

WetterOnline Wetter-API – Schnittstellendokumentation

Version: v=1

Inhalt

WetterOnline Wetter-API – Schnittstellendokumentation	1
1 Einleitung	2
2 Registrierung.....	2
3 Grundlagen	2
3.1 Versionsnummer	2
3.2 API-Anfragen	2
4 Authentifizierung	3
5 Suchen	4
6 Wetter.....	6
6.1 Aktuelles Wetter / package=current.....	7
6.2 Tägliche Vorhersage / package=daily8, package=daily14	8
6.3 Intervallbasierte Vorhersage / package=hourly72, package=day6parts.....	10
6.4 Wetterzustand der Vortage / package=yesterday, package=past3days.....	14
6.5 Vorhersagetexte.....	17
7 Weitere Formate.....	18
7.1 XML-Format	18
8 Kodierung des Wetterzustandes.....	18
9 Beispiel-Code	20
9.1 PHP-Beispiel für Suchanfrage.....	20
9.2 PHP-Beispiel für Wetterabfrage	21
9.3 PHP-Beispiel für Abfrage des Vorhersagetextes	22
10 Beispiel-Checksummen-Berechnung.....	24
10.1 Beispielberechnung einer Checksumme für eine Suchanfrage.....	24
10.2 Beispielberechnung einer Checksumme für einen package-Request	24

1 Einleitung

Mit der Wetter-API von WetterOnline können aktuelle Wetterinformationen und Prognosen für beliebige Orte weltweit abgefragt werden. Die Wetterinformationen stehen dabei für ortsspezifische Kennungen (Geo-IDs) zur Verfügung. Die Geo-IDs können über die zur Verfügung stehende [Suchfunktion für Geokoordinaten, Ortsnamen und Postleitzahlen \(Deutschland\)](#) abgefragt werden.

Die API-Abfragen sind über eine [Authentifizierungsmethode](#) geschützt.

Um die passenden Wetterinformationen für Ihre Anwendung zu erhalten, stehen verschiedene [Pakete](#) zur Verfügung.

2 Registrierung

Zur Nutzung der Wetter-API werden Ihnen von WetterOnline Zugangsdaten, bestehend aus einer Nutzer-ID und einem geheimen Schlüssel, mitgeteilt. Der geheime Schlüssel darf Dritten nicht zugänglich gemacht werden.

3 Grundlagen

3.1 Versionsnummer

Die aktuelle Versionsnummer ist v=1. Dieser Parameter ist Bestandteil jeder API-Anfrage.

3.2 API-Anfragen

Anfragen an die WetterOnline Wetter-API erfolgen per http-GET-Methode. Die Schnittstelle gibt die Daten im JSON-Format zurück.

Ablauf zum Abruf von Wetterdaten:

1. [Suchen](#)
Suche nach gewünschtem Ort / Rückgabe der Geo-ID
Syntax:
`https://api.wetteronline.de/geo?name=<Ortsname/PLZ(Deutschland)>&v=<Versionsnummer>`
`https://api.wetteronline.de/geo?lat=<Breitengrad>&lon=<Längengrad>&v=<Versionsnummer>`
2. [Wetterinformationen abfragen](#)
Die gewünschten Wetterinformationen (package) für eine oder mehrere bekannte Geo-IDs abfragen.
Syntax:
`https://api.wetteronline.de/weather?package=<packagename>&gid=<GeoID>&v=<Versionsnummer>`

Jeder Anfrage müssen weitere Parameter zur [Authentifizierung](#) hinzugefügt werden.

4 Authentifizierung

Die API-Anfragen sind mittels einer Authentifizierungsmethode geschützt.

Zur Authentifizierung werden der geheime Schlüssel sowie alle Parameter des API-Aufrufes, die Base64-kodierte Nutzer-ID (Parametername: **uid**) und der aktuelle Zeitstempel in UTC (Parametername: **date**) im Format YYYY-MM-DD-hh benötigt. Mit den Werten der Parameter und dem geheimen Schlüssel wird eine Checksumme (Parametername: **checksum**) generiert und dem API-Aufruf als weiterer Parameter angehängt. Alle Parameter des API-Aufrufes müssen URL-kodiert (Prozentkodierung) werden.

Die Checksumme wird mit folgendem Verfahren generiert:

1. Alle Parameter des API-Aufrufs werden nach dem Parameternamen alphabetisch sortiert.
Syntax:
`https://api.wetteronline.de/weather?package=<packagename>&gid=<Geo-ID>&v=<Versionsnummer>&uid=<Base64(Nutzer-ID)>&date=<YYYY-MM-DD-hh>`
Sortierung: date, gid, package, uid, v
2. Die Werte dieser Parameter werden aneinandergehängt und durch ein "|" (Pipe) getrennt.
hier: `<YYYY-MM-DD-hh>|<Geo-ID>|<packagename>|<Base64(Nutzer-ID)>|<Versionsnummer>`
3. Zusätzlich wird der geheime Schlüssel angehängt, ebenfalls durch "|" getrennt.
hier:
`<YYYY-MM-DD-hh>|<Geo-ID>|<packagename>|<Base64(Nutzer-ID)>|<Versionsnummer>|<geheimer Schlüssel>`
4. Erstellen Sie anschließend den MD5-Hash über die Byte-Sequenz (im Raw Binary Format, nicht Hexadezimal-Darstellung nutzen!) dieser Zeichenkette und konvertieren Sie das Ergebnis in einen Base64-kodierten String.
5. Der erhaltene String wird URL-kodiert (Prozentkodierung) in der API-Anfrage als Wert für den Parameter **checksum** eingetragen.
`https://api.wetteronline.de/weather?package=<packagename>&gid=<Geo-ID>&v=<Versionsnummer>&uid=<Base64(Nutzer-ID)>&date=<YYYY-MM-DD-hh>&checksum=<Checksumme>`

[Beispielberechnungen für die Checksumme](#) finden sich im Anhang.

Anmerkungen:

Das Datum zur Verschlüsselung ist immer in UTC-Zeit. Diese kann auch über <https://api.wetteronline.de/weather?package=date> abgerufen werden (für diesen Aufruf ist keine Authentifizierung notwendig).

5 Suchen

Um Wetterinformationen abzufragen, wird die Geo-ID des gewünschten Ortes benötigt. Dazu wird eine Suchfunktion angeboten, die es ermöglicht, alle Orte über den Namen, die Postleitzahl (Deutschland) oder die Geo-Position in Längen- und Breitengrad zu finden.

Der Aufruf der Suche erfolgt über:

`https://api.wetteronline.de/geo?name=<Ortsname/Postleitzahl(Deutschland)>&v=<Versionsnummer>`
`https://api.wetteronline.de/geo?lat=<Breitengrad>&lon=<Längengrad>&v=<Versionsnummer>`

Parametername	Bedeutung	Gültige Werte
name	Name oder Postleitzahl (Deutschland) des gesuchten Ortes	alpha-numerisch, UTF-8
lat	Breitengrad	Float mit bis zu 4 Nachkommastellen 90.0, ... -90.0
lon	Längengrad	Float mit bis zu 4 Nachkommastellen 180.0, ... -180.0

Für einen gültigen Aufruf müssen wie oben die Nutzer-ID, das aktuelle Datum und eine MD5-Checksumme als Parameter mitgeliefert werden (Siehe [Authentifizierung](#)).

