

Spezifikation des Anwendungs-Protokolls  
für den Client-Server-Chat RNP02 (github.com/rep0sit)

Autoren:  
Nelli Welker  
Etienne Onasch

Simple Chat Server Communication Protocoll (SCSCP)

1. Einführung	1
2. Befehle	1
3. Ablauf der Kommunikation	2
4. User Befehle	3

**1. Einführung:**

Ziel des SCSCP die Kommunikation zwischen einem Client und einem Server in einem Chatsystem zu ermöglichen.

Das SCSCP ist komplett Textbasiert, weshalb auf Server Seite einige Voraussetzungen erfüllt sein müssen, damit z.B. Clients keine Server-Befehle senden können.

**2. Befehle:**

**Server Kommandos**

"<<300"

übermittelt dem Client den Befehl seinen Usernamen zu senden

**Server Kommandos die den Zustand des Clients ändern**

>>>>601\_<optionale Nachricht>

Zwingt den Client zum disconnect

**Positive Server Antworten**

>>100\_<optionale Nachricht>

Der Username des Clients ist gültig.

>>101\_<optionale Nachricht>

Der User wurde erfolgreich eingeloggt

**Negative Server Antworten**

>>000\_<optionale Nachricht>

Der Username ist ungültig

>>001\_<optionale Nachricht>

Der Username ist bereits in Verwendung

>>002\_<optionale Nachricht>

Der Server ist voll

### 3. Ablauf der Kommunikation

Alle gesendeten Nachrichten sind Textnachrichten und bestehen immer aus einer Zeile, die mit "\n" beendet werden.

Nach dem Verbindungsaufbau sendet der Server den Befehl:

"<<300", welcher den Client auffordert einen Benutzernamen zu übermitteln.

Die nächste vom Client gesendete Textnachricht wird automatisch als Nutzernamen interpretiert. Der Server kann nun folgende Kommandos an den Client senden:

---

**XOR:**

">>002\_<optionale Nachricht>" : Der Server ist voll. Verbindung zum Chat nicht möglich.

Der Client sollte damit reagieren sich selbst mit einer geeigneten User Nachricht zu beenden.

**XOR:**

">>100\_<optionale Nachricht>" : Der Username ist gültig und entspricht den Regularien des Servers, gefolgt von:

">101\_<optionale Nachricht>" : Der User wurde erfolgreich eingeloggt.

Beide Kommandos haben hier allerdings nur einen informativen Charakter und der Client kann je nach Implementierung diese Informationen an den User weiterleiten.

**XOR:**

">>001\_<optionale Nachricht>" : Der Username ist bereits in Verwendung.

**XOR:**

">>000\_<optionale Nachricht>" : Der Nutzernamen ist ungültig und passt nicht in die Regularien des Servers. Der Server sollte die Regularien in der optionalen Nachricht übermitteln.

(z.B. ">>000\_ Usernamen müssen mindestens 3 Zeichen lang sein")

---

Es ist zu empfehlen, dass der Server nachdem er eines der letzten beiden Kommandos (>>000\_, >001\_) übermittelt hat, erneut auf die Übermittlung eines Usernamen wartet.

Mittels einer Schleife sollte der Server diesen Vorgang mehrmals durchführen können.

Alternativ, oder nach erfolgloser Beendigung aller Schleifendurchgänge, sendet der Server den Befehl:

">>>>601\_<optionale Nachricht>",

welcher den Client veranlasst die Verbindung zum Server zu beenden.

Dieser Befehl sollte unabhängig davon vom Client jederzeit als eine Aufforderung zum Verbindungsabbruch interpretiert werden.

#### **4. User Kommandos**

Nachdem der User erfolgreich eingeloggt wurde, kann er folgenden Befehle an den Server senden. Dieser sollte darauf automatisch mit einer passenden Nachricht antworten.

`/commands`

listet alle möglichen User Befehle auf

`/rooms`

listet alle existierenden Chaträume auf

`/goto<leerzeichen><room>`

Gehe in den Chatraum <room>

`/users`

show a list of every user in the room

`/whisper<leerzeichen><username>`

gehe in den Flüstermodus mit User <username>

`/unwhisper`

deaktiviert den Flüstermodus. Nachrichten gehen an alle User des gleichen Chatrooms.

`/log`

zeigt das log aller den User betreffenden Nachrichten dieses Chatraumes.