

# API-Dokumentation

## Goodurl.de

### 1. Einführung

Eine Api (Application programming interface) ist eine Schnittstelle, die von der Webseite goodurl.de anderen Programmen zur softwaremäßigen Anbindung zur Verfügung gestellt wird. Somit können also andere Webseiten oder Anwendungen auf Features von goodurl.de zugreifen, ohne die Webseite selbst aufrufen und anzeigen zu müssen.

Um die Api nutzen zu können, müssen Sie sich zuvor kostenlos als Mitglied anmelden. Desweiteren ist zu beachten, dass die Zugriffszahl auf 1.000 Anforderungen je Kalendertag begrenzt ist.

Der Zugriff erfolgt in PHP über cURL. Nachfolgend finden Sie eine ausführliche Dokumentation des Api-Zugriffs. Wenn Sie Fragen, Anmerkungen und Hinweise auf Fehler mitteilen möchten, schreiben Sie bitte an [api@goodurl.de](mailto:api@goodurl.de)

### 2. Arbeiten mit cURL

#### 2.1 cURL-Status ermitteln

Zunächst gilt es festzustellen, ob cURL auf dem eigenen Server überhaupt eingesetzt werden kann. Dazu befragen wir die PHP-Funktion [phpinfo](#), welche viele Informationen zu den aktuellen Einstellungen liefert.

Dies kann zum Beispiel durch folgende einzeilige PHP-Datei geschehen:

```
<? echo phpinfo(); ?>
```

In der Ausgabe suchen wir dann nach dem Text 'cURL support'. Wenn dort 'Enabled' zu lesen ist, ist alles in Ordnung und Sie können die Api nutzen. Wenn nicht, muss cURL erst installiert und der Server neu gestartet werden. Eine genauere Anleitung hierzu würde den Rahmen dieser Api-Dokumentation sprengen, informieren Sie sich daher gegebenenfalls im Internet oder bei Ihrem Provider.

#### 2.2 cURL anwenden

Eine Standard-Anwendung, wie Sie in dieser Api zum Einsatz kommt, sieht etwa folgendermaßen aus:

```
1 $curl_handle = curl_init();
2 curl_setopt($curl_handle, CURLOPT_URL, "http://goodurl.de/lib/api.php");
3 curl_setopt($curl_handle, CURLOPT_CONNECTTIMEOUT, 2);
4 curl_setopt($curl_handle, CURLOPT_RETURNTRANSFER, 1);
5 curl_setopt($curl_handle, CURLOPT_POST, 1);
6 curl_setopt($curl_handle, CURLOPT_POSTFIELDS, "var1=ab&var2=12");
7 $buffer = curl_exec($curl_handle);
8 curl_close($curl_handle);
```

(Anmerkung: die Zeilennummern dienen ausschließlich der Übersichtlichkeit innerhalb dieser Dokumentation und sind nicht Bestandteil des Quellcodes.)

- Zeile 1: Die cURL-Anwendung wird gestartet und ein Handle zurückgegeben, über welches in den nachfolgenden Zeilen die Identifizierung erfolgt.
- Zeile 2: Es wird festgelegt, welche Datei über cURL angesprochen wird. Dies ist in dieser API immer **http://goodurl.de/lib/api.php**
- Zeile 3: Hier wird die Dauer in Sekunden festgelegt, die der Verbindungsaufbau dauern darf. Der Wert 2 ist hier normalerweise ausreichend.
- Zeile 4: Diese Option legt fest, dass die Rückgabe in Form eines Strings erfolgen soll.
- Zeile 5: Die Werte werden als POST-Request übergeben, also so, wie dies üblicherweise bei HTML-Formularen geschieht.
- Zeile 6: Die zu übergebenden Variablen, jeweils in der Form Variablenname=Wert - mehrere Variablen sind durch das &-Zeichen zu trennen.
- Zeile 7: Hier erfolgt der Aufruf der cURL-API-Anwendung mit den zuvor festgelegten Variablen und Optionen. Wenn eine Rückgabe von Werten erfolgt, so findet sich diese als String in der Variable \$buffer. Der Aufbau des Strings ist Variablenname=Ergebniswert, wobei mehrere zurückgegebene Variablen durch das &-Zeichen voneinander getrennt sind.
- Zeile 8: cURL wird beendet und die zugehörigen Ressourcen werden wieder freigegeben.

Hier kann leider nur in groben Zügen eine Erläuterung zu cURL gegeben werden. Weitere Infos finden Sie z.B. hier: <http://de3.php.net/manual/de/book.curl.php> und an vielen weiteren Stellen im Internet.

## 3. Nutzung der API

### 3.1 Vorgehensweise

Die Nutzung der API erfolgt mittels cURL-Zugriff auf die Datei **http://goodurl.de/lib/api.php**, wobei dieser Datei verschiedene Variable übergeben werden. Einige Variable sind dabei Pflichtvariable, die bei jedem Aufruf angegeben werden müssen. Diese werden im nachfolgenden Absatz beschrieben. Bei einzelnen Funktionen können dann weitere erforderliche Variable hinzukommen. Die Rückgabe der Variablen erfolgt stets in Form eines Strings. Dabei zeigt die Rückgabvariable error stets an, ob ein Fehler eingetreten ist (0=kein Fehler).

