnInforInquireSvc/getCtprvnRltmMesureDnsty";
15
16        // url에 parameter 정보들을 추가하시오.
17        // 이 부분의 코딩 내용 추가하시오
18
19        // 이의 내용을 주입 할 필요
20        URL requestUrl = new URL(url);
21        HttpURLConnection urlConnection = (HttpURLConnection)requestUrl.openConnection();
22        urlConnection.setRequestMethod("GET");
23
24        BufferedReader br = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));
25
26        String responseText = "";
27        String line;
28        while((line = br.readLine()) != null) {
29            responseText += line;
30        }
31        br.close();
32        urlConnection.disconnect();
33        // System.out.println(responseText);
34        return responseText;
35    }
36 }
```

IT 게임 영상

1544-9970

전국대표문의전화(연중무휴)



IT·게임·영상

지점별 위치 찾기

필수 매개변수 (serviceKey, sidoName 등) 를 포함하여, url을 완성시키는 조치가 필요하다.

매개변수의 확인과 관련된 정보는 링크의 공식문서를 통해 확인이 가능하다.

조치내용 [EDIT](#)

조치. url 완성

필수 매개변수인 serviceKey와 sidoName이 필요하며, 요구사항에서 제시된 대로 return type을 json으로 명시한다. 주의할 사항으로는 sidoName 은 한글이기에 인코딩이 필요하다는 점이다.

[전체코드]

```
public static final String serviceKey = "a53t3HEZpk6x58lQICFNhQSjKd29BFTqVXBEubF%2Fs

@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/getCtprvnRltm

    // url에 parameter 정보들을 추가
    url += "?serviceKey=" + serviceKey;
    url += "&sidoName=" + URLEncoder.encode(location, "utf-8");
    url += "&returnType=json";

    // 수정 불 필요 영역
    URL requestUrl = new URL(url);
    HttpURLConnection urlConnection = (HttpURLConnection)requestUrl.openConnection()
    urlConnection.setRequestMethod("GET");

    BufferedReader br = new BufferedReader(new InputStreamReader(urlConnection.getIn

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

    return responseText;
}
```

```
package com.kh.opendata.controller;
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
```

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
```

```
@Controller
```

```
public class AirPollutionController {
```

```
    public static final String serviceKey = "a53t3HEZpk6x58lQICFNhQSjKd29BFTqVXBEubF%2FsQet6BotMcCJddwmyUUjllCq2rHsborCNqanqKFUuOkNDw%3D%3
    @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/getCtprvnRltmMeasureDnsty";
        // url에 parameter 정보들을 추가
        url += "?serviceKey=" + serviceKey;
        url += "&sidoName=" + URLEncoder.encode(location, "utf-8");
```

```

url += "&returnType=json";

// 수정 불 필요 영역
URL requestUrl = new URL(url);
URLConnection urlConnection = (URLConnection)requestUrl.openConnection();
urlConnection.setRequestMethod("GET");

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

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

```

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 화면에 보여지도록 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>
```

원인 [EDIT](#)

원인. 구현 과정

select 박스를 통해 선택한 값이 넘어오면, 해당 value 값을 ajax 를 통해 controller 로 넘겨주고, 이를 통해 조회된 데이터의 item에 접근하여 반복문을 통해 화면에 나타낼 수 있도록 처리해준다.

위 과정을 위해, 측정소명 ~ 오존농도 까지의 영문명을 공식 홈페이지 링크 혹은 기술문서를 통해 확인 가능하다.

조치내용 [EDIT](#)

조치. index.jsp 완성

[버튼 클릭 결과]

실시간 대기오염 정보

지역 : 서울 [해당 지역 대기오염 정보](#)

측정소명	측정일시	통합대기환경수치	미세먼지농도	일산화탄소농도	이산화질소농도	아황산가스농도	오존농도
중구	2024-04-17 10:00	132	69	0.6	0.032	0.004	0.019
한강대로	2024-04-17 10:00	-	108	-	-	-	-
종로구	2024-04-17 10:00	189	84	0.7	0.027	0.003	0.021
청계천로	2024-04-17 10:00	188	94	0.8	0.031	0.003	0.017
종로	2024-04-17 10:00	101	83	0.8	0.029	0.004	0.016
용산구	2024-04-17 10:00	164	92	0.7	0.032	0.003	0.016
광진구	2024-04-17 10:00	179	92	0.7	0.033	0.003	0.018
성동구	2024-04-17 10:00	175	89	0.6	0.030	0.003	0.015
강변북로	2024-04-17 10:00	216	118	0.9	0.037	0.003	0.008
중랑구	2024-04-17 10:00	124	76	0.7	0.025	0.002	0.020

[전체코드]

```
$.ajax({
  url:"air.do",
  data:{
    location:$("#location").val()
  },
  success:function(data){
    console.log(data.response.body);
    const itemArr = data.response.body.items;
    let value = "";
    for(let i in itemArr){
      // for문 안에 있는 내용을 코딩하시오.
      let item = itemArr[i];
      value += "<tr>"
      + "<td>" + item.stationName + "</td>"
      + "<td>" + item.dataTime + "</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);
  }
})
```

```
<%@ 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(){
  $("#btn1").click(function(){
    console.log($("#location").val());
    $.ajax({
      url:"air.do",
      data:{
        location:$("#location").val()
      },
      success:function(data){
        console.log(data.response.body);
      }
    });
  });
});
```

```
const itemArr = data.response.body.items;
let value = "";
for(let i in itemArr){
  // for문 안에 있는 내용을 코딩하시오.
  let item = itemArr[i];
  value += "<tr>"
    + "<td>" + item.stationName + "</td>"
    + "<td>" + item.dataTime + "</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>
```

제출하기