

Concurrency Stock Application

Functional requirements

- A user can place a request to sell/buy stocks at a certain price.
- A user can see the current price of the stock they're looking for.

Non-functional requirements

1. Our service needs to be available 24/7.
2. It needs to be able to process requests as fast as possible because these influences the prices and difference in prices between price at request time and price at processing/buy time should be as small as possible.
3. Prices should be updated fast as stocks are bought or sold.
4. Out of stock or unavailability are possible problems unless a large capacity of stocks are considered (minimum amount set for companies).

Concurrency Issues in our project:

Problem: How do we manage having multiple threads?
Answer: We are using a Fixed Java Thread Pool.

Problem: How do we simulate a user request?
Answer: We are using events, therefore in our application a request is interchangeable with an event. Our preferred framework is RabbitMQ.

Problem: Multiple threads accessing the same company (resource).
Answer: We are using a synchronized block on that bit of code.

Problem: Exposing companies data to all of our threads.
Answer: We implemented a custom thread safe repository, which is a singleton.

High-level component design

