Martin Andersson

SOFTWARE ENGINEER

martinandersson.com/f/cover-letter.pdf
martinandersson.com/f/resume.pdf
stackoverflow.com/users/1268003
github.com/martinanderssondotcom
linkedin.com/in/martinanderssondotcom

This paper presents an overview of a few of my personal projects. Screenshots and code are provided on my personal website and/or on GitHub. Private code can be screenshared. The most common tech baseline is Java, Gradle, TestNG and Mockito.

CODE GALLERY

Queue Service Benchmark

2017.04 > 2017.06
public, GitHub

A queue service API- and framework with implementations that mostly differ in how thread-safety is accomplished. One interesting finding was that for sufficiently short read-operations, the serialized keyword is faster than ReadLock. Uses JMH as benchmarking framework.

Money Profiling

2016.? > 2016.07
public, GitHub

Profiles serialization- time cost and byte size footprint when writing models of a monetary amount to Chronicle Map (uses JMH for benchmarking). Some models tested was provided by Moneta (reference implementation of Java's Money and Concurrency API). Serialization frameworks: Java's serialization protocol, Kryo and FST.

Golden Egg

2015 > 2017
private, GitLab

Backtesting of computer generated investment strategies against data provided by sources such as the Swedish government and Quandl. Key was modeling abstractions and performance. I rolled my own solutions for mathematically described ranges and streams of extreme length which could further be linked, combined and compounded.

Secure Login File Transfer

2010.03 > 2012.03
public, GitHub

An end-to-end example of secure user authentication and encrypted binary file transfer over WebSocket without using public keys or a third party certificate authority. Used the SRP protocol with AES/GCM encryption. The client is written in JavaFX, backend in Java EE.

Live Chat

2012 > 2015
private, GitLab

A complete live chat solution with two front-end clients; one web client is written in HTML/CSS/TypeScript and one fat client written in JavaFX. The backend uses Java EE. The live chat solution was at the time the world's only that featured an optional live typing mode (see *this video*).

Java EE Concepts

2015.03 > Ongoing
public, GitHub

Tons of client-server/in-container test cases using Arquillian that together with killer docs explore and demonstrate core- and edge-cases of Java EE technologies such as JPA, EJB, CDI and JTA.

webmaster@martinandersson.com +1 (434) 288-0795