

기상청 API를 활용한 온/습도 예보 조회 과제
개발 완료 보고서

2021.08.06

양승한

1. 개요

○ 과제 기획 의도

원하는 날짜의 온/습도 예보데이터를 조회할 때 데이터를 시각화 하여
간편하고 쉽게 날씨 예보를 볼 수 있는 웹을 만드는 것.

○ 과제 목표

기상청 API를 활용해 6시간 간격으로 요청하여 받아온 온/습도 예보데이터를
DB에 저장한 뒤 저장된 데이터를 조회하여 JqGrid와 Google Chart를 활용,
Ajax 적용하여 페이지 전환없이 비동기 방식 조회 기능을 구현하고
Excel파일로 다운로드 기능 및 JqGrid 셀 선택을 통해 값 수정/저장 기능을 구현한다.

○ 개발 소요 기간

2021-07-01 ~ 2021-08-06 (약 1개월)

○ 개발 환경

개발 언어 – Java

개발 도구 – Eclipse, SSMS, VS code

DB – MSSQL

WAS – Apache Tomcat v9.0

프레임워크 – Spring framework, Mybatis

빌드 관리 – Maven

데이터 시각화 – JqGrid, Google Chart

○ DB 테이블 구조

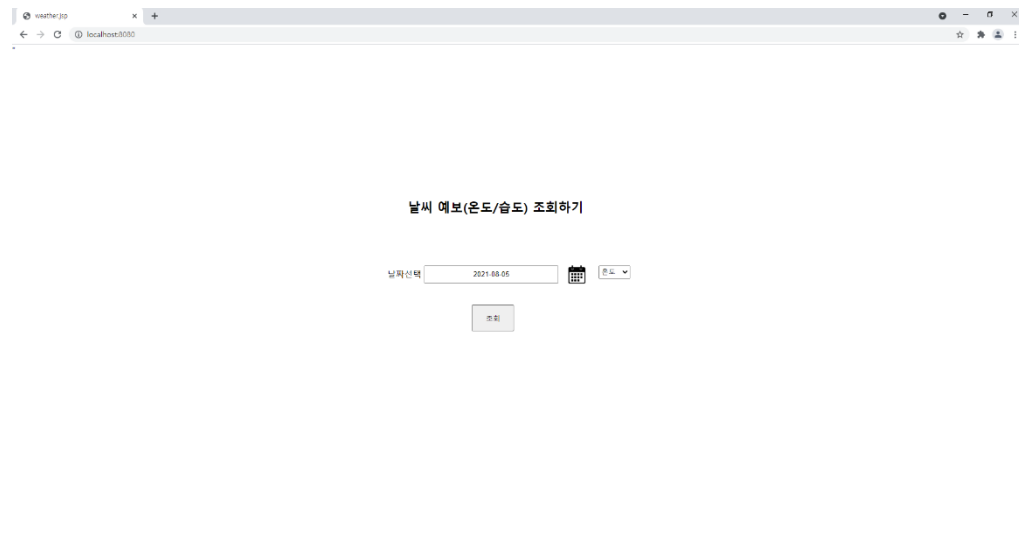


dbo.dbo.WeatherHourlyData	
PK	dataType
sid	(PK, int, Null이 아님)
dataType	(int, Null)
inputdatetime	(varchar(50), Null)
T00	(decimal(16,4), Null)
T01	(decimal(16,4), Null)
T02	(decimal(16,4), Null)
T03	(decimal(16,4), Null)
T04	(decimal(16,4), Null)
T05	(decimal(16,4), Null)
T06	(decimal(16,4), Null)
T07	(decimal(16,4), Null)
T08	(decimal(16,4), Null)
T09	(decimal(16,4), Null)
T10	(decimal(16,4), Null)
T11	(decimal(16,4), Null)
T12	(decimal(16,4), Null)
T13	(decimal(16,4), Null)
T14	(decimal(16,4), Null)
T15	(decimal(16,4), Null)
T16	(decimal(16,4), Null)
T17	(decimal(16,4), Null)
T18	(decimal(16,4), Null)
T19	(decimal(16,4), Null)
T20	(decimal(16,4), Null)
T21	(decimal(16,4), Null)
T22	(decimal(16,4), Null)
T23	(decimal(16,4), Null)

