Postman-App - HTTPful

Inhalt:

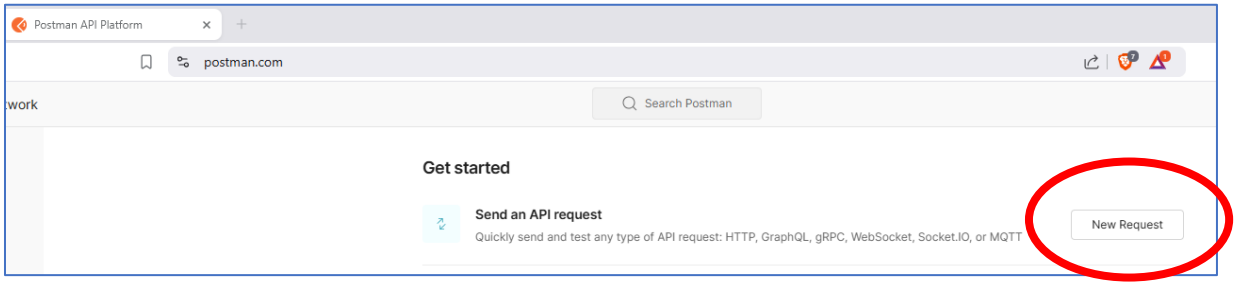
1. Theorie
2. Erstes Beispiel für ein Webservice mit PHP (Localhost - xampp).
3. Beispiel mit Zugriff auf eine externe API bei www.github.com
4. Beispiel mit einer externe API bei https://jsonplaceholder.typicode.com/

1.)Installation

Zum Testen von APIs. Sendet Requests und holt Daten im JSON-Format.

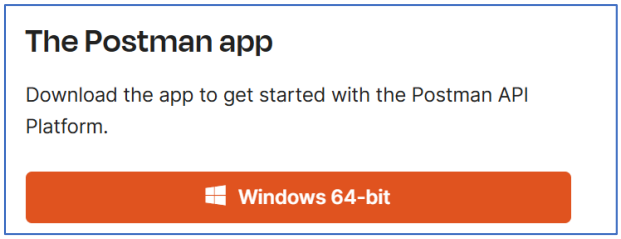
Ursprünglich als Erweiterung zum Chrome-Browser entstanden, ist es nun auch separat als Chrome-App verfügbar.

Möglichkeit a) online mit einer Registrierung (E-Mail, Username und Passwort) – inkl. E-Mail - Verify
https://www.postman.com/ oder www.postman-echo.com
Klicke auf „New Request“



Möglichkeit b) Installation:

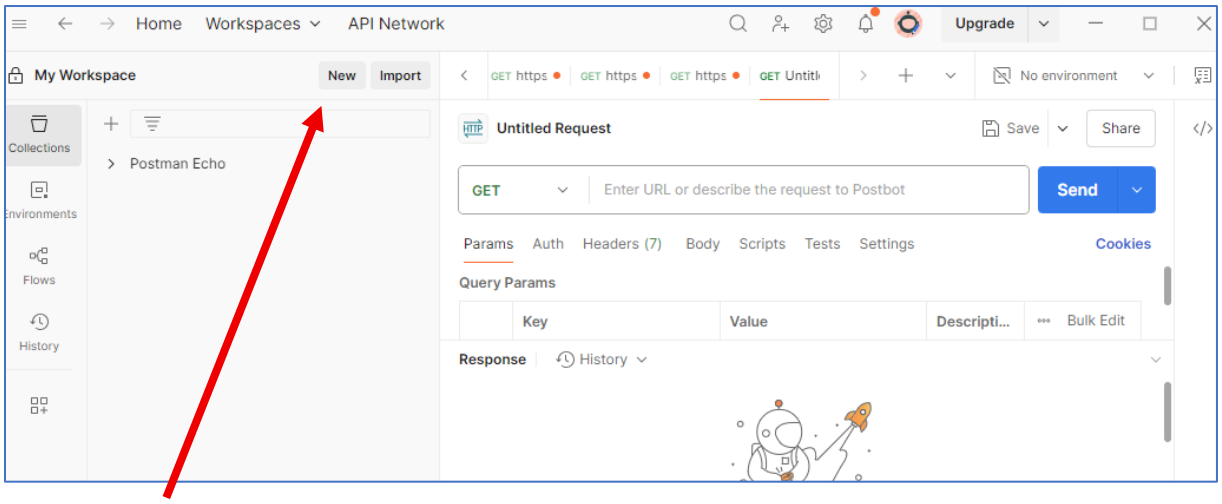
Download von https://www.postman.com/downloads/



