

WINSOME : REWARDING SOCIAL MEDIA

Progetto di Reti di Calcolatori – Laboratorio

Mario Pietrunti 559997

5 settembre 2022

Introduzione

Il progetto rappresenta l'utilizzo di un social media, dove ciascun utente può effettuare delle funzionalità principali per poter interagire con gli altri utenti registrati a Winsome. Ogni utente può seguire gli altri utenti per poter visualizzare i contenuti (post) degli utenti che egli stesso segue, inoltre ha la possibilità di essere curatore di questi contenuti votando o commentando i vari post che vengono visualizzati sul proprio feed . Il social inoltre permette il calcolo di ricompense in base al successo che un post ha riscosso con i voti e i commenti ricevuti. Le ricompense vengono calcolate, tenendo conto dei voti positivi (gli utenti che votano negativamente non vengono ricompensati) e dei commenti pubblicati inerenti al post.

Scelte implementative

Dal punto di vista implementativo il Server è stato realizzato con JAVA I/O e threadpool, di tipo CachedThreadPool, per poter gestire i Client in modo più ottimizzato e gestire simultaneamente e in modo indipendente le richieste che provengono da molteplici Client.

Il server utilizza connessione TCP e RMI in base alla richiesta che riceve per poter poi rispondere al Client.

Sono presenti due sistemi di notifica :

1. RMI callback per i vari aggiornamenti sulla lista dei followers dell'utente.
2. Servizio Multicast per notifiche del calcolo delle ricompense e aggiornamento del wallet dell'utente.

Architettura del sistema e gestione threads

WinsomeServerMain

In WinsomeServerMain il server riceve un file di configurazione per poter effettuare il parsing dei parametri necessari al suo funzionamento, ciò avverrà tramite il passaggio di una URI che individua il file di configurazione all'interno della cartella configurazione.

Successivamente recupera i dati (se presenti) su i due file di backup (usersBackup.json e postsBackup.json) per garantire la persistenza dei dati di Winsome. Essi saranno deserializzati per essere inseriti nelle rispettive strutture dati dedicate.

Vengono avviati 3 thread per poter gestire i task di backup, calcolo della ricompensa e chiusura del server. Inoltre viene creato un threadpool per la gestione delle richieste provenienti dai client.

Viene creata la ServerSocket, il server starà in attesa di richieste di connessione da parte dei client, qual'ora dovesse arrivarne uno, la socket del client sarà affidata al threadpool

ExecutorRequest

In ExecutorRequest vengono eseguite tutte le richieste ricevute del client, attraverso l'utilizzo della classe Winsome Social che contiene tutte le funzionalità principali delle quali verrà e successivamente viene inviata la risposta in base al tipo di richiesta ricevuta. Nel momento in cui non arrivano ulteriori richieste da parte del client il server chiude la connessione.

UsersRewardTask

In UsersRewardTask viene effettuato il calcolo periodico delle ricompense secondo un timeout impostato nella configurazione del server. La ricompensa viene calcolata utilizzando la formula specificata nella traccia del progetto e poi viene suddivisa tra gli autori e i curatori dei post in base ad una percentuale scelta all'avvio del Server. Ogni volta che viene assegnata una ricompensa, viene aggiornato il portafoglio dell'utente e al contempo viene inviata una notifica tramite UDP Multicast.

FilesBackupTask

In FileBackupTask viene effettuata la serializzazione(usando metodi synchronized per evitare inconsistenza tra i file di backup e lo stato del social) in JSON dei dati presenti nel social media Winsome in 2 file, usersBackup.json per salvare tutte le informazioni riguardanti gli utenti e postsBackup.json per salvare tutti i post che vengono creati su Winsome.

Viene utilizzata la libreria gson-2.9.1. Il backup viene effettuato periodicamente secondo un timeout che viene impostato nel momento della configurazione del server.

ClosureServerTask

In ClosureServerTask viene terminato il server quando viene inserito il comando 'chiudi'(chiusura lenta del threadpool) o 'chiudi subito' (chiusura immediata del threadpool con shutdownNow).

In fase di terminazione vengono chiuse le socket TCP e UDP, interrotto il thread che si occupa della ricompensa e il thread del backup e viene svolto l'ultimo backup.

WinsomeClientMain

WinsomeClientMain riceve un file di configurazione per poter effettuare il parsing dei parametri necessari per le connessioni e timeout, successivamente è presente la configurazione della connessione TCP e quella multicast. Inoltre viene effettuata la configurazione RMI per utilizzare i servizi del registry del Server e viene inizializzato il servizio di notifiche di aggiornamento dei followers della classe NotificationFollow che contiene metodi dove viene appunto aggiornata la lista dei followers dell'utente.

In fine è presente un'interfaccia a linea di comando con il quale l'utente può effettuare richieste e svolgere azioni, dove vengono effettuati controlli sull'idoneità dei comandi inseriti.

Funzionalità disponibili :

- register <username> <password> <tags> : l'utente deve fornire username, password e una lista di tag (massimo 5 tag) per la registrazione. Il server può dare errore se la password non contiene almeno 8 caratteri.

