

FAQ Simple-Social

LPR-A, LPR-B - 2016

Generale

- *Quando vengono chiuse le connessioni TCP?*
Tutte le connessioni TCP vengono chiuse al completamento di ogni funzionalità.
- *Quando si indica di usare TCP, si può usare RMI (che usa TCP).*
No, si devono usare socket TCP direttamente.
- *Cosa accade se l'esecuzione del server termina improvvisamente?*
Il server deve essere in grado di ricostruire, almeno parzialmente, lo stato della rete sociale (lista utenti, rete amici, rete follower, messaggi da inviare) , quando viene riavviato.

Registrazione/Login

- *Se l'utente è già registrato, SocialClient deve mostrare solo la funzionalità di login?*
All'avvio dell'applicazione vengono mostrate entrambe le funzionalità. La scelta se fare Login o Register spetta all'utente.
- *Cosa succede se lo stesso utente fa login due volte o da computer diversi?*
Non si deve gestire la situazione in cui un utente effettua il login da macchine distinte oppure quella in cui effettua il login più di una volta

Logout

- *Cosa accade se cliente chiude l'applicazione senza fare Logout o se SocialClient termina la sua esecuzione in modo improvviso?*
In questo caso, il Logout avviene automaticamente sul server (attraverso la funzionalità di Keep-Alive). La prossima volta che l'applicazione viene avviata, si mostrano le due opzioni: Login e Register

Richiesta amicizia

- *E' ridondante verificare che un cliente è online usando la seguente modalità: "SocialServer verifica se l'utente B è online, aprendo una connessione TCP.", visto che esiste già un meccanismo di keep-alive?*

Per verificare se B è online, non è sufficiente che il server scorra la sua lista di utenti online (aggiornata con il meccanismo di keep-alive), perché questa si aggiorna solo periodicamente, e la frequenza di aggiornamento potrebbe essere bassa. L'invio della richiesta, da parte del server, a SocialClient(B) assicura che B sia sicuramente online.

- *La richiesta di amicizia dell'utente A viene visualizzata immediatamente all'utente B?*
SocialClient(B) non mostra immediatamente la richiesta di amicizia a B, per facilitare l'uso di un'interfaccia testuale. B stesso deve richiedere di visualizzare le richieste (questo comportamento può però essere modificato nel caso si decida di utilizzare un'interfaccia grafica).

Conferma amicizia

- *Quali sono le azioni eseguite sulla lista delle richieste di amicizia?*
Un utente B sceglie un altro utente (diciamo A) per cui confermare/negare l'amicizia. SocialClient(B) invia la scelta dell'utente al server, il quale verifica se la richiesta di amicizia iniziale esiste nel sistema. Se la richiesta esiste, il server modifica la rete di amici, risponde a B con un messaggio di conferma e rimuove la richiesta di amicizia dalla lista sul server. Se la richiesta non esiste, risponde a B con un messaggio di errore
- *Cosa accade se un'utente A lascia l'applicazione senza visualizzare la lista di richieste di amicizia?*
Se A fa logout senza visualizzare la lista di richieste, la lista viene salvata localmente sul disco e ricaricata quando l'utente torna e richiede la visualizzazione (se l'utente effettua login su un altro computer, non può vedere le richieste pendenti). **Anche le richieste visualizzate ma non confermate/respinte vengono salvate.**

Keep-alive

- *La risposta al messaggio spedito dal server sul multicast si può inviare su TCP?*
I clienti devono rispondere al server con un datagram, quindi non su TCP e non su multicast.
- *Non è sufficiente che i clienti inviino un datagram ogni 10 secondi?*
Sarebbe una possibilità (anche più efficiente), ma per esigenze didattiche, richiediamo di utilizzare il multicast comunque.

Contenuti: pubblicazione e follower

- *Possiamo registrare la callback usando TCP?*
La registrazione della callback deve essere eseguita mediante l'uso di RMI. Il server RMI offre due funzionalità - due metodi remoti diversi: (1) registrazione callback e (2) registrazione dell'interesse per i contenuti di un amico. SocialClient registra la callback subito dopo il login. Al momento del logout del client la callback viene rimossa dal

server, e inviata di nuovo al prossimo login. La registrazione dell'interesse per un amico può avvenire in un qualsiasi momento. Per ogni utente il server memorizza i suoi follower.

