

## 단계별 여행경보 API

<https://www.data.go.kr/data/15025430/openapi.do>

—

1. [api신청 키 발급](#)
2. [제공되는 sample코드 실행해 보기](#)
3. [제공되는 소스를 실행하고 api결과를 보면서 어떻게 데이터를 사용할지 고민함](#)
4. [소스는 압축파일형태로 제공됨](#)
5. [service 클래스 코드 설명, 주석부분 확인](#)

```
package com.FinalProject.controller;
```

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import org.json.JSONArray;
import org.json.JSONObject;
import org.json.XML;
import org.springframework.stereotype.Component;
```

```
@Component
public class TravelWarningService {
```

메서드명: **travelInfo()**  
반환값: Map정보  
(여행경보별 국가정보객체를 담은 리스트를 담은 Map객체)

```
    public Map<String, Object> travelInfo() throws Exception {
```

```
        StringBuilder urlBuilder = new StringBuilder("http://apis.data.go.kr/1262000/TravelWarningService/getTravelWarningList");    /*URL*/
        urlBuilder.append("? " + URLEncoder.encode("serviceKey", "UTF-8") +
            "=Nwx14kKv7ybp5MV%2BRxteAhgHCTABREF6CpLmwMO3tf942PIQmVEuemOSDtKJDbfK5RxPB9Ubm0yaAD3WJ1e7A%3D%3D"); /*Service Key*/
        urlBuilder.append("& " + URLEncoder.encode("numOfRows", "UTF-8") + " = " + URLEncoder.encode("300", "UTF-8")); /*한 페이지 결과 수*/
```

```
        URL url = new URL(urlBuilder.toString());
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
```

```
        //요청헤더정보 담기
```

```
        conn.setRequestMethod("GET");
        conn.setRequestProperty("Content-type", "application/json");
```

```
        // 응답결과상태확인하기
```

```
        //System.out.println("Response code: " + conn.getResponseCode());
```

```
        //응답스트림얻어오기, 보조스트림의 생성자 매개변수로 넘겨줌
```

```
        //보조스트림을 통해서 편리하게 읽어 옴
```

```
        BufferedReader rd;
```

```
        if(conn.getResponseCode() >= 200 && conn.getResponseCode() <= 300) { // 서버가 준 데이터가 정상인지 확인
```

```
            rd = new BufferedReader(new InputStreamReader(conn.getInputStream()));
```

```
        } else {
```

```
            rd = new BufferedReader(new InputStreamReader(conn.getErrorStream()));
```

```
        }
```

```
        //반복문을 통해 응답데이터 하나의 string으로 만들기
```

```
        // StringBuilder는 문자열을 연결할 때 사용하는 객체이다, StringBuffer랑 같은 역할을 함
```

```
        StringBuilder sb = new StringBuilder();
```

```
        String line;
```

```
        while ((line = rd.readLine()) != null) {
```

```
            sb.append(line);
```

```
        }
```

```
        rd.close();
```

```
        conn.disconnect();
```

```
//1. sb에 요청에 대한 결과가 담긴 변수
//System.out.println(sb.toString()); 1차 결과 눈으로 확인
```

```
//2. xml을 다루기 쉬운 JSON으로 변환
```

```
JSONObject jsonObj = XML.toJSONObject(sb.toString());
```

```
// System.out.println( jsonObj ); => 이 결과를 출력해 보고 데이터를 어떻게 꺼낼지 고민해본다
```

API로부터 받은 결과를 내가 원하는 방식으로 담기

```
/* { response:{ body: { items: { item : [ {국가정보=>원하는 정보}, {국가여행정보==>원하는 정보} , ...] }}}
이런식으로 데이터를 받음 */
```

**주의사항: items꺼내기 위해 json 객체로 부터 여러번을 통해 꺼내는 작업을 해야 한다. \*/**

```
JSONObject response = jsonObj.getJSONObject("response"); //1. response 키로 object 꺼내기
JSONObject body = response.getJSONObject("body"); //2. body키로 object 꺼내기
JSONObject items = body.getJSONObject("items"); //3. items키로 object 꺼내기
JSONArray item = items.getJSONArray("item"); //4. item키로 objectArray 꺼내기
```

```
System.out.println( "국가여행정보 건 수=" + item.length());
```

```
System.out.println( "원하는 국가 정보 리스트 확인 함");
```

```
for(int i = 0; i < item.length(); i++) {
    System.out.println(item.get(i));
}
```

```
//각 여행경로별 국가정보를 담은 list 준비 , 4개
```

```
List<NavyDto> navyDtoList = new ArrayList<NavyDto>();
List<YellowDto> yellowDtoList = new ArrayList<YellowDto>();
List<RedDto> redDtoList = new ArrayList<RedDto>();
List<BlackDto> blackDtoList = new ArrayList<BlackDto>();
```

```
JSONObject obj;
```

```
for(int i = 0; i < item.length(); i++) {
```

```
    //여행가능지역정보별 담기 ( attentionNote, controlNote, LimitaNote , 그 밖 )
```

```
    obj = (JSONObject)item.get(i);
```

```
    //System.out.println(obj.optString("continent"));
```

```
    if(! obj.optString("attentionNote").equals("")){
```

```
        String continent = obj.optString("continent");
```

```
        String countryName = obj.optString("countryName");
```

```
        String attentionNote = obj.optString("attentionNote");
```

```
        NavyDto navyDto = new NavyDto(continent, countryName, attentionNote);
```

```
        // attentionNote 국가만 담기
```

```
        navyDtoList.add(navyDto);
```

```
    }
```

```
    else if( !obj.optString("controlNote").equals("") ) {
```

```
        String continent = obj.optString("continent");
```

```
        String countryName = obj.optString("countryName");
```

```
        String controlNote = obj.optString("controlNote");
```

```
        YellowDto yellowDto = new YellowDto(continent, countryName, controlNote);
```

```
        // controlNote국가만 담기
```

```
        yellowDtoList.add(yellowDto);
```

```

    }
    else if( !obj.optString("limitaNote").equals("")) {
        String continent = obj.optString("continent");
        String countryName = obj.optString("countryName");
        String limitaNote = obj.optString("limitaNote");
        RedDto redDto = new RedDto(continent, countryName, limitaNote);

        //limitaNote 국가만 담기
        redDtoList.add(redDto);
    }
    else {

        String continent = obj.optString("continent");
        String countryName = obj.optString("countryName");
        BlackDto blackDto = new BlackDto(continent, countryName);
        //그밖의 국가만 담기
        blackDtoList.add(blackDto);
    }
}

```

```

// 각 경보단계별 국가 정보가 잘 담겼는지 확인하기
System.out.println(navyDtoList);
System.out.println("=====");
System.out.println(yellowDtoList);
System.out.println("=====");
System.out.println(redDtoList);
System.out.println("=====");
System.out.println(blackDtoList);
System.out.println("=====");

```

```

Map<String, Object> map = new HashMap<String, Object>();

```

```

// 각 경고단계별 국가정보 arrayList를 map 에 다 담는다.
map.put("navy", navyDtoList);
map.put("yellow", yellowDtoList);
map.put("red", redDtoList);
map.put("black", blackDtoList);

```

```

return map;

```

```

}

```

// 서비스 클래스 제공메서드 **travellInfo()** 실행 결과 확인을 위해서 main메서드 추가

```

public static void main(String[] args ) throws Exception {
    TravelWarningService s = new TravelWarningService();
    Map<String, Object > result = s.travellInfo();

    System.out.println("실행결과");
    System.out.println( result);

```

```

}

```

```

}

```