WhatsChat - Programmieraufgabe 1

Timon Baldow	384023
Leonard Kinzinger	393510
Jan Tiegges	393523
Mika Dietz	394284

AWSys HA 01: Gruppe 52

Overview

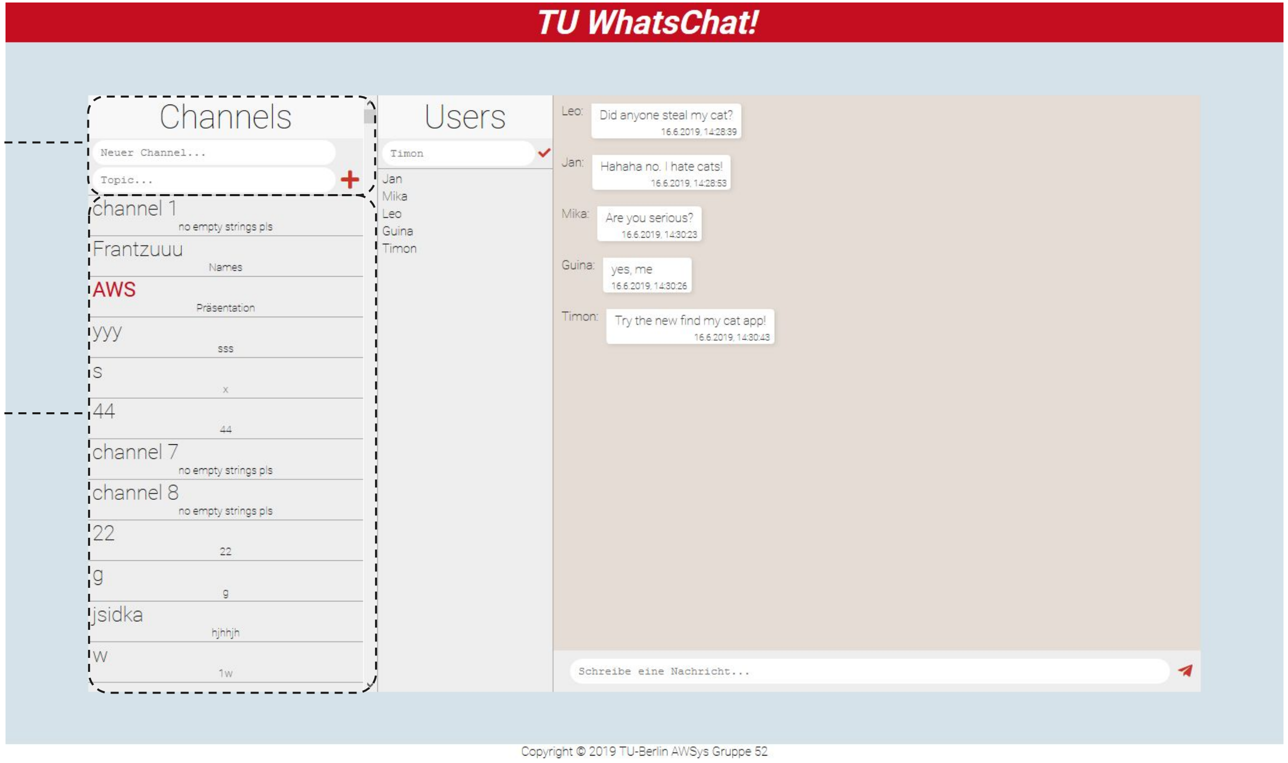
Option to add Channel with **Channel name** and **topic**

Click the + button to create channels on Server

All channels that exist on the server are displayed in the scrollable list below.

The currently viewed channel is marked in red.

The user can switch channels by selecting a channel.



TU WhatsChat!

Channels

Neuer Channel...

Topic...

channel 1 — no empty strings pls
Frantzuuu — Names
AWS — Präsentation
yyy — sss
s — x
44 — 44
channel 7 — no empty strings pls
channel 8 — no empty strings pls
22 — 22
g — g
jsidka — hjhhjh
w — 1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Schreibe eine Nachricht...

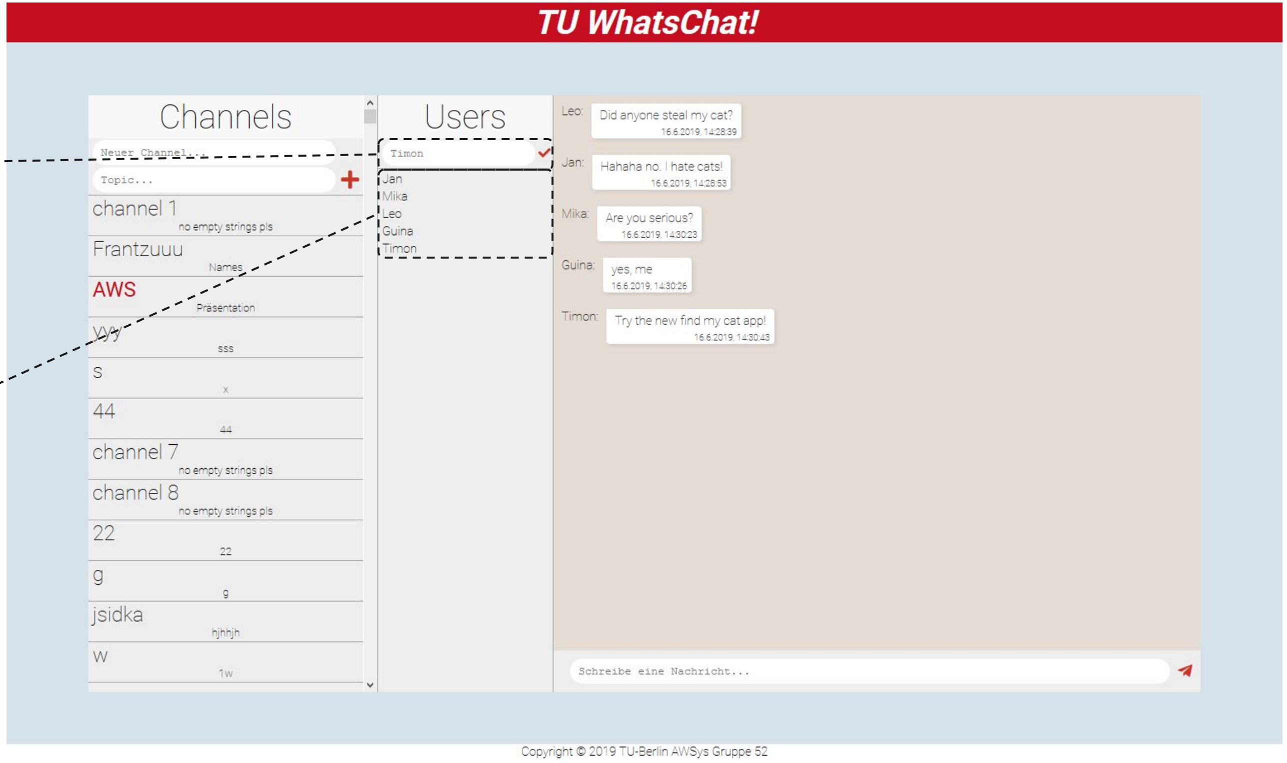
Copyright © 2019 TU-Berlin AWSys Gruppe 52

Overview

Option to add a **user name** in input field. The placeholder and **default value** for the user name is **anonymous**.

Click the check button to apply the new username to the chat.

All the current users with their corresponding user names are displayed in the list.



TU WhatsChat!

Channels

Neuer Channel...

Topic...

channel 1	no empty strings pls
Frantzuuu	Names
AWS	Präsentation
yyy	sss
s	x
44	44
channel 7	no empty strings pls
channel 8	no empty strings pls
22	22
g	g
jsidka	hjhhjh
w	1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

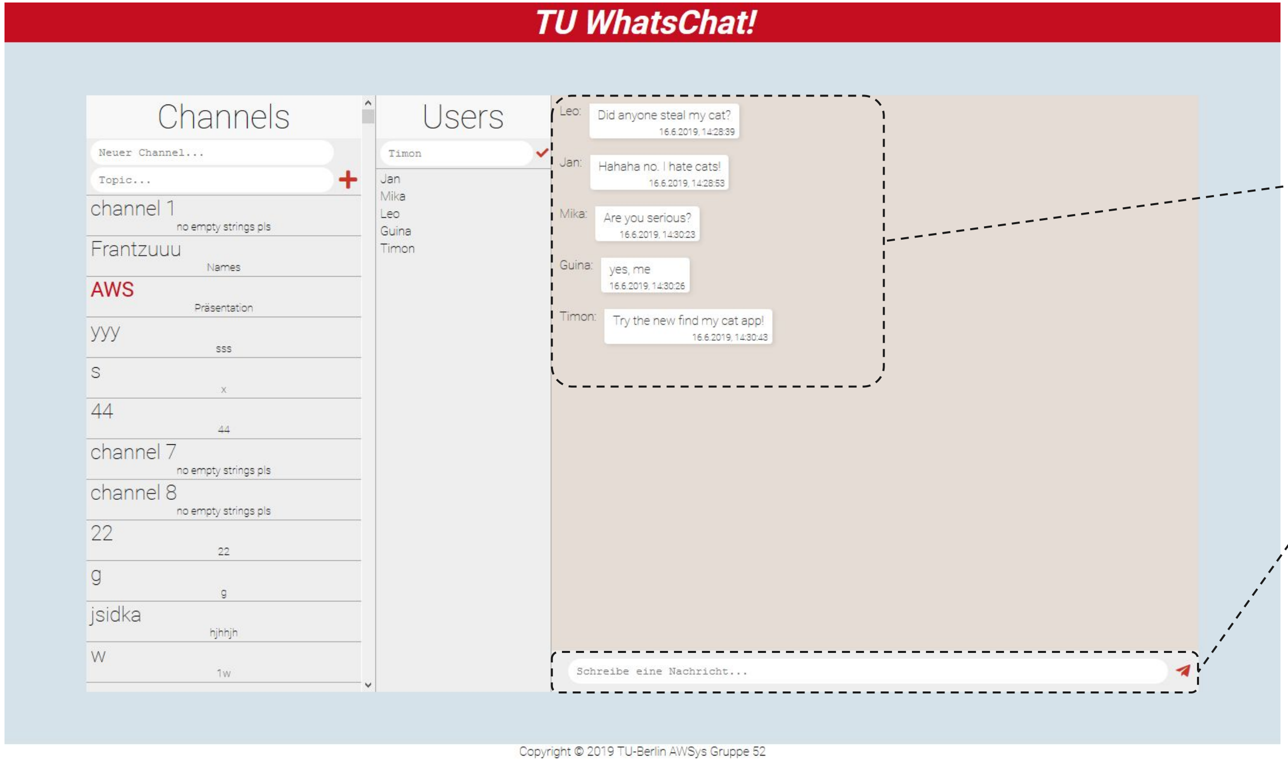
Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Schreibe eine Nachricht...