`https://api.wetteronline.de/geo?name=<Ortsname/Postleitzahl(Deutschland)>&v=<Versionsnummer>&uid=<Nutzer-ID>&date=<YYYY-MM-DD-hh>&checksum=<Checksumme>`
`https://api.wetteronline.de/geo?lat=<Breitengrad>&lon=<Längengrad>&v=<Versionsnummer>&uid=<Nutzer-ID>&date=<YYYY-MM-DD-hh>&checksum=<Checksumme>`

Als Antwort auf die Suchanfrage wird ein String zurückgeliefert. Dieser kann mit einem JSON-Parser in ein Array von JSON-Objekten umgewandelt werden.

Die Rückgabe enthält folgende Parameter:

Parametername	Bedeutung	Gültige Werte
locationName	Name des Ortes	alpha-numerisch
subLocationName	Name des Ortsteils	alpha-numerisch Defaultwert: null
gid	WetterOnline Geo-ID des Ortes	5-stellig, alpha-numerisch
subStateID	Kürzel des Bundeslandes, Bundesstaates (kein ISO-Code)	alpha-numerisch Defaultwert: null
subStateName	Name des Bundeslandes, Bundesstaates	alpha-numerisch Defaultwert: null
stateID	Kürzel des Landes (kein ISO-Code)	2-stellig, alpha-numerisch
stateName	Name des Landes	alpha-numerisch

latitude	Breitengrad des Ortes	Float mit vier Nachkommastellen
longitude	Längengrad des Ortes	Float mit vier Nachkommastellen
altitude	Höhe über NN in Metern	Integer
zip	Postleitzahl des Ortes / Ortsteils	5-stellig, Integer Defaultwert: null

Beispiel: Suche nach Neukölln

Das Ergebnis ist wie folgt aufgebaut:

```
[
  {
    "locationName": "Berlin",           // Name des Ortes
    "subLocationName": "Neukölln",     // Name des Ortsteils (falls bekannt)
    "gid": "10382",                    // WetterOnline Geo-ID
    "subStateID": "BER",                // Kürzel des Bundeslandes (falls bekannt)
    "subStateName": "Berlin",          // Name des Bundeslandes (falls bekannt)
    "stateID": "DL",                   // Kürzel des Landes
    "stateName": "Deutschland",        // Name des Landes
    "latitude": 52.5166,                // Breitengrad
    "longitude": 13.3999,               // Längengrad
    "altitude": 34,                     // Höhe über NN
    "zip": "10965",                    // Postleitzahl (falls bekannt)
  },
  {
    "match": "yes"
  }
]
```

Sollte ein Wert nicht bekannt sein, wird als Defaultwert null geliefert.

Mit einem zusätzlichen Aufruf des Parameters **function=url** kann außerdem die URL zur Landingpage des gesuchten Ortes bei WetterOnline abgerufen werden.

Der Parametername in der Rückgabedatei lautet **url**.

Optional kann die gesuchten Orte auch in der Ortssprache zurückgegeben werden. Um diese Option zu nutzen wird der Link um den Parameter **lang=<Sprache als ISO 639-1 Code>** erweitert. Auf diese Weise können Orte auch in der jeweiligen Landessprache gefunden und angezeigt werden.

Sollte es kein Suchergebnis geben, dann enthält das Array nur ein Objekt mit der Eigenschaft "match" und dem Wert "no". Bei einem oder mehreren Suchergebnissen werden die einzelnen JSON-Objekte hintereinander gehängt und das letzte JSON-Objekt enthält die Eigenschaft "match" mit dem Wert "yes".

WICHTIG: Die Suche akzeptiert nur Aufrufe im UTF-8 Format

Geo-IDs ändern sich nicht. Es bietet sich daher an, die Geo-ID für bekannte Orte in der Anwendung zu speichern, so dass bei späteren Abfragen der Wetterinformationen des gleichen Ortes nicht eine erneute Suche notwendig ist.

Ein [Beispielcode](#) für die Suchfunktion findet sich am Ende des Dokumentes.

6 Wetter

Der allgemeine Aufruf zur Abfrage von Wetterinformationen lautet:

`https://api.wetteronline.de/weather?gid=<Geo-ID>&package=<Packagename>&v=<Versionsnummer>`

Bei der Abfrage nach dem Wetter wird zwischen verschiedenen Paketen (Parametername: *package*) unterschieden. Es gibt die Pakete **current**, **daily8**, **daily14**, **day6parts**, **hourly72**, **yesterday** und **past3days**.

Alle Aufrufe erwarten als Parameter eine Geo-ID (Parametername: *gid*). Die Geo-ID erhält man über die [Suche](#). Bei allen Paketen können die Daten für eine oder mehrere Geo-IDs (maximal 20) abgefragt werden. Mehrere Geo-IDs sind mit Kommata getrennt aneinander zu hängen. Werden mehr als 20 Geo-IDs in einem Aufruf abgefragt, werden nur die ersten 20 Geo-IDs berücksichtigt.

Geo-IDs ändern sich nicht. Es bietet sich daher an, die Geo-ID für bekannte Orte in der Anwendung zu speichern, so dass bei späteren Abfragen der Wetterinformationen des gleichen Ortes nicht eine erneute Suche notwendig ist.

Für einen gültigen Aufruf müssen noch die Nutzer-ID (*uid*), das Datum (*date*) und eine MD5-Checksumme als *get*-Parameter mitgeliefert werden (Siehe [Authentifizierung](#)).

Die Rückgabe dieser Abfrage erfolgt im JSON-Format.

Dabei ist der Aufbau der Rückgabe wie folgt:

```
{
  "<Geo-ID1>": {
    "data": {                                     // Enthält ein Objekt oder ein Array von Objekten mit
                                                    // Wetterinformationen
      "Eigenschaft1": "<Eigenschaftswert1>", (Eigenschaften) zu der Geo-ID
      ...,
      "EigenschaftN": "<EigenschaftswertN>",
    },
    "meta": {                                     // Enthält Metainformationen zur Geo-ID
      "local_date": "YYYY-MM-DD hh:mm:ss"      // Zeitstempel der Daten: Datum und Uhrzeit in Ortszeit der
                                                    // angefragten geoID
    }
  },
  "<Geo-ID2>": {
    "data": { ...
    },
    "meta": { ...
    }
  },
  ...
  "<Geo-IDN>": {
    "data": { ...
    },
    "meta": { ...
    }
  },
  "info": {                                     // Enthält Metainformationen zu dieser Datei
    "locations": [
      "<Geo-ID1>",
      ...
      "<Geo-IDN>"
    ],
    "type": "<type-Wert>"                       // Produktionstyp bei WetterOnline
  }
}
```

Ein Datenobjekt kann dabei auch selbst ein Array von weiteren Datenobjekten enthalten.

