

Weather APIs

Thursday, 24. September 2015

Letzte Aktualisierung Tuesday, 20. October 2015

Wetter APIs Überblick und Vergleich

Um in einer App oder Webseite Wetterdaten verarbeiten zu können, stellt sich zunächst die Frage nach der Quelle der Daten. Dazu konkurrieren einige API-Anbieter miteinander, einige davon kostenlos. In diesem Artikel lege ich den Schwerpunkt auf die kostenlos verfügbaren APIs.

Verglichen werden (bisher)...

OpenWeather

Yahoo Weather

Metwit Weather

Wunderground

Wetter.com

Wetter-API.de

Angeregt zu dieser Gegenüberstellung hatte mich mein Vorhaben, in meiner Foto-Filter-App Kleckrica 44 Filter anzubieten, die Informationen zum aktuellen Wetter in den Fotos anzeigen. Ergo stand ich vor genau diesem Problem. Was nehmen? Wie geht das? Was kostet das? Gibt's da auch etwas Kostenloses?

Open Weather

OpenWeatherMap.org war eigentlich die zweite Alternative nach Yahoo. Allerdings war es die erste, die ich vollständig implementiert hatte. Deshalb stelle ich es hier als erstes vor.

Die API bietet die Resultate als XML oder JSON an (oder HTML). Als Input können Geo-Koordinaten oder ein Stadtname verwendet werden. Die Aktualität liegt zwischen zehn Minuten und zwei Stunden für die kostenlose API. Die Daten stehen unter CC BY-SA 2.0, d.h. die Daten können kommerziell verwendet werden.

Inzwischen bietet OpenWeatherMap vier verschiedene Pricing-Modelle bzw. SLAs an. Das kostenlose Angebot "FREE" ist limitiert auf 1200 Abrufe/Minute und die Daten werden im Abstand von zwei Stunden aktualisiert. Hier Aktualisierungsraten, SLAs und Kontingente sind kostenpflichtig. Allerdings sind die festen monatlichen Preise deutlich planbarer als die Kosten/1000 Requests, die bei Yahoo schnell unkontrollierbar steigen können.

Angeboten wird das aktuelle Wetter, Vorhersagen für 5 Tage per 3 Stunden, 16 Tage täglich sowie historische Wetterdaten und Daten einzelner Stationen. Ausserdem werden diverse Wetterkarten angeboten. Insgesamt, finde ich, ist es das umfassendste Angebot der verglichenen Dienste.

Die Wetterdaten können unmittelbar wahlweise per Geo-Koordinaten, Name einer Stadt oder PLZ abgerufen werden. Zur eindeutigen Identifizierung einer Stadt steht ebenfalls ein City-Code zur Verfügung, wie das bei vielen anderen Diensten (s.u.) auch der Fall ist. Die Abfrage kann für mehrere Orte gleichzeitig erfolgen, wobei diesbezüglich das Preismodell zu beachten ist. Dieser Service steht nicht kostenfrei zur Verfügung.

Das Ergebnis steht wahlweise in XML oder JSON zur Verfügung. Die API ist multilingual. Literale können in mehreren Sprachen angefordert und dann direkt verwendet werden. Als optionale Einheiten stehen Celsius und Fahrenheit zur Verfügung. Default ist Kelvin.

Ausserdem ist die API auf der Webseite ausgesprochen gut dokumentiert. (Englisch)

Beispielsweise für die Abfrage mit Geo-Koordinaten von Achill Island

<http://api.openweathermap.org/data/2.5/weather?lat=53.9667&lon=-10&appid=MyAPIKey>

(Testweise kann der API-Key scheinbar auch weggelassen werden.)