Copyright © 2019 TU-Berlin AWSys Gruppe 52

Overview


TU WhatsChat!

Channels

Neuer Channel...

Topic...

channel 1
no empty strings pls

Frantzuuu
Names

AWS
Präsentation

yyy
sss

s
x

44
44

channel 7
no empty strings pls

channel 8
no empty strings pls

22
22

g
g

jsidka
hjhhjh

w
1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Schreibe eine Nachricht...

Copyright © 2019 TU-Berlin AWSys Gruppe 52

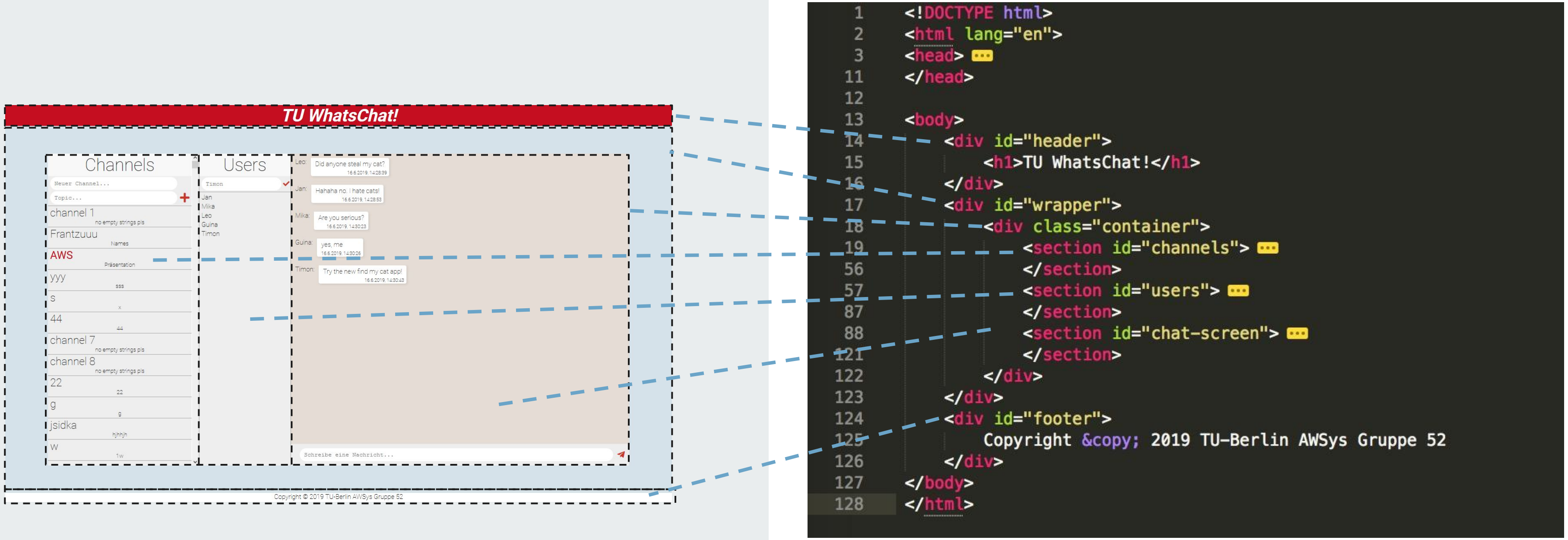
The last 10 messages (implemented on server) are displayed in the chat window.

Each message contains a **user name**, **content** and a **timestamp**.

The text area enables one to send messages to the current channel. You can either send the messages by clicking on the paperplane or hitting enter.

General HTML

HTML



TU WhatsChat!

Channels

Neuer Channel...

Topic...

channel 1
no empty strings pls

Frantzuuu
Names

AWS
Präsentation

yyy
sss

s
x

44
44

channel 7
no empty strings pls

channel 8
no empty strings pls

22
22

g
g

jsidka
hjhhjh

w
1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat?
16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats!
16.6.2019, 14:28:53

Mika: Are you serious?
16.6.2019, 14:30:23

Guina: yes, me
16.6.2019, 14:30:26

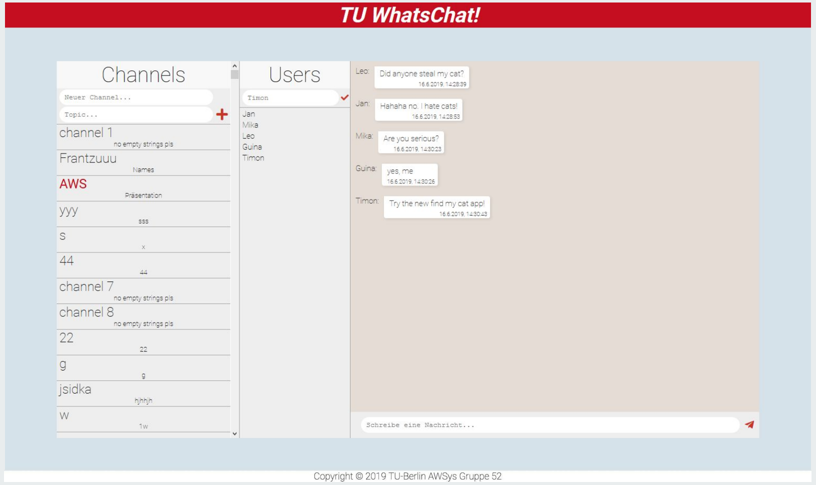
Timon: Try the new find my cat app!
16.6.2019, 14:30:43

Schreibe eine Nachricht...

Copyright © 2019 TU-Berlin AWSys Gruppe 52

```html
<!DOCTYPE html>
<html lang="en">
<head> ···
</head>

<body>
    <div id="header">
        <h1>TU WhatsChat!</h1>
    </div>
    <div id="wrapper">
        <div class="container">
            <section id="channels"> ···
            </section>
            <section id="users"> ···
            </section>
            <section id="chat-screen"> ···
            </section>
        </div>
    </div>
    <div id="footer">
        Copyright &copy; 2019 TU-Berlin AWSys Gruppe 52
    </div>
</body>
</html>
```

General CSS



TU WhatsChat!

Channels

Neuer Channel...

Topic...

channel 1
no empty strings pls

Frantzuuu
Names

AWS
Präsentation

yyy
sss

s
x

44
44

channel 7
no empty strings pls

channel 8
no empty strings pls

22
22

g
g

jsidka
hjhhjh

w
1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

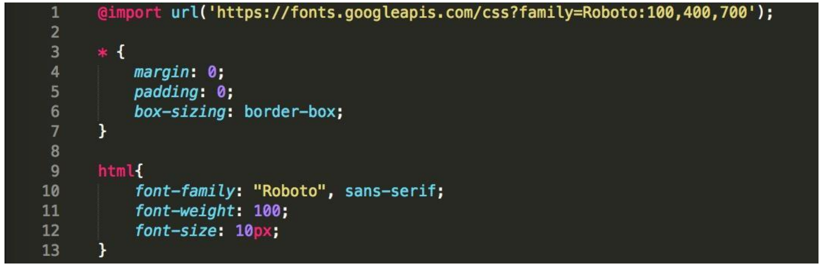
Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Schreibe eine Nachricht...

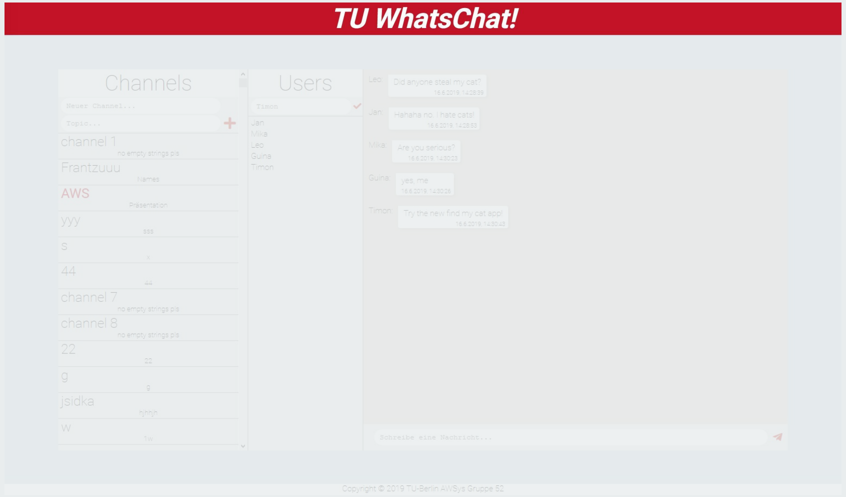
Copyright © 2019 TU-Berlin AWSys Gruppe 52

CSS



```css
@import url('https://fonts.googleapis.com/css?family=Roboto:100,400,700');

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

html{
    font-family: "Roboto", sans-serif;
    font-weight: 100;
    font-size: 10px;
}
```

Header


TU WhatsChat!

