

공공데이터 활용하기

<https://www.data.go.kr>

데이터목록

기상청_단기예보

Q

연관

제공기관별 검색

상세검색

"기상청_단기예보"에 대해 총 639건이 검색되었습니다.

조건검색

초기화

분류체계	서비스유형	제공기관유형	태그	확장자
------	-------	--------	----	-----

국가중점데이터 분류 조건 추가하기 +

조건열기

전체(639건)	파일데이터(516건)	오픈 API(123건)	표준데이터셋0개(0건)
----------	-------------	--------------	--------------

정확도순

10개씩

오픈 API (123건)

과학기술국기행경기관

XMLJSON

기상청_단기예보 ((구)_동네예보) 조회서비스

초단기실황, 초단기예보, 단기((구)동네)예보, 예보버전 정보를 조회하는 서비스입니다.

제공기관 기상청 수정일 2022-03-31 조회수 49123 활용신청 7890 키워드 단기예보,초단기실황,초단기예보

의견수렴
게시판

활용신청

OpenAPI 개발계정 신청

JSON+XML

기상청_단기예보 ((구)_동네예보) 조회서비스

제공기관	기상청	서비스유형	REST
심의여부	자동승인	신청유형	개발계정 활용신청
처리상태	신청	활용기간	승인일로부터 24개월 간 활용가능

공공데이터 제공제도

- * 공공데이터중 위치정보를 포함한 서비스를 사용하고자 하는 사업자는 '위치정보의 보호 및 이용 등에 관한 법률'에 따라 방송통신위원회에 '위치정보서비스 허가'를 받거나 '위치기반 서비스사업 신고'를 하여야 합니다.
- * 이에 해당하는 사업자인 경우에는 첨부파일에 '위치기반서비스사업신고필증'을 첨부해 주시기 바랍니다.
- * 활용신청 시 '위치기반서비스사업신고필증'이 등록되지 않으면 반려가 될 수 있으니 참고 하시기 바랍니다.

활용목적 선택

*표시는 필수 입력항목입니다.

*활용목적

- ☒ 웹 사이트 개발
- ☐ 앱개발 (모바일,솔루션등)
- ☐ 기타
- ☐ 참고자료
- ☐ 연구(논문 등)

개발계정

<div>신청 0건 ></div> <div>신청중인 단계</div> <div><div>· 보류0건</div><div>· 반려0건</div></div>	<div>활용 1건 ></div> <div>승인되어 활용중인 단계</div> <div><div>· 변경신청0건</div></div>	<div>중지0건 ></div> <div>중지신청하여 운영이 중지된 단계</div>
--	--	---

상세검색 열기 ▾

총1건

과학기술

기상청

활용신청

[승인] 기상청_단기예보 ((구)_동네예보) 조회서비스

신청일

2022-05-01

만료예정일

2024-05-01

기본정보

데이터명	기상청_단기예보 ((구)_동네예보) 조회서비스 상세설명		
서비스유형	REST	심의여부	자동승인
신청유형	개발계정 활용신청	처리상태	승인
활용기간	2022-05-01 ~ 2024-05-01		

서비스정보

참고문서	기상청41 단기예보 조회서비스 오픈API활용가이드 최종.zip
데이터포맷	JSON+XML
End Point	http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0

API 환경 또는 API 호출 조건에 따라 인증키가 적용되는 방식이 다를 수 있습니다.

포털에서 제공되는 **Encoding/Decoding** 된 인증키를 적용하면서 구동되는 키를 사용하시기 바랍니다.

* 향후 포털에서 더 명확한 정보를 제공하기 위해 노력하겠습니다.

일반 인증키 (Encoding)	J5%2FThyhm4DvmlWk5EqTakaX7ebIJRjlo7NGylmQ2DeA8qutXiaW6K2mlk9yKXgZMm%2BleKw%2FFt9QQEZ7UsvP3lg% D%3D
일반 인증키 (Decoding)	J5/Thyhm4DvmlWk5EqTakaX7ebIJRjlo7NGylmQ2DeA8qutXiaW6K2mlk9yKXgZMm+leKw/Ft9QQEZ7UsvP3lg==

```
<response>
  <header>
    <resultCode>00</resultCode>
    <resultMsg>NORMAL_SERVICE</resultMsg>
  </header>
  <body>
    <dataType>XML</dataType>
    <items>
      <item>
        <baseDate>20220501</baseDate>
        <baseTime>0600</baseTime>
        <category>PTY</category>
        <nx>55</nx>
        <ny>127</ny>
        <obsrValue>0</obsrValue>
      </item>
      <item>
        <baseDate>20220501</baseDate>
        <baseTime>0600</baseTime>
        <category>REH</category>
        <nx>55</nx>
        <ny>127</ny>
        <obsrValue>79</obsrValue>
      </item>
      <item>
        <baseDate>20220501</baseDate>
        <baseTime>0600</baseTime>
        <category>RN1</category>
        <nx>55</nx>
        <ny>127</ny>
        <obsrValue>0</obsrValue>
      </item>
      <item>
        <baseDate>20220501</baseDate>
        <baseTime>0600</baseTime>
        <category>T1H</category>
        <nx>55</nx>
        <ny>127</ny>
        <obsrValue>8.9</obsrValue>
      </item>
      <item>
        <baseDate>20220501</baseDate>
        <baseTime>0600</baseTime>
        <category>UUU</category>
        <nx>55</nx>
        <ny>127</ny>
        <obsrValue>0</obsrValue>
      </item>
    </items>
  </body>
</response>
```

```
<item>
<baseDate>20220501</baseDate>
<baseTime>0600</baseTime>
<category>T1H</category>
<nx>68</nx>
<ny>101</ny>
<obsrValue>8.5</obsrValue>
</item>
```

