9.3.2023

Elena Chuyan, Fatemeh Sharifiasl

Linköping University

Twidder User’s Guide and API description

Table of Contents

[User guide. 1](#_Toc129213658)

[API description. 3](#_Toc129213659)

# User guide.

1. What is Twidder?

Twidder is a social media Web-application that allows users to exchange messages on so called Walls. Users are able to sign-up, login, post a message on their own wall or on another user’s wall. It is possible for a user to change a password or to reset a password in case if it forgotten.

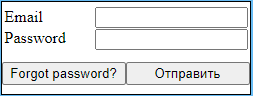
1. How to sign-up?

On the welcome page of the application fill in all fields in specially designated “Sign up here” area click a Submit button. All fields are required to be filled. Password cannot be shorter than 6 symbols.



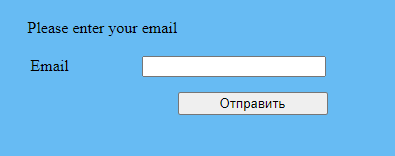
1. How to log-in?

On the welcome page of the application fill in all fields in specially designated area and click a Submit button.



1. How to reset a password?

On the welcome page of the application click a “Forgot password?” button. You will be redirected to a new page where you can fill in your email and request password reset. New automatically generated password will be sent to you via email.

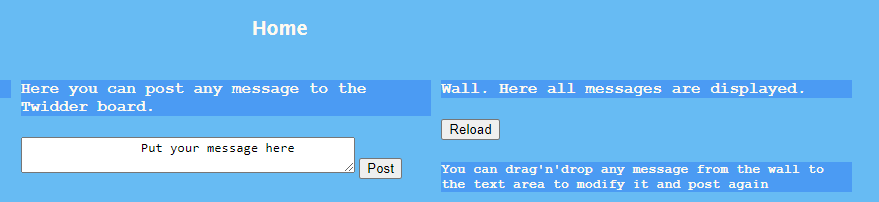


1. How to post a message on your own wall and see the messages posted on your wall?

After log-in you are redirected to the Home tab of your profile. Fill in the “Put your message here” text area and click the Post button. Newly posted message will automatically be displayed on the Wall to the right.

You can click the Reload button to refresh the Wall and check for the messages addressed to you.

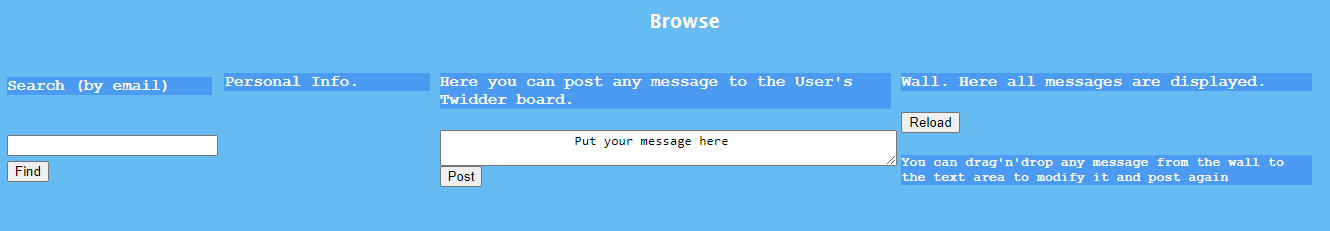
All messages can be dragged and dropped from the wall to the text box area.



1. How to search for another user and post a message to their wall?

Click on the Browse tab and you will be transferred to the form that allows to search for another user. Enter the valid email of another user and click the Find button. If such user is registered on the platform, their personal information will be displayed.

As well as on your own wall, you can post the messages on another user’s wall. Messages are also draggable and droppable.

