

chat.js - код для работы с чатом в комнате

- 1. Обертка для кода
- 2. Локальные переменные
- 3. Подписка на события комнаты
 - SFU_ROOM_EVENT.MESSAGE
 - SFU_ROOM_EVENT.JOINED
 - SFU_ROOM_EVENT.LEFT
- 4. Send message to other participants
- 5. Локальное отображение сообщений чата

Данный модуль содержит код для работы с чатом в комнате

1. Обертка для кода

createChat() [code](#)

Функция-обертка для вызова из основной логики, ограничивает область видимости

```
const createChat = function(room, messages, input, sendButton) {
```

2. Локальные переменные

[code](#)

Объявление локальных переменных: константы и цвета

```
const constants = SFU.constants;
const chatSelfColour = "green";
const chatTextColour = "black";
const chatOtherColour = "red";
const chatEventColour = "navy";
```

3. Подписка на события комнаты

[code](#)

Подписка на события комнаты, имеющие отношение к функциям чата

```
room.on(constants.SFU_ROOM_EVENT.MESSAGE, function(e) {
    appendMessage(e, chatOtherColour, chatTextColour);
}).on(constants.SFU_ROOM_EVENT.JOINED, function(e) {
    appendMessage({
        nickName: e.name,
        message: e.type
    }, chatOtherColour, chatEventColour);
}).on(constants.SFU_ROOM_EVENT.LEFT, function(e) {
    appendMessage({
        nickName: e.name,
        message: e.type
    }, chatOtherColour, chatEventColour);
});
```

SFU_ROOM_EVENT.MESSAGE

Подписка на событие SFU_ROOM_EVENT.MESSAGE. При получении события, сообщение отображается в локальном чате

[code](#)

```
.on(constants.SFU_ROOM_EVENT.MESSAGE, function(e) {
  appendMessage({
    nickName: getNickName(e.message),
    message: getMessage(e.message)
  }, chatOtherColour, chatTextColour);
})
```

SFU_ROOM_EVENT.JOINED

Подписка на событие SFU_ROOM_EVENT.JOINED. При получении, сообщение о входе участника отображается в локальном чате

[code](#)

```
.on(constants.SFU_ROOM_EVENT.JOINED, function(e) {
  appendMessage({
    nickName: e.name,
    message: e.type
  }, chatOtherColour, chatEventColour);
})
```

SFU_ROOM_EVENT.LEFT

Подписка на событие SFU_ROOM_EVENT.LEFT. При получении, сообщение о выходе участника отображается в локальном чате

[code](#)

```
.on(constants.SFU_ROOM_EVENT.LEFT, function(e) {
  appendMessage({
    nickName: e.name,
    message: e.type
  }, chatOtherColour, chatEventColour);
});
```

4. Send message to other participants

`sendMessage()` [code](#)

Функция `sendMessage` реализует разбор пользовательского ввода, отправку сообщения серверу и отображение сообщения в локальном чате

Отметим, что сообщения отправляются по WebRTC data channels

```
const sendMessage = function() {
  let message = input.value;
  input.value = "";
  room.sendMessage(message);
  appendMessage({
    nickName: nickName.value,
    message: message
  }, chatSelfColour, chatTextColour);
}
```

Привязка кнопки Send к функции `sendMessage`

[code](#)

```
sendButton.addEventListener("click", sendMessage);
```

Отправка сообщения по нажатию Enter

[code](#)

```
input.onkeyup = function(e) {
  if (e.keyCode === 13) {
    if (e.shiftKey) {

    } else {
      sendMessage();
    }
    return false;
  }
}
```

5. Локальное отображение сообщений чата

Функция реализует форматирование и отображение сообщений в локальном чате

appendMessage() [code](#)

```
const appendMessage = function(msg, nickColour, msgColour) {
  let message = document.createElement('div');
  message.setAttribute("class", "message");
  messages.appendChild(message);
  let nickDiv = document.createElement('div');
  nickDiv.style.color = nickColour;
  nickDiv.innerText = getChatTimestamp() + " " + msg.nickName + ":";
  message.appendChild(nickDiv);
  let msgDiv = document.createElement('div');
  msgDiv.style.color = msgColour;
  msgDiv.innerText = msg.message;
  message.appendChild(msgDiv);
  scrollToBottom();
}
```

Вспомогательная функция для прокрутки списка сообщений вниз

scrollToBottom() [code](#)

```
const scrollToBottom = function() {
  messages.scrollTop = messages.scrollHeight;
}
```

Вспомогательная функция для вывода времени

getTimeStamp() [code](#)

```
const getChatTimestamp = function() {
  let currentdate = new Date();
  return currentdate.getHours() + ":" + currentdate.getMinutes() + ":" + currentdate.getSeconds();
}
```