

# 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

<p><i>/api/posts</i></p> <p>Optionale Parameter: sortby, order, limit</p>	<p><i>GET</i></p>	<p>Liefert die letzten 50 Posts des gesamten Netzwerks.</p> <p><i>sortby</i>=&lt;attribut&gt; Sortiert nach einem Attribut (z.B. likes).</p> <p><i>order=asc</i> bzw. <i>order=desc</i> Sortiert aufsteigend bzw. absteigend.</p> <p><i>limit</i>=&lt;zahl&gt; Beschränkt die Anzahl der Objekte.</p>
<p><i>/api/pinnwand/&lt;username&gt;</i></p> <p>Optionale Parameter: sortby, order, limit</p>	<p><i>GET</i></p>	<p>Liefert die Posts an der Pinnwand des genannten Nutzers</p> <p>Parameter wie oben</p>
<p><i>/api/users</i></p> <p>Optionale Parameter: sortby, order, limit</p>	<p><i>GET</i></p>	<p>Liefert eine Übersicht über alle registrierten Nutzer</p> <p>Parameter wie oben</p>
<p><i>/api/user/update</i></p> <p>Parameter: username, password Optionale Parameter: newUsername, hobbies, about</p>	<p><i>POST</i></p>	<p>Updaten der Profilinformationen wie Nutzernamen, „Hobbies“ und „Über mich“.</p> <p><i>newUsername</i> Benutzernamen ändern <i>hobbies</i> „Hobbies“-Text ändern <i>about</i> „Über mich“-Text ändern</p>
<p><i>/api/post</i></p> <p>Parameter: username, password, message</p>	<p><i>POST</i></p>	<p>Posten eines Beitrags auf der eigenen Pinnwand und der allgemeinen Chronik.</p>
<p><i>/api/post/&lt;username&gt;</i></p> <p>Parameter: username, password, message</p>	<p><i>POST</i></p>	<p>Posten an die Pinnwand eines bestimmten Nutzers.</p>
<p><i>/api/like</i></p> <p>Parameter: username, password, postid</p>	<p><i>POST</i></p>	<p>Ermöglicht automatisiertes Liken eines Posts.</p>
<p><i>/api/unlike</i></p> <p>Parameter: username, password, postid</p>	<p><i>POST</i></p>	<p>Macht like rückgängig.</p>