Die Beschreibung der möglichen Wetterinformationen wird in den folgenden Abschnitten für jedes verfügbare Paket gegeben.

Ein [Beispielcode](#) für die Abfrage der Wetterinformationen findet sich am Ende des Dokumentes.

6.1 Aktuelles Wetter / package=current

Der Aufruf für die Abfrage des aktuellen Wetters lautet:

`https://api.wetteronline.de/weather?package=current&gid=<Geo-ID>&v=<Versionsnummer>`

Für einen gültigen Aufruf müssen noch die Nutzer-ID (uid), das Datum (date) und eine MD5-Checksumme als get-Parameter mitgeliefert werden (Siehe [Authentifizierung](#)).

Die Rückgabe enthält folgende Wetterinformationen für den aktuellen Zeitpunkt:

Parametername	Bedeutung	Gültige Werte
wm	Key für Wetterzustand	Siehe Kapitel 7
tt_C	aktuelle Temperatur in °C	Float mit einer Nachkommastelle
tta_C	gefühlte Temperatur in °C	Float mit einer Nachkommastelle
dd_dir	Windrichtung in Sektoren	n, nw, w, sw, s, so, o, no, 0 (umlaufend)
dd_deg	Windrichtung in Bogengrad	Integer zwischen 0 (windstill) und 360 (Nord) in 5er Schritten
ff_ms	Windgeschwindigkeit in m/s	Float mit einer Nachkommastelle
ff_bft	Windgeschwindigkeit in Beaufort	Integer
fx_kmh	maximale Windböe in km/h	Integer
fx_bft	maximale Windböe in Beaufort	Integer
pp_hpa	Bodendruck in hPa	Float mit einer Nachkommastelle
rh	relative Feuchte in Prozent	Integer zwischen 0 und 100 in 5er Schritten
pop	Niederschlagswahrscheinlichkeit in Prozent	Integer zwischen 0 und 100 in 5er Schritten
rr_mm	Niederschlagsmenge in mm	Float mit einer Nachkommastelle
rad_wm2	Globalstrahlung in W/m ²	Integer in 5er Schritten

Dabei können die Wetterinformation für einen oder mehrere Orte (Geo-IDs) abgefragt werden. Die Rückgabe dieser Abfrage erfolgt im JSON-Format.

Beispiel für eine Antwort des Servers auf eine Abfrage des aktuellen Wetters:

```
{
  "10382" : {
    "data" : {
      "dd_deg" : 140,
      "dd_dir" : "so",
      "ff_bft" : 2,
      "ff_ms" : 2.90,
      "fx_bft" : 4,
      "fx_kmh" : 20,
      "pop" : 5,

```

```

    "pp_hpa" : 1016.0,
    "rad_wm2" : 945,
    "rh" : 90,
    "rr_mm" : 0.0,
    "tt_C" : 25.20,
    "tta_C" : 25.20,
    "wm" : "so____"
  },
  "meta" : {
    "local_date" : "2014-05-21 11:39:57"
  }
},
"info" : {
  "locations" : [ "10382" ],
  "type" : "city"
}
}

```

6.2 Tägliche Vorhersage / package=daily8, package=daily14

Der Aufruf für die Abfrage der Wettervorhersage in täglicher Auflösung lautet:

<https://api.wetteronline.de/weather?package=daily8&gid=<GeoID>&v=<Versionsnummer>>
bzw. <https://api.wetteronline.de/weather?package=daily14&gid=<GeoID>&v=<Versionsnummer>>

Dabei kann das Wetter für **acht Tage** bzw. **vierzehn Tage** abgefragt werden.

Auch bei diesem Aufruf ist eine [Authentifizierung](#) notwendig.

Die Rückgabe enthält folgende Wetterinformationen für den jeweiligen Ortstag:

Parametername	Bedeutung	Gültige Werte
wm	Key für Wetterzustand	Siehe Kapitel 7
tn_C	Minimumtemperatur (in der Nacht zum angegebenen Ortstag) in °C	Float mit einer Nachkommastelle
tg_C	Minimum der Erdbodentemperatur (5 cm über Erdboden, in der Nacht zum angegebenen Ortstag) in °C	Float mit einer Nachkommastelle
tx_C	Maximumtemperatur des Tages in °C	Float mit einer Nachkommastelle
dd_dir	Windrichtung in Sektoren (Mittel des Tages)	n, nw, w, sw, s, so, o, no, 0 (umlaufend)
dd_deg	Windrichtung in Bogengrad (Mittel des Tages)	Integer zwischen 0 (windstill) und 360 (Nord) in 5er Schritten
ff_ms	Windgeschwindigkeit in m/s (Mittel des Tages)	Float mit einer Nachkommastelle
ff_bft	Windgeschwindigkeit in Beaufort (Mittel des Tages)	Integer
fx_kmh	maximale Windböe in km/h	Integer
fx_bft	maximale Windböe in Beaufort	Integer
pp_hpa	Bodendruck in hPa (Mittel des Tages)	Float mit einer Nachkommastelle

rh	relative Feuchte in Prozent (Mittel des Tages)	Integer zwischen 0 und 100 in 5er Schritten
pop	Niederschlagswahrscheinlichkeit in Prozent	Integer zwischen 0 und 100 in 5er Schritten
rr_mm	Niederschlagsmenge in mm	Float mit einer Nachkommastelle
pd_h	Niederschlagsdauer in Stunden	Float mit einer Nachkommastelle
rad_wm2	Globalstrahlung in W/m ²	Integer in 5er Schritten
sd_h	Sonnenscheindauer in Stunden	Float mit einer Nachkommastelle
sr	Sonnenaufgangszeit (Ortszeit des Prognoseortes)	hh:mm
ss	Sonnenuntergangszeit (Ortszeit des Prognoseortes)	hh:mm
mr	Mondaufgangszeit (Ortszeit des Prognoseortes)	hh:mm
ms	Monduntergangszeit (Ortszeit des Prognoseortes)	hh:mm
date	Datum (Ortszeit des Prognoseortes)	YYYY-MM-DD
weekday	Wochentagsbezeichnung von date (Ortszeit des Prognoseortes)	Montag, ..., Sonntag

Beispiel für eine Antwort des Servers auf eine Abfrage der täglichen Vorhersage (gekürzt):

```
{
  "10382": {
    "data": [
      {
        "date": "2014-03-25",
        "dd_deg": 310,
        "dd_dir": "nw",
        "ff_bft": 2,
        "ff_ms": 2.9,
        "fx_bft": 5,
        "fx_kmh": 34,
        "mr": "02:47",
        "ms": "11:59",
        "pd_h": 1,
        "pop": 60,
        "pp_hpa": 1009.4,
        "rad_wm2": 125,
        "rr_mm": 0.8,
        "sd_h": 1.5,
        "sr": "05:58",
        "ss": "18:28",
        "tg_C": 1.8,
        "tn_C": 1.9,
        "tx_C": 8.2,
        "weekday": "Dienstag",
        "wm": "bdr1__"
      }
    ]
  }
}
```

```

        {
            ZWEITER TAG
        },
        ...,
        {
            LETZTER TAG
        }
    ],
    "meta": {
        "local_date": "2014-03-25 08:32:45"
    }
},
"info": {
    "type": "city"
}

```

Das JSON-Objekt kann auch Daten für mehrere Geo-IDs liefern:

```

{
    Geo-ID1:{...}, Geo-ID2:{...}, ..info:{...}
}

```

Das Array **data**, das Teil jedes Geo-ID-Objektes ist, enthält ein JSON-Objekt für jeden Tag, beginnend mit dem aktuellen Tag. **local_date** ist dabei der Zeitstempel der Daten (Datum und Uhrzeit in Ortszeit) der angefragten Geo-ID.

