

Parsen der Antworten des SocialBotNet

In dem Projekt steht dir eine Klasse `AntwortParser` zur Verfügung.

AntwortParser
<code>zuUserArray(String serverAntwort) : User[]</code> <code>zuPostArray(String serverAntwort) : Post[]</code>

Diese kann die Antworten des SocialBotNet zu richtigen Objekten umwandeln:

- `zuUserArray` wandelt die Antwort (z.B. von `/api/users`) in ein Array von User Objekten.
- `zuPostArray` wandelt die Antwort (z.B. von `/api/posts`) in ein Array von Post Objekten.

User	Post
- id : int - username : String - hobbies : String - about : String	- id : int - message : String - user : User - wall : User - publishingDate : Timestamp - likedBy : User[]
getId() : int getUsername() : String getHobbies() : String getAbout() : String	getId() : int getMessage() : String getUser() : User getWall() : User getPublishingDate() : Timestamp getLikedBy() : User[]

Beispiel:

```
1 public void eigenePinnwandLiken() {
2     // Alle Nachrichten auf der Pinnwand von MeinBot abrufen
3     String antwort = socialbotnet.GETAnfrageSenden("/api/pinnwand/MeinBot");
4     // Die Antwort beschreibt mehrere Posts, verwende also AntwortParser.zuPostArray
5     Post[] nachrichten = AntwortParser.zuPostArray(antwort);
6     // Iteriere ueber die einzelnen Nachrichten
7     for(int i=0; i<nachrichten.length; i++) {
8         Post nachricht = nachrichten[i];
9         // Hole die Nachrichten ID aus dem Post Objekt
10        int postid = nachricht.getId();
11        // Diese ID wird als Eingabewert zum Liken benoetigt
12        liken(postid);
13    }
14 }
15
16 public void liken(int postid) {
17     // POST-Anfrage mit Nutzerdaten vorbereiten
18     socialbotnet.POSTAnfrageVorbereiten("username", username);
19     socialbotnet.POSTAnfrageVorbereiten("password", password);
20     // POST-Anfrage mit postid vorbereiten
21     socialbotnet.POSTAnfrageVorbereiten("postid", postid);
22     // Like schicken
23     socialbotnet.POSTAnfrageSenden("/api/like");
24 }
```

Beispielrückgaben des SocialBotNet

Ein *GET*-Request auf `/api/users` liefert folgendes Ergebnis, das mit `AntwortParser.zuUserArray` verarbeitet werden kann:

```
1 [
2   {
3     "id":1,
4     "username":"root",
5     "hobbies":"Filme schauen, Fussball spielen",
6     "about":"I am root."
7   },
8   {
9     "id":2,
10    "username":"Welcome",
11    "hobbies":"",
12    "about":""
13  },
14  ...
15 ]
```

Ein *GET*-Request auf `/api/posts` oder `/api/pinnwand/...` liefert folgendes Ergebnis, das mit `AntwortParser.zuPostArray` verarbeitet werden kann:

```
1 [
2   {
3     "id":2,
4     "message":"Herzlich Willkommen, root!",
5     "user":{"id":2,"username":"Welcome", "hobbies": "...", "about": "..."},
6     "wall":{"id":1,"username":"root", "hobbies": "...", "about": "..."},
7     "publishingDate":"Oct 25, 2014 10:09:55 AM",
8     "likedBy":[
9       {"id":1,"username":"root", "hobbies": "...", "about": "..."},
10      {"id":10,"username":"Hello", "hobbies": "...", "about": "..."}
11    ]
12  },
13  ...
14 ]
```

Aufgabe

Zeichne ein Objektdiagramm für die *GET*-Anfrage `/api/posts?limit=1`.