

# API

Beschreibung der Programmierschnittstelle.

- [Einleitung](#)
- [Code-Beispiele](#)
  - [Verfügbarkeit testen](#)
  - [Benutzer anlegen](#)
  - [Domain anlegen und Benutzer zuweisen](#)

# Einleitung

Die KeyHelp-API-Schnittstelle erlaubt es anderer Software mit KeyHelp zu interagieren und so bestimmte Aktionen auszulösen. Diese Funktion ist dann hilfreich, wenn diverse IT-Prozesse automatisiert werden sollen.

Die im KeyHelp implementierte API arbeitet nach dem **REST** Prinzip. Das bedeutet, es werden HTTP Anfragen für den Zugriff auf die Daten verwendet werden. Über die verschiedenen HTTP-Methoden werden entsprechende Aktionen abgebildet.

- GET - Abfrage von Daten
- PUT - Aktualisierung von Daten
- POST - Erstellen von Daten
- DELETE - Löschen von Daten

Entsprechend antwortet die API je nach Aktion mit den bekannten HTTP-Status-Codes:

- 200 - OK
- 201 - CREATED
- 400 - BAD REQUEST
- 404 - NOT FOUND
- usw.

Um die Gültigkeit einer API-Anfrage zu verifizieren, muss bei jeder Anfrage ein API-Schlüssel als Teil dieser gesendet werden werden. Dieser API-Schlüssel kann direkt über das KeyHelp-Benutzerinterface erstellt werden. Zusätzlich muss die API zunächst aktiviert werden. Alle relevanten Optionen finden Sie im **KeyHelp-Administrationsbereich** unter **Konfiguration** -> **API**.

## Interaktive Dokumentation

Eine Vollständige API-Dokumentation finden Sie unter folgender URL: <https://api.keyhelp.de>  
Dort erhalten Sie eine detaillierte Beschreibung aller API-Endpunkte und erfahren wie API-Anfragen aufgebaut und API-Antworten strukturiert sein können. Darüber hinaus können Sie sämtliche Endpunkte interaktiv, direkt über Ihren Browser testen.

Um die interaktive Test-Umgebung zu verwenden, betätigen Sie den Button **Try it out**, den Sie unter der Erklärung eines jeden Endpunktes finden. Klicken Sie anschließend den Button **Execute** um eine Anfrage abzuschicken. Um Anfragen direkt zu einem Server Ihrer Wahl zu schicken. Füllen Sie im oberen Bereich der API-Dokumentation zunächst die Eingabefelder **host** sowie die nach einem Klick auf dem Button **Authorize** erscheinenden Eingabefelder aus.

# Sicherheit

Die Sicherheit der API-Schnittstelle wird durch verschiedenste Mechanismen sichergestellt, so dass es Angreifern fast unmöglich ist, sich per Brute-Force unberechtigt Zugriff zu verschaffen. Die Schnittstelle sollte dennoch nur als sicher angesehen, wenn der Zugriff auf ein oder mehrere IP-Adressen / IP-Bereiche beschränkt wird.

In der KeyHelp-Oberfläche können Sie über die Eigenschaften eines API-Schlüssels die gewünschten freigegebenen IPs hinterlegen.

## Beispiel-Code

Um Ihnen den Einstieg zu erleichtern finden Sie unter [Code-Beispiele](#) kurze Anleitungen zur API. Beachten Sie, dass der dort gezeigte Code Ihnen lediglich den Umgang mit der API verdeutlichen soll, Fehlerbehandlungsroutinen und Ähnliches müssen entsprechend von Ihnen programmiert werden.

Im Beispielcode finden Sie Platzhalter nach dem Schema **<NAME>**, die Sie ersetzen müssen. Ersetzen Sie **<HOSTNAME>** mit dem Hostnamen oder IP Ihres KeyHelp Servers. Ersetzen Sie **<API-KEY>** mit Ihrem API-Schlüssel Ihres Servers.

# Code-Beispiele

Code-Beispiele um die Verwendung der API zu verdeutlichen.

# Verfügbarkeit testen

Um die Verfügbarkeit der API sowie den eigenen Code zu testen, kann der API Endpunkt **[GET] /ping** kontaktiert werden. Die Verfügbarkeit ist gegeben, wenn die API mit **pong** antwortet.

## Anfrage

### PHP

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://<HOSTNAME>/api/v2/ping');
curl_setopt($curl, CURLOPT_HTTPHEADER, [
    'X-API-KEY: <API-KEY>',
    'Content-Type: application/json',
]);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($curl);
```

### Bash

```
curl -X 'GET' \
  'https://<HOSTNAME>/api/v2/ping' \
  -H 'accept: application/json' \
  -H 'X-API-Key: <API-KEY>'
```

## Antwort

HTTP-Status-Code: 200