- *Cosa accade con i contenuti generati quando un follower A non è online?*
Se A non è online, i contenuti vengono memorizzati sul server e inviati al SocialClient(A) dopo il successivo login, usando la nuova callback registrata dall'utente.
- *I contenuti generati vengono notificati subito ai followers?*
I contenuti generati non vengono mostrati subito ai followers, ma vengono richiesti esplicitamente per facilitare l'implementazione dell'interfaccia testuale (in caso si voglia utilizzare un' interfaccia grafica, i contenuti possono essere mostrati immediatamente).

Visualizzazione contenuti:

- *Cosa accade quando un utente richiede di vedere i contenuti già ricevuti (e memorizzati usando la callback)?*
Il SocialClient del richiedente visualizza i contenuti e li rimuove dalla lista di contenuti.
- *Cosa accade se un utente esce dall'applicazione senza visualizzare i contenuti dei suoi followers?*
Se l'utente esegue il *Logout* o chiude l'applicazione senza visualizzare i contenuti, la lista viene salvata sul disco e ricaricata quando l'utente ne richiede la visualizzazione

Sottomissione

- *Come si sottomette il progetto?*
Per il corso A si deve usare la procedura descritta nel documento del progetto.
Per il corso B la sottomissione è solo elettronica tramite il sito Moodle.

Gestione Token

- *Il client deve inviare il token di autenticazione fornitogli dal server insieme ad ogni richiesta che effettua, o è lui stesso a verificarne la validità e a forzarsi a effettuare nuovamente il login se scade? Non dovrebbe essere il server a verificarne la validità? O eventualmente entrambi?*
Sia il client che il server possono effettuare la procedura di verifica del token. Se il client si rende conto che il token non è più valido, può iniziare la procedura di login, questa soluzione consente di risparmiare un messaggio. Tuttavia il Server deve comunque controllare il token ricevuto. Si consideri, ad esempio, il seguente scenario: il server ha annullato il token di un client, perchè la risposta, da parte del client, alla sua richiesta di

keep-alive, è arrivata in ritardo. Tuttavia il client non si è scollegato dalla applicazione. La prossima volta che il client presenta il token al server, verifica la sua validità. Anche se il token risulta sempre valido per il client, esso è stato annullato dal server (vedi risposta successiva).

- *Il gruppo di keep-alive permette di mantenere traccia degli utenti online. Se un SocialClient precedentemente online non risponde ai messaggi di keep-alive e quindi viene messo offline, se vuole tornare ad effettuare richieste, deve necessariamente effettuare nuovamente il login o viene semplicemente rimesso online e la richiesta viene processata? (a patto che l'OAuth sia valido).*

Il Social Client deve effettuare nuovamente il login. Se tenta di richiedere un servizio senza effettuare il login, (ad esempio perchè la sua risposta al server è arrivata in ritardo, ma in realtà esso è sempre attivo), il Server gli invia un messaggio di tipo "Token non valido, occorre effettuare di nuovo il Login".

- *Deve essere previsto un comando specifico per il rinnovamento del token o questo si intende rinnovato al momento del login?*

Non ci esistono comandi specifici per rinnovare il token, esso viene rinnovato dal server al momento del login del client

Utilizzo Porte

- *Riguardo le diverse connessioni TCP corrispondenti ai vari servizi offerti dal server (login, logout, registrazione etc...), queste devono essere implementate ognuna su una porta differente o va bene anche utilizzare un'unica porta?*

Sono possibili entrambe le soluzioni

Varie

- *Se il client effettuare logout devo terminare anche tutti i thread relativi al client e poi riavviarli?*
Questo comportamento è alla discrezione dello sviluppatore.
- *Nel caso si crash del server, quando questo viene riavviato mantiene la lista degli iscritti ma perde quella degli utenti online e quindi gli utenti devono effettuare nuovamente il login?*
Sì, gli utenti devono rifare il login.