

# wetter.com openweather Doku



## Was ist wetter.com openweather?

openweather ist ein Webservice, mit dem Wettervorhersagen von wetter.com mithilfe von beliebigen Programmiersprachen abgerufen werden können. Hiermit können Sie die Wettervorhersage in eigene Programme, Widgets und Webseiten einbinden.

## Für wen ist wetter.com openweather?

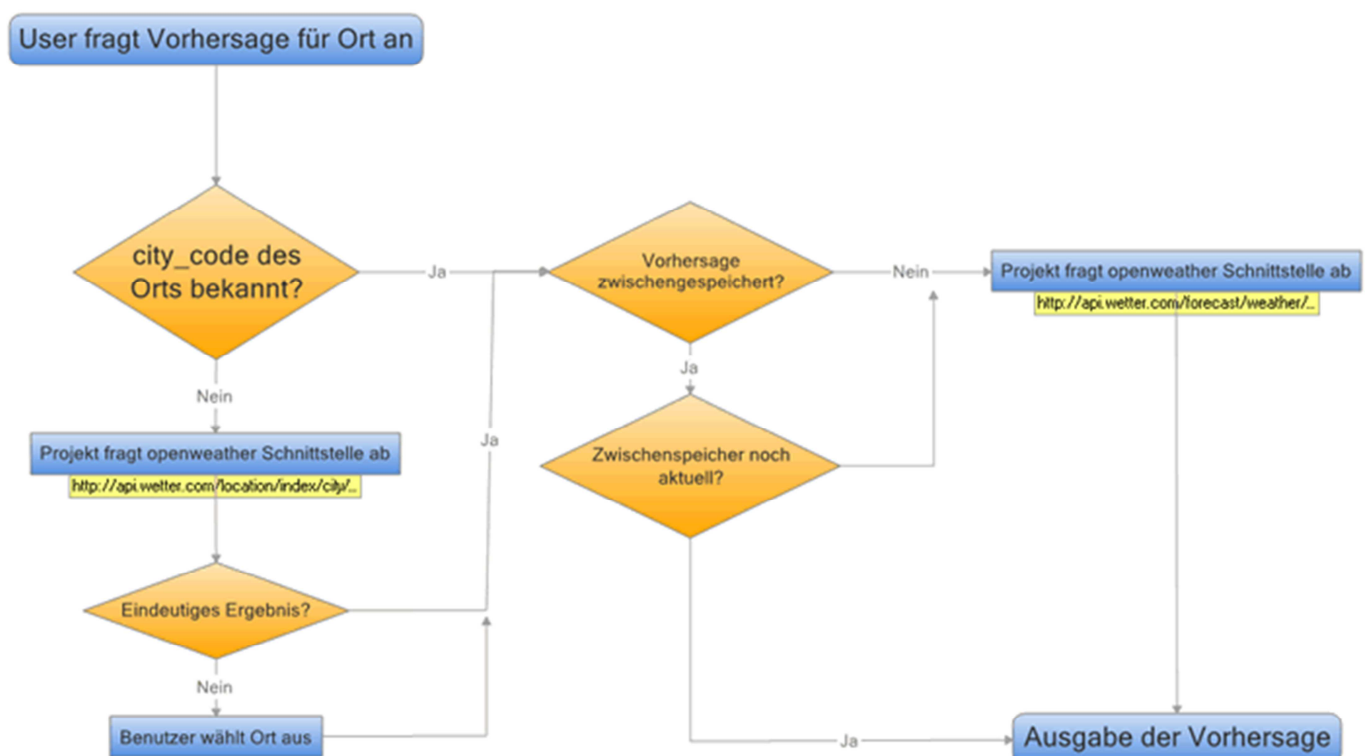
openweather ist für Privatpersonen, die damit Projekte ohne kommerziellen Hintergrund verwirklichen möchten. Unser Ziel ist es, Menschen die kostenlos etwas Gutes für Andere tun in ihrem Vorhaben zu unterstützen – und wie könnten wir das besser als durch Bereitstellung unserer Wetterdaten?