- login <username> <password> : login di un utente già registrato per accedere al servizio. Il server dà errore se la password inserita non è corretta o se l'username non esiste.
- logout : l'utente effettua il logout da Winsome.
- list users : consente all'utente di visualizzare gli utenti di Winsome che hanno almeno un tag in comune con l'utente che effettua la richiesta.
- list followers : consente di visualizzare i followers dell'utente. Ogni volta che c'è un nuovo follow o unfollow, il Server aggiorna tramite callback la lista.
- list following : consente di visualizzare la lista degli utenti seguiti dall'utente che effettua la richiesta.
- follow <username> : consente all'utente di iniziare a seguire l'utente specificato dall'username. Il server restituisce errore se l'utente vuole seguire se stesso, se già segue l'utente con quell'username oppure l'username non esiste.
- unfollow <username> : consente all'utente di smettere di seguire l'utente specificato dall'username. Il server restituisce errore se l'utente vuole smettere di seguire se stesso, se non segue già l'utente con quell'username oppure l'username non esiste.
- blog : consente all'utente di visualizzare la lista dei post di cui è autore.
- post <title> <content> : consente all'utente di creare un post. Il server restituisce errore se il titolo è più lungo di 15 caratteri oppure se il contenuto è più lungo di 500 caratteri
- show feed : consente all'utente di visualizzare la lista di tutti i post degli utenti che egli segue dei rewin fatti dagli stessi.
- show post <id> : consente di visualizzare il post con quel determinato id con titolo, contenuto, voti positivi, voti negativi e commenti. Viene restituito errore se l'id non esiste.
- delete <idPost> : consente all'utente di cancellare il post con quel determinato id solamente se egli è l'autore del post e inoltre viene restituito errore se l'id non esiste.

- *rewin <idPost>* : consente all'utente di ricondividere un post di un altro utente. Viene restituito errore se il post non esiste, se l'utente è autore del post o se il post non è presente nel feed dell'utente.
- *rate <idPost> <vote>* : consente all'utente di assegnare un voto positivo o negativo (1 oppure -1). Il Server restituisce errore se il post non esiste, se l'utente è l'autore del post, se il post non è nel feed dell'autore, se l'utente ha già votato quel post o se il voto è diverso da 1 o -1;
- *comment <idPost> <comment>* : consente all'utente di commentare il post con quell'id. Il commento deve essere massimo di 100 caratteri. Viene restituito errore se l'id del post non esiste, se l'utente che vuole commentare è l'autore del post o se il post non è presente nel suo feed.
- *wallet* : consente all'utente di visualizzare il totale del portafoglio in wincoin e la lista delle transazioni relative alle ricompense assegnategli fino a quel momento
- *wallet btc* : consente di visualizzare il totale del portafoglio in bitcoin.
- *help* : consente all'utente di visualizzare tutti i comandi che può digitare.
- *quit* : consente all'utente per terminare il Client.

UDPMulticast

In UDPMulticast vengono ricevuti messaggi multicast inviati dal server ogni volta che viene calcolata una nuova ricompensa e l'aggiornamento del portafoglio dell'utente.

Entità e strutture dati

WinsomeSocial

In Winsome Sociale vengono gestite tutte le informazioni degli utenti e dei post di Winsome. Contiene l'implementazione dei metodi che rappresentano le funzionalità principali di Winsome. Sono state utilizzate delle CurrentHashMap per gestire gli utenti e i post per poter svolgere azioni come add, get e remove senza una sincronizzazione esplicita. Le sincronizzazioni sono gestite dalle classi User e Post.

Post

In Post è possibile trovare le informazioni riguardanti un post di Winsome come id (univoco), username dell'utente autore, titolo, contenuto, numero di volte che è stato valutato il post, numero dei commenti, lista dei voti ricevuti(positivi e negativi), lista dei commenti e la data dell'ultima ricompensa. Sono state utilizzate due ReentrantLock per i voti e per i commenti. Inoltre sono presenti metodi per la modifica e la gestione di ogni post di Winsome.

User

In User è possibile trovare le informazioni riguardanti un utente di Winsome come username (univoco), password, lista dei post pubblicati dall'utente, il feed, lista dei followers (seguaci), lista dei followed (seguiti), lista dei tag dell'utente, lista dei post votati dall'utente e il portafoglio dell'utente (Classe Wallet). Sono state utilizzate delle ConcurrentHashMap per quanto riguarda il blog e il feed dell'utente, invece per i followers è stata usata una ReentrantLock in modo da non avere inconsistenze quando viene modificata o visualizzata la lista dei followers. Inoltre sono presenti metodi per la gestione dell'utente e un costruttore per quando un nuovo utente si registra e un altro per rigenerare gli utenti nel file di backup.

Wallet

In Wallet è possibile trovare le informazioni riguardanti il portafoglio di un utente come proprietario, lista delle transazioni e ammontare totale del portafoglio in wincoin. Inoltre viene implementato il metodo per la conversione del totale del portafoglio da wincoin a bitcoin e il metodo per l'aggiornamento del portafoglio quando viene assegnata una nuova ricompensa.

Transaction

In Transaction è possibile trovare le informazioni riguardanti una transazione come importo, e causale della transazione.

Vote

In Vote è possibile trovare le informazioni riguardanti un voto come autore del voto, voto e data del voto

Comment

In Comment è possibile trovare le informazioni riguardanti un commento come autore, contenuto e data della pubblicazione del commento.

Indicazioni per l'utilizzo

Compilazione codice

1. Aprire il terminale nella cartella src;
2. Inserire il comando:
`javac -cp ../librerie\gson-2.9.1.jar *.java`

Eseguire WinsomeServerMain

1. Aprire il terminale nella cartella src;
2. Inserire il comando:
`java -cp ../librerie\gson-2.9.1.jar WinsomeServerMain.java
../configurazione\Server.txt`

Eseguire WinsomeClientMain

1. Aprire il terminale nella cartella src;
2. Inserire il comando:
`java WinsomeClientMain.java ../configurazione\Client.txt`

Eseguire il file Server.jar:

1. Aprire il terminale nella cartella jar;
2. Inserire il comando: `java -jar Server.jar ../configurazione\Server.txt`

Eseguire il file Client.jar

1. Aprire il terminale nella cartella jar;
2. Inserire il comando: `java -jar Client.jar ../configurazione\Client.txt`