Always leave the code base a little better than you found it

Johan Sjöblom

Mobile +46 704-582192

Email sjoblom.johan@gmail.com

Introduction

Johan has always had a great interest in computers and first began programming at the age of twelve. He often has a few software projects going on, and a constant hunger for learning more and improving his craftsmanship.

Johan has substantial experience from working with several different types of systems, with Kotlin, Java, the Spring Framework and Kafka being his core competence. He has worked with business-critical software, such as core software-components in the automotive industry, financial systems and governmental systems. He has worked on and designed cutting edge microservice architectures, as well as maintained and improved ten-year-old Java monoliths. Recently he has discovered the joy of modern front-end development, having some experience in Javascript and frameworks, as well as data visualization using d3.js. He always strives to work test-driven, and apart from unit testing, he has also worked with and created end-to-end test suites and load tests. No matter the project, he strongly believes that code must be clean and clear. Code quality, readability, robustness, testing and well thought-through solutions are keywords in Johan's deliveries.

Johan has held presentations and coding dojos for team members. He has taken the initiative to introduce new tools to the technology stack, and to improve the team's way of working. Software is always built by people for people, and to understand the problems that are to be solved as well as the interactions between those who build the solution, is something Johan believes is very important. Most problems and complications arise from inadequate communication.

As a person, Johan is social and outgoing. He is driven by creative joy and takes on challenges with passion. Tasks exist in order to be solved and processes to be simplified!

Programming Languages / level

Java	Expert
Kotlin	Very good
C	Very good
JavaScript	Very good
Bash	Very good
Tcl/Tk	Very good
HTML	Very good
C++	Good
Python	Good
Haskell	Good
PHP	Good
CSS	Good
Assembler	Good
MATLAB	Good

Technologies

Spring Framework / Spring Boot
REST
Micro services
Kafka, Kafka Streams
Hibernate, Liquibase
Cloud (GCP, aws)
JUnit, Mockito, Wiremock, pitest
AMQ/WMQ/ActiveMQ
Guava/Caffeine cache
d3.js
Jboss, Wildfly
Json, XML
Maven, Gradle
git, svn, GitHub Actions, Jenkins, TravisCI

Databases / SQL languages

Oracle	Very good
Postgres, MySQL	Good
Prometheus	Good
InfluxDb	Good
SPARQL / Stardog	Good

Tools and Software

Docker
Grafana
SonarQube
ElasticSearch
Kibana
Gatling
Fluentd
Nginx
Kong
IntelliJ, Eclipse
Linux, Windows

Languages

Swedish	Native – C2
English	Fluent – C2
Norwegian	Fluent – C2
German	Understanding – A2

Education/Certificates

2009 – 2015: Master of Science, Computer Science – Algorithms, Languages and Logic, Chalmers University of Technology

2019: Docker Course for busy DevOps and Developers

2016: Certificate: Java SE Programmer I (Oracle 1Z0-808)

2009: Programming C++, Uppsala University

2009: Web design with JavaScript and the DOM, Umeå University

2007 – 2008: Mathematics, Gothenburg University

Employment

Squeed AB
2015-03-30 – ongoing

Ruag Space
2012 + 2013 Projects employee

Reference projects

NAV, Team Søknad
Role: Architect and developer
2019 May – Ongoing

Johan joined NAV as the team he was part of was formed, and they were handed a number of critical legacy systems almost without any handover. The systems were part of the Norwegian government's welfare programs, and handed citizens requests for social security. Downtime or malfunction of the systems ment that people would be without income, which in the long run would have political consequences. The software was therefore referred to as critical to society.

While stable, the systems were in dire need of attention. The tasks they performed were relatively simple, but legacy solutions and unmaintained libraries were sprinkled all over the systems. Business logic was mixed with logic for integrating with the old libraries and frameworks.