## Sind Programmierkenntnisse nötig?

Die Abfrage der Daten an sich ist einfach, allerdings werden grundlegende Programmierkenntnisse vorausgesetzt (zum Beispiel um eine MD5 Checksumme zu erzeugen). Auch werden die Daten im XML oder JSON Format geliefert, wer sie also schön anzeigen möchte muss da noch ein bisschen Logik hineinstecken.

## Wie funktioniert wetter.com openweather?

Folgendem Diagramm kann man den Ablauf einer Vorhersage mit der Schnittstelle entnehmen.



Der Benutzer Ihres Projekts möchte die Wettervorhersage für einen Ort sehen.

Um die Wettervorhersage zu holen, benötigen Sie den city\_code dieses Orts (z.B. DE0001020 für Berlin). Ist dieser nicht bekannt, müssen Sie erst bei der openweather Schnittstelle nachfragen. Mehr dazu erfahren Sie im Kapitel "Suchanfrage".

Haben Sie den city\_code können Sie damit die Vorhersage für den Ort anfragen. Lesen Sie dazu das Kapitel "Wettervorhersage".

## Suchanfrage

Um einen Ort aus dem wetter.com Datenbestand eindeutig zu identifizieren benötigt man einen sogenannten wetter.com city code.

Der city code ist eine Mischung aus Buchstaben und Zahlen. Der city code von Berlin ist DE0001020.

DE zeigt an, dass die Stadt in Deutschland liegt, die restlichen Stellen identifizieren den Ort. Stadtteile einer Stadt haben einen längeren city code, der von Berlin Tempelhof lautet zum Beispiel DE0001020466

Soll eine Anwendung nur das Wetter für eine bestimmte, vorher festgelegte Stadt liefern, ist es nicht nötig den city code für diese Stadt jedesmal dynamisch zu holen. Die einfachere Methode wäre hier, auf [www.wetter.com](http://www.wetter.com) den Ort über die Schnellsuche zu suchen und den Code aus der URL zu kopieren.

Beispiel:

[http://www.wetter.com/wetter\\_aktuell/aktuelles\\_wetter/deutschland/berlin/DE0001020.html](http://www.wetter.com/wetter_aktuell/aktuelles_wetter/deutschland/berlin/DE0001020.html)

Für komplexere Anwendungen empfiehlt es sich oft, dem Benutzer selbst die Auswahl des Orts zu überlassen.

Hierzu können Sie eine Anfrage an die openweather Schnittstelle schicken, die Informationen für alle Suchergebnisse eines Begriffs zurück liefert. Diese Liste kann dem Benutzer präsentiert werden, um die Auswahl zu erleichtern.

Sie sollten in Ihrer Programmierung auf jeden Fall Sorge dafür tragen, dass ein Fehler als Antwort entsprechend behandelt wird. In der Projektverwaltung können Sie einstellen, welche Werte von der Suche zurück geliefert werden und erfahren auch, was diese bedeuten.

## Suchmöglichkeiten

Je nach Anwendungsfall macht es Sinn, die Suche so offen wie möglich zu halten oder sie einzuschränken. Hierfür gibt es mehrere Suchmöglichkeiten mit der wetter.com openweather Schnittstelle.

1. Suche nach dem citycode, der einen Ort in der wetter.com Datenbank eindeutig identifiziert.
2. Suche nach dem Ortsnamen oder nach einem Teil des Ortsnamens
3. Suche nach einer Postleitzahl

<b>URL</b>	<code>http://api.wetter.com/location/index/search/&lt;suchbegriff&gt;/project/&lt;projektname&gt;/cs/&lt;checksumme&gt;</code>
<b>Controller</b>	Forecast
<b>Action</b>	weather
<b>Benötigte Parameter</b>	city
<b>Optionale Parameter</b>	output
<b>Checksumme</b>	md5 aus zusammengesetztem String aus projektname, API key und citycode
<b>Ausgabeformate</b>	Json, XML

## Parameterbeschreibung

Parametername	Benötigt	Beschreibung
search	Ja	Suchbegriff nach dem gesucht werden soll
output	Nein	json/xml, gibt die Daten im jeweiligen Format aus. Standardmäßig wird XML ausgegeben.

