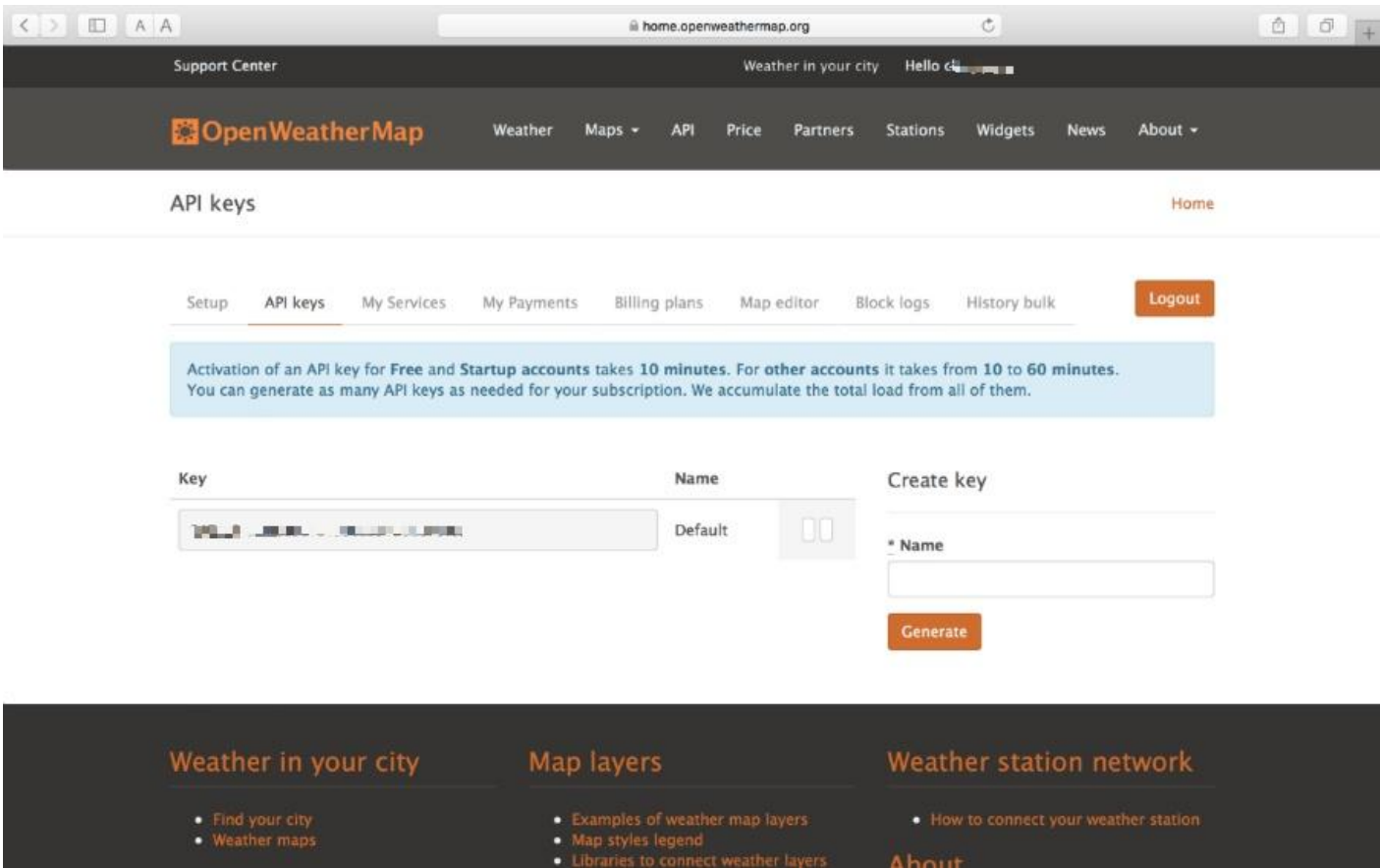


Javascript에서 날씨를 불러오기위해 구글에 검색했더니 구글날씨, 기상청, openweathermap 등이 나왔다.
구글날씨 api는 사용이 종료되었다고 해서 openweathermap api를 이용하기로했다.

- Openwathermap API 사용법

먼저 openweathermap 홈페이지에 들어가서 가입을 한다.(https://openweathermap.org)
가입을하면 API key를 받게되는데 이것을 이용해서 api를 사용하게된다.
가입을하고 my home페이지에서(my home 페이지는 로그인상태에서 상단의 sign in이나 sign up 버튼을 누르면 나온다.) API keys버튼을 누르면
자신이 api key를 볼수있다. 무료, 스타트업 계정은 api key가 다시 활성화될때까지 10분 걸린다고한다.
따라서 api key를 사용하고 10분후에 다시사용할수있다.