The team realized that large parts of the systems could be deprecated and had to be rewritten. Johan acted as the architect for designing and implementing a Kafka based replacement. By basing the system on Kafka, reliability and robustness could be built into the infrastructure. The system was Event Sourced, meaning that the state could always be recreated by replaying the Kafka stream. Much care was taken into the fault tolerance and reliability of the system, making sure it would run automatically, with many replicas yet leaderless, and handle error scenarios.

Thorough care was put into making extensive automatic end-to-end tests, load tests, continuous integration and state of the art solutions.

Technologies: Kotlin, Java, Kafka, Kafka Streams, GCP, Kubernetes, Maven, Postgres, Spring Boot, REST, Docker, Github Actions, JUnit, Wiremock, Mockito, Testcontainers

Shipmember
Role: Architect and developer
2019 January – 2019 March
https://github.com/sjoblomj/shipmember

Johan worked on a project to send out invitations and information to members and subscribers. After parsing a list of membership information, the application was capable of sending out personalized emails, generating personalized PDF files using the LaTeX typesetting system and sending different types of messages depending on the member properties. The application automated what used to be a manual laborious process, and the personalized messages were easily customizable using the Thymeleaf templating engine.

Technologies: Kotlin, Gradle, JUnit, pitest, LaTeX, SMTP, CSV, Jodd Email, Thymeleaf

Travelling Bruteforcer
Role: Architect and developer
2018 September
https://github.com/sjoblomj/TravellingAntColony

In order to learn Kotlin, Johan wrote an application to brute-force a solution to the Travelling Salesman Problem. The algorithm initially greedily calculates a decent path by picking the nearest unvisited node, after which it will evaluate every other permutation of nodes. When determining a permutation, it could be that after visiting only a subset of nodes, the path already exceeds the record. In that case, the algorithm will stop early and not waste time calculating the rest of the permutation.

Technologies: Kotlin, Maven, JUnit, Immutable data structures, extension functions, typealiases, tail recursion, sequence generation, smart casting, local functions, default parameter values

Difi, Direktoratet for Forvaltning og Ikt, eInnsyn
Role: Senior Developer, Infrastructural engineer
2018 August – 2019 March

Johan worked as a senior developer on a governmental agency, within einnsyn.no, which is a website exposing documents from authorities to the general public. Before publicizing, documents were handled by a number of Java microservices that would enrich the data, analyze it and make the content searchable.

The back-end consisted of micro services chained though an ActiveMQ. Documents were saved to a graph database called Stardog, and retrieved using SPARQL. ElasticSearch was used to make documents searchable. All components in the system were dockerised. The data was made available to the general public and journalists through a web portal, available at einnsyn.no

The project was in a maintenance and issue fixing phase when Johan joined, so his tasks were mainly to tweak, bug fix and perform various infrastructural improvements. That meant a lot of exposure to a vast array of technologies, and apart from coding, Johan worked on and improved robustness of the application security, logging flow, server configuration, docker setup, the graph database, fields to expose through ElasticSearch and code quality analysis.

The project and the team's efforts were well received and praised. Seeing as how access to documents from public agencies is a vital cornerstone in a democracy, the team was entrusted a high degree of freedom and could choose technologies freely. This resulted in a sophisticated technology stack, well suited to solve the task at hand.

The team was distributed across four different locations, so Johan got first hand experience of the challenges and opportunities brought by remote work. Apart from strengthening the team, improving code, tests and processes, Johan helped solidify the underlying technologies.

Technologies: Java, Spring Framework, Docker, ActiveMQ, SPARQL, Nginx, Kong, Fluentd, Kibana, SonarQube, ElasticSearch, Stardog, JUnit, Wiremock, Mockito, REST, Micro services, Bash, Linux, Travis CI

PayEx
Role: Architect and Senior Developer
2017 September – 2018 June

Johan was the architect and senior developer for a series of Java microservices, for a financial system. Major technologies used were Spring Framework, Oracle databases, Rest communication and ElasticSearch caches. The system handled value codes, i.e. discounts for certain products, given to certain customers. The software was business-critical, where customers would stand in a store and scan their value codes, so the system had to be very responsive with minimal latency.