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

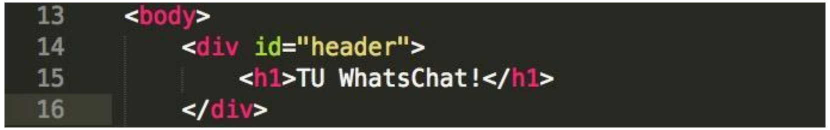
Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

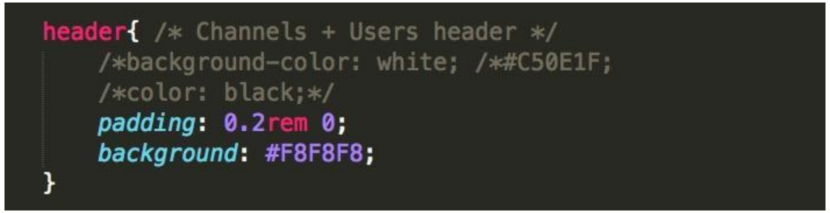
Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Copyright © 2019 TU-Berlin AWSys Gruppe 52

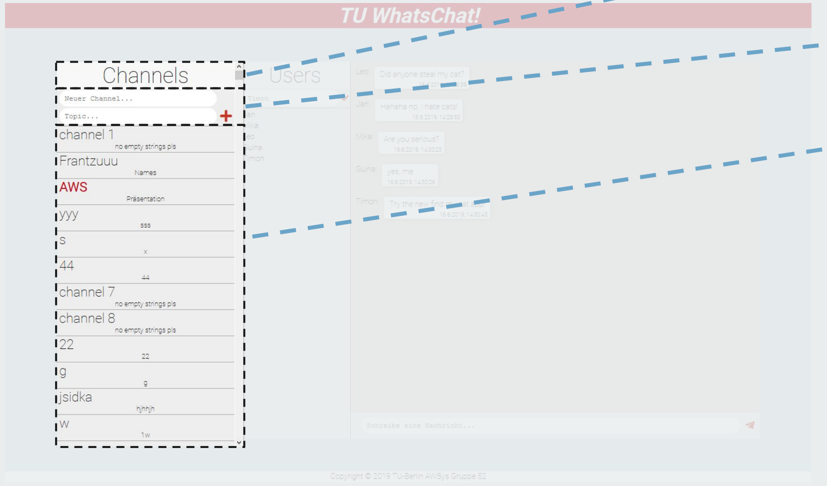
HTML


```html
    <body>
        <div id="header">
            <h1>TU WhatsChat!</h1>
        </div>
```

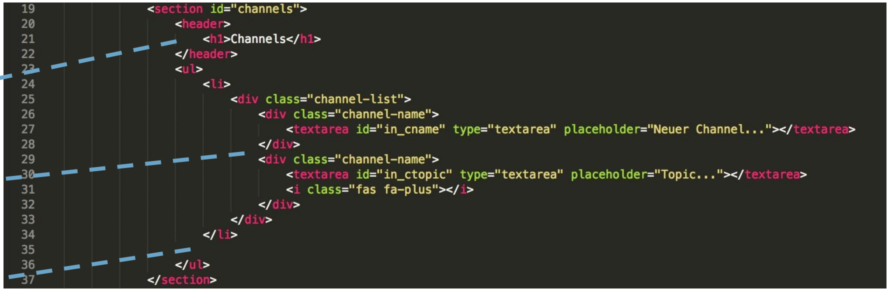
CSS


```css
header{ /* Channels + Users header */
    /*background-color: white; /*#C50E1F;
    /*color: black;*/
    padding: 0.2rem 0;
    background: #F8F8F8;
}
```

Channels

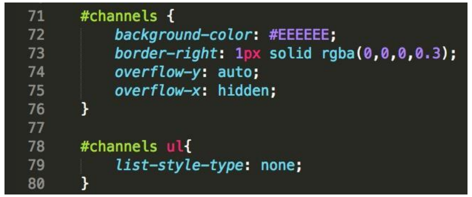


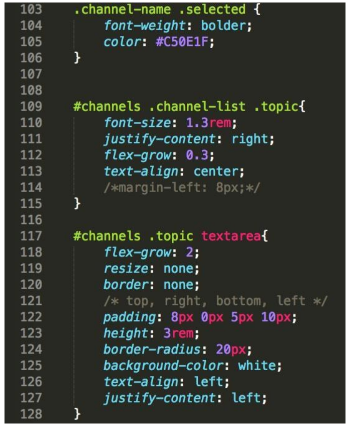
HTML


```html
<section id="channels">
    <header>
        <h1>Channels</h1>
    </header>
    <ul>
        <li>
            <div class="channel-list">
                <div class="channel-name">
                    <textarea id="in_cname" type="textarea" placeholder="Neuer Channel..."></textarea>
                </div>
                <div class="channel-name">
                    <textarea id="in_ctopic" type="textarea" placeholder="Topic..."></textarea>
                    <i class="fas fa-plus"></i>
                </div>
            </div>
        </li>

    </ul>
</section>
```

- Channels will be appended to unordered list by Javascript (see Channels Javascript - _addChannelToScreen())

CSS


```css
#channels {
    background-color: #EEEEEE;
    border-right: 1px solid rgba(0,0,0,0.3);
    overflow-y: auto;
    overflow-x: hidden;
}

#channels ul{
    list-style-type: none;
}
```


```css
.channel-name .selected {
    font-weight: bolder;
    color: #C50E1F;
}


#channels .channel-list .topic{
    font-size: 1.3rem;
    justify-content: right;
    flex-grow: 0.3;
    text-align: center;
    /*margin-left: 8px;*/
}

#channels .topic textarea{
    flex-grow: 2;
    resize: none;
    border: none;
    /* top, right, bottom, left */
    padding: 8px 0px 5px 10px;
    height: 3rem;
    border-radius: 20px;
    background-color: white;
    text-align: left;
    justify-content: left;
}
```

- Not all CSS is displayed here, for more, see style.css

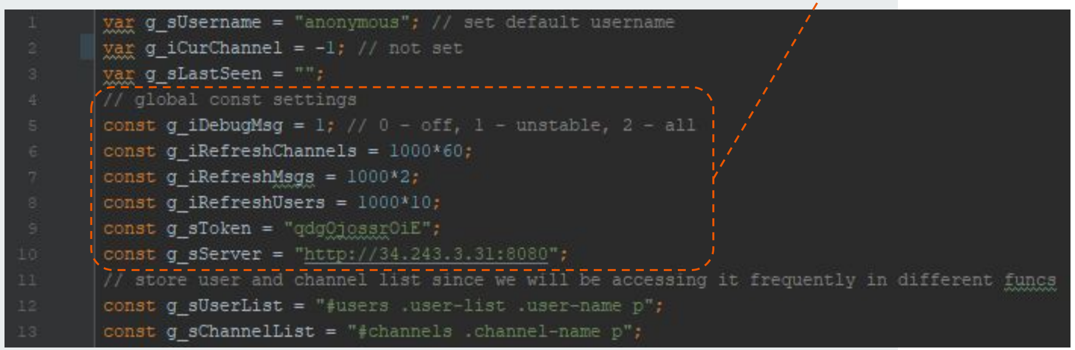
Basic JavaScript

→ Global script constants are used to ensure efficiency.

→ Script-wide unified variable names.

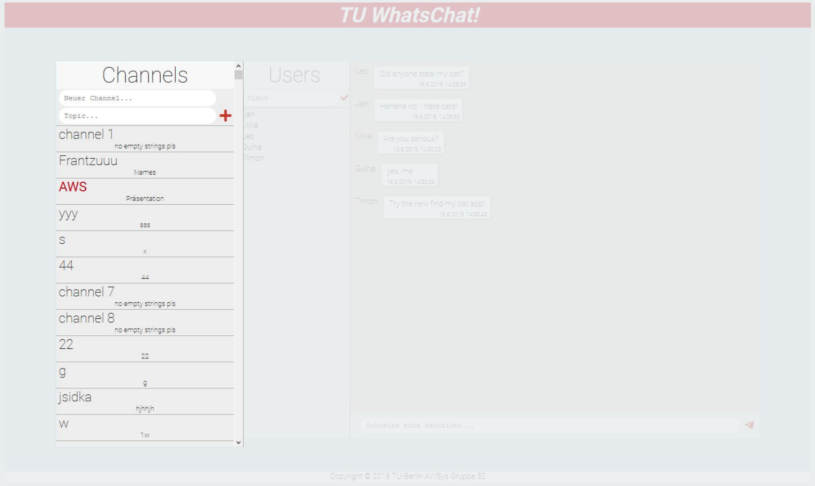
Standard: A variable starts with a lowercase letter indicating the proposed variable type (since JS uses variants this helps avoid accidental casting). For example: s → String, i → Integer, a → Array.
After the indicator the actual name follows starting with a capitalized character

Global variables: Start with „g_“ followed by their standard unified variable name.



```javascript
var g_sUsername = "anonymous"; // set default username
var g_iCurChannel = -1; // not set
var g_sLastSeen = "";
// global const settings
const g_iDebugMsg = 1; // 0 - off, 1 - unstable, 2 - all
const g_iRefreshChannels = 1000*60;
const g_iRefreshMsgs = 1000*2;
const g_iRefreshUsers = 1000*10;
const g_sToken = "qdgOjossrOiE";
const g_sServer = "http://34.243.3.31:8080";
// store user and channel list since we will be accessing it frequently in different funcs
const g_sUserList = "#users .user-list .user-name p";
const g_sChannelList = "#channels .channel-name p";
```