어떤 API를 사용할수 있는지 상단메뉴에서 API버튼을 클릭해봤다.
현재 날씨, 5일/3시간날씨, 16일날씨가있는데, 나는 오늘부터 7일간의 날씨가 필요해서 16 day / daily forecast의 API doc버튼을 눌러보았다.
Json이나 xml포맷으로 사용가능하다고한다.

openweathermap.org

OpenWeatherMap

WeatherMapsAPIPricePartnersStationsWidgetsNewsAbout

Our weather API is simple, clear and free. We also offer higher levels of support, please see our [paid plan options](#). To access the API you need to sign up for an [API key](#) if you are on a free or paid plan.

Current weather data

API docSubscribe

- Access current weather data for any location including over 200,000 cities
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations
- Data is available in JSON, XML, or HTML format
- Available for Free and all other paid accounts

5 day / 3 hour forecast

API docSubscribe

- 5 day forecast is available at any location or city
- 5 day forecast includes weather data every 3 hours
- Forecast is available in JSON and XML
- Available for Free and all other paid accounts

16 day / daily forecast

API docSubscribe

- 16 day forecast is available at any location or city
- 16 day forecast includes daily weather
- Forecast is available in JSON and XML
- Available for Developer, Professional and Enterprise accounts

Historical data

API docSubscribe

- Through our API we provide city historical weather data for 37,000+ cities
- Historical data is available for 1 month previous in Starter account, for 1 year previous in Medium accounts, and for 5 years in History Bulk

History Bulk NEW!

API docGET

- We have recently launched History Bulk functionality that provides weather data for 37,000+ cities.
- Historical data is available for 5 years and Pricing is simple and easy - just \$10 for one city, no matter how much data will you receive.

Weather map layers

API docSubscribe

- Weather maps include precipitation, clouds, pressure, temperature, wind, and more
- Connect our weather maps to your mobile applications and websites
- Use as layers in Direct Tiles, OpenLayers, Leaflet, and Google Maps
- Available for Free and all other paid accounts

Json을 이용한 간단한 테스트 코드이다.(인터넷익스플로러제외) 브라우저에서 실행하면 Rain이라는 글자가 뜬다.

```

<html>
<head>
  <meta charset = "UTF-8">
  <title>Using weather api</title>
</head>

<script>
  function parseWeather()
  {
    loadJSON(function(response)
    {
      var jsonData = JSON.parse(response);
      document.getElementById("todaysWeather").innerHTML = jsonData["list"][0]["weather"][0]["main"];
    });
  }

  function loadJSON(callback) //url의 json 데이터 불러오는 함수
  {
    var url = "http://api.openweathermap.org/data/2.5/forecast/daily?q=Busan,KR&cnt=7&APPID=6";
    var request = new XMLHttpRequest();
    request.overrideMimeType("application/json");
    request.open('GET', url, true);

    request.onreadystatechange = function ()
    {
      if (request.readyState == 4 && request.status == "200")
      {
        callback(request.responseText);
      }
    };
    request.send(null);
  }

  window.onload = function()
  {
    parseWeather();
  }
</script>

<body>
  <span id = "todaysWeather"></span>
</body>
</html>

```

loadJSON 메소드의 url변수에는 url을 넣으면 되는데 마지막 APPID에 자신의 api키를 넣으면된다.
url에 대한설명은 api doc버튼을 눌러보면 자세히 설명되어있다.
테스트해본 코드는 부산(q)의 현재부터 7일간(cnt)의 날씨이다.

parseWeather메소드에서 JSON.parse()함수를 이용해 string을 json객체로 바꾸어준다.

(JSON.parse) - parse 메소드는 string 객체를 json 객체로 변환 / JSON.stringify - stringify 메소드는 json 객체를 String 객체로 변환)

Json객체는 배열처럼 쓸수있다.

