

Реализация консольного приложения для чата в реальном времени

Ервлева Регина Викторовна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Ервлев Павел Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье будут представлены пример реализации консольного приложения для чата в реальном времени. Работа будет происходить с помощью языка программирования JavaScript.

Ключевые слова: JavaScript, Node, Чат

Implementing a Console Application for Chat in the Atmosphere of Time

Eroleva Regina Viktorovna

Sholom-Aleichem Priamursky State University

Student

Erolev Pavel Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article will present an example implementation of a real-time chat console application. The work will be done using the JavaScript programming language.

Keywords: JavaScript, Node, Chat

Сейчас все большую популярность набирают мессенджеры и чаты для общения в сети, появляются большое количество разнообразных чатов в сети. Так же для возможности использования на работе чата между всей структурой организации имеется возможность использовать сетевые чаты, которые работают только в определенной сети. Это гораздо безопаснее, ведь данные не передаются дальше внутренней сети организации.

Цель работы – создать сетевой консольный чат, состоящий из сервера и клиента.

Исследованиями в данной теме занимались следующие авторы.

В своей работе С.А. Беляева рассматривает ключевые вопросы разработки однопользовательских двумерных игр на языке JavaScript [1]. А.В.Диков исследовал объект Math языка JavaScript, который обладает

множеством математических свойств и методов. Таким образом открывается возможность использования языка веб программирования для решения математических задач [2]. Т.В. Зудилова, М. Л. Буркова в своем пособии рассматривают необходимые для создания игры объекты `math` и `document`, а также рассказывают о способе создания и вызова функций [3]. В своей работе Н. Прохоренок и В. Дронов рассмотрели вопросы создания интерактивных Web-сайтов с помощью HTML, JavaScript [4]. При создании игры были изучены и прочитаны следующие источники. В учебном пособии В. В. Захаркина рассмотрены основы применения алгоритмического языка JavaScript при создании интерактивных веб-документов [5].

В данном примере сервер и клиент будут реализованы в одном и том же проекте. Сначала создадим каталог с именем «chat-cli» и инициализируем проект «npm» с расширением «npm init -y».

Теперь установим необходимые модули «socket.io» и «socket.io-client» с «npm install socket.io socket.io-client». Создадим два файла в проекте «index.js» который будет содержать код клиента и «server.js» который будет содержать код сервера (рис.1).

```
D:\Projects\NodeJs\chat_client>npm install socket.io socket.io-client
npm WARN chat_client@1.0.0 No description
npm WARN chat_client@1.0.0 No repository field.

+ socket.io-client@4.1.2
+ socket.io@4.1.2
updated 2 packages and audited 30 packages in 2.78s
found 0 vulnerabilities

D:\Projects\NodeJs\chat_client>
```

Рисунок 1 – Установка модулей

Необходим сервер «socket.io» для обработки всех событий и отправки их клиенту (рис.2).

```

const io = require("socket.io") ();

const PORT = process.env.PORT || 3000;

const users = {};

io.on("connection", (socket) => {
  console.log("New Connection: " + socket.id);
  socket.on('new user', (name) => {
    users[socket.id] = name;
    socket.broadcast.emit("message", `${name} joined the chat.`)
  });

  socket.on('message', (text) => {
    socket.broadcast.emit("message", `${users[socket.id]}> ${text}`);
  });
});

io.listen(PORT);

```

Рисунок 2 – Реализация Server.js

Теперь нужно создать клиента и позволить ему подключиться к серверу (рис.3).

```

const io = require('socket.io-client');
const socket = io("http://localhost:3000");

const readline = require('readline');

const rl = readline.createInterface({
  input: process.stdin,
});

socket.on("message", (text) => {
  process.stdout.write("\r\n\b[K]");
  console.log(text);
  process.stdout.write("> ");
});

console.log("What is your name?");
rl.question("What is your name?", (text) => {
  socket.emit('new user', text.trim());
  console.log("You joined the chat");
  process.stdout.write("> ");
});

rl.prompt();

rl.on('line', (text) => {
  socket.emit('message', text.trim());
  process.stdout.write("> ");
  rl.prompt(true);
});

```

Рисунок 3 – Реализация клиентской части

Теперь можно запускать проект. Сначала откроем три терминала или cmd в каталоге проекта. Первым запустим сервер, набрав node server.js. После запустим двух клиентов, с которыми будем общаться, набрав «node index.js» два раза в терминале или cmd (рис.4-6).

```
D:\Projects\NodeJs\chat_client>node server.js
New Connection: mGD4AXUSK4ipONLGAAAB
New Connection: rw1gsesAobfyBtZkAAAD
|
```

Рисунок 4 – Запуск сервера

```
D:\Projects\NodeJs\chat_client>node index
What is your name?
Programmer101N joined the chat.
> Prgmaz
You joined the chat
> Hello World!
Programmer101N> Wassup?
> |
```

Рисунок 5 – Запуск первого клиента

```
D:\Projects\NodeJs\chat_client>node index
What is your name?
Programmer101N
You joined the chat
Prgmaz joined the chat.
Prgmaz> Hello World!
> Wassup?
> |
```

Рисунок 6 – Запуск второго клиента

В данной статье был рассмотрен пример реализации консольного сетевого чата в реальном времени.

Библиографический список

1. Беляев С.А. Разработка игр на языке JavaScript: Учебное пособие. СПб.: Лань, 2016. 128с.
2. Диков А.В. Математические алгоритмы на JavaScript // Известия Пензенского государственного педагогического университета им. В.Г. Белинского. 2009. № 17. С. 84-88.
3. Зудилова Т.В., Буркова М.Л. Web-программирование JavaScript. // СПб: НИУ ИТМО, 2012. 68 с
4. Прохоренок Н., Дронов В. HTML, JavaScript, PHP и MySQL. // СПб.: БХВПетербург, 2015. 766с.
5. Захаркина В. В. JavaScript. Основы клиентского программирования: Учебное пособие. СПб.: Ф-т филологии и искусств СПбГУ, 2007. 73 с.