Future Chat

Foundations of Distributed Systems
Lab Guide 3

2020/2021

Consider a chat server using Java and sockets, where lines sent by any client are broadcast to all currently connected clients.

Steps

1. Implement the server using CompletableFuture asynchronous sockets.
2. Run clients with different delay configurations.
3. Refactor the server to work on strings internally.
4. Add a login/password validation step to connections.

Questions

1. How does this code map to the equivalent threaded and callback-based program?
2. What version of the login layer is easier to develop/understand?

Learning Outcomes Apply event-driven programming based on monadic asynchronous constructs. Recognize the equivalence between threaded and various approaches to asynchronous programming.

Maven dependency for FutureSockets

```
<repositories>
    <repository>
        <id>jitpack.io</id>
        <url>https://jitpack.io</url>
    </repository>
</repositories>

<dependencies>
    <dependency>
        <groupId>com.github.spullara</groupId>
        <artifactId>java-future-jdk8</artifactId>
        <version>master</version>
    </dependency>
</dependencies>
```