Channels



channel 1 — no empty strings pls
Frantzuuu — Names
AWS — Präsentation
yyy — sss
s — x
44 — 44
channel 7 — no empty strings pls
channel 8 — no empty strings pls
22 — 22
g — g
jsidka — hjhhjh
w — 1w

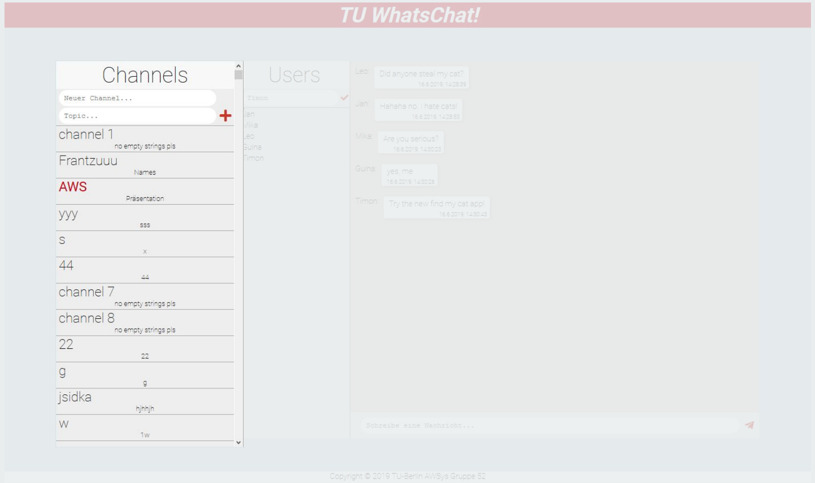
Leo: Did anyone steal my cat? 16.6.2019, 14:28:39
Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53
Mika: Are you serious? 16.6.2019, 14:30:23
Guina: yes, me 16.6.2019, 14:30:26
Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Copyright © 2019 TU-Berlin AWSys Gruppe 52

JS

- All available Channels will be loaded from server by the **_getChannels()** function
- **setInterval**(_getChannels, g_iRefreshChannels) is used for background polls every 60s (g_iRefreshChannels)
- If user clicks + button to create Channel $("#channels i").click(function () will be called which then calls **_createNewChannel(sChannel, sTopic)** to perform Ajax POST call to the server
- To add the channels to the list, the function _getChannels() calls **_addChannelToScreen(iID, sChannel, sTopic = '')**
- If user switches the channel by clicking on any channel item in the list the function **_switchChannel(sChannel)** will be called by the .click event
- We also implemented a function **_prepChannelName()** to trim whitespaces, set type to String and block code injection

Channels


TU WhatsChat!

Channels

Neuer Channel...

Topic...

channel 1
no empty strings pls

Frantzuuu
Names

AWS
Präsentation

yyy
sss

s
x

44
44

channel 7
no empty strings pls

channel 8
no empty strings pls

22
22

g
g

jsidka
hjhhjh

w
1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

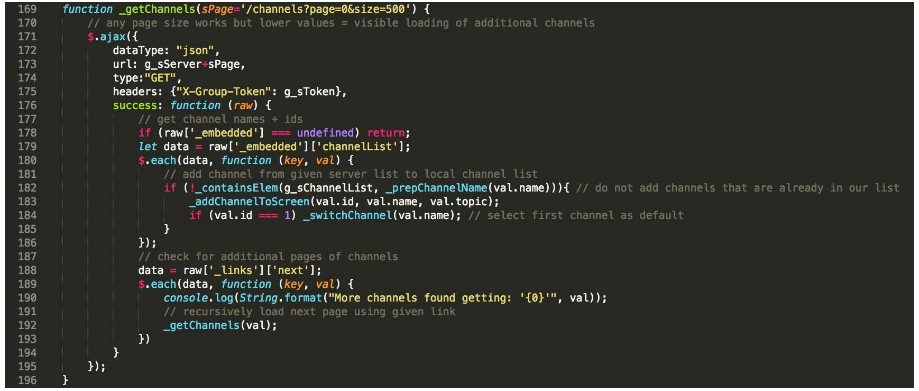
Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

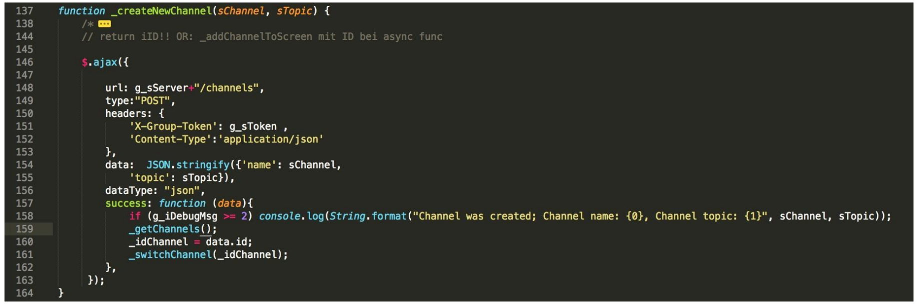
Schreibe eine Nachricht...

Copyright © 2019 TU-Berlin AWSys Gruppe 52

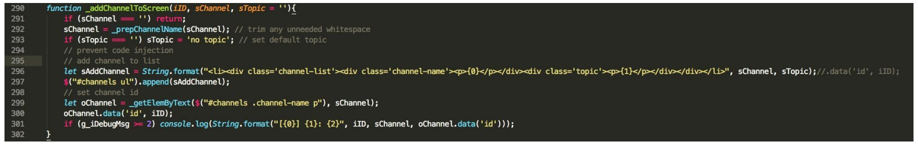
_getChannels()


```javascript
function _getChannels(sPage='/channels?page=0&size=500') {
    // any page size works but lower values = visible loading of additional channels
    $.ajax({
        dataType: "json",
        url: g_sServer+sPage,
        type:"GET",
        headers: {"X-Group-Token": g_sToken},
        success: function (raw) {
            // get channel names + ids
            if (raw['_embedded'] === undefined) return;
            let data = raw['_embedded']['channelList'];
            $.each(data, function (key, val) {
                // add channel from given server list to local channel list
                if (!_containsElem(g_sChannelList, _prepChannelName(val.name))){ // do not add channels that are already in our list
                    _addChannelToScreen(val.id, val.name, val.topic);
                    if (val.id === 1) _switchChannel(val.name); // select first channel as default
                }
            });
            // check for additional pages of channels
            data = raw['_links']['next'];
            $.each(data, function (key, val) {
                console.log(String.format("More channels found getting: '{0}'", val));
                // recursively load next page using given link
                _getChannels(val);
            })
        }
    });
}
```

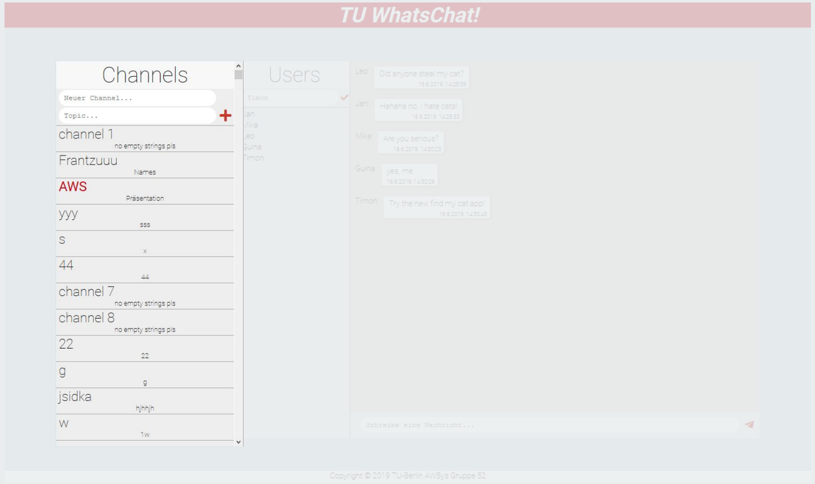
_createNewChannel()


```javascript
function _createNewChannel(sChannel, sTopic) {
    /* ···
    // return iID!! OR: _addChannelToScreen mit ID bei async func

    $.ajax({

        url: g_sServer+"/channels",
        type:"POST",
        headers: {
            'X-Group-Token': g_sToken ,
            'Content-Type':'application/json'
        },
        data:  JSON.stringify({'name': sChannel,
            'topic': sTopic}),
        dataType: "json",
        success: function (data){
            if (g_iDebugMsg >= 2) console.log(String.format("Channel was created; Channel name: {0}, Channel topic: {1}", sChannel, sTopic));
            _getChannels();
            _idChannel = data.id;
            _switchChannel(_idChannel);
        },
    });
}
```

_addChannelToScreen ()


```javascript
function _addChannelToScreen(iID, sChannel, sTopic = ''){
    if (sChannel === '') return;
    sChannel = _prepChannelName(sChannel); // trim any unneeded whitespace
    if (sTopic === '') sTopic = 'no topic'; // set default topic
    // prevent code injection
    // add channel to list
    let sAddChannel = String.format("<li><div class='channel-list'><div class='channel-name'><p>{0}</p></div><div class='topic'><p>{1}</p></div></div></li>", sChannel, sTopic);//.data('id', iID);
    $("#channels ul").append(sAddChannel);
    // set channel id
    let oChannel = _getElemByText($("#channels .channel-name p"), sChannel);
    oChannel.data('id', iID);
    if (g_iDebugMsg >= 2) console.log(String.format("[{0}] {1}: {2}", iID, sChannel, oChannel.data('id')));
}
```

Channels


TU WhatsChat!

Channels

Neuer Channel...