## Rückgabewerte

Wertname	Beschreibung												
maxhits	Anzahl wie viele Treffer insgesamt gefunden wurden												
hits	Ist beim openweather generell auf 30 beschränkt												
exact_match	boolean, ob ein Ort gefunden wurde, der exakt der Suchanfrage entspricht												
search_string	Der Begriff nach dem gesucht wurde												
start	Offset von welcher Position aus die Ergebnisse starten												
credit	Um die kostenlose wetter.com openweather API benutzen zu dürfen, müssen zwei von den drei Hinweisen auf wetter.com angezeigt werden. <table><tr><th>Wertname</th><th>Beschreibung</th></tr><tr><td>Text</td><td>Powered by wetter.com</td></tr><tr><td>Link</td><td>Link auf wetter.com</td></tr><tr><td>Logo</td><td>Eine Sammlung von offiziellen wetter.com Logos</td></tr></table>	Wertname	Beschreibung	Text	Powered by wetter.com	Link	Link auf wetter.com	Logo	Eine Sammlung von offiziellen wetter.com Logos				
Wertname	Beschreibung												
Text	Powered by wetter.com												
Link	Link auf wetter.com												
Logo	Eine Sammlung von offiziellen wetter.com Logos												
result	<table><tr><td>name</td><td>Name des Ortes</td></tr><tr><td>country_code</td><td>Abkürzung des Landes</td></tr><tr><td>Region</td><td>Name der Region</td></tr><tr><td>region_code</td><td>Abkürzung der Region</td></tr><tr><td>Plz</td><td>PLZ des Ortes</td></tr><tr><td>city_code</td><td>city_code, eindeutige ID eines Orts in der wetter.com Datenbank. Der city_code kann über eine Suchanfrage ermittelt werden</td></tr></table>	name	Name des Ortes	country_code	Abkürzung des Landes	Region	Name der Region	region_code	Abkürzung der Region	Plz	PLZ des Ortes	city_code	city_code, eindeutige ID eines Orts in der wetter.com Datenbank. Der city_code kann über eine Suchanfrage ermittelt werden
name	Name des Ortes												
country_code	Abkürzung des Landes												
Region	Name der Region												
region_code	Abkürzung der Region												
Plz	PLZ des Ortes												
city_code	city_code, eindeutige ID eines Orts in der wetter.com Datenbank. Der city_code kann über eine Suchanfrage ermittelt werden												

## Ausgabe XML

```
<search>
  <search_string>
    <![CDATA[ berlin ]]>
  </search_string>
  <maxhits>30</maxhits>
  <hits>24</hits>
  <exact_match>no</exact_match>
  <credit>
    <!--
```

In order to use the free weather data from [wetter.com](http://www.wetter.com) you HAVE TO display at least two out of three of the following possibilities: text, link, logo

```
-->
<text>Powered by wetter.com</text>
<link>http://www.wetter.com</link>
<logo>
Download at http://www.wetter.com/api/downloads/#logos
</logo>
</credit>

<result>
  <item value="1">
    <adm_1_name>Deutschland</adm_1_name>
    <city_code>DE0001020</city_code>
    <plz>10115</plz>
    <quarter/>
    <adm_1_code>DE</adm_1_code>
    <adm_2_name>Berlin</adm_2_name>
    <adm_4_name>Berlin</adm_4_name>
    <name>Berlin</name>
  </item>
  <item value="2">
    <adm_1_name>Vereinigte Staaten von Amerika</adm_1_name>
    <city_code>US0NH0019</city_code>
    <plz/>
    <quarter/>
    <adm_1_code>US</adm_1_code>
    <adm_2_name>New Hampshire</adm_2_name>
    <adm_4_name/>
    <name>Berlin</name>
  </item>
</result>
</search>
```