xml data format

```
{"response":{"header":{"resultCode":"00","resultMsg":"NORMAL_SERVICE"},"body":{"dataType":"JSON",
"items":{"item":[{"baseDate":"20220501","baseTime":"0600","category":"PTY","nx":68,"ny":101,"obsrValue":"0"},
{"baseDate":"20220501","baseTime":"0600","category":"REH","nx":68,"ny":101,"obsrValue":"84"},
{"baseDate":"20220501","baseTime":"0600","category":"RN1","nx":68,"ny":101,"obsrValue":"0"},
{"baseDate":"20220501","baseTime":"0600","category":"T1H","nx":68,"ny":101,"obsrValue":"8.5"},
{"baseDate":"20220501","baseTime":"0600","category":"UUU","nx":68,"ny":101,"obsrValue":"0.4"},
{"baseDate":"20220501","baseTime":"0600","category":"VEC","nx":68,"ny":101,"obsrValue":"205"},
{"baseDate":"20220501","baseTime":"0600","category":"VVV","nx":68,"ny":101,"obsrValue":"0.9"},
{"baseDate":"20220501","baseTime":"0600","category":"WSD","nx":68,"ny":101,"obsrValue":"1"}]}},"pageNo":1,"numOfRows":1000,"totalCount":8}}}
```

Json data format

```
{
  "response": {
    "header": {
      "resultCode": "00",
      "resultMsg": "NORMAL_SERVICE"
    },
    "body": {
      "dataType": "JSON",
      "items": {
        "item": [
          {
            "baseDate": "20220501",
            "baseTime": "0600",
            "category": "PTY",
            "nx": 55,
            "ny": 127,
            "obsrValue": "0"
          },
          {
            "baseDate": "20220501",
            "baseTime": "0600",
            "category": "REH",
            "nx": 55,
            "ny": 127,
            "obsrValue": "79"
          },
          {
            "baseDate": "20220501",
            "baseTime": "0600",
            "category": "RN1",
            "nx": 55,
            "ny": 127,
            "obsrValue": "0"
          },
          {
            "baseDate": "20220501",
            "baseTime": "0600",
            "category": "T1H",
            "nx": 55,
            "ny": 127,
            "obsrValue": "8.9"
          },
          {
            "baseDate": "20220501",
            "baseTime": "0600",
            "category": "UUU",
            "nx": 55,
            "ny": 127,
            "obsrValue": "0"
          },
          {
            "baseDate": "20220501",
            "baseTime": "0600",
            "category": "VEC",
            "nx": 55,
            "ny": 127,
            "obsrValue": "107"
          },
          {
            "baseDate": "20220501",
            "baseTime": "0600",
            "category": "VVV",
            "nx": 55,
            "ny": 127,
            "obsrValue": "0"
          },
          {
            "baseDate": "20220501",
            "baseTime": "0600",
            "category": "WSD",
            "nx": 55,
            "ny": 127,
            "obsrValue": "0.1"
          }
        ]
      }
    },
    "pageNo": 1,
    "numOfRows": 100,
    "totalCount": 8
  }
}
```

<https://jsonformatter.jsonparseronline.com/>

```
{
  response: {
    header: {
      resultCode: "00",
      resultMsg: "NORMAL_SERVICE"
    },
    body: {
      dataType: "JSON",
      items: {
        item: [
          {
            baseDate: "20220501",
            baseTime: "0600",
            category: "PTY",
            nx: 55,
            ny: 127,
            obsrValue: "0"
          },
          {
            baseDate: "20220501",
            baseTime: "0600",
            category: "REH",
            nx: 55,
            ny: 127,
            obsrValue: "79"
          }
        ]
      }
    }
  }
}
```

