TypefastServer Documentation

Overview

`TypefastServer` is a Java-based server for the Typefast game, which allows multiple clients to register, login, join games, and view scores. The server handles client connections using socket programming and manages game logic, user authentication, and scoring.

Main Components

- **Networking Components**: `ServerSocket`, `Socket`
- **Threading and Concurrency**: `ExecutorService`, `ClientHandler`
- **Data Structures**: `ConcurrentHashMap`, `ArrayList`, `List`
- **File I/O**: `Files`, `Paths`

Constants

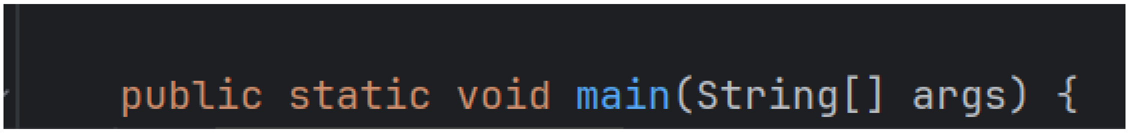
- `**PORT**`: The port number on which the server listens (default is 12345).
- `**INITIAL_TIME_LIMIT**`**:** The initial time limit for each word (default is 30 seconds).
- `**GROUP_SIZE**`: The number of clients required to start a game (default is 3).
- `**WORDS_PER_GAME**`: The number of words per game (default is 5).
- `**WORDS_FILE_PATH**`: Path to the file containing the list of words.

Fields

- `**userDatabase**`: A concurrent map storing username and password pairs.
- `**authenticatedUsers**`: A concurrent map storing authenticated users and their handlers.
- `**waitingClients**`: A synchronized list of clients waiting to join a game.
- `**words**`: A list of words loaded from a file.
- `**pool**`: An executor service for handling client threads.

Methods

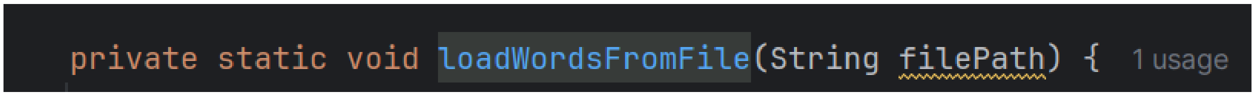
Main Method



```java
public static void main(String[] args) {
```

- Starts the server, loads words from the file, and listens for client connections.

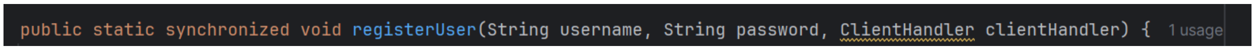
Word Loading



```java
private static void loadWordsFromFile(String filePath) {
```

- Loads words from the specified file and stores them in the `words` list.

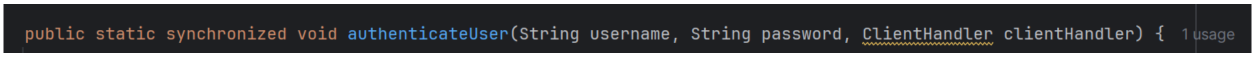
User Registration



```java
public static synchronized void registerUser(String username, String password, ClientHandler clientHandler) {
```

- Registers a new user if the username does not already exist.

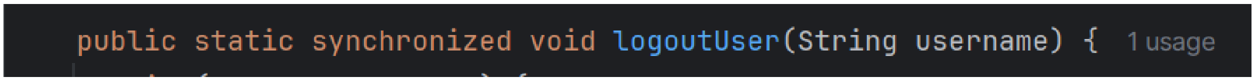
User Authentication



```java
public static synchronized void authenticateUser(String username, String password, ClientHandler clientHandler) {
```

- Authenticates a user if the username and password match.

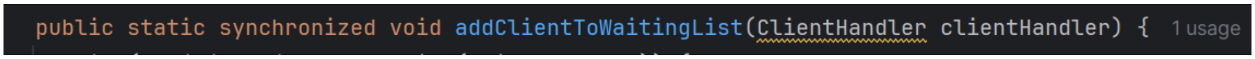
User Logout



```java
public static synchronized void logoutUser(String username) {
```

- Logs out the user and removes them from the authenticated users list.

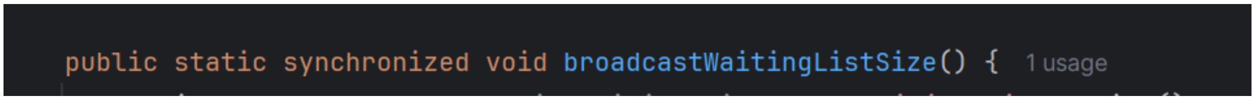
Waiting List Management



```
public static synchronized void addClientToWaitingList(ClientHandler clientHandler) {  1 usage
```

- Adds a client to the waiting list and starts a game if enough clients are waiting.

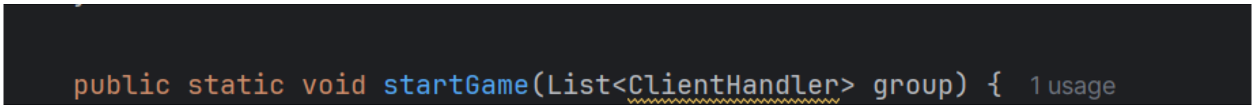
Broadcast Waiting List Size



```
public static synchronized void broadcastWaitingListSize() {  1 usage
```

- Broadcasts the number of clients in the waiting list to all waiting clients.

Game Management



```
public static void startGame(List<ClientHandler> group) {  1 usage
```

- Starts a game for the specified group of clients, sending words and handling scoring.

ClientHandler Class

- Inner class that handles individual client connections.
- Fields: \`socket\`, \`in\`, \`out\`, \`username\`, \`currentWord\`, \`startTime\`, \`wordTypedCorrectly\`, \`totalScore\`, \`exitFlag\`.
- Methods: \`run\`, \`handleClientMessage\`, \`sendMessage\`, \`setUsername\`, \`setWord\`, \`setStartTime\`, \`isWordTypedCorrectly\`, \`getTotalScore\`, \`calculateScore\`, \`sendScoreboard\`.