실제 코드에있는 url에 브라우저로 들어가보면

```

{"city":{"id":1838524,"name":"Busan","coord":
{"lon":129.0403,"lat":35.1028},"country":"KR","population":0,"cod":"200","message":4.6271179
,"cnt":7,"list":[{"dt":1499396400,"temp":
{"day":300.15,"min":297.69,"max":300.15,"night":297.89,"eve":297.91,"morn":300.15},"pressure":
1004.19,"humidity":88,"weather":[{"id":501,"main":"Rain","description":"moderate
rain","icon":"10d"}],"speed":2.52,"deg":219,"clouds":92,"rain":11.29},
{"dt":1499482800,"temp":
{"day":300.16,"min":297.03,"max":300.16,"night":297.03,"eve":299.62,"morn":297.25},"pressure":
1001.17,"humidity":87,"weather":[{"id":500,"main":"Rain","description":"light
rain","icon":"10d"}],"speed":2.36,"deg":223,"clouds":92,"rain":0.88},{"dt":1499569200,"temp":
{"day":297.89,"min":296.39,"max":298.26,"night":296.39,"eve":298.26,"morn":297.12},"pressure":
999.77,"humidity":99,"weather":[{"id":501,"main":"Rain","description":"moderate
rain","icon":"10d"}],"speed":1.86,"deg":215,"clouds":92,"rain":7.28},{"dt":1499655600,"temp":
{"day":297.13,"min":296.62,"max":297.13,"night":296.62,"eve":296.83,"morn":297.03},"pressure":
1012.62,"humidity":0,"weather":[{"id":503,"main":"Rain","description":"very heavy

```

```
rain","icon":"10d"}],{"speed":7.5,"deg":195,"clouds":68,"rain":55.45},{ "dt":1499742000,"temp":
{"day":298.76,"min":296.09,"max":298.76,"night":296.09,"eve":297.12,"morn":297.42},"pressure"
:1013.19,"humidity":0,"weather":[{"id":500,"main":"Rain","description":"light
rain","icon":"10d"}],{"speed":4.37,"deg":224,"clouds":42,"rain":2.96},{ "dt":1499828400,"temp":
{"day":298.8,"min":296.76,"max":298.8,"night":296.76,"eve":296.98,"morn":297.87},"pressure":1
014.26,"humidity":0,"weather":[{"id":501,"main":"Rain","description":"moderate
rain","icon":"10d"}],{"speed":2.53,"deg":224,"clouds":43,"rain":6.68},{ "dt":1499914800,"temp":
{"day":299.09,"min":296.32,"max":299.09,"night":296.32,"eve":297.19,"morn":297.59},"pressure"
:1010.52,"humidity":0,"weather":[{"id":500,"main":"Rain","description":"light
rain","icon":"10d"}],{"speed":4.48,"deg":210,"clouds":22,"rain":1.94}}}]
```

이런 데이터가있다.

jsonData["list"][0]["weather"][0]["main"] 코드를 보자. (대괄호를 이용할수도있고 .을 이용할수도 있
다. ex) jsonData.city.name)

jsonData의 리스트는 0부터 6까지 총 7개가있는데 오늘부터 7일간의 날씨를 의미한다.

위코드는 list의 0번 즉 오늘 날씨의 main을 의미한다. 오늘 날씨의 main은 Rain이므로 Rain이 브
라우저상에 나타난다.

(파라미터에대한 설명은 홈페이지에가면 알수있다.)

jsonData를 정렬해서 보고싶다면 <http://json.parser.online.fr> 와 같은 사이트를 이용하면된다.

api가 10분마다 활성화되기때문에 테스트를 10분에 한번씩 할 수 없으니 아래와같이 임시로 json객체를 만들어 사용할
수있다.

```
var jsonString={"city":{"id":1838524,"name":"Busan","coord":{"lon":129.0403,"lat":35.1028},"country":"KR","population":0},"cod":"200","message":0.6703125,"
cnt":7,"list":[{"dt":1499050800,"temp":{"day":300.15,"min":297.64,"max":300.15,"night":297.64,"eve":298.64,"morn":300.15},"pressure":1000.36,"humidity"
:71,"weather":[{"id":500,"main":"Rain","description":"light rain","icon":"10d"}],{"speed":4.97,"deg":233,"clouds":88,"rain":0.25},{ "dt":1499137200,"temp"
":{"day":296.69,"min":294.58,"max":297.42,"night":294.58,"eve":295.05,"morn":297.42},"pressure":1000.28,"humidity":100,"weather":[{"id":503,"main":"Rain"
,"description":"very heavy rain","icon":"10d"}],{"speed":1.26,"deg":92,"clouds":100,"rain":70.98},{ "dt":1499223600,"temp":{"day":296.76,"min":295.69,"max
":296.76,"night":295.81,"eve":295.8,"morn":295.69},"pressure":1018.19,"humidity":0,"weather":[{"id":502,"main":"Rain","description":"heavy intensity
rain","icon":"10d"}],{"speed":2.03,"deg":124,"clouds":68,"rain":15.96},{ "dt":1499310000,"temp":{"day":296.78,"min":296.23,"max":296.78,"night":296.23,"
eve":296.32,"morn":296.39},"pressure":1017.8,"humidity":0,"weather":[{"id":502,"main":"Rain","description":"heavy intensity rain","icon":"10d"}],{"speed"
:3.6,"deg":197,"clouds":47,"rain":35.81},{ "dt":1499396400,"temp":{"day":297.56,"min":296.05,"max":297.56,"night":296.05,"eve":296.44,"morn":296.68},"
pressure":1015.21,"humidity":0,"weather":[{"id":501,"main":"Rain","description":"moderate rain","icon":"10d"}],{"speed":4.45,"deg":225,"clouds":62,"rain"
:9.96},{ "dt":1499482800,"temp":{"day":298.35,"min":297,"max":298.35,"night":297,"eve":297.32,"morn":297.16},"pressure":1015.63,"humidity":0,"weather":[{"
id":501,"main":"Rain","description":"moderate rain","icon":"10d"}],{"speed":6.47,"deg":214,"clouds":70,"rain":8.24},{ "dt":1499569200,"temp":{"day":299.
66,"min":297.67,"max":299.66,"night":297.67,"eve":298.18,"morn":297.88},"pressure":1016.93,"humidity":0,"weather":[{"id":500,"main":"Rain","description"
:"light rain","icon":"10d"}],{"speed":6.64,"deg":222,"clouds":56,"rain":0.87}}];
var jsonData = JSON.parse(JSON.stringify(jsonString));
```