Topic...

channel 1 — no empty strings pls
Frantzuuu — Names
AWS — Präsentation
yyy — sss
s — x
44 — 44
channel 7 — no empty strings pls
channel 8 — no empty strings pls
22 — 22
g — g
jsidka — hjhhjh
w — 1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39
Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53
Mika: Are you serious? 16.6.2019, 14:30:23
Guina: yes, me 16.6.2019, 14:30:26
Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Schreibe eine Nachricht...

Copyright © 2019 TU-Berlin AWSys Gruppe 52

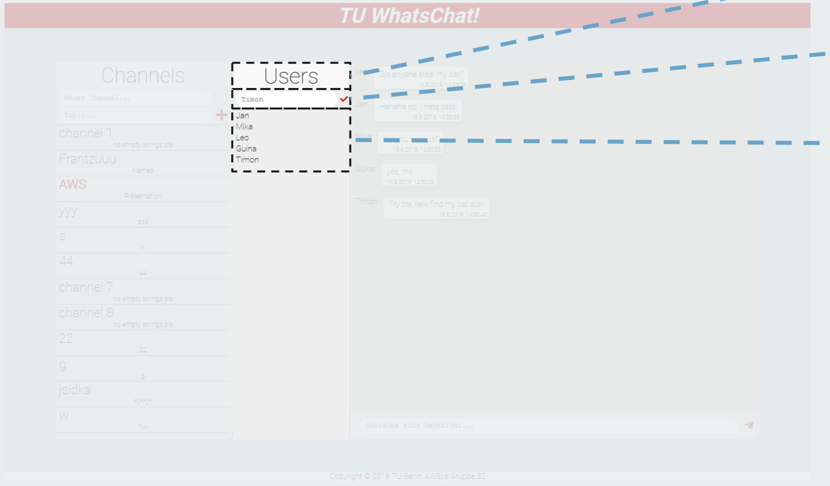
_switchChannel()

```javascript
function _switchChannel(sChannel) {
    $("#channels .channel-name p.selected").removeClass("selected"); // remove class selected from currently selected channe
    let oChannel = _getElemByText($("#channels .channel-name p"), _prepChannelName(sChannel));
    let iChannelId = oChannel.data('id');
    oChannel.addClass("selected");

    console.log(String.format("Switching channel to '{0}' with id {1}", sChannel, iChannelId));

    // show channel messages
    _clearChatScreen();
    _getMessages(iChannelId);
    g_curChannel = iChannelId;

    // show channel users
    _getUserList(g_curChannel);
}
```

_prepChannelName()

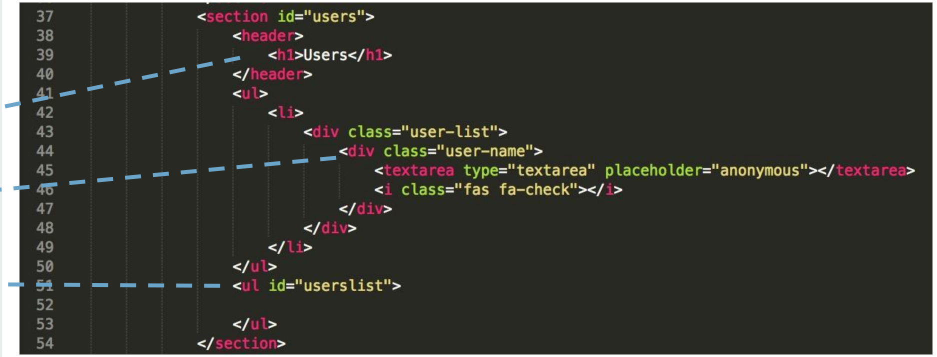
```javascript
function _prepChannelName(sChannel) {
    /*
        Function:    _prepChannelName
        Description: prepares the name of a channel (as string) to be added into the list
        Params:      sChannel - String
        Returns:     Updated String
     */

    if (typeof sChannel !== 'string') return 'invalid_ChannelName';
    sChannel = sChannel.trim(); // trim any unneeded whitespace
    // block code injection
    return sChannel.replace(/(?:<script>)?(?:alert)\(?'?(.+?)'\)?(?:<\/script>)?/, "$1");
}
```

Users



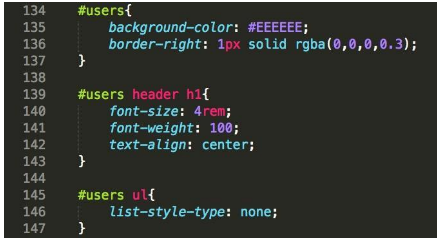
Leo: Did anyone steal my cat? 16.6.2019, 14:28:39
Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53
Mika: Are you serious? 16.6.2019, 14:30:23
Guina: yes, me 16.6.2019, 14:30:26
Timon: Try the new find my cat app! 16.6.2019, 14:30:43

HTML

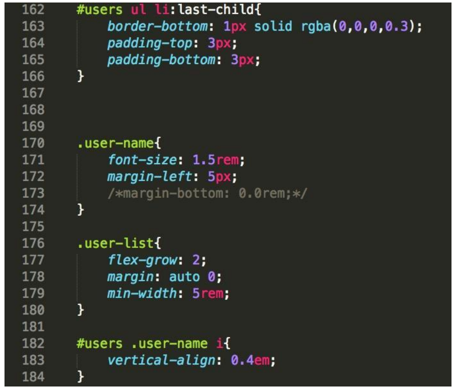


```html
<section id="users">
    <header>
        <h1>Users</h1>
    </header>
    <ul>
        <li>
            <div class="user-list">
                <div class="user-name">
                    <textarea type="textarea" placeholder="anonymous"></textarea>
                    <i class="fas fa-check"></i>
                </div>
            </div>
        </li>
    </ul>
    <ul id="userslist">

    </ul>
</section>
```

CSS

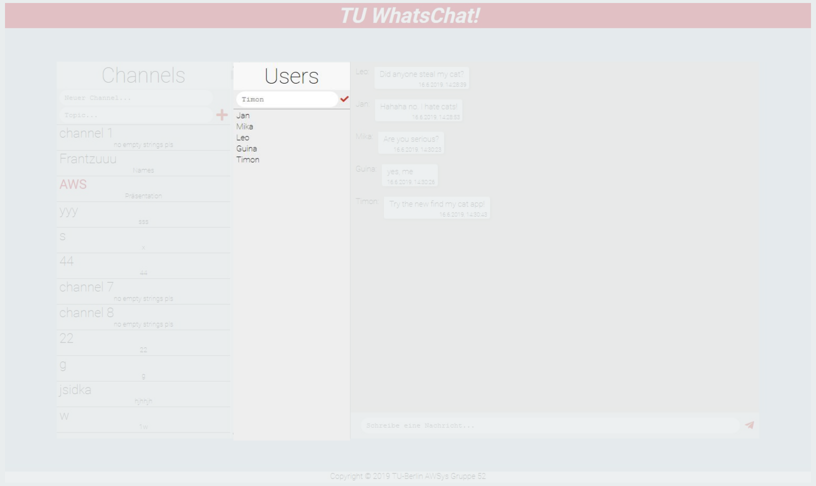


```css
#users{
    background-color: #EEEEEE;
    border-right: 1px solid rgba(0,0,0,0.3);
}

#users header h1{
    font-size: 4rem;
    font-weight: 100;
    text-align: center;
}

#users ul{
    list-style-type: none;
}
```



```css
#users ul li:last-child{
    border-bottom: 1px solid rgba(0,0,0,0.3);
    padding-top: 3px;
    padding-bottom: 3px;
}



.user-name{
    font-size: 1.5rem;
    margin-left: 5px;
    /*margin-bottom: 0.0rem;*/
}

.user-list{
    flex-grow: 2;
    margin: auto 0;
    min-width: 5rem;
}

#users .user-name i{
    vertical-align: 0.4em;
}
```

Users


TU WhatsChat!

Channels

Neuer Channel...

Topic...

channel 1
no empty strings pls

Frantzuuu
Names

AWS
Präsentation

yyy
sss

s
x

44
44

channel 7
no empty strings pls

channel 8
no empty strings pls

22
22

g
g

jsidka
hjhhjh

w
1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

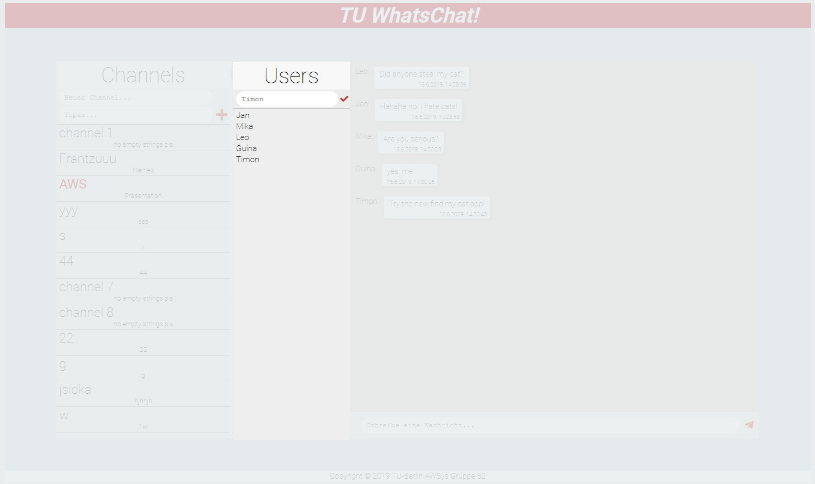
Schreibe eine Nachricht...