<http://api.openweathermap.org/data/2.5/weather?lat=53.9667&lon=-10>)

zu diesem beispielhaften Ergebnis in JSON.

```
{
  "coord":{
    "lon":-9.72,
```

```

{
  "lat":53.9
},
"weather":[
  {
    "id":800,
    "main":"Clear",
    "description":"Sky is Clear",
    "icon":"01d"
  }
],
"base":"stations",
"main":{
  "temp":283.499,
  "pressure":1039.19,
  "humidity":96,
  "temp_min":283.499,
  "temp_max":283.499,
  "sea_level":1048.75,
  "grnd_level":1039.19
},
"wind":{
  "speed":1.66,
  "deg":97.508
},
"clouds":{
  "all":0
},
"dt":1443688483,
"sys":{
  "message":0.0184,
  "country":"IE",
  "sunrise":1443681639,
  "sunset":1443723322
},
"id":2961666,
"name":"Rosturk",
"cod":200
}

```

Weiterführende Links

Weather Conditions und Icons: <http://openweathermap.org/weather-conditions>

Angaben zu den Kosten: <http://openweathermap.org/price>

Yahoo Weather

Yahoo war eigentlich meine erste Wahl, als ich mich nach Wetter-APIs umschaute. Es gibt reichlich Tutorials und ist auch gar nicht so wahnsinnig schwierig. Die API wird als RSS in XML angeboten. Das Resultat kann auch in JSON angefordert werden.

Der Knackpunkt bei Yahoo ist, dass man das Wetter nicht direkt zu Geo-Koordinaten abfragen kann. Man muss zunächst als regionale Referenz eine sog. WOEID anfordern und kann dann die Wetterdaten zu dieser WOEID anfordern. Allerdings wurde die kostenlose API zur Abfrage der WOEID zu Geokoordinaten (oder auch Ortsnamen) vor einiger Zeit eingestellt. Der Dienst wurde durch Yahoo Boss ersetzt. Boss ist allerdings kostenpflichtig bzw. bietet für nicht-gewerbliche Projekte eine kostenfreie Nutzung bis zu 2000 Requests pro Tag an. Wer also auf etwas Erfolg mit seiner App hofft, auch wenn sie keinen gewerblichen Hintergrund hat, riskiert, seine Kreditkarte nicht unerheblich belasten zu lassen.

(Leider enthält Yahoos Webseite keine Angaben zur Zeitspanne, für welche das 2000er Limit gelten soll. Die meisten Quellen sprechen von 2000/Tag in Bezug auf die Places-Tabelle und 2000/Stunde in Bezug auf die Anzahl Request per IP für den Endpunkt.)

Damit keine Missverständnisse auftreten: Die Yahoos Wetter-API steht weiterhin kostenlos zur Verfügung, allerdings setzt sie die WOEID voraus. Und die Suche nach der WOEID ausgehend von Geo-Daten ist kostenpflichtig. Um Wetterdaten für einen bestimmten Ort anzuzeigen, wäre die API immer noch eine gute alternative. Auch für

Vohersagen bei denen der Anwender einen Ortsnahmen als Suchbegriff eingibt, ist die API sicher gut einsetzbar. Deshalb möchte ich dennoch das Vorgehen kurz erläutern. Umweg über yahoo search

Um der kostenpflichtigen bzw. begrenzten BOSS API aus dem Weg zu gehen, bietet sich für Mobile Apps der Umweg über das integrierte Geo-Coding an. Die SDK für iOS, Android und Windows Phone bieten native Reverse-Geo-Coding APIs, deren Verwendung kostenfrei und unlimitiert ist. (Für Google/Android könnte es Begrenzungen geben, falls die App zu einem kostenpflichtigen Dienst gehört, nicht aber falls die App etwas Geld kostet.) Ermittelt man auf diesem Weg den Ort zu einer Geo-Location, dann kann man den Ort über die Search-API suchen und bekommt im Ergebnis u.a. die woeid zurück.

```
Dazu ein Code-Beispiel für iOS/Objective-C bzw. Cocoa Touch CLGeocoder *geocoder = [[CLGeocoder alloc] init];
[geocoder reverseGeocodeLocation:location
 completionHandler:^(NSArray *placemarks, NSError *error) {
    if (error){
        NSLog(@"Geocode failed with error: %@", error);
        return;
    }
    CLPlacemark *placemark = [placemarks objectAtIndex:0];
    NSLog(@"cityname - %@", placemark.locality);
}];
```