```
{  
  "response": "pong"  
}
```

# Benutzer anlegen

Um einen Benutzer anzulegen, verwendet man den API-Endpunkt **[POST] /clients**. Über das Feld **id\_hosting\_plan** bestimmen Sie, welche Ressourcen und Berechtigungen dieser neue Benutzer besitzen darf.

Über den Endpunkt **[GET] /hosting-plans** kann ermittelt werden, welche ID, welcher Konto-Vorlage zugeordnet ist. Die ID der gewünschten Konto-Vorlage verwenden Sie dann als Wert für das **id\_hosting\_plan** Feld. Existieren noch keine Konto-Vorlagen, müssen diese zunächst erstellt werden.

Im Folgenden soll das Anlegen eines Benutzers **webhosting-001** und das Übertragen der Einstellungen einer Konto-Vorlage mit dem Namen **HostingPlus** demonstriert werden.

## Konto-Vorlagen auslesen

Über den API-Endpunkt **[GET] /hosting-plans** können alle im KeyHelp angelegten Konto-Vorlagen abgefragt werden.

Über den API-Endpunkt **[GET] /hosting-plans/name/<NAME>** können gezielt die Daten einer Konto-Vorlage mit dem gewünschten Namen abgefragt werden, wie in der folgenden Anfrage verdeutlicht.

### Anfrage

#### PHP

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://<HOSTNAME>/api/v2/hosting-
plans/name/HostingPlus');
curl_setopt($curl, CURLOPT_HTTPHEADER, [
    'X-API-KEY: <API-KEY>',
    'Content-Type: application/json',
```

```
]);  
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);  
$result = curl_exec($curl);
```

## Bash

```
curl -X 'GET' \  
  'https://<HOSTNAME>/api/v2/hosting-plans/name/HostingPlus' \  
  -H 'accept: application/json' \  
  -H 'X-API-Key: <API-KEY>'
```

## Antwort

HTTP Status-Code: 200

```
{  
  "id": 3,  
  "name": "HostingPlus",  
  "resources": { ... },  
  "permissions": { ... },  
  "php": { ... },  
  "php_fpm": { ... }  
}
```

# Benutzer anlegen

Zum Anlegen des Benutzers verwenden wir den Endpunkt **[POST] /clients**.

Aus der vorangegangenen Anfrage der Konto-Vorlage übernehmen wir den Wert des Feldes **id** und geben ihn als Wert für das Feld **id\_hosting\_plan** an.

Als API-Antwort erhalten wir die ID des soeben angelegten Benutzers, sowie dessen zufällig generiertes Passwort.

# Anfrage

## PHP

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://<HOSTNAME>/api/v2/clients');
curl_setopt($curl, CURLOPT_HTTPHEADER, [
    'X-API-KEY: <API-KEY>',
    'Content-Type: application/json',
]);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode([
    'username' => 'webhosting-001',
    'id_hosting_plan' => 3,
]));
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($curl);
```

## Bash

```
curl -X 'POST' \
  'https://<HOSTNAME>/api/v2/clients' \
  -H 'accept: application/json' \
  -H 'X-API-Key: <API-KEY>' \
  -d '{ "username": "webhosting-001", "id_hosting_plan": 3 }'
```

# Antwort

HTTP Status-Code: 201

```
{
  "id": 23,
  "password": "ry[Pm- ^Hv+5E"
}
```



# Domain anlegen und Benutzer zuweisen

Um einen neue Domain anzulegen und einem bestehenden Benutzer zuzuweisen, verwendet man den API-Endpunkt **[POST] /domains**. Über das Feld **id\_user** bestimmen Sie, welchem Benutzer die Domain zugewiesen werden soll.

Über den Endpunkt **[GET] /clients** können zunächst alle vorhandenen Benutzer auf dem Server ermittelt werden. Die ID des gewünschten Benutzers verwenden Sie dann als Wert für das Feld **id\_user**. Existiert noch kein Benutzeraccount, muss zunächst mindestens ein Benutzer angelegt werden.

Im Folgenden soll das Anlegen einer Domain **example.com** und dessen Zuweisung an einen Benutzer **webhosting-001** demonstriert werden.

## Benutzer auslesen

Über den API-Endpunkt **[GET] /clients** können alle im KeyHelp angelegten Benutzerkonten abgefragt werden.

Über den API-Endpunkt **[GET] /clients/name/<NAME>** können gezielt die Daten eines Benutzerkontos mit dem gewünschten Namen abgefragt werden, wie in der folgenden Anfrage verdeutlicht.

### Anfrage

#### PHP

```
$curl = curl_init();  
curl_setopt($curl, CURLOPT_URL, 'https://<HOSTNAME>/api/v2/clients/name/webhosting-001');  
curl_setopt($curl, CURLOPT_HTTPHEADER, [
```

```
' X-API-KEY: <API-KEY>',  
' Content-Type: application/json',  
]);  
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);  
$result = curl_exec($curl);
```

## Bash

```
curl -X 'GET' \  
  'https://<HOSTNAME>/api/v2/clients/name/webhosting-001' \  
  -H 'accept: application/json' \  
  -H 'X-API-Key: <API-KEY>'
```

## Antwort

HTTP Status-Code: 200

```
{  
  "id": 23,  
  "username": "webhosting-001",  
  ...  
}
```

# Domain anlegen

Zum Anlegen einer Domain verwenden wir den Endpunkt **[POST] /domains**.

Aus der vorangegangenen Anfrage der Konto-Vorlage übernehmen wir den Wert des Feldes **id** und geben ihn als Wert für das Feld **id\_user** an.

Als API-Antwort erhalten wir die ID der soeben hinzugefügten Domain.

## Anfrage

## PHP

```
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, 'https://<HOSTNAME>/api/v2/domains');
curl_setopt($curl, CURLOPT_HTTPHEADER, [
    'X-API-KEY: <API-KEY>',
    'Content-Type: application/json',
]);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode([
    'domain' => 'example.com',
    'id_user' => 23,
]));
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($curl);
```

## Bash

```
curl -X 'POST' \
  'https://<HOSTNAME>/api/v2/domains' \
  -H 'accept: application/json' \
  -H 'X-API-Key: <API-KEY>' \
  -d '{ "domain": "example.com", "id_user": 23 }'
```

## Antwort

HTTP Status-Code: 201

```
{
  "id": 193
}
```