Copyright © 2019 TU-Berlin AWSys Gruppe 52

JS

- All available Users in the current channel will be loaded from server by **_getUserList()** upon channel join
- **setInterval**(_updateUserList, g_iRefreshUsers) is used for background polls every 10s (g_iRefreshUsers)
- **_updateUserList()** fetches the list of online users from the server for the current channel (if a channel has been joined)
- The users menu allows one to set the user name for the current channel. This is done by adding the preferred name in the input field and clicking the check button
- This triggers the **on click event** which reads the input, changes the internal value, and sets the placeholder attribute to the input value
- If no user name is set **anonymous** is used as **default name**
- The **_getUserList(iChannel)** function clears the user list pulls the current users of a given channel from the server using an ajax call
- For every user in the ajax response the **_addUserToScreen(sUsername)** function will be called to display the name
- The **_addUserToScreen(sUsername)** function than adds the content in an html wrapper to the list (see JS)

Users



TU WhatsChat!

Channels

Neuer Channel...
Topic...

channel 1
no empty strings pls
Frantzuuu
Names
AWS
Präsentation
yyy
sss
s
x
44
44
channel 7
no empty strings pls
channel 8
no empty strings pls
22
22
g
g
jsidka
hjhhjh
w
1w

Users

Timon

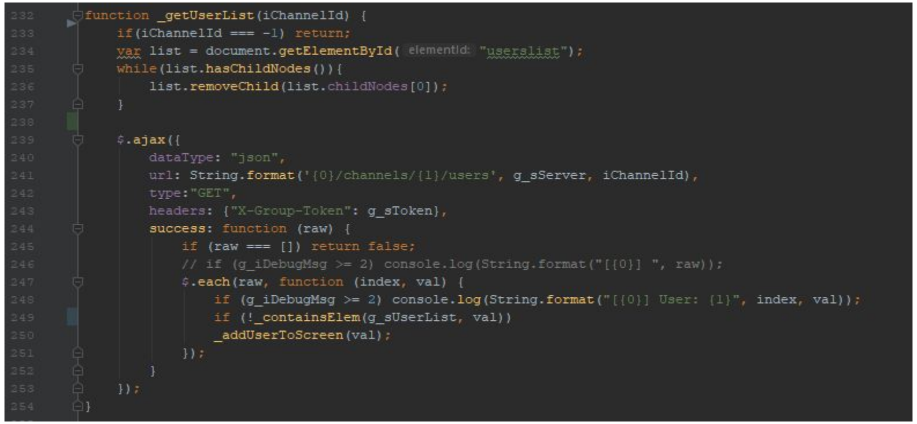
Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39
Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53
Mika: Are you serious? 16.6.2019, 14:30:23
Guina: yes, me 16.6.2019, 14:30:26
Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Schreibe eine Nachricht...

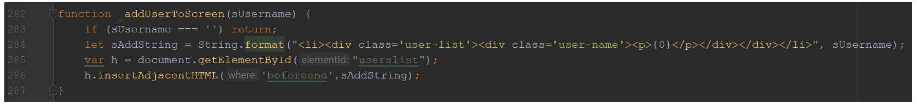
Copyright © 2019 TU-Berlin AWSys Gruppe 52

_getUserlist()



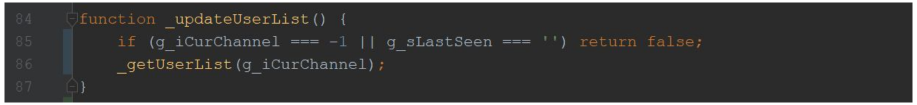
```
function _getUserList(iChannelId) {
    if(iChannelId === -1) return;
    var list = document.getElementById( elementId: "userslist");
    while(list.hasChildNodes()){
        list.removeChild(list.childNodes[0]);
    }

    $.ajax({
        dataType: "json",
        url: String.format('{0}/channels/{1}/users', g_sServer, iChannelId),
        type:"GET",
        headers: {"X-Group-Token": g_sToken},
        success: function (raw) {
            if (raw === []) return false;
            // if (g_iDebugMsg >= 2) console.log(String.format("[{0}] ", raw));
            $.each(raw, function (index, val) {
                if (g_iDebugMsg >= 2) console.log(String.format("[{0}] User: {1}", index, val));
                if (!_containsElem(g_sUserList, val))
                _addUserToScreen(val);
            });
        }
    });
}
```

_addUsersToScreen()



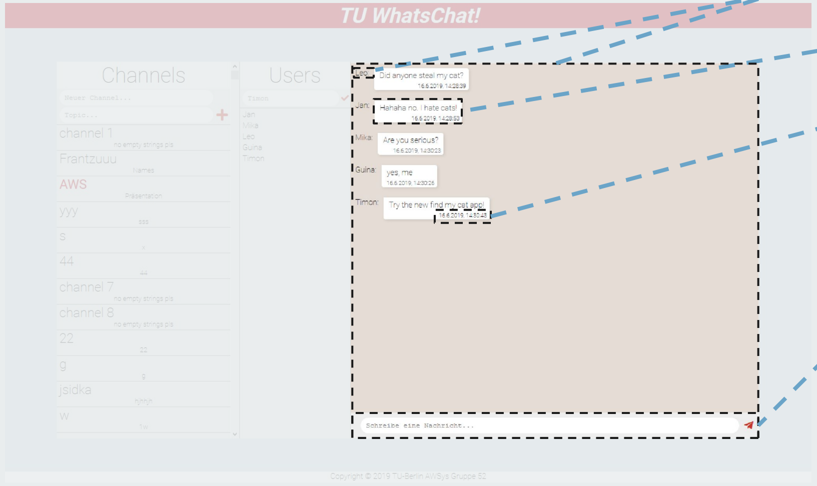
```
function _addUserToScreen(sUsername) {
    if (sUsername === '') return;
    let sAddString = String.format("<li><div class='user-list'><div class='user-name'><p>{0}</p></div></div></li>", sUsername);
    var h = document.getElementById( elementId: "userslist");
    h.insertAdjacentHTML( where: 'beforeend',sAddString);
}
```

_updateUserList()



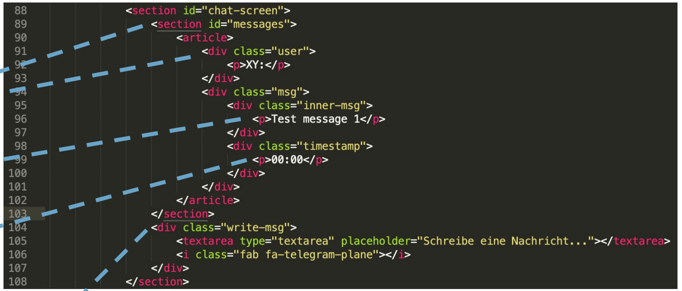
```
function _updateUserList() {
    if (g_iCurChannel === -1 || g_sLastSeen === '') return false;
    _getUserList(g_iCurChannel);
}
```

Messages



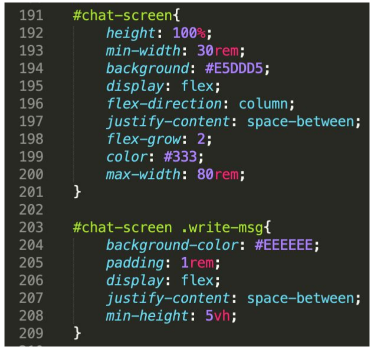
Leo: Did anyone steal my cat? 16.6.2019, 14:28:39
Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53
Mika: Are you serious? 16.6.2019, 14:30:23
Guina: yes, me 16.6.2019, 14:30:26
Timon: Try the new find my cat app! 16.6.2019, 14:30:43

HTML

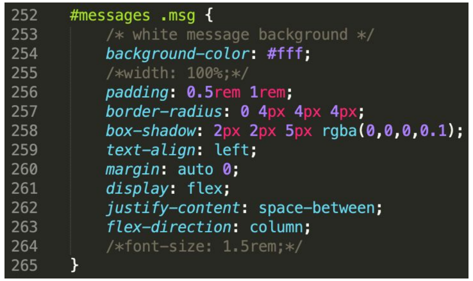


```html
<section id="chat-screen">
    <section id="messages">
        <article>
            <div class="user">
                <p>XY:</p>
            </div>
            <div class="msg">
                <div class="inner-msg">
                    <p>Test message 1</p>
                </div>
                <div class="timestamp">
                    <p>00:00</p>
                </div>
            </div>
        </article>
    </section>
    <div class="write-msg">
        <textarea type="textarea" placeholder="Schreibe eine Nachricht..."></textarea>
        <i class="fab fa-telegram-plane"></i>
    </div>
</section>
```

CSS



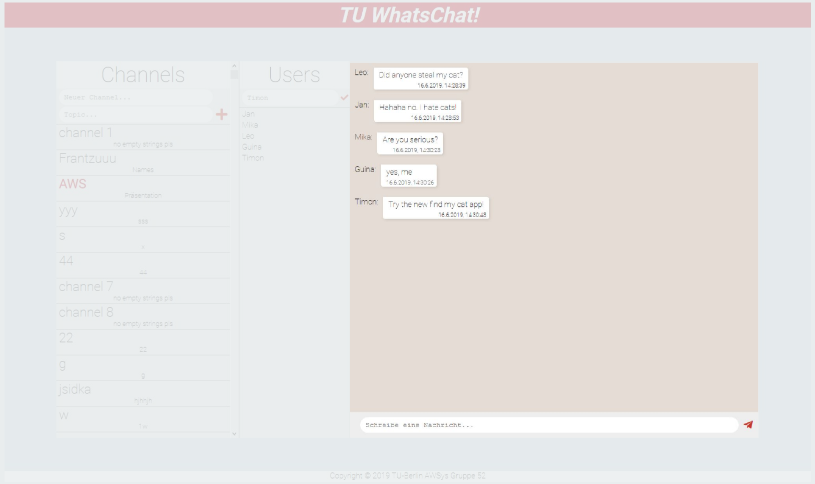
```css
#chat-screen{
    height: 100%;
    min-width: 30rem;
    background: #E5DDD5;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    flex-grow: 2;
    color: #333;
    max-width: 80rem;
}

#chat-screen .write-msg{
    background-color: #EEEEEE;
    padding: 1rem;
    display: flex;
    justify-content: space-between;
    min-height: 5vh;
}
```



```css
#messages .msg {
    /* white message background */
    background-color: #fff;
    /*width: 100%;*/
    padding: 0.5rem 1rem;
    border-radius: 0 4px 4px 4px;
    box-shadow: 2px 2px 5px rgba(0,0,0,0.1);
    text-align: left;
    margin: auto 0;
    display: flex;
    justify-content: space-between;
    flex-direction: column;
    /*font-size: 1.5rem;*/
}
```