Dann das Setup ausführen.

HTTP GET lässt sich im Browser testen, indem wir den URL aufrufen, aber für alle anderen Verben benötigen wir ein Tool zur Unterstützung. **Hier bietet sich Postman an, der uns eine grafische Oberfläche zur Verfügung stellt**, mit der sich HTTP-Anfragen in allen Formen, Farben und Varianten modellieren lassen. Postman steht als Chrome-App für alle Plattformen zur Verfügung.

Mit Postman hat man eine Möglichkeit, wie man eine API ohne Frontend und ohne Browser auf Herz und Nieren testen kann.



Klicke auf New und dann HTTP.

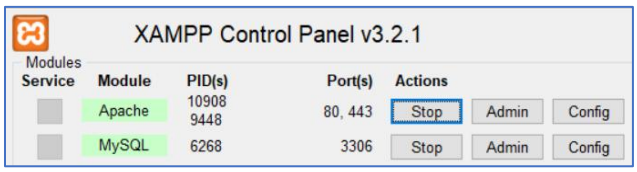
Weiter Info: https://www.getpostman.com/docs/

2)Erstes Beispiel für ein kleines Webservice mit PHP (Localhost - XAMPP).

Erstelle in xampp/htdocs einen Ordner „postman.test“ und darin ein PHP-Script mit einfacher echo-Ausgabe. Das soll später mit Postman abgerufen werden.

