



Wender Inc

API

API description

Summary

I.	Changelog	2
II.	General	3
III.	Public routes	3
	Home.....	3
	GCU	3
	Signup.....	3
	Login.....	4
IV.	Private routes	5
	User.....	5
	Research.....	5
	Edit information.....	5
	Friendships	6
	Friendship send request	6
	Friendship requests recuperation	6
	Friendship sent requests	6
	Accept a friendship request.....	7
	Reject or delete a friendship.....	7
	Friends.....	8
	Get your friends list	8
	Get a specific friend.....	8
V.	Websocket.....	9
	Messages	9
	Output (client to server)	9
	Input (server side to client).....	9
	Sequence diagrams of communications.....	10
	Connection to the websocket server	10
	Friends gps coordinate update.....	10
	Update the user gps coordinate	11

Changelog

- New routes:
 - A new route allow deleting a friendship: `/friendships/{id}` with http method DELETE. It can be used to reject a friend request or to delete an existing friendship between 2 users
 - A route now allow to edit the user information (including avatar): `/users/{id}` with http method PUT. The token must correspond to the id passed to the request.
- Routes that have changed:
 - `/friendships/request` is now replaced by `/friendships/sendrequest`
 - `/friendships/requests` is now replaced by `/friendships/sent`
 - `/friendships/pendingrequests` is now replaced by `/friendships/received`
 - `/friendships/acceptrequest` is now replaced by `/friendships/accept/{id}`. It has also changed of http method from POST to PUT
- Deleted routes:
 - `/friends/{id}` is now replaced by `/friendships/{id}` with http method DELETE
- Other:
 - The routes `/friends` and `/friends/{id}` now take an optional parameter “optimized” that can be set in order to receive the user info without the avatar

General

Server	
IP	193.250.96.232
Protocol	HTTPS

Public routes

Home

Route	/
Method	GET
Description	Goto Home
Params	
Returns	

GCU

Route	/gcu
Method	GET
Description	Terms and conditions
Params	
Returns	

Signup

Route	/signup
Method	POST
Description	Subscription to wender
Params	
username	Username
email	User's email
password	User's password (md5)
Returns	
201	User created
400	A user with this email address or username already exists
400	One of the parameters is missing
520	Unable to create a new user (db problem)

Login

Route	/login
Method	POST
Description	Login to wender
Params	
email	User's email
password	User's password
Returns	
200	Access token is given
400	Email / password is not valid
400	One of the parameters is missing

It can be a GET field, a POST field or set in x-access-token request header.

Route	/users/username/{username}
Method	GET
Description	Search for user by username
Params	
{username}	The username to search
Returns	
200	List of the 20 users (light info) whose username starts with {username}

Route	/users/{id}
Method	PUT
Description	Update a user information
Params	
{id}	Id of the user to update (need to match with the given token)
password	The password of the user
newPassword (opt)	Optional: The new password of the user
username (opt)	Optional: The new username of the user
email (opt)	Optional: The new email of the user
avatar (opt)	Optional: The new avatar of the user in base64 format ex: {avatar: "data:image/jpeg;base64,/9j/4RjORXhpZgAATU0AKgAAAAGADAEEAAAMAAAA..."}
Returns	
200	User information updated
200	User information updated but some field may contains errors (see returned message)
400	DB error
400	Given params format error (ex: email not valid)
400	Username and / or email address already used by another account
400	Wrong password
401	No password provided
400	You are not allowed to update this user information
400	The url id field must be an integer

Friendships

Friendship send request

Route	/friendships/sendrequest
Method	POST
Description	Request a friendship relation
Params	
targetUserId	The id of the user to send the friend request
Returns	
201	Friend request sent
400	Friendship request has already been sent to this user
400	The target user has already sent a friendship request to the user
400	Can't find target user
400	Can't find current user
400	One of the parameters is missing

Friendship requests recuperation

Route	/friendships/received
Method	GET
Description	Get the friendship which the user is the target
Params	
Returns	
200	Array of users (light info) who have sent you a request and where the request state is no confirmed

Friendship sent requests

Route	/friendships/sent
Method	GET
Description	Get the unconfirmed friendship requests made by the user
Params	
Returns	
200	Array of users (light info) to whom you have sent a request and where the request state is no confirmed

Accept a friendship request

Route	/friendships/accept/{id}
Method	PUT
Description	Accept a friendship request
Params	
{id}	The id of the user who sent the friend request
Returns	
200	Friend request accepted (note: returned even if no request has been made => no effect in this case)
400	The url id field must be an integer

Reject or delete a friendship

Route	/friendships/{id}
Method	DELETE
Description	Delete a friendship between two users
Params	
{id}	The id of the user to delete the friendship with
Returns	
200	Friend request accepted (note: returned even if no request has been made => no effect in this case)
400	The url id field must be an integer

Friends

Get your friends list

Route	/friends
Method	GET
Description	Get all the user's friends information
Params	
optimized (opt)	Optional: If set to true, the avatar will not be returned. Default value is false
Returns	
200	Array of users (full info) (all your friends)

Get a specific friend

Route	/friends/{id}
Method	GET
Description	Get one user's friend information
Params	
{id}	Id of the friend to get
optimized (opt)	Optional: If set to true, the avatar will not be returned. Default value is false
Returns	
200	User (full info) with id = {id} information
400	There is no confirmed friendship with this id
400	The {id} field must be an integer

Websocket

Messages

Output (client to server)

Authentication

Message	authenticate
Description	Authentication message
Params	
token	The user connection token

Update the user's position

Message	update-position
Description	Send the user's gps coordinates (json object as string. ie {latitude: xx, longitude: xx})
Params	
latitude	the user's latitude
logitude	the user's longitude