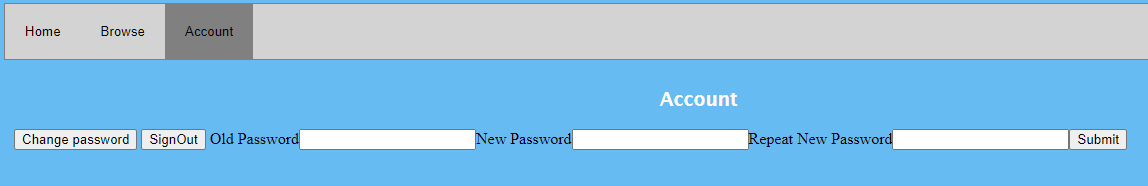


1. How to change a password?

Click on the Account tab and you will be transferred to the form that allows you to change a password. To do so click the Change password button.

Type in the old password and a new password (twice). New password cannot be shorter than 6 symbols.

Click the Submit button.



1. How to sign out from the application?

Click on the Account tab and you will be transferred to the form that allows you to sign ouy. To do so click the SignOut button.

# API description.

|  |
| --- |
| Functionality: Sign-up  Method: POST  URL: /user/create/  Headers: "Content-type", "application/json;charset=UTF-8"  Body: JSON  user = {          email,          password,          firstname,          familyname,          gender,          city,          country      }  Response (success): 201 HTTP Status Code  Error codes:  409, if such user already exists  400, if data format is wrong or any of the required fields are omitted |
| Functionality: Sign-in  Method: POST  URL: /user/login/  Headers: "Content-type", "application/json;charset=UTF-8"  Body: JSON  user = {          email,          password      }  Response (success): token in plain text + 200 HTTP Status Code  Error codes:  404, if such user doesn’t exist  400, if data format is wrong or any of the required fields are omitted |
| Functionality: LogOut  Method: POST  URL: /user/logout/  Headers: "Authorization", token  Body: -  Response (success): 200 HTTP Status Code  Error codes:  404, if such user doesn’t exist  400, if data format is wrong or invalid token is provided |
| Functionality: Post a message on user’s own wall  Method: POST  URL: /post/  Headers: "Authorization", token  "Content-type", "application/json;charset=UTF-8"  Body: JSON  body ={                message,                email,                location,              }  Response (success): 200 HTTP Status Code  Error codes:  401, if invalid token is provided  400, if data format is wrong |
| Functionality: Get user’s personal information by token  Method: GET  URL: /users/current  Headers: "Authorization", token  Body: -  Response (success): 200 HTTP Status Code  {      "city": "",      "country": "",      "email": "",      "family\_name": "",      "firstname": "",      "gender":  ""  }  Error codes:  401, if invalid token is provided  400, if an error occurred |
| Functionality: Get user’s personal information by email  Method: GET  URL: /users/find/<email>  Headers: "Authorization", token  Body: -  Response (success): 200 HTTP Status Code  {      "city": "",      "country": "",      "email": "",      "family\_name": "",      "firstname": "",      "gender":  ""  }  Error codes:  401, if invalid token is provided  404, if email is not found  400, if an error occurred |
| Functionality: Get user’s messages by token  Method: GET  URL: /users/current/messages  Headers: "Authorization", token  Body: -  Response (success): 200 HTTP Status Code  [      {          "location": "",          "message": "",          "writer": ""      },          {          "location": "",          "message": "",          "writer": ""      },      ……  ]  Error codes:  401, if invalid token is provided  400, if an error occurred |
| Functionality: Get user’s messages by email  Method: GET  URL: /user/messages/<email>  Headers: "Authorization", token  Body: -  Response (success): 200 HTTP Status Code  [      {          "location": "",          "message": "",          "writer": ""      },          {          "location": "",          "message": "",          "writer": ""      },      ……  ]  Error codes:  401, if invalid token is provided  400, if an error occurred |
| Functionality: Change password  Method: PUT  URL: /change\_password/  Headers: "Content-type", "application/json;charset=UTF-8"  "Authorization", token  Body: JSON  pass ={          oldPassword,          newPassword,        }  Response (success): 201 HTTP Status Code  Error codes:  401, if invalid token is provided  400, if data format is wrong or any of the required fields are omitted or password is not long enough or old password is wrong |
| Functionality: Reset password  Method: PUT  URL: /reset\_password/  Headers: "Content-type", "application/json;charset=UTF-8"  Body: JSON  user ={          email,         }  Response (success): 200 HTTP Status Code  Error codes:  400, if an error occurred |