6.3 Intervallbasierte Vorhersage / package=hourly72, package=day6parts

Die intervallbasierte Vorhersage-Abfrage lautet:

<https://api.wetteronline.de/weather?package={hourly72|day6parts}&gid=<GeoID>>

Diese Pakete bestehen aus einer täglichen Vorhersage (siehe [package=daily8](#)) und zusätzlichen Intervallinformationen (periods).

hourly72 enthält für 3 Tage zusätzlich stündliche Wetterinformationen, beginnend mit dem aktuellen Tag um 0 Uhr Ortszeit (gesamt 72 Stunden).

day6parts enthält für 6 Tage zusätzliche 6h-Intervall-Informationen. Auch hier beginnt der abgedeckte Zeitraum mit dem aktuellen Tag um 0 Uhr Ortszeit.

Der Aufbau der Daten ist identisch zu dem der [täglichen Vorhersage](#). Das Array **data** wird dabei um ein Array **periods** erweitert. Dieses enthält JSON-Objekte für jede Stunde des Tages bzw. jeden Tagesabschnitt.

period_duration gibt dabei die Länge des Intervalls in Stunden an. Der Beginn des Intervalls wird durch das Element **hour** definiert (z. B.: hour=3 -> Intervall beginnt um 3 Uhr). Die Bezeichnung des Intervalls ist in **periodname** enthalten.

Die Rückgabe enthält folgende Wetterinformationen:

Für jeden Tag:

Parametername	Bedeutung	Gültige Werte
wm	Key für Wetterzustand	Siehe Kapitel 7
tn_C	Minimumtemperatur (in der Nacht zum angegebenen Ortstag) in °C	Float mit einer Nachkommastelle
tg_C	Minimum der Erdbodentemperatur (5 cm über Erdboden, in der Nacht zum angegebenen Ortstag) in °C	Float mit einer Nachkommastelle

tx_C	Maximumtemperatur des Tages in °C	Float mit einer Nachkommastelle
dd_dir	Windrichtung in Sektoren (Mittel des Tages)	n, nw, w, sw, s, so, o, no, 0 (umlaufend)
dd_deg	Windrichtung in Bogengrad (Mittel des Tages)	Integer zwischen 0 (windstill) und 360 (Nord) in 5er Schritten
ff_ms	Windgeschwindigkeit in m/s (Mittel des Tages)	Float mit einer Nachkommastelle
ff_bft	Windgeschwindigkeit in Beaufort (Mittel des Tages)	Integer
fx_kmh	maximale Windböe in km/h	Integer
fx_bft	maximale Windböe in Beaufort	Integer
pp_hpa	Bodendruck in hPa (Mittel des Tages)	Float mit einer Nachkommastelle
rh	relative Feuchte in Prozent (Mittel des Tages)	Integer zwischen 0 und 100 in 5er Schritten
pop	Niederschlagswahrscheinlichkeit	Integer zwischen 0 und 100 in 5er Schritten
rr_mm	Niederschlagsmenge in mm	Float mit einer Nachkommastelle
pd_h	Niederschlagsdauer in Stunden	Float mit einer Nachkommastelle
rad_wm2	Globalstrahlung in W/m ² (Mittel des Tages)	Integer in 5er Schritten
sd_h	Sonnenscheindauer in Stunden	Float mit einer Nachkommastelle
sr	Sonnenaufgangszeit (Ortszeit des Prognoseortes)	hh:mm
ss	Sonnenuntergangszeit (Ortszeit des Prognoseortes)	hh:mm
mr	Mondaufgangszeit (Ortszeit des Prognoseortes)	hh:mm
ms	Monduntergangszeit (Ortszeit des Prognoseortes)	hh:mm
date	Datum (Ortszeit des Prognoseortes)	YYYY-MM-DD
weekday	Wochentagsbezeichnung von date (Ortszeit des Prognoseortes)	Montag, ..., Sonntag
periods	enthält ein Array von weiteren Wetterinformationen, die für das durch hour und period_duration definierte Zeitintervall gelten	

Inhalt von "periods":

wm	Key für Wetterzustand	Siehe Kapitel 7
tt_C	Temperatur in °C	Float mit einer Nachkommastelle

tta_C	gefühlte Temperatur in °C	Float mit einer Nachkommastelle
dd_dir	Windrichtung in Sektoren (Mittel des Intervalls)	n, nw, w, sw, s, so, o, no
dd_deg	Windrichtung in Bogengrad (Mittel des Intervalls)	Integer zwischen 0 und 360 in 5er Schritten
ff_ms	Windgeschwindigkeit in m/s (Mittel des Intervalls)	Float mit einer Nachkommastelle
ff_bft	Windgeschwindigkeit in Beaufort (Mittel des Intervalls)	Integer
fx_kmh	maximale Windböe in km/h	Integer
fx_bft	maximale Windböe in Beaufort	Integer
pp_hpa	Bodendruck in hPa (Mittel des Intervalls)	Float mit einer Nachkommastelle
rh	relative Feuchte in Prozent (Mittel des Intervalls)	Integer zwischen 0 und 100 in 5er Schritten
pop	Niederschlagswahrscheinlichkeit in Prozent	Integer zwischen 0 und 100 in 5er Schritten
rr_mm	Niederschlagsmenge in mm	Float mit einer Nachkommastelle
pd_min* / pd_h**	Niederschlagsdauer in Minuten* / in Stunden**	Integer / Float mit einer Nachkommastelle
rad_wm2	Globalstrahlung in W/m ² (Mittel des Intervalls)	Integer in 5er Schritten
sd_min* /sd_h**	Sonnenscheindauer in Minuten* / in Stunden**	Integer / Float mit einer Nachkommastelle
hour	Uhrzeit des Intervall-Beginns (in Ortszeit)	Integer
period_duration	Dauer des Intervalls (in Stunden)	Integer
periodname	Intervall-Name	Vormittag, Nachmittag, Abend, Nacht, 00:00, ..., 23:00

* gilt bei package=hourly72

** gilt bei package=day6parts

Auch bei diesen Aufrufen ist eine Authentifizierung notwendig.