So resultiert die GET-Abfrage an <https://search.yahoo.com/sugg/gossip/gossip-gl-location/?appid=weather&output=sd1&p2=pt&command=Dortmund>

```
für Dortmund: {
  "q":"Dortmund",
  "l":{
    "gprid":"kID_92z1SOahl3R9Y.7fAA"
  },
  "r":[
    {
      "k":"Dortmund",
      "d":"pt:iso=DE&woeid=645458&lon=7.45829&lat=51.5166&s=North Rhine-
Westphalia&c=Germany&country_woeid=23424829&n=Dortmund, North Rhine-Westphalia, Germany"
    }
  ]
}
```

Mit der WOEID für Dortmund 645458 lässt sich dann das Wetter erfragen.

Die URI <http://weather.yahooapis.com/forecastrss?w=23424829&u=c> führt zu folgendem beispielhaften Ergebnis:

Dabei ist 23424829 die woeid für Dortmund (siehe obige response) und c gibt an, dass das Ergebnis in Celsius ausgegeben werden soll. Default ist Fahrenheit, wenn man &u= nicht mit angibt. {

```
"query":{
  "count":1,
  "created":"2015-09-24T17:28:31Z",
  "lang":"de",
  "results":{
    "channel":{
      "title":"Yahoo! Weather - Dortmund, DE",

"link":"http://us.rd.yahoo.com/dailynews/rss/weather/Dortmund__DE/*http://weather.yahoo.com/forecast/GMXX0024_f.html",
    "description":"Yahoo! Weather for Dortmund, DE",
    "language":"en-us",
    "lastBuildDate":"Thu, 24 Sep 2015 6:49 pm CEST",
    "ttl":"60",
    "location":{
      "city":"Dortmund",
      "country":"Germany",
      "region":"NW"
    },
    "units":{
      "distance":"mi",
      "pressure":"in",
      "speed":"mph",
```

```

    "temperature": "F"
  },
  "wind": {
    "chill": "61",
    "direction": "210",
    "speed": "8"
  },
  "atmosphere": {
    "humidity": "63",
    "pressure": "30.03",
    "rising": "0",
    "visibility": "6.21"
  },
  "astronomy": {
    "sunrise": "7:18 am",
    "sunset": "7:23 pm"
  },
  "image": {
    "title": "Yahoo! Weather",
    "width": "142",
    "height": "18",
    "link": "http://weather.yahoo.com",
    "url": "http://l.yimg.com/a/i/brand/purplelogo//uh/us/news-wea.gif"
  },
  "item": {
    "title": "Conditions for Dortmund, DE at 6:49 pm CEST",
    "lat": "51.52",
    "long": "7.46",

"link": "http://us.rd.yahoo.com/dailynews/rss/weather/Dortmund__DE/*http://weather.yahoo.com/forecast/GMXX0024_f.html",
    "pubDate": "Thu, 24 Sep 2015 6:49 pm CEST",
    "condition": {
      "code": "28",
      "date": "Thu, 24 Sep 2015 6:49 pm CEST",
      "temp": "61",
      "text": "Mostly Cloudy"
    },
    "description": "\n<img src=\"http://l.yimg.com/a/i/us/we/52/28.gif\"/><br />\nCurrent Conditions:<br />\nMostly Cloudy, 61  

/>\n<BR />Forecast:<BR />\nThu - Cloudy. High: 62 Low: 51<br />\nFri - Partly Cloudy. High: 62 Low: 44<br />\nSat - Partly  

Cloudy. High: 61 Low: 43<br />\nSun - Sunny. High: 61 Low: 43<br />\nMon - Sunny. High: 62 Low: 43<br />\n<a  

href=\"http://us.rd.yahoo.com/dailynews/rss/weather/Dortmund__DE/*http://weather.yahoo.com/forecast/GMXX0024_f.ht  

ml\">Full Forecast at Yahoo! Weather</a><BR/><BR/>\n(provided by <a href=\"http://www.weather.com\">The Weather  

Channel</a><br/>\n",
    "forecast": [
      {
        "code": "26",
        "date": "24 Sep 2015",
        "day": "Thu",
        "high": "62",
        "low": "51",
        "text": "Cloudy"
      },
      {
        "code": "32",
        "date": "28 Sep 2015",
        "day": "Mon",
        "high": "62",
        "low": "43",
        "text": "Sunny"
      }
    ],
    "guid": {
      "isPermaLink": "false",
      "content": "GMXX0024_2015_09_28_7_00_CEST"
    }
  }
}