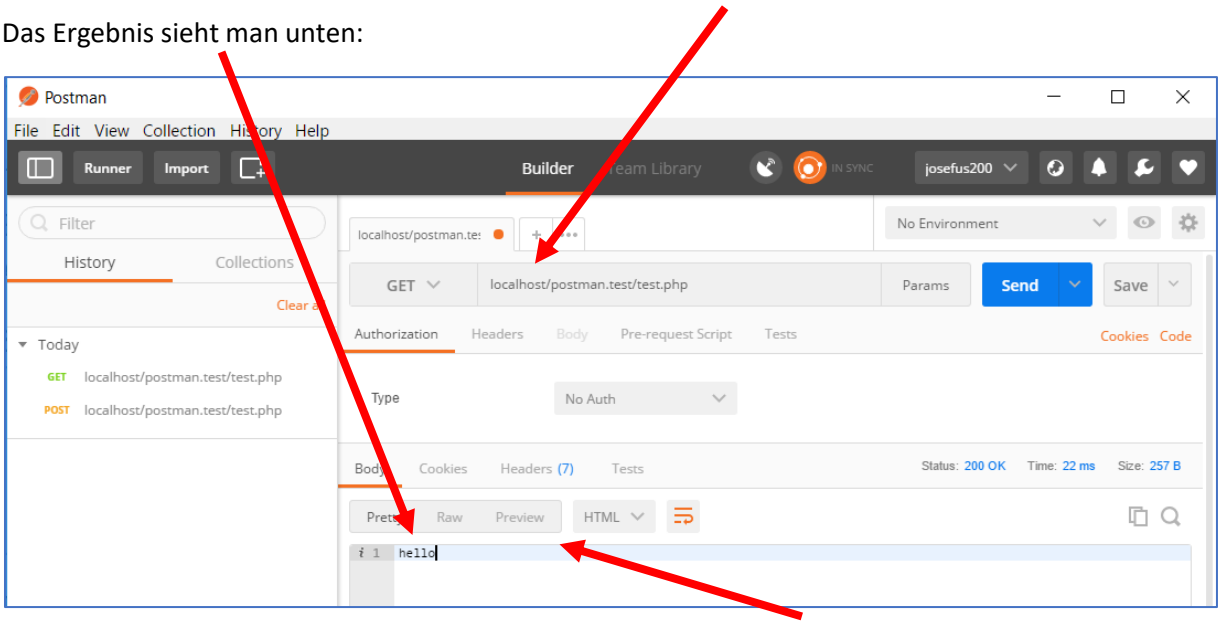
```php
<?php

echo "hallo";
?>
```

Starte auch den XAMPP-Server.



Dann ist die URL einzugeben. Beachte den Pfad zum htdocs Ordner und zum php-File. Dann „Send“.

Das Ergebnis sieht man unten:



Wir haben hier mit der „get“-Methode gearbeitet. Mit Klick auf „Preview“ sieht es noch besser aus.

2)Erstellen von JSON – Inhalt für eine JSON-Response

Die JSON-Response kann der Kunde nutzen. Er kann eine Abfrage durchführen.
Beispiel: eine Android-App will „results“ von der Datenbank, wird zuerst eine Anfrage („request“) an das API gesendet, welche dann eine Antwort („response“) sendet. Diese ist JSON-Format und kann gelesen werden.
Lösche das „echo“ und schreibe zuerst ein Array: $arr mit einem Status-Code und einer Message.
Dann Daten aus der Datenbank, die man anbietet.

```php
<?php
//DB Ergebnisse
$results = ['Josef','Daniel','Franz'];

$arr['status_code'] = 1;
$arr['message'] = "Daten erfolgreich gefunden";
$arr['data'] = $results;

print_r($arr);
?>
```

Speichern: test2.php

Code:

```
<?php
//DB Ergebnisse
$results = ['Josef','Daniel','Franz'];
$arr['status_code'] = 1;
$arr['message'] = "Daten erfolgreich gefunden";
$arr['data'] = $results;
print_r($arr);
?>
```

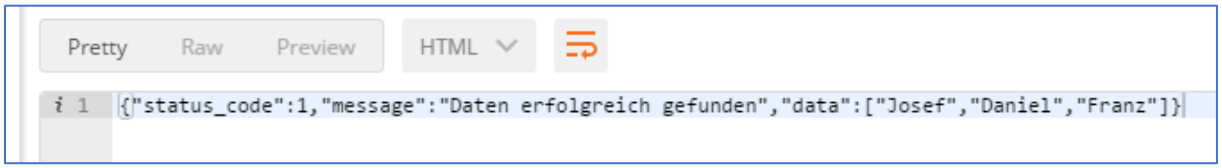
Ergebnis in postman:

Aber: Das Ergebnis ist nicht nutzbar, da der Client dieses Format nicht lesen kann. Daher muss es lesbar gemacht werden:

Ändere den Code, um die Daten in JSON-Format zu ändern für die Ausgabe:
Zuerst statt der „print()“-Funktion benutze die „echo json_encode()“-Funktion.

```
$arr['data'] = $results;

echo json_encode($arr);
?>
```

Resultat in lesbaren JSON-Format:



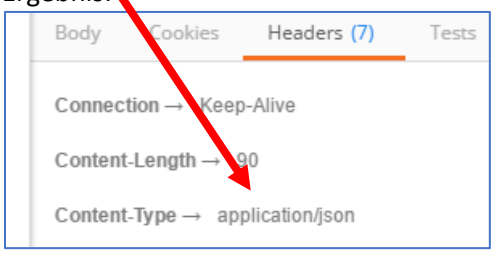
In JSON betrachten: klicke auf die Option „JSON“:

Hier kann man auch den „Content-Type“ ansehen, der HTML ist:
Klicke auf „Headers“

Der sichere Weg, um Antworten zu senden ist aber, den Type in JSON zu ändern. Dazu schreibe folgenden Code dazu:
header('Content-Type: application/json');

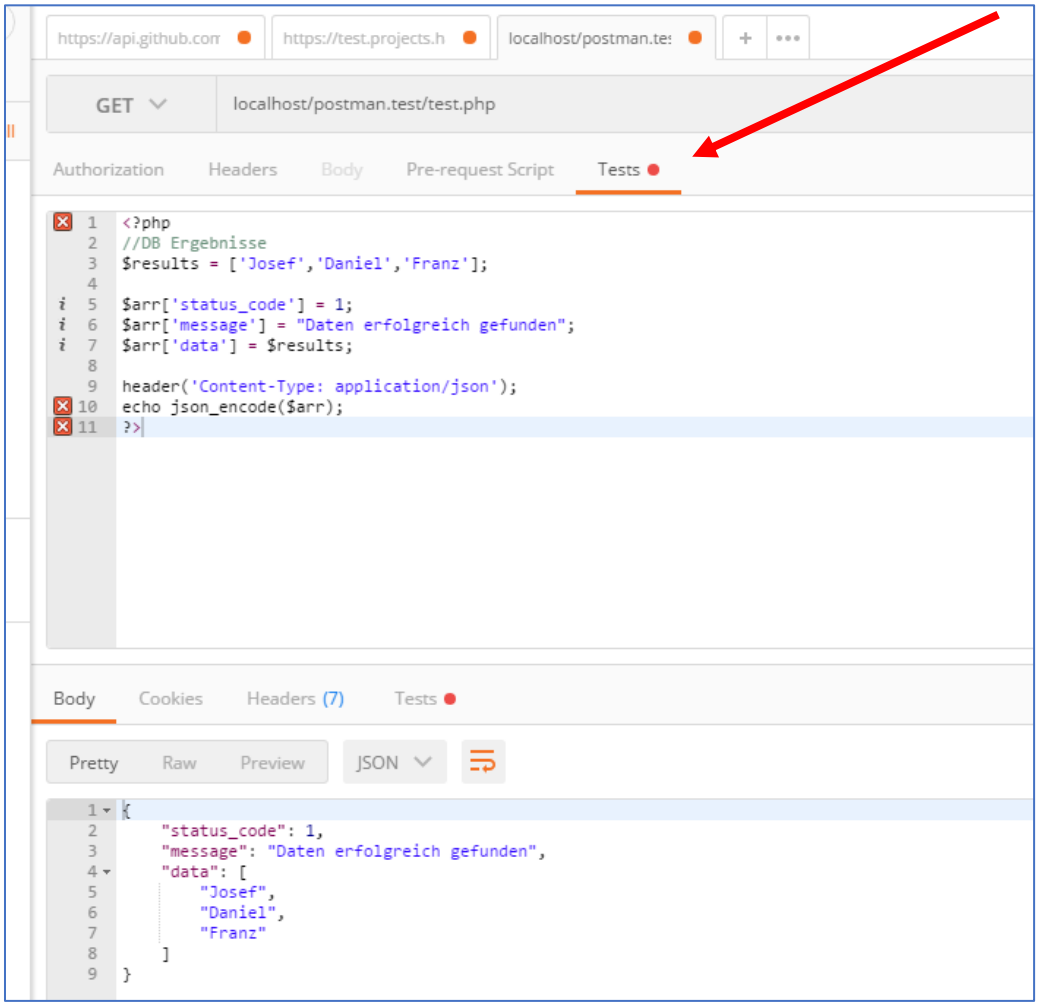
```
$arr['data'] = $results;

header('Content-Type: application/json');
echo json_encode($arr);
?>
```

Ergebnis:



INFO:

Man benötigt keinen Editor, da man in Postman selbst den Code schreiben kann. Klicke auf „Tests“:



3)Beispiel mit Zugriff auf eine externe API bei www.github.com

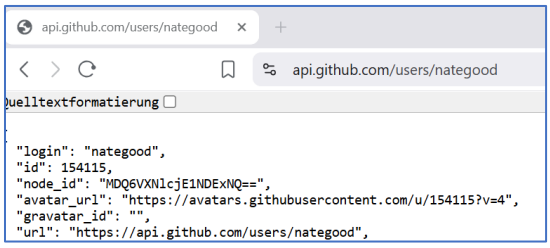
Benötigt immer eine sichere Verbindung https://

Github.com stellt ein API zur Verfügung, mit dem auf Daten zugegriffen werden kann. Das nutzen wir, um mit Requests darauf zuzugreifen und Abfragen zu bekommen.

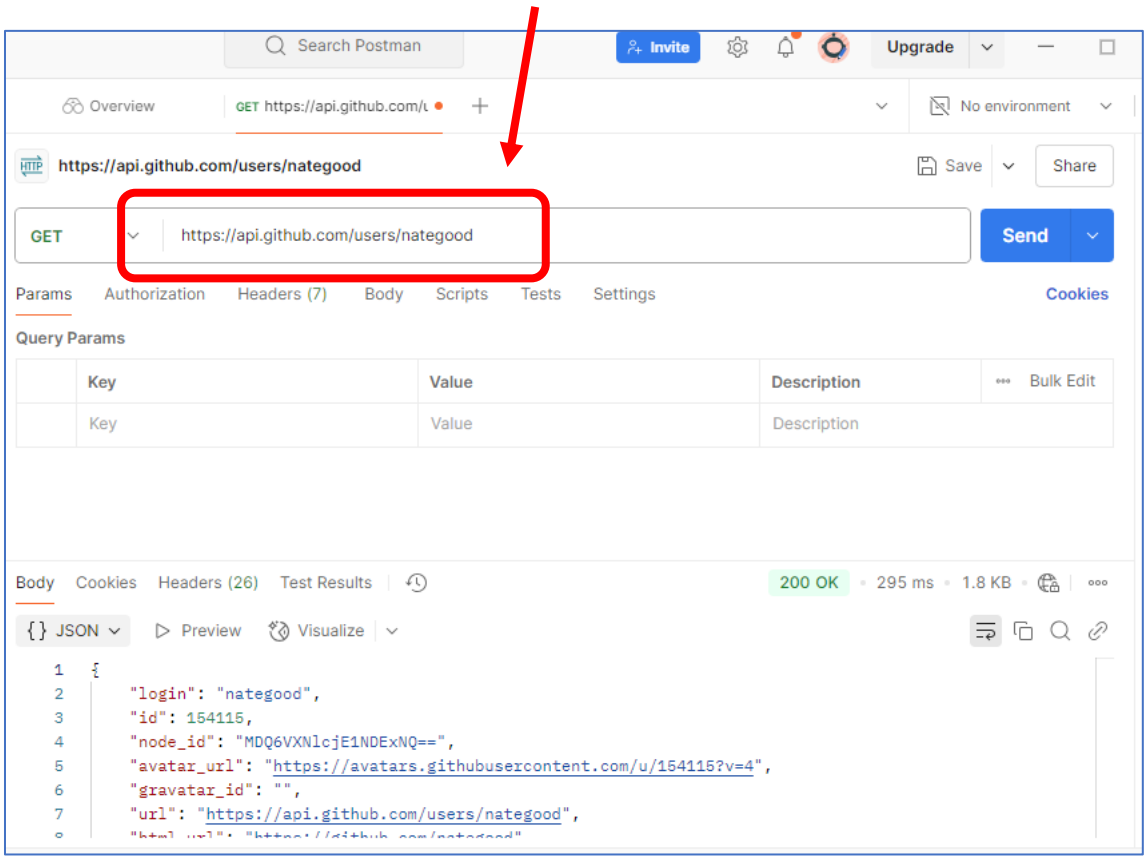
Quelle und Info: https://developer.github.com/
https://docs.github.com/de/rest?apiVersion=2022-11-28

Gib folgende Adresse ein:
https://api.github.com/users/nategood und klicke auf „Send“.