## Ausgabe Json

```
{
  search: {
    search_string: "berlin",
    maxhits: 30,
    hits: 24,
    exact_match: false,
```

```

    credit: {
        info: "In order to use the free weather data from wetter.com you HAVE TO display
at least two out of three of the following possibilities: text, link, logo",
        text: "Powered by wetter.com",
        link: "http://www.wetter.com",
        logo: "Download at http://www.wetter.com/api/downloads/#logos"
    },
    result: [
        {
            adm_1_name: "Deutschland",
            city_code: "DE0001020",
            plz: "10115",
            quarter: "",
            adm_1_code: "DE",
            adm_2_name: "Berlin",
            adm_4_name: "Berlin",
            name: "Berlin"
        },
        {
            adm_1_name: "Vereinigte Staaten von Amerika",
            city_code: "US0NH0019",
            plz: "",
            quarter: "",
            adm_1_code: "US",
            adm_2_name: "New Hampshire",
            adm_4_name: "",
            name: "Berlin"
        },
        {
            adm_1_name: "Vereinigte Staaten von Amerika",
            city_code: "US0NJ0041",
            plz: "",
            quarter: "",
            adm_1_code: "US",
            adm_2_name: "New Jersey",
            adm_4_name: "",
            name: "Berlin"
        }
    ]
}

```

# Wettervorhersage

Hat man den [wetter.com](http://wetter.com) city code eines Ortes, kann man für diesen die Wettervorhersage abfragen. Wie man den city code zu einem Ort findet, ist im Kapitel "Suchanfrage" beschrieben.

Die Abfrage einer Wettervorhersage läuft genauso ab, wie die Suchabfrage. Eine Anwendung ruft eine URL mit bestimmten Parametern auf und bekommt als Antwort ein Dokument in einem vorher festgelegten Format.

Die Wettervorhersage wird maximal für Heute plus zwei Tage geliefert. Benötigt man weniger Daten, kann dies in den Projekteinstellungen entsprechend konfiguriert werden. Für jeden Tag gibt es vier Vorhersagezeitpunkte: morgens, mittags, abends und nachts. Dies ist definiert durch einen Zeitpunkt in lokaler Zeit für diesen Ort und ein Attribut, welches die Gültigkeitsdauer dieses Vorhersagezeitpunkts beschreibt.

Um ein Beispiel für diese Vorhersagepunkte zu sehen, kann man auf [www.wetter.com](http://www.wetter.com) in der Schnellsuche nach einem Ort suchen. Im Reiter „Heute“ sieht man die Vorhersage mit verschiedenen Messzeitpunkten (5 Uhr, 8 Uhr, 11 Uhr, etc). In der openweather API sind nur 4 Tagesabschnitte vorhanden. Zusätzlich zu den Abschnitten gibt es einen aus der Vorhersage errechneten Tagesdurchschnitt, der direkt unterhalb des Tagesknotens zu finden ist.

Ausführen einer Vorhersageanfrage

Um eine Wettervorhersage zu erhalten, schickt man einen HTTP Request an

`http://api.wetter.com/forecast/weather/city/<citycode>/project/<Projektname>/cs/<Checksumme>`

Die in spitzen Klammern gefassten Variablen müssen durch die entsprechenden Werte für das Projekt und die Anfrage ersetzt werden.

CityCode ist der [wetter.com](http://wetter.com) City Code, den man entweder statisch gespeichert hat oder dynamisch über die openweather Schnittstelle beziehen kann. Projektname ist der Name des Projekts und kann in den Projekteinstellungen nachgelesen werden. Die Checksumme ist ein MD5 Hash über einen zusammengefassten String aus Projektname, API Key und City Code, siehe auch "Authentifizierungsverfahren".

## Fakten

URL	<code>http://api.wetter.com/forecast/weather/city/&lt;citycode&gt;/project/&lt;projektname&gt;/cs/&lt;checksumme&gt;/</code>
Controller	Forecast
Action	Weather
Benötigte Parameter	City
Optionale Parameter	language output
Checksumme	md5 aus zusammengesetztem String aus projektname, API key und citycode
Ausgabeformate	Json, XML

## Parameterbeschreibung

Parametername	Benötigt	Beschreibung
city	Ja	city_code, eindeutige ID eines Orts in der wetter.com Datenbank. Der city_code kann über eine Suchanfrage ermittelt werden
language	Nein	en/es, übersetzt die Parameter w_txt_short und w_txt in die jeweilige Sprache
output	Nein	json/xml, gibt die Daten im jeweiligen Format aus. Standardmäßig wird XML ausgegeben.