2. 화면 및 기능

- 구현 화면

○ 메인 화면



○ 날짜 선택 창

날씨 예보(온도/습도) 조회하기

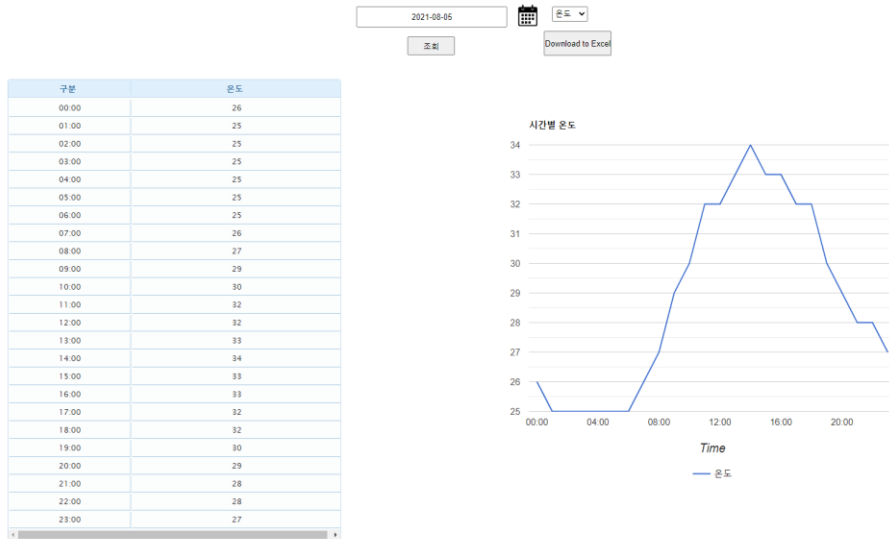


○ 온/습도 카테고리 선택 창

날씨 예보(온도/습도) 조회하기



○ 데이터 조회 화면



- 구현 기능

○ Spring Scheduler를 이용해 6시간 간격 예보데이터 요청 실행 및 요청 URL 설정 코드

```
// serviceKey는 기상청에서 신청시 발급되고 신청마다 고유 키값이 발급됨
// 위도와 경도는 본당시 구미동으로 고정 하기위해 final static을 사용
final static String serviceKey = "WBnK0gPLv12z8n%2FBujlX6vtsLzzZG1BJYhNB6MvacORuj8GAQy5jKOh5HbgitXV6mvVB02U0xY9PyR1CYxGuw%3D%3D";
final static String nx = "62";
final static String ny = "122";
static List<JSONObject> reh = new ArrayList<>();
static List<JSONObject> tmp = new ArrayList<>();

static List<JSONObject> result2 = new ArrayList<>();

// 기상청 API에 예보데이터를 요청하여 받아와 DB에 저장하는 Controller 클래스
// 매일 6시간 간격으로 실행되는 스케줄러
@Scheduled(cron = "0 0 05,11,17,23 * * *")
@RequestMapping("/weather2")
public void restApiWeather() throws Exception {

    SimpleDateFormat format = new SimpleDateFormat("yyyyMMdd");
    SimpleDateFormat format2 = new SimpleDateFormat("HH");

    Date now = new Date();

    String date2 = "20210805";
    String time2 = "0800";
    String date1 = format.format(now); // 발표 날짜
    String time1 = format2.format(now) + "00"; // 발표 시간
    String type = "json"; // 데이터 타입
    String pageNo = "1"; // 페이지 수
    String numOfRows = "1000"; // 한페이지에 표시할 데이터

    WeatherDTO dto = new WeatherDTO();

    String apiURL = "http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getVilageFcst";

    StringBuilder urlBuilder = new StringBuilder(apiURL);
    urlBuilder.append("?") + URLEncoder.encode("ServiceKey", "UTF-8") + "=" + serviceKey;
    urlBuilder.append("&") + URLEncoder.encode("numOfRows", "UTF-8") + "=" + numofRows;
    urlBuilder.append("&") + URLEncoder.encode("pageNo", "UTF-8") + "=" + pageNo;
    urlBuilder.append("&") + URLEncoder.encode("nx", "UTF-8") + "=" + URLEncoder.encode(nx, "UTF-8");
    urlBuilder.append("&") + URLEncoder.encode("ny", "UTF-8") + "=" + URLEncoder.encode(ny, "UTF-8");
    urlBuilder.append("&") + URLEncoder.encode("base_date", "UTF-8") + "=" + URLEncoder.encode(date1, "UTF-8");
    urlBuilder.append("&") + URLEncoder.encode("base_time", "UTF-8") + "=" + URLEncoder.encode(time1, "UTF-8");
    urlBuilder.append("&") + URLEncoder.encode("dataType", "UTF-8") + "=" + URLEncoder.encode(type, "UTF-8");
}
```

○ 날씨 정보 수정

구분	온도
00:00	26
01:00	<input type="text" value="0"/>
02:00	0



localhost:8080 내용:
데이터가 변경되었습니다.

확인

구분	온도
00:00	26
01:00	27
02:00	0
03:00	0

시간별 온도

30



구분	온도
00:00	26
01:00	27
02:00	0

○ Excel 파일로 다운로드 (Download to Excel 버튼 클릭시)

localhost:8080 내용:
현재 조회된 데이터를 Excel 파일로 다운로드 합니다.

확인



2021-08-05 온도....xlsx

Xlsx 확장자로 다운로드