

# Telegram Chatbot



Ein Chatbot ist ein Programm, dass sich in einem Chatroom von selbst mit einem User unterhält. Wie der Bot auf den User reagiert und was er sagt, hängt davon ab, wie das Programm geschrieben ist. Wir wollen einen Chatbot für die Messenger-App Telegram programmieren.

## 1 Wie funktioniert ein Bot?

Ein Bot liest nacheinander die Nachrichten, die an ihn gesendet werden, und versucht den Inhalt ihrer Nachricht zu erkennen, indem er die Nachricht mit Sätzen und Wörtern abgleicht, die er bereits kennt. Dass dabei sinnvolle Antworten rauskommen, ist Aufgabe des Programmierers.  
Beispiel: Der Bot erkennt, dass das Wort “Hallo” in der Nachricht vorkommt und antwortet mit “Hallo, \*Name des Benutzers\*“

## 2 Wie kann ich selbst einen Bot programmieren?

### 2.1 Vorbereitung

Die Grundfunktionen eines Chatbots stehen Dir schon zur Verfügung. Um die Funktionen nutzen zu können, musst du die Klasse `TelegramBot` aus dem Modul `chatter` importieren.

```
from chatter.TelegramBot import TelegramBot
```

Danach kannst du ein `TelegramBot`-Objekt erzeugen. Das geht so:

```
deinBot = TelegramBot([oauth]) (# Argument optional --> nur bei neuem Bot-Account)
```

### 2.2 Nachrichten, Chats und User

Dein Bot weiß bereits, was Nachrichten, Chats und User sind. Objekte dieser Klassen haben folgende Attribute:

#### 2.2.1 Nachrichten

- `id` – Nummer, mit der eine bestimmte Nachricht identifiziert werden kann
- `sender` – User, der die Nachricht gesendet hat
- `datum` – Zeitpunkt zu dem die Nachricht gesendet wurde (in UNIX-Zeit)
- `chat` – Chat, in dem die Antwort gesendet wurde
- `antwort` – Originalnachricht, falls die Nachricht eine Antwort war
- `inhalt` – Der Inhalt der Nachricht

#### 2.2.2 Chats

- `id` – Nummer, mit der ein bestimmter Chat identifiziert werden kann
- `typ` – Typ des Chats: “private“, “group“, “supergroup“, oder “channel“
- `titel` – Titel des Gruppenchats (optional)

### 2.2.3 User

- id – Nummer, mit der ein bestimmter Chat identifiziert werden kann
- vorname – Vorname des Users
- nachname – Nachname des Users (optional)
- username – Username des Users (optional)

## 2.3 Die TelegramBot-Klasse

### 2.3.1 Attribute

Dein Bot hat folgende Attribute:

- id – Nummer, mit der der Bot identifiziert werden kann
- name – Name des Bots
- username – Username des Bots
- oauth – Authorisierungskennung zur Kommunikation mit dem Server
- online – zeigt an ob der Bot on- oder offline ist (Boolean-Wert)

Konstruktor:

### 2.3.2 Funktionen

Folgende Funktionen kennt dein Bot bereits:

`hole_updates()`

Rückgabewert: Eine Liste aller Nachrichten, die der Bot seit der letzten Abfrage erhalten hat

`gehe_online()`

Setzt online-attribut auf "True"

`gehe_offline()`

Setzt online-attribut auf "False"

`sende_nachricht(text, chat_id, antwort_id)`

Sendet eine Nachricht an den Chat.

Parameter:

- text – Die zu sendende Nachricht
- chat\_id – Die ID des Chats an den die Nachricht gesendet werden soll
- antwort\_id – Nachricht-ID der Nachricht, auf die der Bot antwortet (optional)

Rückgabewert: Die verschickte Nachricht

`sende_bild(bild, chat_id)`

Sendet ein Bild an den Chat.

Parameter:

- bild – Pfad des Bildes, das verschickt werden soll
- chat\_id – Die ID des Chats an den das Bild gesendet werden soll

### 3 Arbeitsauftrag

- Erstelle eine neue Python-Datei, importiere die Klasse TelegramBot und erzeuge eine Klasse für deinen Chatbot (s. Vorbereitung)
- definiere eine Methode, die folgendes macht:
  - den Bot aktiviert
  - so lange er aktiviert ist, immer wieder checkt ob der Bot neue Nachrichten erhalten hat (while-Schleife!)
  - für jede Nachricht checkt, ob der Bot eine passende Antwort hat und diese dann sendet (implementiere am besten eine eigene Methode hierfür und rufe sie dann auf)
    - \* auf die Nachricht „hallo“ soll der Bot mit „hallo, \*Name\*“ antworten
    - \* auf „wie heißt du?“ soll der Bot mit seinem Namen antworten
    - \* auf „tschüss“ soll der Bot mit „bis bald, \*Name\*“ antworten
  - außerhalb der Klasse, erzeuge ein Objekt deiner Bot-Klasse und starte den Bot
  - erweitere die Funktionalität deines Bots mit deinen eigenen Ideen