```

```
}  
}  
}  
}
```

} (Die Antwort enthält fünf Vorhersage-Elemente im entsprechenden JSON-Array. Die Antwort wurde zugunsten der Lesbarkeit des Artikels dort gekürzt.
Weiterführende Links

<https://developer.yahoo.com/boss/geo/>

https://developer.yahoo.com/yql/guide/usage_info_limits.html (Limits for Endpoints; Additional Limitations to the Table apply)

<https://developer.yahoo.com/weather/documentation.html> (Doku inkl. weather condition codes) Metwit Weather

Auf Metwit wurde ich aufmerksam, als ich nach Alternativen suchte für Yahoo's nicht mehr kostenfrei verfügbaren Yahoo Placefinder API.

Nun, ihre Homepage <http://www.metwit.com> scheint aktuell (3.10.2015) nicht richtig funktionieren zu wollen.
Weiterführende Links

<http://www.metwit.com>

<http://soup.metwit.com/post/47181933854/an-alternative-to-yahoo-weather-api> Wunderground (weather.com)

Kostenlos ist die Verwendung scheinbar ausschließlich für Entwickler mit die Produktion unbrauchbaren Limitierungen.

<http://www.wunderground.com/weather/api?apikey=3be21450e864dc6e> Weiterführende Links

http://www.chrisge.org/blog/2013-03-09/wunderground_api

Wetter.com

Bei Wetter.com gliedert sich die Suche wieder in zwei Bereiche. Zunächst steht die Suche nach einem City-Code zu einem Ort. Leider gibt es wohl keine Möglichkeit, nach Geo-Koordinaten zu suchen. Den aktuellen Standort zu finden dürfte sich genauso fehleranfällig gestalten wie bei Yahoo.com.

Die API kann grundsätzlich kostenlos verwendet werden. Dazu müssen zwei der drei Verweise auf Wetter.com implementiert werden:

1. "Powered by wetter.com"
2. Link zu wetter.com
3. Wetter.com Logo wird angezeigt. API Key erhalten

Nach der Registrierung bei wetter.com legt man ein Projekt an und erhält für dieses Projekt einen API-Key. Ortsname ermitteln

Um vom den Geo-Koordinaten des aktuellen Standorts auf den Ortsnamen kommen, kann zunächst analog wie beim Workaround für YahooWeather vorgegangen werden. D.h. man kann das im mobilen OS nativ angebotene Reverse-Geocoding verwenden. City-Code ermitteln

Mit dem Ortsnamen lässt sich nun die URL ermitteln um zum Ort u.a. den City-Code zu erhalten. In das URL-Schema (s. Dokumentation) fließen der Ortsname, der Projektname und ein aus Projektname, API-Key und Ortsname bzw. Suchbegriff (in dieser Reihenfolge ohne Trennzeichen) ermittelten MD5-Hash ein. Am Beispiel Mettmann ergibt sich daraus aus dieser Anfrage