## Rückgabewerte

### Ortsdaten

Wertname	Beschreibung								
name	Name des Orts								
post_code	PLZ des Ortes								
url	Http Link auf die jeweilige Ortsseite auf wetter.com								
Credit	<p>Um die kostenlose wetter.com openweather API benutzen zu dürfen, müssen zwei von den drei Hinweisen auf wetter.com angezeigt werden.</p> <table> <tr> <th>Wertname</th><th>Beschreibung</th></tr> <tr> <td>Text</td><td>Powered by wetter.com</td></tr> <tr> <td>Link</td><td>Link auf wetter.com</td></tr> <tr> <td>Logo</td><td>Eine Sammlung von offiziellen wetter.com Logos</td></tr> </table>	Wertname	Beschreibung	Text	Powered by wetter.com	Link	Link auf wetter.com	Logo	Eine Sammlung von offiziellen wetter.com Logos
Wertname	Beschreibung								
Text	Powered by wetter.com								
Link	Link auf wetter.com								
Logo	Eine Sammlung von offiziellen wetter.com Logos								

### Wetterdaten

Wertname	Beschreibung
du	Unixtimestamp Unicode [int unsigned]
d	Unixtimestamp lokal [int unsigned]
dhu	ISO 8601 formatiertes Datum und Zeit, UTC [string]
dhl	ISO 8601 formatiertes Datum und Zeit, UTC [string]
p	Gültigkeitszeitraum der Prognose (24 Stunden, 1 Stunde, 3 Stunden, ...) [int unsigned, 0-24]
w	Code für Wetterzustand [int unsigned, 0-999] Siehe „Definition Wetterzustände“
pc	Änderung des Luftdrucks [float 1 Nachkommastelle]
wd	Windrichtung in Grad [int, 0=N, 90=O, 180=S, 270=W]
ws	Windgeschwindigkeit in Kilometer/Stunde [float unsigned, 1 Nachkommastelle]
wd_txt	Windrichtung in Textform (N,NO,S,SW ...) [string]
w_txt	Wetterzustand in Textform [string]
tn	Mindesttemperatur
tx	Maximaltemperatur

forecast time part	Für jeden Tag gibt es 4 verschiedene Tagesabschnitte.
	Morgens 6 -11Uhr
	Mittags 11 - 17Uhr
	Abends 17 - 23Uhr
	Nachts 23 - 6 Uhr
	Da es sein kann, dass sich die Abschnitte in Zukunft ändern, ist es wichtig nicht fest auf die jeweiligen Zeiträume zu mappen.

## Ausgabe Json

```
{
  {
    city: {
      city_code: "DE0002538",
      name: "Elmshorn",
      url: "deutschland/elmshorn/DE0002538.html",
      credit: {
        info: "In order to use the free weather data from wetter.com you
HAVE TO display at least two out of three of the following possibilities: text, link, logo",
        text: "Powered by wetter.com",
        link: "http://www.wetter.com",
        logo: "Download at http://www.wetter.com/api/downloads/#logos"
      },
      forecast: {
        2014-02-17: {
          wd: "225",
          w: "3",
          tx: "8",
          pc: "30",
          06:00: {
            wd: "225",
            w: "2",
            tx: "5",
            pc: "30",
            tn: "2",
            p: "5",
            dhl: "2014-02-17 06:00",
            ws: "9",
            dhu: "2014-02-17 05:00",
            du: "1392613200",
            d: "1392616800",
            wd_txt: "SW",
            w_txt: "wolkig"
          },
          11:00: {},
          17:00: {},
          23:00: {},
          d: "1392616800",
          tn: "2",
          p: "24",
          dhl: "2014-02-17 06:00",
          ws: "8",
        }
      }
    }
  }
}
```



```

        dhu: "2014-02-17 05:00",
        du: "1392613200",
        wd_txt: "SW",
        w_txt: "bedeckt"
    }
}
}