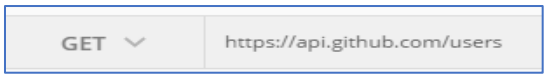
Die GET-Abfrage kann man auch im Browser abfragen, ABER mit postman sind auch alle anderen Möglichkeiten vorhanden (POST, DELETE)

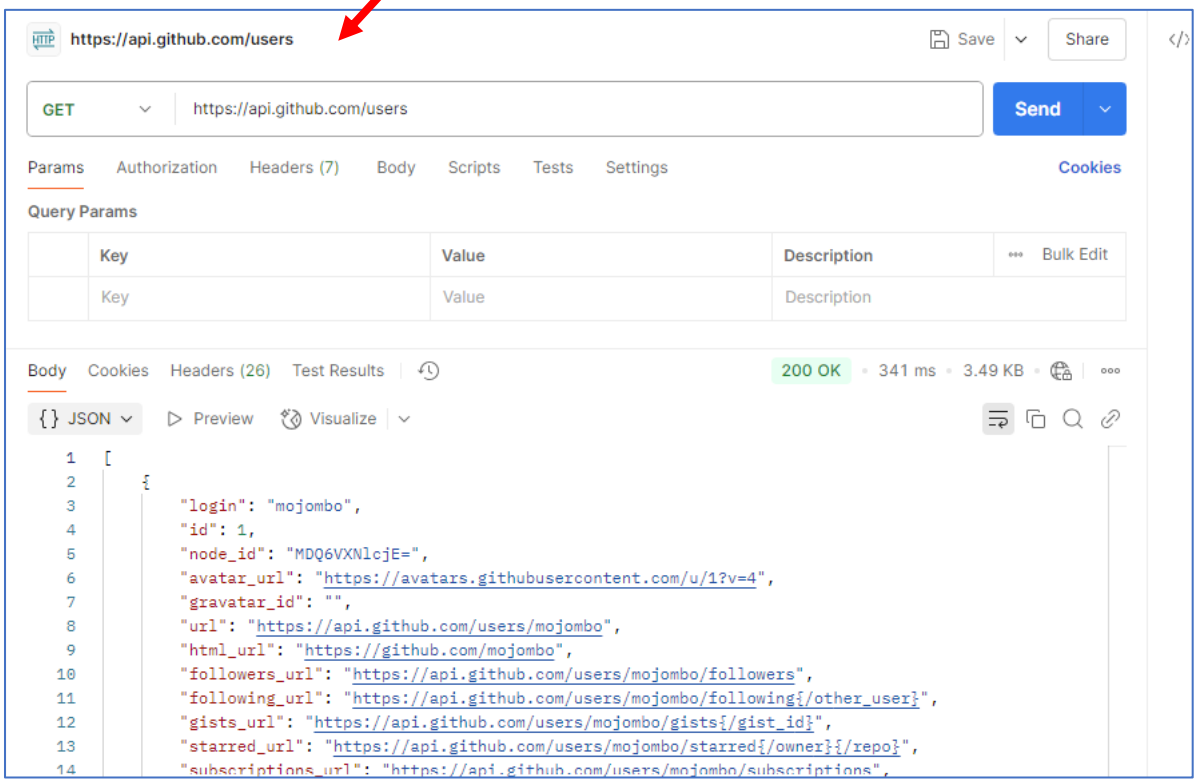


Ist „JSON“ eingestellt erhält man eine schöne Darstellung für diesen User:



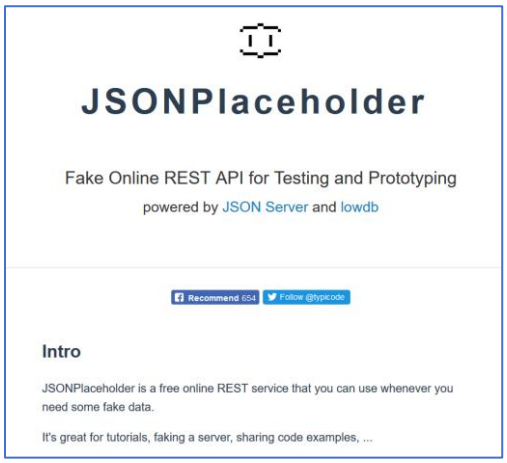
Will man alle User anzeigen lassen, lasse den User „nategood“ weg:

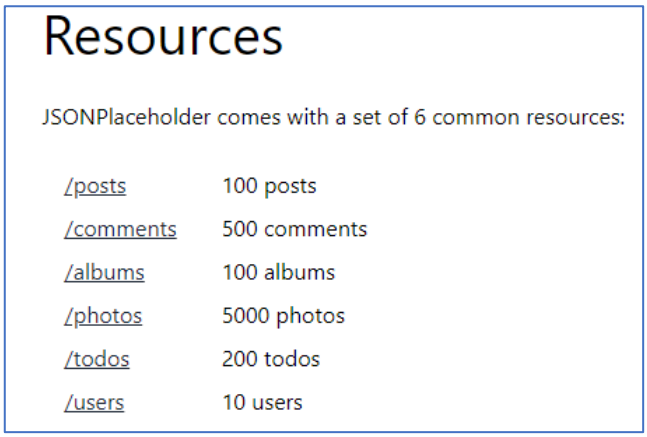

GET
https://api.github.com/users


https://api.github.com/users
Save
Share
GET
https://api.github.com/users
Send
Params
Authorization
Headers (7)
Body
Scripts
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Key
Value
Description
Body
Cookies
Headers (26)
Test Results
200 OK
341 ms
3.49 KB
{} JSON
Preview
Visualize
[
{
"login": "mojombo",
"id": 1,
"node_id": "MDQ6VXNlcjE=",
"avatar_url": "https://avatars.githubusercontent.com/u/1?v=4",
"gravatar_id": "",
"url": "https://api.github.com/users/mojombo",
"html_url": "https://github.com/mojombo",
"followers_url": "https://api.github.com/users/mojombo/followers",
"following_url": "https://api.github.com/users/mojombo/following{/other_user}",
"gists_url": "https://api.github.com/users/mojombo/gists{/gist_id}",
"starred_url": "https://api.github.com/users/mojombo/starred{/owner}{/repo}",
"subscriptions_url": "https://api.github.com/users/mojombo/subscriptions",

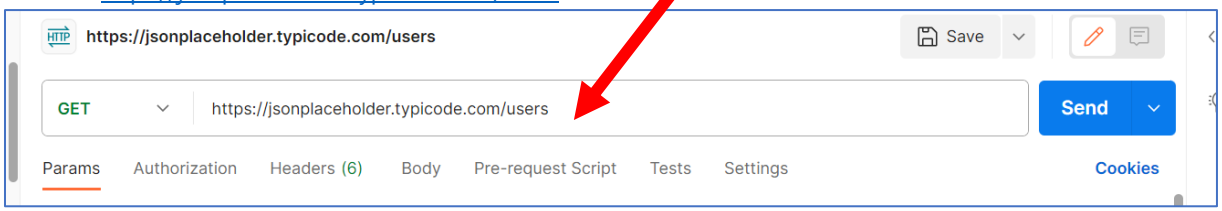
4)Beispiel mit einer externe API bei https://jsonplaceholder.typicode.com/

Öffne die Seite und scrolle nachunten bis zu „Resources“ für 6 Quellen:





Gib ein: https://jsonplaceholder.typicode.com/users



Ergebnis:

Wähle auch