

Programmierschnittstellen im Web

Webseiten bieten Programmierern oft Schnittstellen, sogenannte *APIs*. An diese Schnittstellen können Programme Anfragen stellen. Dabei gibt es zwei wichtige Arten:

- **POST-Anfragen**, um dem Server Informationen zum Verarbeiten zu **senden**.
- **GET-Anfragen**, um Informationen vom Server **abzufragen**

POST-Anfragen

POST-Anfragen benötigen Daten, die an das Netzwerk geschickt werden müssen. Als Hilfestellung steht die Klasse `NetzwerkZugriff` zur Verfügung.

NetzwerkZugriff
<code>NetzwerkZugriff(String domain)</code> <code>GETAnfrageSenden(String url) : String</code> <code>POSTAnfrageVorbereiten(String parameterName, String parameterWert): void</code> <code>POSTAnfrageSenden(String url) : void</code>

Du kannst durch die Methode `POSTAnfrageVorbereiten` Werte für die nächste Anfrage setzen und mit `POSTAnfrageSenden` abgeschicken. Z.B:

```
1 NetzwerkZugriff socialbotnet = new NetzwerkZugriff("https://www.socialbotnet.de");
2
3 // Login Daten setzen
4 socialbotnet.POSTAnfrageVorbereiten("username", "MeinBot");
5 socialbotnet.POSTAnfrageVorbereiten("password", "secure");
6
7 // Nachricht setzen
8 socialbotnet.POSTAnfrageVorbereiten("message", "Hallo Welt!");
9
10 // POST-Anfrage abschicken
11 socialbotnet.POSTAnfrageSenden("/api/post");
```

GET-Anfragen

GET-Anfragen können mit Objekten der Klasse `NetzwerkZugriff` durch die Methode `GETAnfrageSenden` gesendet werden. Z.B.

```
1 NetzwerkZugriff socialbotnet = new NetzwerkZugriff("https://www.socialbotnet.de");
2
3 // Alle User abfragen
4 String antwort = socialbotnet.GETAnfrageSenden("/api/users");
```

Auch bei GET-Anfragen können Parameter benutzt werden. Diese werden jedoch einfach in der URL angehängt:

```
1 // Posts abfragen, mit den Parametern "sortBy=likes" und "limit=1"
2 String antwort = socialbotnet.GETAnfrageSenden("/api/posts?sortBy=likes&limit=1");
```

Übersicht der Schnittstellen

<i>/api/users</i>	<i>GET</i>	Liefert eine Übersicht über alle registrierten Nutzer
<i>/api/posts</i> Optionale Parameter: <i>sortBy</i> , <i>limit</i>	<i>GET</i>	Liefert die letzten 50 Posts des gesamten Netzwerks <i>sortBy=likes</i> nach Likes sortiert, <i>sortBy=time</i> nach Datum sortiert. Limit beschränkt die Anzahl, z.B. <i>limit=1</i> für nur einen Post
<i>/api/pinnwand/<username></i> Optionale Parameter: <i>sortBy</i> , <i>limit</i>	<i>GET</i>	Liefert die Posts an der Pinnwand des genannten Nutzers <i>sortBy=likes</i> , <i>sortBy=time</i> , <i>limit=zahl</i> wie oben
<i>/api/user/update</i> Parameter: <i>username</i> , <i>password</i> Optionale Parameter: <i>newUsername</i> , <i>hobbies</i> , <i>about</i>	<i>POST</i>	Updaten der Profilinformationen wie <i>username</i> , „Hobbies“ und „Über mich“. <i>newUsername</i> Benutzernamen ändern, <i>hobbies</i> und <i>about</i> setzen den „Hobbies“ und „Über mich“ Text.
<i>/api/post</i> Parameter: <i>username</i> , <i>password</i> , <i>message</i>	<i>POST</i>	Posten eines Beitrags auf der eigenen Pinnwand und der allgemeinen Chronik.
<i>/api/post/<username></i> Parameter: <i>username</i> , <i>password</i> , <i>message</i>	<i>POST</i>	Posten an die Pinnwand eines bestimmten Nutzers.
<i>/api/like</i> Parameter: <i>username</i> , <i>password</i> , <i>postid</i>	<i>POST</i>	Ermöglicht automatisiertes Liken eines Posts.
<i>/api/unlike</i> Parameter: <i>username</i> , <i>password</i> , <i>postid</i>	<i>POST</i>	Macht like rückgängig.