Monadic Futures

Distributed Systems Paradigms
Lab Guide 4

2021/2022

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

Steps

1. Implement a layer to convert from bytes to strings strings.
2. Implement a layer to do login/password validation on incoming connections.
3. Advanced: Refactor the server to work on strings internally using futures.

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>
```