Beispiel für eine Antwort des Servers auf eine Abfrage der stündlichen Vorhersage (gekürzt):

```
{
  "10382": {
    "data": [
      {
        "date": "2014-03-25",
        "dd_deg": 310,
        "dd_dir": "nw",
        "ff_bft": 2,
        "ff_ms": 2.9,
        "fx_bft": 5,
        "fx_kmh": 34,
        "mr": "02:47",
        "ms": "11:59",
        "pd_h": 1,

```

```

    "periods": [
      {
        "dd_deg": 340,
        "dd_dir": "n",
        "ff_bft": 2,
        "ff_ms": 2,
        "fx_bft": 3,
        "fx_kmh": 14,
        "hour": 0,
        "pd_min": 0,
        "period_duration": 1,
        "periodname": "00:00",
        "pop": 35,
        "pp_hpa": 1011.5,
        "rad_wm2": 0,
        "rh": 85,
        "rr_mm": 0,
        "sd_min": 0,
        "tt_C": 4.5,
        "tta_C": 1.9,
        "wm": "md"
      }, { STUNDE 2}, ... {STUNDE 24}
    ],
    "pop": 60,
    "pp_hpa": 1009.4,
    "rad_wm2": 125,
    "rr_mm": 1.1,
    "sd_h": 1.5,
    "sr": "05:58",
    "ss": "18:28",
    "tg_C": 1.8,
    "tn_C": 1.9,
    "tx_C": 8.2,
    "weekday": "Dienstag",
    "wm": "bws1__"
  }, {TAG 2}, {TAG3}
],
  "meta": {
    "local_date": "2014-03-25 09:23:25"
  }
},
"info": {
  "type": "city"
}
}

```

Beispiel für eine Antwort des Servers auf eine Abfrage der Tagesabschnittvorhersage:

```

{
  "10382": {
    "data": [
      {
        "date": "2014-03-25",
        "dd_deg": 310,
        "dd_dir": "nw",
        "ff_bft": 2,
        "ff_ms": 2.9,
        "fx_bft": 5,
        "fx_kmh": 34,
        "mx": "02:47",
        "ms": "11:59",
        "pd_h": 1,
        "periods": [
          {
            "dd_deg": 335,
            "dd_dir": "nw",
            "ff_bft": 2,
            "ff_ms": 2.2,
            "fx_bft": 3,
            "fx_kmh": 14,

```

```

        "hour": 0,
        "pd_h": 0,
        "period_duration": 6,
        "periodname": "Nacht",
        "pop": 35,
        "pp_hpa": 1010.4,
        "rad_wm2": 0,
        "rh": 95,
        "rr_mm": 0,
        "sd_h": 0,
        "tt_C": 2.6,
        "wm": "mw____"
    }, { TAGESABSCHNITT2}, ... {TAGESABSCHNITT 4}
  ],
  "pop": 60,
  "pp_hpa": 1009.4,
  "rad_wm2": 125,
  "rr_mm": 1.1,
  "sd_h": 1.5,
  "sr": "05:58",
  "ss": "18:28",
  "tg_C": 1.8,
  "tn_C": 1.9,
  "tx_C": 8.2,
  "weekday": "Dienstag",
  "wm": "bws1__"
}, {TAG 2}, ..., {TAG6}
  ],
  "meta": null
},
"info": {
  "type": "city"
}
}

```

6.4 Wetterzustand des Vortages / package=yesterday, package=past3days

Die Abfrage der Wetterparameter des Vortages/der Vortage lautet:

<https://api.wetteronline.de/weather?package={yesterday|past3days}&gid=<Geo-ID>>

Diese Pakete liefern die analysierten Wetterparameter des Vortages/der Vortage als Tageswerte und in stündlicher Auflösung.

Der Aufbau der Daten ist identisch zu dem der [intervallbasierten Vorhersage](#). Das Array **data** enthält dabei das Array **periods**, welches die JSON-Objekte für jede Stunde des Vortages enthält. **period_duration** gibt die Länge des Intervalls in Stunden an. Der Beginn des Intervalls wird durch das Element **hour** definiert (z. B.: hour=3 -> Intervall beginnt um 3 Uhr). Die Bezeichnung des Intervalls ist in **periodname** enthalten.

Die Rückgabe enthält folgende Wetterinformationen:

Für den Vortag/die Vortage:

Parametername	Bedeutung	Gültige Werte
wm	Key für Wetterzustand	Siehe Kapitel 7
tn_C	Minimumtemperatur (in der Nacht zum angegebenen Ortstag) in °C	Float mit einer Nachkommastelle
tg_C	Minimum der Erdbodentemperatur (5 cm über Erdboden, in der Nacht zum angegebenen Ortstag) in °C	Float mit einer Nachkommastelle
tx_C	Maximumtemperatur des Tages in °C	Float mit einer Nachkommastelle

dd_dir	Windrichtung in Sektoren (Mittel des Tages)	n, nw, w, sw, s, so, o, no, 0 (umlaufend)
dd_deg	Windrichtung in Bogengrad (Mittel des Tages)	Integer zwischen 0 (windstill) und 360 (Nord) in 5er Schritten
ff_ms	Windgeschwindigkeit in m/s (Mittel des Tages)	Float mit einer Nachkommastelle
ff_bft	Windgeschwindigkeit in Beaufort (Mittel des Tages)	Integer
fx_kmh	maximale Windböe in km/h	Integer
fx_bft	maximale Windböe in Beaufort	Integer
pp_hpa	Bodendruck in hPa (Mittel des Tages)	Float mit einer Nachkommastelle
rh	relative Feuchte in Prozent (Mittel des Tages)	Integer zwischen 0 und 100 in 5er Schritten
pop	Niederschlagswahrscheinlichkeit	Integer zwischen 0 und 100 in 5er Schritten
rr_mm	Niederschlagsmenge in mm	Float mit einer Nachkommastelle
pd_h	Niederschlagsdauer in Stunden	Float mit einer Nachkommastelle
rad_wm2	Globalstrahlung in W/m ² (Mittel des Tages)	Integer in 5er Schritten
sd_h	Sonnenscheindauer in Stunden	Float mit einer Nachkommastelle
sr	Sonnenaufgangszeit (Ortszeit des Prognoseortes)	hh:mm
ss	Sonnenuntergangszeit (Ortszeit des Prognoseortes)	hh:mm
mr	Mondaufgangszeit (Ortszeit des Prognoseortes)	hh:mm
ms	Monduntergangszeit (Ortszeit des Prognoseortes)	hh:mm
date	Datum (Ortszeit des Prognoseortes)	YYYY-MM-DD
weekday	Wochentagsbezeichnung von date (Ortszeit des Prognoseortes)	Montag, ..., Sonntag
periods	enthält ein Array von weiteren Wetterinformationen, die für das durch hour und period_duration definierte Zeitintervall gelten	

