

Chat Application - API

The api location starts with /api/v[versionNumber].

User management

The user management can be found under /users

What is a user.

A user is an entity which can have the following attributes:

- id
- username (unique)
- password (must be at least 10 digits long)
- icon

The icon should be requestable in a small and a big format.

Routes

In this specification new User is always an object like this:

```
user = {
  id: [String],
  username: [String],
  password: [String],
  smallIcon: [String] (url),
  bigIcon: [String] (url)
}
```

And the routes have to look like:

Route	Arguments	Response	Description
POST /users	username, password	new User	Register a user.
POST /users/:username	username, password	new User	Log in a user.
PUT /users/:username	(all attributes)	new User	Update an user.
DELETE /users/:username	username, password	status	Delete an user.
GET /users/:username		user	Get the information about a user.

```
| GET /users/logout          | status | Log out the user. |
|
|-----|-----|-----|-----|
|-----|
```

Contact Management

What is a contact

A contact is identified by its id. It basically is a user, which is in one room with the user. Therefore a contact is represented by a room. A room has the following form:

```
room = {
  id: [String],
  members: [Array of users],
  messages: [Array of Message]
}
```

And the routes have to look like:

Route	Arguments	Response	Description
POST /rooms	members [Array]	room	Create a new Room
PUT /rooms/:id	room	room	Update a room
GET /rooms	username	rooms [Array]	Get all rooms of a user.
DELETE /rooms/:id		status	Delete a room.

Messages

What is a message

A Message is text which can be send in a room.

It has the following form:

```
message = {
  type: [String],
  payload: [Object],
  roomId: [String] (if not specified -> for all rooms),
  createdAt: [JS Timestamp]
}
```

The type can be:

- TEXT_MESSAGE

- text [String]
- USER_CONNECTED
 - username
 - candidate [Object]
- USER_DISCONNECTED
 - username
- VIDEO_CALL_START
- VIDEO_CALL_END

How to exchange

Messages will be exchanged via websockets.