

Chat Applikation

Installation

1. Github Repo klonen

```
git clone --branch master https://github.com/prashanttsachan/realtime-chat-application-using-nodejs-expressjs-and-web-socket.git
```

2. In Verzeichnis gehen

```
cd realtime-chat-application-using-nodejs-expressjs-and-web-socket
```

3. Installieren

```
npm install
```

4. Datenbank erstellen

```
CREATE DATABASE chat;
```

5. Benutzer erstellen, welcher alle Rechte auf der **chat** DB hat

```
CREATE USER 'chat'@'localhost' IDENTIFIED BY 'chatPassword1234!';  
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP ON chat.* TO  
'chat'@'localhost';
```

6. User verändern, dass er sich mit dem alten Mysql-Plugin anmelden kann

```
ALTER USER 'chat'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'chatPassword1234!';
```

7. Datenbank Konfig File öffnen

```
sudo nano realtime-chat-application-using-nodejs-expressjs-and-web-socket/database/db.js
```

8. Datenbankinfos anpassen

```
GNU nano 7.2 db.js *
/**
 * Created by Prashant on 14-07-2019.
 * https://github.com/psachan190
 * https://www.linkedin.com/in/psachan190
 */

var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "chat",
  password: "chatPassword1234!",
  database: "chat"
});
con.connect(function (err) {
  if (err) throw err;
  var sql = "CREATE TABLE IF NOT EXISTS message (id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, message VARCHAR(2550) )";
  con.query(sql, function (err, result) {
    if (err) throw err;
  });
  var sql = "CREATE TABLE IF NOT EXISTS login (id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY, username VARCHAR(250) , password VARCHAR(250) )";
  con.query(sql, function (err, result) {
    if (err) throw err;
  });
});
module.exports = con;
```

9. Applikation starten

```
cd realtime-chat-application-using-nodejs-expressjs-and-web-socket
node server.js
```

10. Logins in der DB erstellen

```
INSERT INTO login VALUES ('1','user1','user1');
INSERT INTO login VALUES ('2','user2','user2');
```

Testing

1. Zweiten Browser installieren

```
sudo apt install chromium
```

2. Seite öffnen

```
http://localhost:3000/
```

3. Beim einem Browser mit User 1 und beim anderen mit User 2 anmelden

Jetzt kann man Nachrichten an den andere schicken

Stored Procedure

Folgendes SQL sollte via Stored Procedure ausgeführt werden können.

```
// ===== Send and Save Messages=====
socket.on('send-message', function (data, user) {
    //console.log(user);
    var sql = "INSERT INTO message (message , user) VALUES ('" + data+ "' , '"+user+"')";
    con.query(sql, function (err, result) {
        if (err) throw err;
        //console.log("1 record inserted");
    });
    io.sockets.emit('new-message', {msg: data , username : user});
})
})
}
chat_start();
```

```
DELIMITER //
CREATE PROCEDURE save_message
(
    IN data varchar(500),
    IN user varchar(250)
)
BEGIN
    INSERT INTO message
    (
        message,
        user
    ) VALUES
    (
        data,
        user
    );
END//
DELIMITER ;
```

Chat-User auf die Stored Procedure berechtigen

```
GRANT EXECUTE ON PROCEDURE save_message TO 'chat'@'localhost';
FLUSH PRIVILEGES;
```

Aufruf:

```
mysql> CALL save_message ('TestEnea', 'Enea');
Query OK, 1 row affected (0.09 sec)

mysql> show tables;
+-----+
| Tables_in_chat |
+-----+
| login          |
| message        |
+-----+
2 rows in set (0.00 sec)

mysql> select * from message;
+----+-----+-----+
| id | message | user |
+----+-----+-----+
| 1  | hellou  | user2 |
| 2  | salü    | user1 |
| 3  | TestEnea | Enea  |
+----+-----+-----+
3 rows in set (0.07 sec)
```

Angepasstes Server.js:

```
// ===== Send and Save Messages=====
socket.on('send-message', function (data, user) {
    //console.log(user);
    var sql = "CALL save_message ('" + data + "' , '"+user+"')";
    con.query(sql, function (err, result) {
        if (err) throw err;
        //console.log("1 record inserted");
    });
    io.sockets.emit('new-message', {msg: data , username : user});
})
})
}
chat_start();
```

Views

Folgendes SQL sollte via View ausgeführt werden können.

```
// =====initilize data and show=====
socket.on('initial-messages', function (data) {
  var sql = "SELECT * FROM message ";
  con.query(sql, function (err, result, fields) {
    var jsonMessages = JSON.stringify(result);
    // console.log(jsonMessages);
    io.sockets.emit('initial-message', {msg: jsonMessages});
  });
});
socket.on('username', function (data) {
  socket.emit('username', {username: username});
  //io.sockets.emit('username', {username: username});
});
```

```
CREATE VIEW get_message AS
SELECT *
FROM message;
```

Angepasstes Server.js:

```
// =====initilize data and show=====
socket.on('initial-messages', function (data) {
  var sql = "SELECT * FROM get_message ";
  con.query(sql, function (err, result, fields) {
    var jsonMessages = JSON.stringify(result);
    // console.log(jsonMessages);
    io.sockets.emit('initial-message', {msg: jsonMessages});
  });
});
socket.on('username', function (data) {
  socket.emit('username', {username: username});
  //io.sockets.emit('username', {username: username});
});
```