- Not all CSS is displayed here, for more, see style.css

Messages



Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

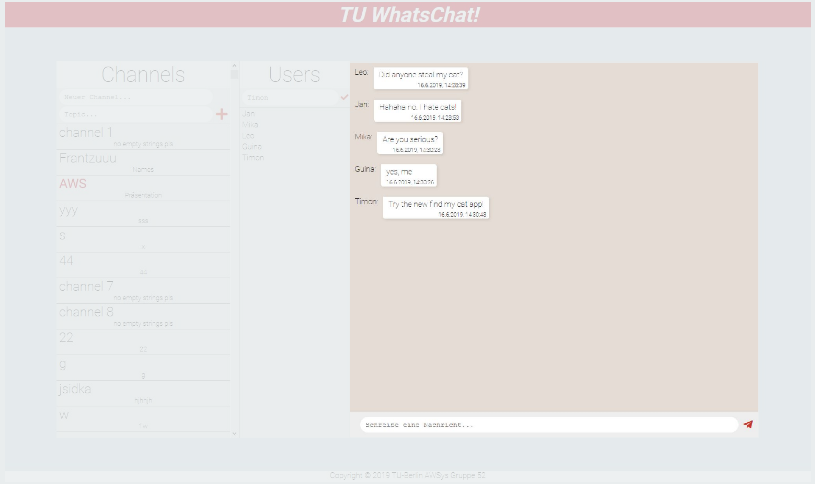
Schreibe eine Nachricht...

Copyright © 2019 TU-Berlin AWSys Gruppe 52

JS

- All available Messages in the current channel will be loaded, when joined, from server by **_getMessages(iCurChannel, sLastSeen)**
- **setInterval**(_updateMessageList, g_iRefreshMsgs) is used for background poll every 1s (see g_iRefreshMsgs)
- This triggers the **_updateMessageList()** to check if user has joined any channel yet - if this is case the case the **_getMessages()** function will be called with the '?lastSeenTimestamp=' parameter of the last message and the encoded-to-url timestamp
- The benefit is, that only the last messages will have to be loaded and the chat screen is continuous
- To send messages users can use the textarea at the bottom of the chat window - either hitting enter or clicking the paperplane-telegram button. That will trigger the **_inputHelper_ReadTextarea()** function
- This function checks for a valid entry and triggers the **_sendMsg(sMsg, sSender)** function and resets the input field
- The **_sendMsg()** function will then perform a POST ajax call to the server and will update the message list and user list, if successful
- This is done to have a smooth chat experience

Messages



Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

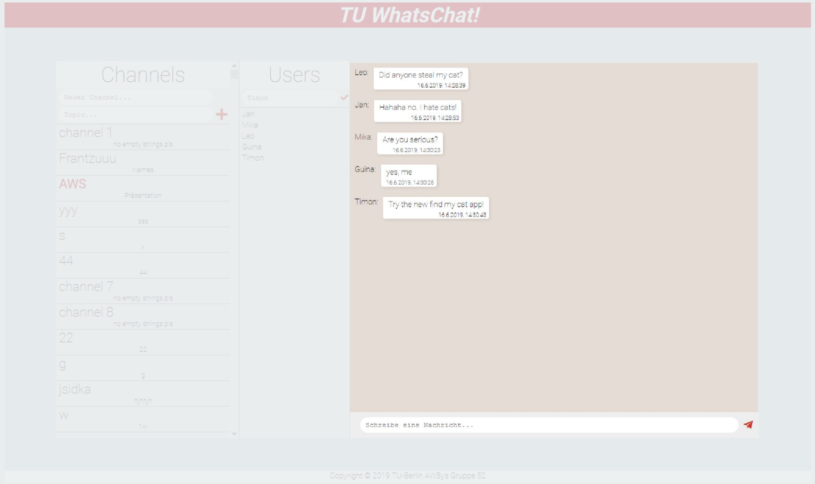
Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Copyright © 2019 TU-Berlin AWSys Gruppe 52

JS

- The **_getMessages()** will trigger the **_addMsgToScreen(sMsg, sSender, timestamp = new Date(), bReverse=false)** function.
- This function checks if the input is valid and puts it into an html wrapper, that will be added to the unordered list from the html file.
- It also checks if the function was called with the ?lastSeenTimestamp=' - in this case it adds the messages in reverse to show the messages in the correct order.
- The bReverse parameter changes the order in which the messages will be added to the screen. This is useful when parsing the messages from the server since in that case the newest messages will be the first ones in the object - opposed to the standard case where only one new message is appended to the end of the screen.

Messages



Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

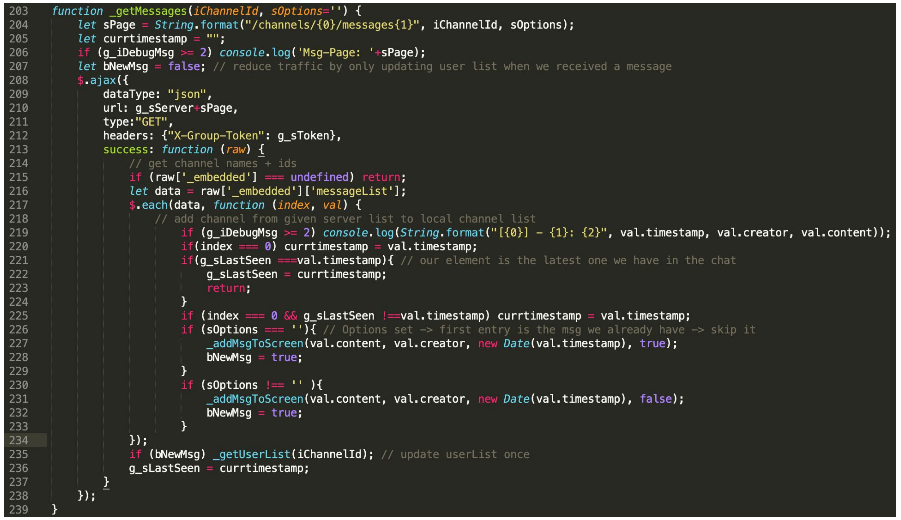
Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

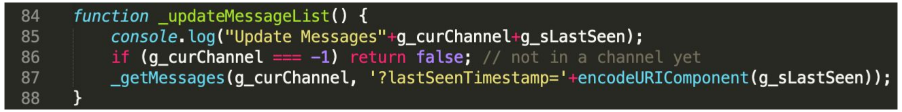
Copyright © 2019 TU-Berlin AWSys Gruppe 52

_getMessages()



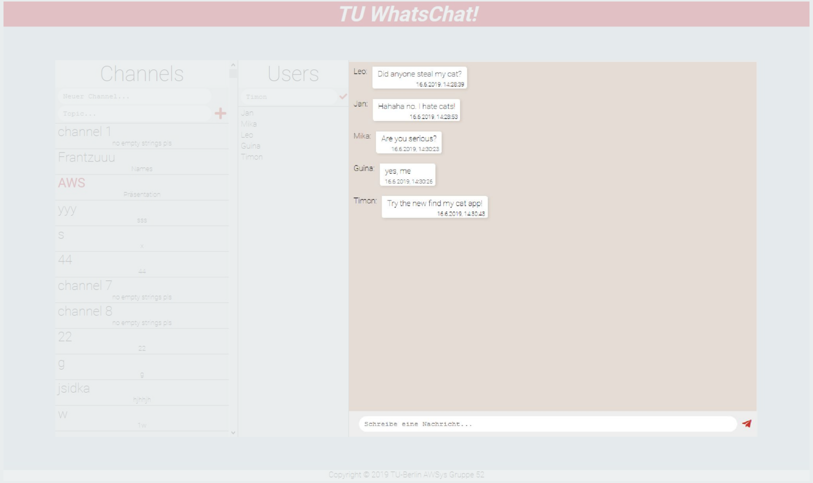
```
function _getMessages(iChannelId, sOptions='') {
    let sPage = String.format("/channels/{0}/messages{1}", iChannelId, sOptions);
    let currtimestamp = "";
    if (g_iDebugMsg >= 2) console.log('Msg-Page: '+sPage);
    let bNewMsg = false; // reduce traffic by only updating user list when we received a message
    $.ajax({
        dataType: "json",
        url: g_sServer+sPage,
        type:"GET",
        headers: {"X-Group-Token": g_sToken},
        success: function (raw) {
            // get channel names + ids
            if (raw['_embedded'] === undefined) return;
            let data = raw['_embedded']['messageList'];
            $.each(data, function (index, val) {
                // add channel from given server list to local channel list
                    if (g_iDebugMsg >= 2) console.log(String.format("[{0}] - {1}: {2}", val.timestamp, val.creator, val.content));
                    if(index === 0) currtimestamp = val.timestamp;
                    if(g_sLastSeen ===val.timestamp){ // our element is the latest one we have in the chat
                        g_sLastSeen = currtimestamp;
                        return;
                    }
                    if (index === 0 && g_sLastSeen !==val.timestamp) currtimestamp = val.timestamp;
                    if (sOptions === ''){ // Options set -> first entry is the msg we already have -> skip it
                        _addMsgToScreen(val.content, val.creator, new Date(val.timestamp), true);
                        bNewMsg = true;
                    }
                    if (sOptions !== '' ){
                        _addMsgToScreen(val.content, val.creator, new Date(val.timestamp), false);
                        bNewMsg = true;
                    }
            });
            if (bNewMsg) _getUserList(iChannelId); // update userList once
            g_sLastSeen = currtimestamp;
        }
    });
}
```

