

# Python als Webclient

---

Dominik Pataky

10. Dezember 2017

Python-Kurs 2017

## 1. urllib.request

- Öffnen einer URL

- Request Klasse

- HTTPResponse Klasse

## 2. Andere Module

## 3. Das Requests-Modul

- Installation

Das Package `urllib` ist eine nützliche Sammlung mehrerer Module zur Arbeit mit URLs.

`urllib.request`

---

Das Modul `urllib.request` enthält Funktionen und Klassen, welche beim Öffnen von URLs (vor allem über HTTP) helfen.

Unterstützt werden:

- verschiedene Authentifizierungsarten
- Weiterleitungen
- Cookies
- und mehr...

# Öffnen einer URL

Das Öffnen einer URL wird über die Funktion `urlopen()` realisiert:

```
1 import urllib.request
2
3 urllib.request.urlopen(url, data=None, [timeout, ] *,
4                          cafile=None, capath=None,
5                          cadefault=False, context=None)
```

# Öffnen einer URL

**url** ein String für simple URLs oder ein **Request** Objekt für komplexere Anfragen

**data** Daten, die an den Server gesendet werden sollen. vom Typ **bytes** oder ein Iterable von **bytes** Objekten

## Rückgabewerte

bei URLs mit http-Requests

`httplib.client.HTTPResponse` Objekt

bei ftp, file und data

`urllib.addinfourl` Objekt

Um komplexere Anfragen stellen zu können, kann man Request Objekte verwenden:

```
1 urllib.request.Request(url, data=None, headers={},  
2                        origin_req_host=None,  
3                        unverifiable=False, method=None)
```



**url** muss String mit gültiger URL sein

**data** wie bei *urlopen*

**headers** dict mit {Header-Name : Header-Value, ...}

oder

list von Tupeln mit [(Header-Name,  
Header-Value), ...]

**method** String, der Art des HTTP Request angibt (HEAD, GET, POST, ...)

# Beispiel

```
1 import urllib.request
2
3 r = urllib.request.Request(
4     'http://python.org',
5     headers={'content-type': 'application/json'},
6     method='PUT'
7 )
```

Die Request Klasse kann man Verwenden zum:

- Kontrollieren der gesendeten Header (z.B. *Content-Type* oder *User-Agent*)
- Kontrollieren der Method **POST**, **PUT** oder **HEAD**

# HTTPResponse Klasse

Objekte dieser Klasse werden nicht direkt vom User erstellt.

```
1 http.client.HTTPResponse(sock, debuglevel=0,  
2                             method=None, url=None)
```

Klasse enthält Funktionen und Variablen wie:

- **read()** - gibt zurückgelieferten Inhalt zurück
- **getheader()** oder **getheaders()** liefert einen/alle Header zurück
- **status** gibt den HTTP Statuscode zurück
- **version** gibt die HTTP Version zurück

## Andere Module

---

Das Package `urllib` enthält außerdem folgende Module:

- `urllib.error`  
Enthält Exceptions, die von `urllib.request` geworfen werden.
- `urllib.parse`  
Zum Parsen von URLs.
- `urllib.robotparse`  
Zum Parsen der `robots.txt` von Webseiten.

# Das Requests-Modul

---

Das Requests Modul ist eine gute und für Menschen verständliche Alternative zu `urllib.requests`, das HTTP Requests vereinfacht.

Das Requests Package lässt sich ganz einfach über **pip** installieren:  
`pip install requests`

# Beispiel

```
1 r = requests.get('https://api.github.com/user', auth=('user', '
    pass'))
2
3 r.status_code    # -> 200
4 r.headers['content-type']
5 # -> 'application/json; charset=utf8'
6 r.encoding      # -> 'utf-8'
7 r.text          # -> u'{"type":"User"...}'
8 r.json()
9 # -> {'private_gists': 419, 'total_private_repos': 77, ...}
```

Ruft die API-Seite von **GitHub** auf und authentifiziert sich mit  
Nutzername und Passwort.