```

## Ausgabe XML

```

<city>
  <city_code>DE0002538</city_code>
  <name>Elmshorn</name>
  <url>deutschland/elmshorn/DE0002538.html</url>
  <post_code>25335</post_code>
  <credit>
    <!--
      In order to use the free weather data from wetter.com you HAVE TO display at
least two out of three of the following possibilities: text, link, logo
    -->
    <text>Powered by wetter.com</text>
    <link>http://www.wetter.com</link>
    <logo>
      Download at http://www.wetter.com/api/downloads/#logos
    </logo>
  </credit>
  <forecast>
    <date value="2014-02-17">
      <wd>225</wd>
      <w>3</w>
      <tx>8</tx>
      <pc>30</pc>
      <time value="06:00">
        <wd>225</wd>
        <w>2</w>
        <tx>5</tx>
        <pc>30</pc>
        <tn>2</tn>
        <p>5</p>
        <dhl>2014-02-17 06:00</dhl>
        <ws>9</ws>
        <dhu>2014-02-17 05:00</dhu>
        <du>1392613200</du>
        <d>1392616800</d>
        <wd_txt>SW</wd_txt>
        <w_txt>wolkig</w_txt>
      </time>
      <time value="11:00">...</time>
      <time value="17:00">...</time>
      <time value="23:00">...</time>
      <d>1392616800</d>
      <tn>2</tn>
      <p>24</p>
      <dhl>2014-02-17 06:00</dhl>
      <ws>8</ws>
      <dhu>2014-02-17 05:00</dhu>
      <du>1392613200</du>
      <wd_txt>SW</wd_txt>
      <w_txt>bedeckt</w_txt>
    </date>
  </forecast></city>

```

## Ausgabe bei leerem Ergebnis oder Fehlerfall

### JSON

```
{
  error: {
    title: "Invalid access data",
    message: "Either your username or the checksum are invalid."
  }
}
```

### XML

```
<error>
  <title>Invalid access data</title>
  <message>Either your username or the checksum are invalid.</message>
</error>
```

## Anmerkungen

Es kann vorkommen, dass manche, besonders die kleineren, unbemannten Messstationen nicht alle Werte liefern können, die bei größeren Stationen verfügbar sind.

## Definition Wetterzustände

Die von der Vorhersage- und Aktuellwetterschnittstelle zurückgegebenen Werte enthalten Codes für den Wetterzustand (Parameter "w"). Diese Codes werden in der Rückgabe auch in Textform dargestellt (Parameter "w\_txt"), oft ist es aber wichtig eine komplette Liste der Codes zu haben.

Code (w)	Wetterzustand (w_txt)
0	sonnig
1	leicht bewölkt
2	wolkig
3	bedeckt
4	Nebel
5	Sprühregen
6	Regen
7	Schnee
8	Schauer
9	Gewitter
10	leicht bewölkt
20	wolkig
30	bedeckt
40	Nebel
45	Nebel
48	Nebel mit Reifbildung
49	Nebel mit Reifbildung"
50	Sprühregen

51	leichter Sprühregen
53	Sprühregen
55	starker Sprühregen
56	leichter Sprühregen, gefrierend
57	starker Sprühregen, gefrierend
60	leichter Regen
61	leichter Regen
63	mäßiger Regen
65	starker Regen
66	leichter Regen, gefrierend
67	mäßiger od. starker Regen, gefrierend
68	leichter Schnee-Regen
69	starker Schnee-Regen
70	leichter Schneefall
71	leichter Schneefall
73	mäßiger Schneefall
75	starker Schneefall
80	leichter Regen - Schauer
81	Regen - Schauer
82	starker Regen - Schauer
83	leichter Schnee / Regen - Schauer
84	starker Schnee / Regen - Schauer
85	leichter Schnee - Schauer
86	mäßiger od. starker Schnee - Schauer
90	Gewitter
95	leichtes Gewitter
96	starkes Gewitter
999	Keine Angabe

Die Codes von 0-9 sind hierbei die Hauptwetterzustände. Alle anderen Codes sind Varianten dieser Zustände. Zum Beispiel ist "65: starker Regen" eine Variante von "6: Regen". Sollen nur die Hauptzustände angezeigt werden kann einfach die erste Stelle des Codes verwendet werden.

## Authentifizierung

Um eine Städtesuche oder eine Wettervorhersage für eine eigene Anwendung mit der `wetter.com` `openweather` API auszuführen, müssen die folgenden Voraussetzungen erfüllt sein:

- der Anwendungsersteller muss auf `www.wetter.com` registriert sein
- der Anwendungsersteller muss ein Projekt für seine Anwendung erstellt haben
- die Anfrage muss sich durch Übergabe verschiedener Parameter im GET-Request (in der URL) authentifizieren.

