

Client-Server communications protocol

New user registration

client->server

```
{
  "username": "",
  "password": "",
  "nickname": "",
  "description": "",
  "avatarUrl": ""
}
```

server -> client

On success

```
{
  "messageType": "authSuccess",
  "username": {
    "username": "",
    "nickname": "",
    "description": "",
    "avatarUrl": ""
  },
  "subscribedChannels": [
    {
      "channelId": "",
      "channelThread": [],
      "users": [
        {
          "nickname": "",
          "description": "",
          "avatarUrl": ""
        }
      ],
      "description": "",
      "numberOfSubscribers": 3, /* assuming some int */
      "isPublic": true
    }
  ],
  "privateChannels": []
}
```

On failure

```
{
  "messageType": "authFailure",
```

```
    "error": ""
}
```

User enters a system (Login)

client->server

```
{
  "username": "",
  "password": ""
}
```

server -> client

Same parameters as on login success and same for failure

New channel creation

client->server

```
{
  "messageType": "createChannel",
  "messageContent": {
    "channelId": "",
    "description": "",
    "username": "" /* only if private channel */
  }
}
```

server -> client

On success

```
{
  "messageType": "channelSuccess",
  "channel": {
    "channelId": "",
    "channelThread": [],
    "usernames": [
      {
        "nickname": "",
        "description": "",
        "avatarUrl": ""
      }
    ],
    "description": "",
    "numberOfSubscribers": 3,
    "isPublic": true
  }
}
```

On failure

```
{
  "messageType": "channelFailure",
  "channel": "",
  "error": ""
}
```

New subscription

client->server

```
{
  "messageType": "subscribe",
  "messageContent": {
    "channelId": ""
  }
}
```

server -> client

On success, to the user made operation

```
{
  "messageType": "subscribeSuccess",
  "channel": {
    "channelId": "",
    "channelThread": [],
    "usernames": [
      {
        "nickname": "",
        "description": "",
        "avatarUrl": ""
      }
    ],
    "description": "",
    "numberOfSubscribers": 3,
    "isPublic": true
  }
}
```

On success, to other users, except one who subscribed

```
{
  "messageType": "userSubscribed",
  "channel": "",
}
```

```
    "username": {
      "username": "",
      "nickname": "",
      "description": "",
      "avatarUrl": ""
    }
  }
}
```

Unsubscription

client->server

```
{
  "messageType": "unsubscribe",
  "messageContent": {
    "channelId": ""
  }
}
```

server -> client

On success, to user who unsubscribed

```
{
  "messageType": "unsubscribe",
  "channelId": "",
  "error": "" /* null if success */
}
```

On success, to other users, except user himself, if he has additional session

```
{
  "messageType": "userUnsubscribed",
  "channelId": "",
  "username": ""
}
```

Channel discovery

client->server

```
{
  "messageType": "channelDiscovery",
  "messageContent": {
```

```
        "query": ""
    }
}
```

server -> client

```
{
  "messageType": "channelDiscovery",
  "channels": [
    {
      "channelId": "",
      "channelThread": [],
      "users": [],
      "description": "",
      "numberOfSubscribers": 3,
      "isPublic": true
    }
  ]
}
```

Channel viewing

client->server

```
{
  "messageType": "channelViewing",
  "messageContent": {
    "channelId": ""
  }
}
```

server -> client

No response

Download messages from server

client->server

```
{
  "messageType": "downloadMessages",
  "messageContent": {
    "channelId": ""
  }
}
```

server -> client

```
{
  "messageType": "downloadMessages",
  "channelId": "",
  "channelThread": [
    {
      "message": {
        "id": 1,
        "user": {
          "username": "",
          "nickname": "",
          "description": "",
          "avatarUrl": ""
        },
        "messageTime": "time in milliseconds",
        "repliedToId": -1,
        "content": ""
      },
      "replies": []
    }
  ]
  "unreadmessages": 0,
  "unreadMentionedmessages": 0
}
```

Message sending/receivment

client->server

```
{
  "messageType": "sendMessage",
  "messageContent": {
    "message": {
      "channel": "",
      "repliedToId": -1, /* here id of the message, -1 by default */
      "content": ""
    }
  }
}
```

server -> client

Response with downloadMessages json.

Update counters

client->server

This json is uni-directed and is only sent from server.

server -> client

```
{
  "messageType": "sendMessage",
  "messageContent": {
    "channelId": ""
    "unreadmessages": 0,
    "unreadMentionedmessages": 0
  }
}
```