

New Facebook - RESTful API

HTTP Method	URL	Input	Output
POST	/newFacebook/signup.php רישום למשתמש. פלט: סטטוס 201 והודעה שהמשתמש נוצר	NewUserForm	--
GET	/newFacebook/login.php?username&password לוגין למשתמש. פלט: סטטוס 200 ואת שם המשתמש והסימא נכון לעכשיו (בלי records). data.username. data.password	--	UserTO
GET	/newFacebook/users/search.php?username חיפוש משתמשים לפי שם דומה (LIKE). פלט: רשימת שמות משתמשים בתוך שדה records	--	UserTO[]
GET	/newFacebook/users/request.php?reqUser&resUser שינוי סטטוס חברות בין שני משתמשים, המבקש - reqUser (לא קריטי). אם הם כבר חברים, מבטל את החברות, אחרת יוצר אותה. פלט: סטטוס 200 והודעה שהחברות נוצרה/נמחקה	--	--
GET	/newFacebook/users/details.php?reqUser&resUser בדיקה האם שני משתמשים הם חברים והאם נשלחה הזמנה למשחק. כאשר reqUser = המשתמש הנוכחי, resUser = מי שרוצים לבדוק האם שלח הזמנה. פלט: אובייקט JSON בעל 3 תכונות בוליאניות - friends, invited, inviting. האם שלח לי הזמנה - invited האם שלחתי לו הזמנה - Inviting	--	{ 'friends': boolean, 'invited': boolean, 'inviting': boolean }
POST	/newFacebook/posts/create.php יצירת פוסט חדש. פלט: סטטוס 201 והודעה שהפוסט נוצר בהצלחה	NewPostForm	--
GET	/newFacebook/posts/update.php?username&id שינוי פוסט פרטי לציבורי ולהפך. אם פוסט הוא פרטי, הוא ישתנה לציבורי. פלט: סטטוס 200 והודעה שהפוסט עודכן	--	--
GET	/newFacebook/posts/like.php?username&id לייק או ביטול לייק לפוסט. אם המשתמש אהב את הפוסט, הפעלת ה-URL תבטל את הלייק ולהפך. פלט: סטטוס 200 והודעה שהמשתמש עשה לייק/אנלייק	--	--

GET	/newFacebook/ posts /feed.php?username קבלת הפיד של משתמש מסוים. הפיד כולל את הפוסטים הציבוריים של החברים של אותו משתמש וכל הפוסטים שלו עצמו. פלט: רשימת פוסטים בתוך שדה records, שים לב שכל פוסט מכיל בנוסף את מספר הלייקים ואם המשתמש המבקש אוהב את הפוסט.	--	PostTO[]
POST	/newFacebook/images/upload.php?id העלאת תמונה בודדת. id - מזהה של הפוסט. פלט: URL חלקי (בלי baseUrl) של התמונה (דוגמא images/uploads/fileName.jpg), והודעה שהועלתה בהצלחה קוד להעלאת תמונה	formData	URL(string)
GET	/newFacebook/ posts /profile.php?reqUser&resUser קבלת פוסטים של משתמש מסוים (resUser). אם המבקש (reqUser) רוצה לראות את הפוסטים של עצמו, הוא יקבל גם את הפרטיים, אחרת ניתן לצפות רק בציבוריים. פלט: רשימת פוסטים בתוך שדה records	--	PostTO[]
GET	/newFacebook/ comments /read.php?post_id צפייה בתגובות של פוסט מסוים. פלט: רשימת תגובות בתוך שדה records	--	CommentTO[]
POST	/newFacebook/ comments /create.php כתיבת תגובה חדש לפוסט. פלט: סטטוס 201 והודעה שהתגובה נוצרה	NewCommentForm	--
GET	/newFacebook/game/invite.php?reqUser&resUser יצירה/ביטול הזמנה למשחק בין שני משתמשים, המבקש - reqUser. אם השולח כבר שלח הזמנה, היא תבוטל, אחרת תישמר חדשה. פלט: סטטוס 200 והודעה שההזמנה למשחק נוצרה/נמחקה	--	--

/newFacebook/**posts**/update.php - update only privacy value

UserTO

- username (string)

PostTO

- id (int)
- username (string)
- date (string)
- private (boolean)
- content (string)
- likes (int)
- meLike (boolean)

- comments(sent as string) : [
 - {
 - id,
 - author,
 - date,
 - content
-]
- images(string[] of file names - sent as string) : [
 - `fileName1.jpg`, `fileName2.jpg`
-]

Load images from <http://address/newFacebook/images/uploads/fileName1.jpg>

CommentTO

- id (string)
- post_id (string)
- username (string)
- date (string)
- content (string)

NewUserForm

- username (string)
- password (string)

NewPostForm

- username (string)
- private (boolean) - currently required
- content (string)

NewCommentForm

- username (string)
- post_id (string)
- content (string)

```
const uploadUrl = `images/upload.php?${post.id}`;
const file = event.target.files[0];
const formData = new FormData();
formData.append('image', file); // must be 'image!'
const { data } = await axios({
  url: uploadUrl,
  baseURL: baseUrl,
  method: 'POST',
  data: formData,
  headers: {
    'content-type': 'multipart/form-data'
  }
});
console.log(data.url);
console.log(data.message);
```

```
const uploadUrl = `images/upload.php?${post.id}`;
const file = event.target.files[0];
const formData = new FormData();
formData.append('image', file); // must be 'image!'
axios({
  url: uploadUrl,
  baseURL: baseUrl,
  method: 'POST',
  data: formData,
  headers: {
    'content-type': 'multipart/form-data'
  }
}).then(res => res.data)
.then(data => {
  console.log(data.url);
  console.log(data.message);
});
```