

Emergency Social Network API

This is the API for the ESN server

Access

Methods

[[Jump to Models](#)]

Table of Contents

[Chat](#)

- [DELETE /chats/{chatId}](#)
- [GET /chats/{chatId}](#)
- [GET /chats/{username}](#)
- [GET /chats](#)
- [POST /chats](#)
- [PUT /chats/{chatId}](#)

[User](#)

- [POST /users](#)
- [DELETE /users/{username}](#)
- [GET /users/{username}](#)
- [GET /users](#)
- [GET /users/login](#)
- [GET /users/logout](#)
- [PUT /users/{username}](#)

Chat

DELETE /chats/{chatId}

[Up](#)

Deletes a chat (**deleteChat**)

Path parameters

chatId (required)

Path Parameter — Chat id to delete format: int64

Responses

400

Invalid ID supplied

404

Chat not found

GET /chats/{chatId}

[Up](#)

Find chat by ID (**getChatById**)

Returns a single chat

Path parameters

chatId (required)

Path Parameter — ID of chat to return format: int64

Return type

[Chat](#)

Example data

Content-Type: application/json

```
{
  "name" : "Be the first to write something",
  "id" : 0,
  "status" : "available"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

successful operation [Chat](#)

400

Invalid Chat ID

404

Chat not found

GET /chats/{username}

[Up](#)

Find chats by username (**getChatByUsername**)

Returns chats posted by a user

Path parameters

username (required)

Path Parameter — ID of chat to return

Return type

array[[Chat](#)]

Example data

Content-Type: application/json

```
[ {
  "name" : "Be the first to write something",
  "id" : 0,
  "status" : "available"
}, {
```

```
{
  "name" : "Be the first to write something",
  "id" : 0,
  "status" : "available"
} ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses**200**

successful operation

400

Invalid username

404

Username not found

GET /chats

[Up](#)

Get a list of all chats (**getchats**)

This can only be done by the logged in user.

Return type

array[[Chat](#)]

Example data

Content-Type: application/json

```
[ {
  "name" : "Be the first to write something",
  "id" : 0,
  "status" : "available"
}, {
  "name" : "Be the first to write something",
  "id" : 0,
  "status" : "available"
} ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses**200**

List chat object

POST /chats

[Up](#)

Post a chat (**postChat**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [Chat](#) (required)

Body Parameter — Chat object that needs to be published

Query parameters

username (required)

Query Parameter — The username of the author

chat (required)

Query Parameter — The content of the chat

Responses

201

Chat created

PUT /chats/{chatId}

[Up](#)

Updates a chat in the store with form data (**updateChatWithForm**)

Path parameters

chatId (required)

Path Parameter — ID of chat that needs to be updated format: int64

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body

body [Chat](#) (required)

Body Parameter — Chat object that needs to be published

Query parameters

chat (required)

Query Parameter — The content of the chat

Responses

200

successful operation

400

Invalid Chat ID

404

Chat not found

405

Invalid input

User

POST /users

[Up](#)

Create user (**createUser**)

This can only be done by the logged in user.

Query parameters

username (required)

Query Parameter — The user name for login

password (required)

Query Parameter — The password for login in clear text

firstName (optional)

Query Parameter — The firstname of the user

lastName (optional)

Query Parameter — The lastname of the user

email (optional)

Query Parameter — The email of the user

phone (optional)

Query Parameter — The phone of the user

Responses

200

user exists

201

successful operation

DELETE /users/{username}

[Up](#)

Delete user (**deleteUser**)

This can only be done by the logged in user.

Path parameters

username (required)

Path Parameter — The name that needs to be deleted

Responses

400

Invalid username supplied

404

User not found

GET /users/{username}

[Up](#)

Get user by user name (**getUserByName**)

Path parameters

username (required)

Path Parameter — The name that needs to be fetched. Use user1 for testing.

Return type

[User](#)

Example data

Content-Type: application/json

```
{
  "firstName" : "firstName",
  "lastName" : "lastName",
  "password" : "password",
  "userStatus" : 6,
  "phone" : "phone",
  "id" : 0,
  "email" : "email",
  "username" : "username"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

successful operation [User](#)

400

Invalid username supplied

404

User not found

GET /users

[Up](#)

Get a list of all users (**getUsers**)

This can only be done by the logged in user.

Return type

array[[User](#)]

Example data

Content-Type: application/json

```
[ {
  "firstName" : "firstName",
  "lastName" : "lastName",
  "password" : "password",
  "userStatus" : 6,
  "phone" : "phone",
  "id" : 0,
```

```
{
  "email" : "email",
  "username" : "username"
}, {
  "firstName" : "firstName",
  "lastName" : "lastName",
  "password" : "password",
  "userStatus" : 6,
  "phone" : "phone",
  "id" : 0,
  "email" : "email",
  "username" : "username"
} ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses**200**

OK

GET /users/login

[Up](#)

Logs user into the system (**loginUser**)

Query parameters**username (required)**

Query Parameter — The user name for login

password (required)

Query Parameter — The password for login in clear text

Return type[User](#)**Example data**

Content-Type: application/json

```
{
  "firstName" : "firstName",
  "lastName" : "lastName",
  "password" : "password",
  "userStatus" : 6,
  "phone" : "phone",
  "id" : 0,
  "email" : "email",
  "username" : "username"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses**200**

successful operation [User](#)

400

Invalid username/password supplied

GET /users/logout

[Up](#)

Logs out current logged in user session (**logoutUser**)

Responses**default**

successful operation

PUT /users/{username}

[Up](#)

Updated user (**updateUser**)

This can only be done by the logged in user.

Path parameters**username (required)**

Path Parameter — name that need to be updated

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request body**body [User](#) (required)**

Body Parameter — Updated user object

Responses**200**

User updated

400

Invalid user supplied

404

User not found

Models

[[Jump to Methods](#)]

Table of Contents

1. [Chat](#)
2. [User](#)

Chat[Up](#)**id (optional)**[Long](#) format: int64**name (optional)**[String](#)*example: Be the first to write something***status (optional)**[String](#) chat ready for the review**Enum:***available**reported**deleted***User**[Up](#)**id (optional)**[Long](#) format: int64**username (optional)**[String](#)**firstName (optional)**[String](#)**lastName (optional)**[String](#)**email (optional)**[String](#)**password (optional)**[String](#)**phone (optional)**[String](#)**userStatus (optional)**[Integer](#) User Status format: int32