In diesem Dokument wird die dritte Voraussetzung näher beschrieben.

### Authentifizierung bei einer Suchanfrage

Einer Suchanfrage müssen drei Informationen übergeben werden, der zu suchende Begriff, der Projektname und eine Checksumme.

Die Checksumme ist ein MD5-Hash aus dem Projektnamen, dem API Key und dem Suchbegriff. Der Projektname sowie der API Key werden beim Anlegen eines Projekts festgelegt und können in den Projektinformationen eingesehen werden.

Das folgende Beispiel zeigt, wie in der Programmiersprache PHP ein Suchrequest zusammengebaut werden würde. Projektname ist hierbei "beispielprojekt", der API Key lautet "geheim" (in der Wirklichkeit wäre er eine längere Zeichenfolge aus HEX-Charakteren). Gesucht wird nach "Berlin".

```
<?php
$searchUrl  = 'http://api.wetter.com/location/index';
$projectName = 'beispielprojekt';
$apiKey     = 'geheim';
$searchString = 'Berlin';

// Generieren der Checksumme, muss für jeden neuen Suchstring berechnet werden
$checksum = md5($projectName . $apiKey . $searchString);

$searchUrl .= '/search/' . $searchString;
$searchUrl .= '/project/' . $projectName;
$searchUrl .= '/cs/' . $checksum;
?>
```

Am Ende des Scripts wäre `$searchUrl` gleich

**`http://api.wetter.com/location/index/search/Berlin/project/beispielprojekt/cs/528f01f0b631d3af6f52707dc8110d15`**

Natürlich sollte man in einem richtigen Projekt das ganze etwas dynamischer halten, aber zu Demonstrationszwecken sollte obiges Beispiel genügen.

Es spielt übrigens keine Rolle ob man, wie in obigem Beispiel, eine gemischte Suche, eine Textsuche oder eine reine Postleitzahlensuche ausführt, das Vorgehen zur Authentifizierung ist dabei immer gleich. Allein der Inhalt des Suchstrings ändert sich. PHP selbst ist keine streng typisierte Sprache, somit ist es egal ob `$searchString` eine Zeichenfolge oder eine Zahl ist. In anderen Programmiersprachen muss eine Postleitzahl, sollte sie als Integer oder ähnliches vorliegen, natürlich zunächst in einen String konvertiert werden.

### Authentifizierung bei einer Wettervorhersage

Das Vorgehen beim Abfragen der Wetterdaten für einen Ort ist dem bei einer Suchanfrage sehr ähnlich. Der URL werden ein wetter.com city code, der Projektname und eine Checksumme als GET-Parameter mitgegeben. Stimmen diese Informationen mit den für das Projekt gespeicherten überein, ist die Anfrage zulässig.

Auch hier wieder ein Beispiel in PHP für den Zusammenbau einer Wettervorhersage URL. Projektname für dieses Beispiel ist "beispielprojekt", der API Key lautet "geheim" und die Wettervorhersage wird für den wetter.com city code "DE0001020" (Berlin) angefordert. Der Projektname sowie der API Key werden beim Anlegen eines Projekts festgelegt und können in den Projektinformationen eingesehen werden.

```
<?php
$sForecastUrl = 'http://api.wetter.com/forecast/weather';
$sProjectName = 'beispielprojekt';
$sApiKey      = 'geheim';
$sCityCode    = 'DE0001020';

// Generieren der Checksumme, muss für jeden City Code neu berechnet werden
$sChecksum = md5($sProjectName . $sApiKey . $sCityCode);

$sForecastUrl .= '/city/' . $sCityCode;
$sForecastUrl .= '/project/' . $sProjectName;
$sForecastUrl .= '/cs/' . $sChecksum;
?>
```

Am Ende des Scripts wäre \$sSearchUrl gleich

```
http://api.wetter.com/forecast/weather/city/DE0001020/project/beispielprojekt/cs/3c551bc20819c19ee88c9ec94280a61d
```

Falls noch Fragen oder Probleme zu unserer Schnittstelle auftauchen, können Sie uns gerne eine Email an [gadget@wetter.com](mailto:gadget@wetter.com) schreiben.