Inhalt von "periods":

wm	Key für Wetterzustand	Siehe Kapitel 7
tt_C	Temperatur in °C	Float mit einer Nachkommastelle
tta_C	gefühlte Temperatur in °C	Float mit einer Nachkommastelle

dd_dir	Windrichtung in Sektoren (Mittel des Intervalls)	n, nw, w, sw, s, so, o, no
dd_deg	Windrichtung in Bogengrad (Mittel des Intervalls)	Integer zwischen 0 und 360 in 5er Schritten
ff_ms	Windgeschwindigkeit in m/s (Mittel des Intervalls)	Float mit einer Nachkommastelle
ff_bft	Windgeschwindigkeit in Beaufort (Mittel des Intervalls)	Integer
fx_kmh	maximale Windböe in km/h	Integer
fx_bft	maximale Windböe in Beaufort	Integer
pp_hpa	Bodendruck in hPa	Float mit einer Nachkommastelle
rh	relative Feuchte in Prozent	Integer zwischen 0 und 100 in 5er Schritten
pop	Niederschlagswahrscheinlichkeit in Prozent	Integer zwischen 0 und 100 in 5er Schritten
rr_mm	Niederschlagsmenge in mm	Float mit einer Nachkommastelle
pd_min	Niederschlagsdauer in Minuten	Integer
rad_wm2	Globalstrahlung in W/m ²	Integer in 5er Schritten
sd_min	Sonnenscheindauer in Minuten	Integer
hour	Uhrzeit des Intervall-Beginns (in Ortszeit)	Integer
period_duration	Dauer des Intervalls (in Stunden)	Integer
periodname	Intervall-Name	00:00, ..., 23:00

Auch bei diesen Aufrufen ist eine Authentifizierung notwendig.

Beispiel für eine Antwort des Servers auf eine Abfrage des Wetterzustandes am Vortag (gekürzt):

```
{
  "10382": {
    "data": [
      {
        "date": "2014-03-24",
        "dd_deg": 310,
        "dd_dir": "nw",
        "ff_bft": 2,
        "ff_ms": 2.9,
        "fx_bft": 5,
        "fx_kmh": 34,
        "mr": "02:47",
        "ms": "11:59",
        "pd_h": 1,
        "periods": [
          {
            "dd_deg": 340,
```



```

        "dd_dir": "n",
        "ff_bft": 2,
        "ff_ms": 2,
        "fx_bft": 3,
        "fx_kmh": 14,
        "hour": 0,
        "pd_min": 0,
        "period_duration": 1,
        "periodname": "00:00",
        "pop": 35,
        "pp_hpa": 1011.5,
        "rad_wm2": 0,
        "rh": 85,
        "rr_mm": 0,
        "sd_min": 0,
        "tt_C": 4.5,
        "tta_C": 1.9,
        "wm": "md____"
    }, { STUNDE 2}, ... {STUNDE 24}
],
"pop": 60,
"pp_hpa": 1009.4,
"rad_wm2": 125,
"rr_mm": 1.1,
"sd_h": 1.5,
"sr": "05:58",
"ss": "18:28",
"tg_C": 1.8,
"tn_C": 1.9,
"tx_C": 8.2,
"weekday": "Montag",
"wm": "bws1__"
}},
"meta": {
    "local_date": "2014-03-25 09:23:25"
}
},
"info": {
    "type": "city"
}
}

```

6.5 Vorhersagetexte

Vorhersagetexte sind mit dem Paket **text** abrufbar. Der allgemeine Aufruf lautet:

```
https://api.wetteronline.de/weather?package=text&gid=<Geo-ID>&locationname=<Ortsname>&v=<Versionsnummer>
```

Der Aufruf erwartet genau eine Geo-ID (Parametername: *gid*). Soll sich der Vorhersagetext auf einen speziellen Ortsteil beziehen, so ist der Name des Ortsteils ebenfalls anzugeben (Parametername: *locationname*).

Sowohl Geo-ID als auch Ortsname erhält man über die Suche. Letzterer steht im Objekt der Rückgabe unter dem Schlüssel *subLocationName*. Ist dieser auf *null* gesetzt, kann der Parameter weggelassen oder der Ortsname dem Schlüssel *locationName* entnommen werden. Es ist darauf zu achten, dass der Parameter *locationname* nicht mit einem unsinnigen Namen gesetzt wird. Im Zweifelsfall ist er besser gar nicht zu setzen.

Für einen gültigen Aufruf müssen noch die Nutzer-ID (*uid*), das Datum (*date*) und eine MD5-Checksumme als get-Parameter mitgeliefert werden (Siehe [Authentifizierung](#)). Der Parameter *locationname* ist ebenfalls in die Checksummenbildung mit einzubeziehen, sofern er nicht weggelassen wird.

Die Rückgabe dieser Abfrage erfolgt im JSON-Format. Dabei ist der Aufbau der Rückgabe wie folgt:

```
{
  "text12hours": "<12-h-Vorhersagetext>", // Vorhersagetext der nächsten 12 Stunden
  "text3days": "<3-Tage-Vorhersagetext>" // Vorhersagetext der nächsten 3 Tage
}
```

Der Vorhersagetext für die nächsten 12 Stunden (*text12hours*) wird erst in Zukunft verfügbar sein. Bis dahin wird hier der Wert auf *null* gesetzt.

WICHTIG: Es werden nur Aufrufe im UTF-8 Format akzeptiert

Beispiel:

Abfrage des Vorhersagetextes für Neukölln. Durch die Suche ist bekannt, dass die Geo-ID 10382 und der Ortsteil *Neukölln* lautet.

Daraus ergibt sich die Anfrage¹:

<https://api.wetteronline.de/weather?package=text&gid=10382&locationname=Neukölln&v=1>

Die Antwort des Servers lautet:

```
{
  "text12hours": null,
  "text3days": "Mal Wolken, mal Sonne - mit beidem ist in den nächsten Tagen beim Wetter in der Region Neukölln zu rechnen. Vor allem am Donnerstag zeigt sich das Wetter im Raum Neukölln vielfach sonnig. Am Donnerstag muss vereinzelt mit Schauern gerechnet werden. Es wird kühler in der Region Neukölln: Bis Donnerstag gehen die Tagestemperaturen auf 17 Grad zurück. Vor allem am Donnerstag weht ein zum Teil starker Wind aus westlicher Richtung."
}
```

Würde bei der Anfrage der Parameter *locationname* weggelassen, so würde sich der Vorhersagetext auf den Ort Berlin statt auf den Ortsteil Neukölln beziehen. Dies kann natürlich auch gewollt sein.

Ein [Beispielcode](#) für die Abfrage des Wettertextes findet sich weiter unten im Text.

7 Weitere Formate

Die Anfrage kann mit dem Parameter „format“ ein gewünschtes Datenformat anfordern. Sämtliche Paketdaten (außer Paket date) werden dann im angefragten Format geliefert. Der format-Parameter ist ebenfalls in die Checksummenbildung mit einzubeziehen.

7.1 XML-Format

Enthält die Anfrage den Parameter „format“ mit dem Wert „xml“, so werden die Daten im XML-Format geliefert. Es existiert zudem eine Anleitung zur Datenlieferung im XML-Format.

8 Kodierung des Wetterzustandes

Der Wetterzustand (Parametername: *wm*) ist gekennzeichnet durch eine Grafik und einen beschreibenden Kurztext.