<http://api.wetter.com/location/index/search/Mettmann/project/my-very-own-project/cs/0413c766560897be194720cb02d43f2b>

folgende Antwort<?xml version="1.0" encoding="UTF-8"?>
<search>


```

& & & & & & & & & & & <time value="11:00">
& & & & & & & & & & & & & & <wd>180</wd>
& & & & & & & & & & & & & & <w>61</w>
& & & & & & & & & & & & & & <tx>19</tx>
& & & & & & & & & & & & & & <tn>16</tn>
& & & & & & & & & & & & & & <p>6</p>
& & & & & & & & & & & & & & <ws>14</ws>
& & & & & & & & & & & & & & <d>1444129200</d>
& & & & & & & & & & & & & & <wd_txt>S</wd_txt>
& & & & & & & & & & & & & & <w_txt>leichter Regen</w_txt>
& & & & & & & & & & & & & & </time>
& & & & & & & & & & & & & & <time value="17:00">
& & & & & & & & & & & & & & <wd>135</wd>
& & & & & & & & & & & & & & <w>61</w>
& & & & & & & & & & & & & & <tx>19</tx>
& & & & & & & & & & & & & & <tn>16</tn>
& & & & & & & & & & & & & & <p>6</p>
& & & & & & & & & & & & & & <ws>12</ws>
& & & & & & & & & & & & & & <d>1444150800</d>
& & & & & & & & & & & & & & <wd_txt>SO</wd_txt>
& & & & & & & & & & & & & & <w_txt>leichter Regen</w_txt>
& & & & & & & & & & & & & & </time>
& & & & & & & & & & & & & & <time value="23:00">
& & & & & & & & & & & & & & <wd>135</wd>
& & & & & & & & & & & & & & <w>3</w>
& & & & & & & & & & & & & & <tx>16</tx>
& & & & & & & & & & & & & & <tn>15</tn>
& & & & & & & & & & & & & & <p>6</p>
& & & & & & & & & & & & & & <ws>11</ws>
& & & & & & & & & & & & & & <d>1444172400</d>
& & & & & & & & & & & & & & <wd_txt>SO</wd_txt>
& & & & & & & & & & & & & & <w_txt>bedeckt</w_txt>
& & & & & & & & & & & & & & </time>
& & & & & & & & & & & & & & <d>1444111200</d>
& & & & & & & & & & & & & & <tn>14</tn>
& & & & & & & & & & & & & & <p>24</p>
& & & & & & & & & & & & & & <ws>12</ws>
& & & & & & & & & & & & & & <wd_txt>S</wd_txt>
& & & & & & & & & & & & & & <w_txt>leichter Regen</w_txt>
& & & & & & & & & & & & & & </date>
& & & </forecast>
</city>

```

Das XML der Antwort enthält eine Vorhersage, wobei der Tag in 4 Teile aufgeteilt wird und es für jeden eine Vorhersage gibt. Neben dem Wettercode ("w") 61 enthält es ein Literal ("w_txt") "leichter Regen", die Höchst- und Tiefst-Temperatur für diesen Tagesabschnitt ("tx", "tn") in °C, die Windrichtung ("wd") in ° und die Windstärke ("ws") in km/h sowie die lokale Zeit als Unix-Timestamp ("d"), u.a. Weiterführende Links

http://www.wetter.com/apps_und_mehr/website/api/dokumentation/

http://api.wetter.com/files/wetter.com_openweather_api.pdf

Logos (Enthält auch Icons) Wetter-Api.de

Kostenlos bis 250 Zugriffe pro Tag.

Stellt Wetterdaten nur für Deutschland bereit.
Liefert die Daten ausschließlich in XML aus.

Anekdote am Rande: Direkt nach der ersten Anmeldung nach Registrierung wird angezeigt, man habe das maximale Limit für den laufenden Tag bereits erreicht. :-)

Die Abfrage geschieht über die Postleitzahl des Ortes. Gut ist daran, dass die PLZ über die Reverse-Geocoding APIs in Android und iOS recht gut und zuverlässig ermittelt werden können.