_updateMessageList()



```
function _updateMessageList() {
    console.log("Update Messages"+g_curChannel+g_sLastSeen);
    if (g_curChannel === -1) return false; // not in a channel yet
    _getMessages(g_curChannel, '?lastSeenTimestamp='+encodeURIComponent(g_sLastSeen));
}
```

Messages



channel 1
no empty strings pls
Frantzuuu
Names
AWS
Präsentation
yyy
sss
s
x
44
44
channel 7
no empty strings pls
channel 8
no empty strings pls
22
22
g
g
jsidka
hjhhjh
w
1w

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

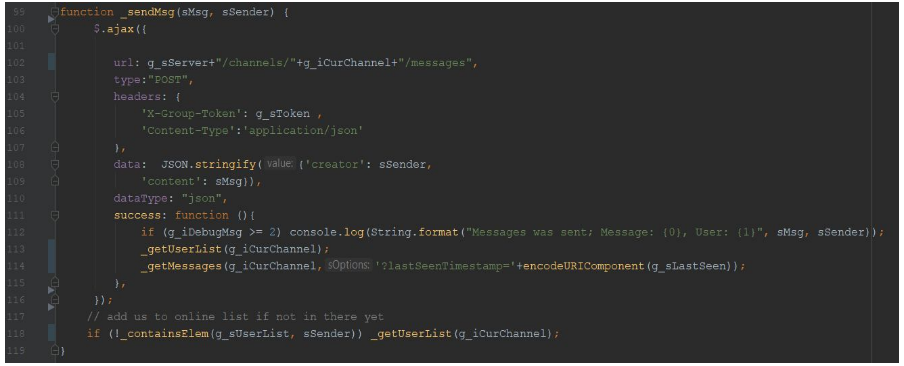
Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

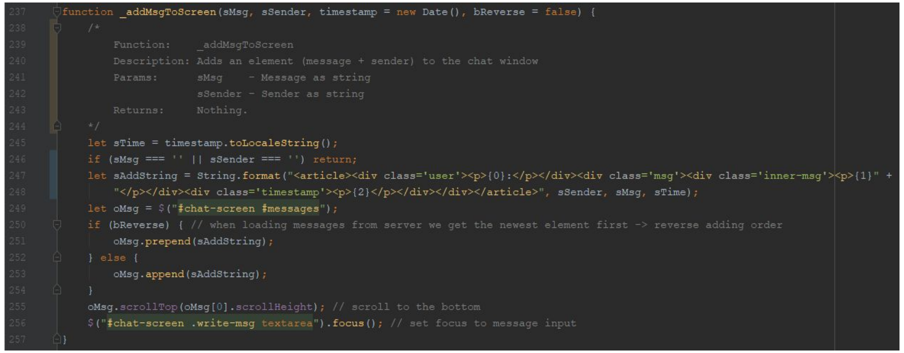
Timon: Try the new find my cat app! 16.6.2019, 14:30:43

_sendMsg(sMsg, sSender)



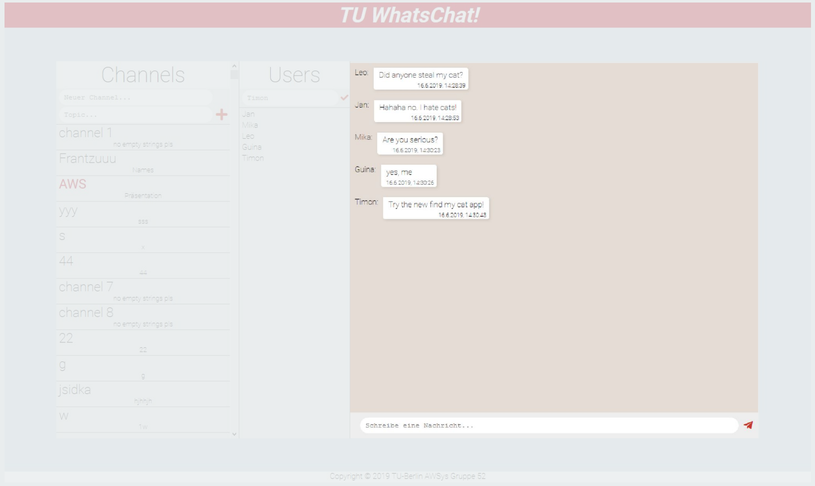
```
function _sendMsg(sMsg, sSender) {
    $.ajax({

        url: g_sServer+"/channels/"+g_iCurChannel+"/messages",
        type:"POST",
        headers: {
            'X-Group-Token': g_sToken ,
            'Content-Type':'application/json'
        },
        data:  JSON.stringify( value: {'creator': sSender,
            'content': sMsg}),
        dataType: "json",
        success: function (){
            if (g_iDebugMsg >= 2) console.log(String.format("Messages was sent; Message: {0}, User: {1}", sMsg, sSender));
            _getUserList(g_iCurChannel);
            _getMessages(g_iCurChannel, sOptions: '?lastSeenTimestamp='+encodeURIComponent(g_sLastSeen));
        },
     });
    // add us to online list if not in there yet
    if (!_containsElem(g_sUserList, sSender)) _getUserList(g_iCurChannel);
}
```

_addMsgToScreen(sMsg, sSender...)



```
function _addMsgToScreen(sMsg, sSender, timestamp = new Date(), bReverse = false) {
    /*
        Function:    _addMsgToScreen
        Description: Adds an element (message + sender) to the chat window
        Params:      sMsg    - Message as string
                     sSender - Sender as string
        Returns:     Nothing.
    */
    let sTime = timestamp.toLocaleString();
    if (sMsg === '' || sSender === '') return;
    let sAddString = String.format("<article><div class='user'><p>{0}:</p></div><div class='msg'><div class='inner-msg'><p>{1}" +
        "</p></div><div class='timestamp'><p>{2}</p></div></div></article>", sSender, sMsg, sTime);
    let oMsg = $("#chat-screen #messages");
    if (bReverse) { // when loading messages from server we get the newest element first -> reverse adding order
        oMsg.prepend(sAddString);
    } else {
        oMsg.append(sAddString);
    }
    oMsg.scrollTop(oMsg[0].scrollHeight); // scroll to the bottom
    $("#chat-screen .write-msg textarea").focus(); // set focus to message input
}
```

Messages


TU WhatsChat!

Channels

Neuer Channel...
Topic...

channel 1
no empty strings pls
Frantzuuu
Names
AWS
Präsentation
yyy
sss
s
x
44
44
channel 7
no empty strings pls
channel 8
no empty strings pls
22
22
g
g
jsidka
hjhhjh
w
1w

Users

Timon

Jan
Mika
Leo
Guina
Timon

Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

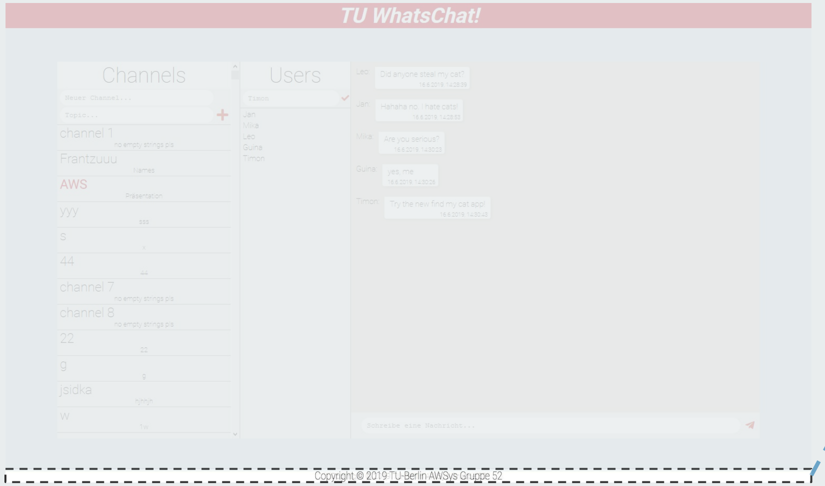
Schreibe eine Nachricht...

Copyright © 2019 TU-Berlin AWSys Gruppe 52

_inputHelper_ReadTextarea()

```
function _inputHelper_ReadTextarea() {
    let o = $("#chat-screen .write-msg textarea");
    let sMsg = o.val(); // read msg from input field
    if (sMsg === '') return;
    _sendMsg(sMsg, g_sUsername);
    o.val(''); // clear msg field
}
```

Footer



Leo: Did anyone steal my cat? 16.6.2019, 14:28:39

Jan: Hahaha no. I hate cats! 16.6.2019, 14:28:53

Mika: Are you serious? 16.6.2019, 14:30:23

Guina: yes, me 16.6.2019, 14:30:26

Timon: Try the new find my cat app! 16.6.2019, 14:30:43

Copyright © 2019 TU-Berlin AWSys Gruppe 52

HTML

```html
<div id="footer">
    Copyright &copy; 2019 TU-Berlin AWSys Gruppe 52
</div>
```

CSS

```css
#footer{
    /* copyright footer */
    font-size: 1.5rem;
    text-align: center;
}
```

Distribution of tasks

- **Timon**: *HTML, CSS, JS, Presentation*
- **Leonard**: *HTML, JS, Presentation*
- **Jan**: *HTML, JS, Presentation*
- **Mika**: *CSS, JS, Presentation*

“Over and out”

- Michael Jackson

Content and Design by: Leonard, Timon