¹ Die Authentifizierungsparameter (*uid*, *date*, *checksum*) wurden zur Übersichtlichkeit weggelassen

Die Zuordnung der Wetterzustände zu den Kurztexten ist unter folgendem Link zu finden:

Auf deutsch:

<https://api.wetteronline.de/weather?package=symbol&format=text>

Auf englisch:

<https://api.wetteronline.de/weather?package=symbol&format=text&lang=en>

Die Grafiken/Symbole der Wetterzustände können unter folgenden Links als ZIP-Files in verschiedenen Größen (40x28, 50x35 oder 60x42 Pixel) heruntergeladen werden:

<https://api.wetteronline.de/weather?package=symbol&format=png&size=40x28>

<https://api.wetteronline.de/weather?package=symbol&format=png&size=50x35>

<https://api.wetteronline.de/weather?package=symbol&format=png&size=60x42>

Die Kodierung des Wetterzustandes besteht immer aus sechs Zeichen.

Die ersten zwei Zeichen beschreiben die Bewölkung und die Tages-/Nachtzeit. Die übrigen Zeichen (3-6) beinhalten die Art und Intensität des Niederschlages. Bei Wetterzuständen ohne Niederschlag werden das 3.-6. Zeichen als „_“ (Unterstrich) dargestellt.

Die ersten beiden Zeichen geben die Bewölkung und die Tages/Nachtzeit an:

Bewölkung	Tagessymbol	Nachtsymbol
Bedeckt (Kein Himmel sichtbar)	bd	md
Bewölkt (Nur vereinzelt Himmel sichtbar)	bw	mw
Teils bewölkt	wb	mb
Leicht bewölkt	ms	mm
Klar (keine Wolken)	so	mo
Nebel		nb
Teils neblig	ns	nm

Bei Nebel, geringer und keiner Bewölkung können keine weiteren Wetterereignisse auftreten.

Die anderen Wetterzustände können durch Regen, Schnee und Gewitter erweitert werden.

Wetter	Leicht / Vereinzelt	Mittel	Stark
Regen	r1	r2	r3
Schauer	s1		s2
Schneeregen	sr1	sr2	sr3
Schneeregenschauer	srs1		srs2
Schnee	sn1	sn2	sn3
Schneeschauer	sns1		sns2
Schneegewitter*		sg	
Gewitter	g1	g2	g3
Gefrierender Regen	gr1		gr2

Diese Zeichen werden an die Bewölkung angehängt. Freie Zeichen werden mit „_“ (Unterstrich) aufgefüllt.

*Schneegewitter haben keine Intensitätsangabe.

Beispiel: Das Symbol für vereinzelt Gewitter bei bedecktem Himmel ist bdg1___. Das Symbol für vereinzelt Schneeschauer bei bewölktem Himmel ist bwsns1.

9 Beispiel-Code

9.1 PHP-Beispiel für Suchanfrage

```
<?php
header('Content-Type: text/html; charset=utf-8');

/*****
 * WetterOnline Geo-API: Beispielzugriff
 *****/

// Basis-Pfad für geo-API-Anfrage
$basepath = "https://api.wetteronline.de/geo?";

// individuelle UserID
$id = "bWFpbEBleGFtcGx1LmNvbQ==";

// "Secret" des Accounts zur UserID
$secret = "TmF0/HJsaWNoIGtlaW4gZWNodGVzIFNlY3JldCE=";

// als Beispiel wird hier das Geoobjekt Bonn angefordert
// $name = "Bonn";
// zweites optionales Beispiel für eine Ergebnis-Mehrfachauswahl
$name = "Aal";

// falls einer der notwendigen Parameter fehlt: Ende
if (isset($basepath) == false || isset($id) == false || isset($secret) == false ||
    isset($name) == false )
{
    echo "at least one of the required parameters is missing";
    exit();
}

// Datum - wird fuer Bildung der Checksumme benoetigt
$date = new DateTime("now", new DateTimeZone('UTC'));

// query-parameter zusammenstellen
$data = array(
    // Aktuelles Datum im Format 2014-04-11-09
    'date' => $date->format('Y-m-d-H'),
    // UserID
    'uid' => $id,
    // Suchbegriff Ortsname
    'name' => $name
);

// array nach keys alphabetisch sortieren:
ksort($data);

/***** checksumme berechnen *****/
- Parameterwerte mit | als Trennzeichen aneinanderhängen
- Secret anhängen, ebenfalls mit | getrennt
- string md5_binary hashen
- md5-gehashten string base64 kodieren
*****/
$data["checksum"] = base64_encode(md5(implode("|", $data) . "|" . $secret, true));

// query string bilden
$q = http_build_query($data, null, '&', PHP_QUERY_RFC3986);

// pfad für API-request zusammenbauen
$apipath = $basepath . $q;

// API-Anfrage taetigen
$result = @file_get_contents($apipath);

// falls die Anfrage nicht erfolgreich war, wird der HTTP-Response ausgegeben
if ($result == false) exit("Error: " . $http_response_header[0]);
```

```
// Ergebnis in ein JSON-Objekt umwandeln
$json = json_decode($result);

/*****
/***** Debugging Output *****/

// Test-Link zum eigenen Abrufen des generierten API-Aufrufes
echo "<a href=\"\$apipath\" target=\"_blank\">\$apipath</a><br><br>\r\n";

// einfache Testausgabe der API-Response
echo "<pre>";
print_r($json);
echo "</pre>";

// Programmende
exit();

?>
```

9.2 PHP-Beispiel für Wetterabfrage

```
<?php
header('Content-Type: text/html; charset=utf-8');

/*****
* WetterOnline Wetter-API: Beispielzugriff
*****/
// Basis-Pfad für API-Anfrage
$basepath = "https://api.wetteronline.de/weather?";

// gewünschtes Package (verfügbar current, hourly72, day6parts, daily8)
$package = "daily8";

// individuelle UserID
$id = "bWFpbEBleGFtcGx1LmNvbQ==";

// "Secret" des Accounts zur UserID
$secret = "TmF0/HJsaWNoIGtlaW4gZWNoZGVzIFN1Y3JldCE=";

// als Beispiel wird hier das Wetter für die Geo-ID von Bonn angefordert
$gid = "10518";

// optional mehrere Geo-IDs durch Komma getrennt
// $gid = "10518,10513,10400";

// falls einer der notwendigen Parameter fehlt: Ende
if (isset($basepath) == false || isset($package) == false || isset($id) == false ||
    isset($secret) == false || isset($gid) == false)
{
    echo "at least one of the required parameters is missing";
    exit();
}

// Datum - wird fuer Bildung der Checksumme benoetigt
$date = new DateTime("now", new DateTimeZone('UTC'));

// query-parameter zusammenstellen
$data = array(
    // Aktuelles Datum im Format 2014-04-11-09
    'date' => $date->format('Y-m-d-H'),
    // Angabe des gewünschten Datenpaketes
    'package' => $package,
    // UserID
    'uid' => $id,
    // Geo-ID des abzufragenden Ortes
    'gid' => $gid,
);

// array nach keys alphabetisch sortieren:
ksort($data);
```

```

/***** checksumme berechnen *****/
- Parameterwerte mit | als Trennzeichen aneinanderhängen
- Secret anhängen, ebenfalls mit | getrennt
- string md5_binary hashen
- md5-gehashten string base64 kodieren
*****/
$data["checksum"] = base64_encode(md5(implode("|", $data) . "|" . $secret, true));

// query string bilden
$qqs = http_build_query($data, null, '&', PHP_QUERY_RFC3986);

// pfad für API-request zusammenbauen
$apipath = $basepath . $qqs;

// API-Anfrage taetigen
$result = @file_get_contents($apipath);

// falls die Anfrage nicht erfolgreich war, wird der HTTP-Response ausgegeben
if ($result == false) exit("Error: " . $http_response_header[0]);

// Ergebnis in ein JSON-Objekt umwandeln
$json = json_decode($result);

/***** Debugging Output *****/

// Test-Link zum eigenen Abrufen des generierten API-Aufrufes
echo "<a href=\"$apipath\" target=\"_blank\">$apipath</a><br><br>\r\n";

// einfache Testausgabe der API-Response
echo "<pre>";
print_r($json);
echo "</pre>";

// Programmende
exit();

?>