(+document.getElementById()에서 html코드에 그 아이디가 없으면 에러나기 때문에 주의해야한
다. 반대로 script에없는 아이디가 html코드에 있는건 가능하다.)

(+jsonData에서 temp는 온도인데 모두 297이렇게 높은 온도로 되어있다. 이것은 절대온도이므
로 273.15를 빼주어야한다.)

그리고 위의 코드에서 콜백함수를 쓰는이유는

- * 콜백(Callback)이란?
디자인 패턴 중 하나인 옵저버(observer) 패턴에서 나온 개념으로써 객체의 **상태 변화(이벤트)**가 발생 하였을 경우에 **이러한 사실을 함수를 통해 전달**
하게 되는데, 이를 콜백 함수라고 한다.
- * 콜백함수를 왜 사용하나?
웹 페이지에서 서버로부터 빠른 데이터를 수신하기 위해서 AJAX(Asynchronous JavaScript and XML)가 아닌 JSON(JavaScript Object Notation)
을 많이 사용하게 된다. 이렇게 하면 문제는 동기화 방식(어떤 함수의 실행이 끝나기 전에 다른 함수가 실행 됨)으로 동작을 하기 때문에 순차적으로
데이터를 전달받게 된다.
특히 AJAX요청과 함께 사용하면 정말 강력하다. 예를 들어, AJAX요청을 해서 어떤 정보를 가져와 그 가져온 정보로 무엇인가를 처리해주고 싶다. 하
지만 AJAX특성상 서버의 응답도 영향이 있기 때문에 같은 요청도 같은 컴퓨터일지라도 네트워크 환경 등 여러가지 변수 때문에 반응 속도가 천차만별
이다.

하지만 콜백함수를 이용하게 되면 "어, 나 이 작업 다 끝났으니까 뭐 할 꺼 있으면 해" 라고 콜백함수가 메시지를 전달함으로써 확실히 가져온 데이터를 가지고 처리를 할 수 있게 된다.

==> 콜백함수 문법은 특정 함수의 동작이 끝남과 동시에, 다른 여러 가지 함수를 호출해야 할 경우에 사용된다.

출처: <http://lovestudycom.tistory.com/entry/콜백함수를-왜-사용하나> [꿈꾸는 개발자]

이라고한다.

접기

openweathermap api 예제코드

```
<html>
<head>
  <meta charset = "UTF-8">
  <title>Using weather api</title>
</head>

<script>
  function parseWeather()
  {
    loadJSON(function(response)
    {
      var jsonData = JSON.parse(response);
      document.getElementById("todaysWeather").innerHTML = jsonData["list"][0]
["weather"][0]["main"];
    });
  }

  function loadJSON(callback) //url의 json 데이터 불러오는 함수
  {
    var url = "http://api.openweathermap.org/data/2.5/forecast/daily?
q=Busan,KR&cnt=7&APPID=047ea91a02e19c4e493ceb04d81879f6";
    var request = new XMLHttpRequest();
    request.overrideMimeType("application/json");
    request.open('GET', url, true);

    request.onreadystatechange = function ()
    {
      if (request.readyState == 4 && request.status == "200")
      {
        callback(request.responseText);
      }
    };
    request.send(null);
  }

  window.onload = function()
  {
    parseWeather();
  }
</script>
```

```
<body>
  <span id ="todaysWeather"> </span>
</body>

</html>
```