개발계정 상세보기

기본정보

데이터명	기상청_단기예보 ((구)_동네예보) 조회서비스			상세설명
서비스유형	REST	심의여부	자동승인	
신청유형	개발계정 활용신청	처리상태	승인	
활용기간	2022-05-01 ~ 2024-05-01			

샘플코드

Java

Javascript

C#

PHP

Curl

Objective-C

Python

Nodejs

R

Python3 샘플 코드

```
import requests
```

```
url = 'http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst'
```

```
params ={'serviceKey' : '서비스키', 'pageNo' : '1', 'numOfRows' : '1000', 'dataType' : 'XML', 'base_date' : '20210628', 'base_time' : '0600', 'nx' : '55', 'ny' : '127' }
```

```
response = requests.get(url, params=params)
```

```
print(response.content)
```



```
import requests
import json

url = 'http://apis.data.go.kr/1360000/VilageFcstInfoService_2.0/getUltraSrtNcst'
service_key="J5/Thyhm4Dvm1Wk5EqTakaX7ebIJRjlo7NGylmQ2DeA8qutXiaW6K2mlk9yKXgZMm+IeKw/Ft9QQEZ7UsvP3lg=="

params ={'serviceKey' : service_key,
        'pageNo' : '1',
        'numOfRows' : '1000',
        'dataType' : 'JSON',
        'base_date' : '20220501',
        'base_time' : '1200',
        'nx' : '68',
        'ny' : '101' }

res = requests.get(url, params=params)
print(res.text)
print(res.status_code, type(res.text),res.url)
```

```
print("==== response json data start ====")
print(res)
print("==== response json data end ====")
print()

r_dict = json.loads(res.text)
r_response = r_dict.get("response")
r_body = r_response.get("body")
r_items = r_body.get("items")
r_item = r_items.get("item")

result = {}
for item in r_item:
    if(item.get("category") == "T1H"):
        result = item
        break

print("==== response dictionary(python object) data start ====")
print(result)
print("==== response dictionary(python object) end ====")
print()
```

JSON (JavaScript Object Notation)

XML vs JSON

JSON (JavaScript Object Notation)

- JavaScript Object Notation라는 의미의 축약어로 데이터를 저장하거나 전송할 때 많이 사용되는 **경량의 DATA 교환 형식**
- Javascript에서 객체를 만들 때 사용하는 표현식을 의미
- JSON 표현식은 사람과 기계 모두 이해하기 쉬우며 용량이 작아서, 최근에는 JSON이 XML을 대체해서 데이터 전송 등에 많이 사용함.
- JSON은 데이터 포맷일 뿐이며 어떠한 통신 방법도, 프로그래밍 문법도 아닌 단순히 데이터를 표시하는 표현 방법

JSON 특징

- 서버와 클라이언트 간의 교류에서 일반적으로 사용
- JavaScript 객체 표기법과 아주 유사
- JavaScript를 이용하여 JSON 형식의 문서를 쉽게 자바스크립트 객체로 변환할 수 있음
- **JSON 문서 형식은 자바스크립트 객체의 형식을 기반으로 만들어짐**
- JavaScript의 문법과 유사하지만 **JSON은 텍스트 형식임**
- 다른 프로그래밍 언어를 이용해서도 쉽게 만들 수 있음
- 특정 언어에 종속되지 않으며, 대부분의 프로그래밍 언어에서 JSON 포맷의 데이터를 핸들링 할 수 있는 라이브러리를 제공

XML

- 데이터 값 양쪽으로 태그가 있다.
 - (HTML을 근본으로 했기에 태그 라는 것이 없을 수가 없는데, 그 태그를 줄인다 해도 최소한 표현하려면 양쪽에 몇 글자씩 있어야 한다.)

JSON

- 태그로 표현하기 보다는 중괄호({}) 같은 형식으로 하고, 값을 ' '로 나열하기에 그 표현이 간단함

JSON 문법

- JSON 형식은 자바스크립트 객체와 마찬가지로 **key / value**가 존재할 수 있으며 **key값**이나 **문자열**은 항상 쌍 따옴표(" ")를 이용하여 표기
- 객체, 배열 등의 표기를 사용할 수 있다.
- 일반 자바스크립트의 객체처럼 원하는 만큼 중첩 시켜서 사용할 수도 있다.
- JSON형식에서는 **null, number, string, array, object, boolean**을 사용할 수 있다.

```
{  
  "employees": [  
    {  
      "name": "Surim",  
      "lastName": "Son"  
    },  
    {  
      "name": "Someone",  
      "lastName": "Huh"  
    },  
    {  
      "name": "Someone else",  
      "lastName": "Kim"  
    }  
  ]  
}
```