```

9.3 PHP-Beispiel für Abfrage des Vorhersagetextes

```

<?php
header('Content-Type: text/html; charset=utf-8');

/*****
* WetterOnline Wetter-API: Beispielzugriff Vorhersagetext
*****/
// Basis-Pfad für API-Anfrage
$basepath = "https://api.wetteronline.de/weather?";

// gewünschtes Package
$package = "text";

// individuelle UserID
$id = "bWFpbEBleGFtcGx1LmNvbQ==";

// "Secret" des Accounts zur UserID
$secret = "TmF0/HJsaWNoIGtlaW4gZWNodGVzIFNlY3JldCE=";

// als Beispiel wird hier der Vorhersagetext für Berlin, Ortsteil Neukölln angefordert
$id = "10518";
$locationname = "Neukölln";
// $locationname = utf8_encode("Neukölln");//Kodierung in UTF8, falls Datei ISO-8859-1 kodiert

// falls einer der notwendigen Parameter fehlt: Ende
if (isset($basepath) == false || isset($package) == false || isset($id) == false ||
isset($secret) == false || isset($gid) == false)
{
    echo "at least one of the required parameters is missing";
    exit();
}

```

```
}
// Datum - wird fuer Bildung der Checksumme benoetigt
$date = new DateTime("now", new DateTimeZone('UTC'));

// query-parameter zusammenstellen
$data = array(
    // Aktuelles Datum im Format 2014-04-11-09
    'date' => $date->format('Y-m-d-H'),
    // Angabe des gewünschten Datenpaketes
    'package' => $package,
    // UserID
    'uid' => $uid,
    // Geo-ID des abzufragenden Ortes
    'gid' => $gid,
    // Name des Ortteil
    'locationname' => $locationname,
);

// array nach keys alphabetisch sortieren:
ksort($data);

/***** checksumme berechnen *****/
- Parameterwerte mit | als Trennzeichen aneinanderhängen
- Secret anhängen, ebenfalls mit | getrennt
- string md5_binary hashen
- md5-gehashten string base64 kodieren
*****/
$data["checksum"] = base64_encode(md5(implode("|", $data) . "|" . $secret, true));

// query string bilden
$qqs = http_build_query($data, null, '&', PHP_QUERY_RFC3986);

// Pfad für API-request zusammenbauen
$apipath = $basepath . $qqs;

// API-Anfrage taetigen
$result = @file_get_contents($apipath);

// falls die Anfrage nicht erfolgreich war, wird der HTTP-Response ausgegeben
if ($result == false) exit("Error: " . $http_response_header[0]);

// Ergebnis in ein JSON-Objekt umwandeln
$json = json_decode($result);

/***** Debugging Output *****/

// einfache Testausgabe der 3-Tage-Vorhersage
echo "<pre>";
print_r($json->text3days);

echo "</pre>";

// Programmende
exit();

?>
```

10 Beispiel-Checksummen-Berechnung

10.1 Beispielberechnung einer Checksumme für eine Suchanfrage

Parameter:

Nutzer-ID: mail@example.com

Base64-kodieren: mail@example.com -> bWFpbEBleGFtcGxlMnVbQ==

Geheimer Schlüssel: TmF0/HJsaWNoIGtlaW4gZWNoGVzIFNIY3JldCE=

uid=bWFpbEBleGFtcGxlMnVbQ==

date=2014-05-23-08

v=1

name=Münster

1. Parameter nach Namen sortieren & Werte-String bilden

date, name, uid, v

2014-05-23-08|Münster|bWFpbEBleGFtcGxlMnVbQ==|1

2. Geheimen Schlüssel anhängen

2014-05-23-

08|Münster|bWFpbEBleGFtcGxlMnVbQ==|1|TmF0/HJsaWNoIGtlaW4gZWNoGVzIFNIY3JldCE=

3. String md5-hashen und die erhaltene Byte-Sequenz Base64- & URL-kodieren

checksum= U8b/JLuoJb8ZdWv8459Ymw==

[https://api.wetteronline.de/geo?date=2014-05-23-](https://api.wetteronline.de/geo?date=2014-05-23-08&name=Münster&uid=bWFpbEBleGFtcGxlMnVbQ%3D%3D&v=1&checksum=U8b/JLuoJb8ZdWv8459Ymw%3D%3D)

[08&name=Münster&uid=bWFpbEBleGFtcGxlMnVbQ%3D%3D&v=1&checksum=U8b/JLuoJb8ZdWv8459Ymw%3D%3D](https://api.wetteronline.de/geo?date=2014-05-23-08&name=Münster&uid=bWFpbEBleGFtcGxlMnVbQ%3D%3D&v=1&checksum=U8b/JLuoJb8ZdWv8459Ymw%3D%3D)

10.2 Beispielberechnung einer Checksumme für einen package-Request

Parameter:

Nutzer-ID: mail@example.com

Base64-kodieren: mail@example.com -> bWFpbEBleGFtcGxlMnVbQ==

Geheimer Schlüssel: TmF0/HJsaWNoIGtlaW4gZWNoGVzIFNIY3JldCE=

uid=bWFpbEBleGFtcGxlMnVbQ==

date=2014-05-23-08

package=daily8

v=1

gid=10518

4. Parameter nach Namen sortieren & Werte-String bilden

date, gid, package, uid, v

2014-05-23-08|10518|daily8|bWFpbEBleGFtcGxlMnVbQ==|1

5. Geheimen Schlüssel anhängen

2014-05-23-
08|10518|daily8|bWFpbEBleGFtcGxLmNvbQ==|1|TmF0/HJsaWNoIGtlaW4gZWNodGVzIFNlY3JldCE=

6. String md5-hashen und die erhaltene Byte-Sequenz Base64- & URL-kodieren

checksum= lb9fpSERMkmdm0m46AR0aw==

<https://api.wetteronline.de/weather?date=2014-05-23-08&gid=10518&package=daily8&uid=bWFpbEBleGFtcGxLmNvbQ%3D%3D&v=1&checksum=lb9fpSERMkmdm0m46AR0aw%3D%3D>