Input (server side to client)

Authentication success

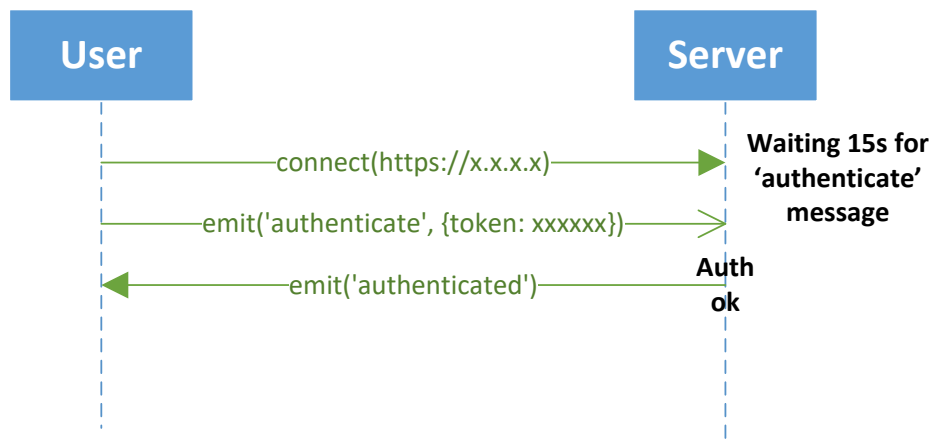
Message	authenticated
Description	receive this message when the user is authenticated by the server
Params	

Use'sr friend updates his gps position

Message	friend-update-position
Description	Be notified of the user's friends gps coordinates (json object received)
Params	
id	the id of the user's friend
latitude	the user's latitude
logitude	the user's longitude

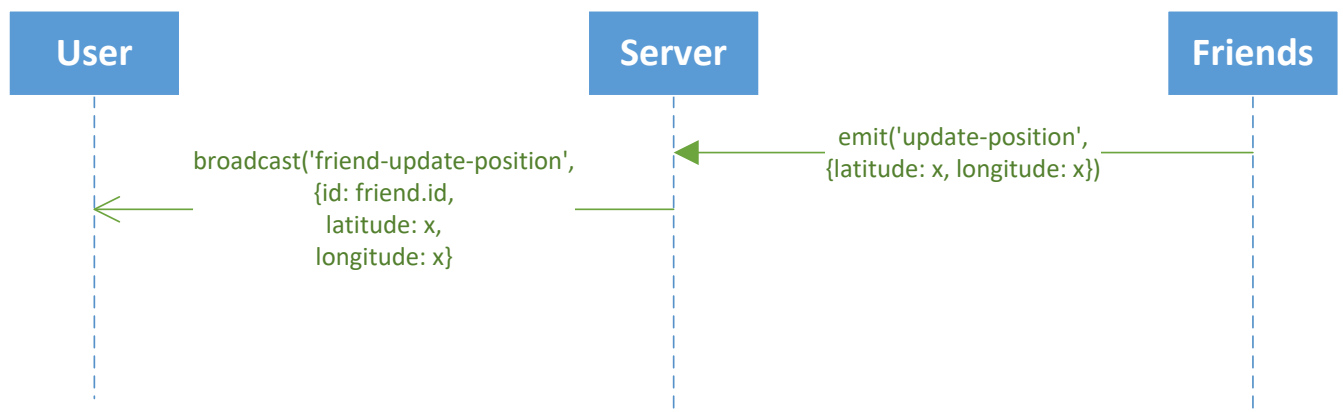
Sequence diagrams of communications

Connection to the websocket server

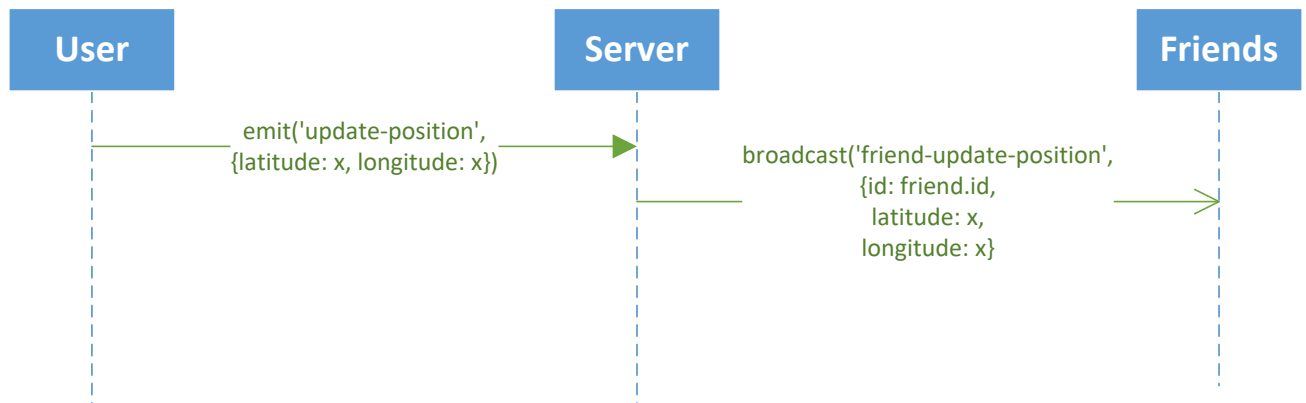


Friends gps coordinate update

Once connected to the web socket server, the user will automatically be informed of all his friends gps update



Update the user gps coordinate



Disconnection

When the user leaves the application, the socket connection will automatically end and the server will broadcast the information to all the user's friends