### 3.2 Pflicht-Variable

Folgende Variable sind bei jedem Aufruf mindestens zu übergeben:

Variable	Beschreibung
user	der Nutzernamen, mit dem Sie bei goodurl.de angemeldet sind
pw	das zu obigem Nutzernamen gehörige Passwort
cmd	der auszuführende Befehl / die auszuführende Funktion

Hierzu kommen bei einzelnen Funktionen weitere erforderliche und/oder optionale Variablen. Diese, sowie die möglichen Rückgabewerte und Fehlernummern finden Sie bei der Dokumentation der einzelnen Funktionen.

## 4. API - Referenz

Die nachfolgenden Funktionen stehen Ihnen in dieser API zur Verfügung. Informationen zu den einzelnen Funktionen finden Sie im Anschluss.

### new\_shorturl

#### 4.1.1 Funktion new\_shorturl

Beschreibung: Diese Funktion erzeugt eine neue Kurzurl.

Erforderliche Variable:

Variable	Beschreibung
cmd	Funktionsaufruf, muss den Wert <b>new_shorturl</b> beinhalten
user	Ihr angemeldeter Benutzername
pw	das zugehörige Passwort
url	Die vollständige (lange) Url, zu der eine Kurzurl erzeugt werden soll

Optionale Variable:

keine

Rückgabe:

Variable	Beschreibung
error	zurückgebene Fehlernummer (0=fehlerfrei).
short	die erzeugte Kurzurl, z.B.: ght34ue

## 5. Fehlermeldungen

Nach der Verarbeitung des Api-Aufrufs werden als Ergebnis Variable zurückgegeben. Eine Variable, die immer zurückgegeben wird, trägt den Namen error. Sie hat den Wert 0, wenn kein Fehler aufgetreten ist oder einen anderen Wert, der den aufgetretenen Fehler beschreibt. In nachfolgender Liste finden Sie alle definierten Fehlernummern.

Fehlernummer	Beschreibung
0	kein Fehler aufgetreten
11	Die erforderliche Variable <b>user</b> (Benutzername) fehlt.
12	Die erforderliche Variable <b>pw</b> (Passwort) fehlt.
21	ungültige Anmeldedaten
22	Nutzer hat E-Mail-Adresse noch nicht bestätigt.
23	Der Benutzer ist gesperrt.
24	Der Benutzer ist abgemeldet.
31	Tageslimit (1.000 Api-Aufrufe) erreicht
101	Die erforderliche Variable <b>cmd</b> fehlt
102	Die übergebene Variable <b>cmd</b> enthält ungültigen Wert
110	Es wurde keine Url (lang) übergeben
111	Übergebene Url zu lang (max. 240 Zeichen möglich)
112	gesperrte Url, der keine Kurzurl zugewiesen werden kann

## 6. API - Anwendungsbeispiel

### 6.1 Hinweise

Nachfolgend ein Beispiel zur Anwendung der goodurl.de-API. Hier wird eine Kurzurl erzeugt. Es wird ein Array mit den erforderlichen Werten an eine Funktion übergeben. Diese bereitet die Werte auf und führt den eigentlichen Api-Aufruf durch. Die API-Ergebniswerte werden durch die Funktion dann wieder zu einem Array aufbereitet und zurückgegeben. Sie können dieses Beispiel unverändert übernehmen, wenn Sie dabei die korrekten Nutzerangaben und die zu kürzende Url übergeben.

### 6.2 Quellcode

```
function gu_callapi() {

    global $gu_request, $gu_response;

    $c_param = 0;
    foreach ($gu_request as $key => $value) {
        $value = urlencode($value);
        $c_param++;
        if ($c_param>1) { $req .= "&"; }
        $req .= "$key=$value"; }

    $curl_handle = curl_init();
    curl_setopt($curl_handle, CURLOPT_URL, "http://goodurl.de/lib/api.php");
    curl_setopt($curl_handle, CURLOPT_CONNECTTIMEOUT, 2);
    curl_setopt($curl_handle, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl_handle, CURLOPT_POST, 1);
    curl_setopt($curl_handle, CURLOPT_POSTFIELDS, $req);
    $buffer = curl_exec($curl_handle);
    curl_close($curl_handle);

    $result = explode("&", $buffer);
    if (count($result)==0) {
        return 0; }

    foreach ($result as $data) {
        $entry = explode("=", $data);
        $gu_response[$entry[0]]=$entry[1]; }

    return count($gu_response); }

// Beispielaufruf und Auswertung der Funktion

$gu_request['cmd']="new_shorturl";
$gu_request['url']="http://www.beispiel.com/seite.html?param=wert";
$gu_request['user']="testuser";
$gu_request['pw']="geheim";

$result = gu_callapi();
```

```
if ($result==0) {  
    echo "Fehler beim Aufruf der Funktion"; }  
else if ($gu_response['error']<>0) {  
    echo "von API zurückgegebene Fehlernummer: ".$gu_response['error']; }  
else {  
    echo "erzeugte Kurz-Url: " . $gu_response['short']; }
```