So fÃ¼hrt die Abfrage <http://www.wetter-api.de/wetter-MyKey-85737.xml> (wobei MyKey durch den jeweiligen API-Key ersetzt werden muss) zu der Antwort:Ã <?xml version="1.0" encoding="UTF-8"?>

```
<vorhersage>
  <ort>Ismaning</ort>
  <aktuell>
    <icon>partlycloudy</icon>
    <icon_url>http://www.wetter-api.de/wettericon/partlycloudy.gif</icon_url>
    <temperatur>12</temperatur>
    <feuchtigkeit>54</feuchtigkeit>
    <wind>33</wind>
    <windrichtung>East</windrichtung>
    <druck>1030</druck>
    <regen>-99990</regen>
    <schnee>0</schnee>
    <sicht>3</sicht>
  </aktuell>
  <vorhersagetage>
    <vorhersagetag>
      <datum>
        <epoche>1443718800</epoche>
        <tag>1</tag>
        <monat>10</monat>
        <jahr>2015</jahr>
        <tagdesjahres>273</tagdesjahres>
        <stunde>19</stunde>
        <min>00</min>
        <sec>0</sec>
      </datum>
      <periode>1</periode>
      <hoch>
        <celsius>17</celsius>
      </hoch>
      <tief>
        <celsius>3</celsius>
      </tief>
      <conditions>Heiter</conditions>
      <icon>clear</icon>
      <icon_url>http://www.wetter-api.de/wettericon/clear.gif</icon_url>
      <regenwahrscheinlichkeit>0</regenwahrscheinlichkeit>
      <regen>
        <mm>0</mm>
      </regen>
      <schnee>
        <cm>0</cm>
      </schnee>
      <wind>
        <kmh>1</kmh>
        <dir>O</dir>
      </wind>
      <feuchtigkeit>5</feuchtigkeit>
    </vorhersagetag>
  <!-- Hier wurden weitere Vorhersagetage ausgeblendet -->
</vorhersagetage>
<warnung>
  <warnungvor />
  <warnstufe />
  <![CDATA[<warninfo></warninfo>]]>
</warnung>
</vorhersage>
```

(Die Antwort enthÃ¤lt 9 Vorhersagetage, von denen 8 zur besseren Lesbarkeit des Artikels ausgeblendet wurden.)
WeiterfÃ¼hrende Links

<http://www.wetter-api.de> Fonts und Icon-Sammlungen

Zu guter letzt muss das Wetter noch dargestellt werden. Dazu habe ich ein paar Icon-Sammlungen und Fonts entdeckt.

Metwit
Sollte Metwit noch funktionieren, dann scheinen sie Icons zu den Wetterkonditionen anzubieten. Links zu den Icons befinden sich im JSON tbd

Lukas Bischoff

Lukas Bischoff bietet seinen Font auf archill.de für nicht-kommerzielle Projekte kostenlos an. artill.de
wetter.com

Auf der Seite, in der die Logos publiziert werden, die man gemeinsam mit den Wetterdaten anzeigen soll, stehen außerdem Sammlungen mit Wetter-Symbolen zum Herunterladen bereit. wetter.com doku

Eric Flowers

222 Icons rund um das Wetter als Font GitHub Daniel Vierich

Sammlung 60 schlichter Wettericons, die im Open-Source-Gedanken frei zur Verfügung stehen. <http://www.danvierich.de>

<http://www.danvierich.de>

Mike Afford

Kostenpflichtige aber bezahlbare Fonts und Icon-Sammlungen für Wetter-Apps

Mike Afford Media

Meteocons Eine kostenfreie Sammlung Icons in diversen Grafikformaten

<http://www.alessioatzeni.com>

Â

Â

Â