When Johan was brought into the project, he made several performance-boosting improvements to the design, such as adding multi-layer caches and buffers. Johan took the initiative and responsibility of bringing in application monitoring to the system. He was the lead developer of the customer facing parts of the system.

Johan also worked on several other core financial systems during his time at PayEx. He held multiple training sessions for his team on testing, test-driven development and code quality, and he held coding dojos. He introduced technologies such as application monitoring and non-intrusive exposure of metrics and meta-data. Johan took the initiative to make everyone focus less on silo-programming and individualism and more on being an effective team together. He received much acclaim for professionalism, technical experience, his will to share knowledge and help the whole team improve, as well as for socializing and quickly becoming a natural part of the team.

Technologies: Java, Spring Framework, REST, Oracle, Docker, Hibernate, SonarQube, ElasticSearch, Caffeine Cache, Grafana, Prometheus, ActiveMQ, json, JUnit, Mockito, Wiremock, Velocity, Jenkins, Micro services

Volvo Group Telematics, Vehicle Service Core
Role: System Developer
2016 December – 2017 August

The Volvo vehicles constantly send a large amount of data to the Volvo Telematics systems. The Vehicle Service Core is one of the first systems in the architecture that receive data, and the systems there were responsible for among other things enriching the data by communicating with other services and routing messages to consumers.

The systems in the Vehicle Service Core were a mixture of a monolith and micro-services. Johan added new features and performed maintenance, and helped with support and deployment. He took the initiative and responsibility for conducting load tests for a component in a restricted environment and wrote a parser that extracted log data and illustrated it in Grafana graphs. The results were vital in finding several important configuration issues.

Technologies: Java, Spring Boot, Docker, Kibana, AMQ, Grafana, JAXB, JUnit, Mockito, REST, Micro services, cloud based applications

Volvo Group Telematics, NGPS
Role: System Developer
2016 August – 2016 December

Johan worked in an enterprise system for fleet management and reports, written in ECMAScript 6 with React.

The NGPS project was a web front end for management of Volvo trucks and heavy-duty vehicles, such as excavators, graders, articulated haulers and the like. The application gave users modern means of fleet management, allowed for tracking of vehicles on a map and fetched data about vehicles, drivers and assets for display. The application was intended to be used for mobile, tablets and desktop and to be used in locations with bad or even no Internet connection.

Johan was part of the project during its initial phase and contributed to and took responsibility for its core parts. He worked with features such as searching, filtering, routing and design.

The application was written in React and ECMAScript. Many of the cutting edge technologies of the day were used, with React Boilerplate as a base, which included projects such as Redux (state container), Hot Reloading (instant results on save), Babel (transpiling Javascript dialects), react-router (keeping URLs in sync with the UI), Offline First (making sure Internet access is not required) and redux-saga (side effect management).

Technologies: ECMAscript, Redux, Hot Reloading, Babel, react-router, Offline First, redux-saga, Immutable.js

Volvo Group Telematics, Caretrack
Role: System Developer
2015 April – 2016 August

Caretrack is a telematics system from Wireless Car, a subsidiary of Volvo. It is a large Java monolith system running on JBoss, developed for more than ten years. The different kinds of Volvo heavy duty vehicles (such as excavators, graders and articulated haulers), all send data to the Caretrack system. Several measurements and metrics are sent, such as fuel consumption, operating hours and vehicle position. Using this data, Volvo can provide customers with services such as subscriptions of reports, setting geofences on vehicles, fleet management, automatic alarms and much more.

Johan spent quite a bit of time with maintenance and refactoring. Some initial steps were taken to move towards a micro-service architecture, and Johan was involved and took responsibility for the development of all components that were created during his time there. When production issues arose, Johan often was the first person to be entrusted to take action.

Technologies: Java, Spring Boot, JBoss, AMQ, WMQ, JAXB, Oracle, Hibernate, Liquibase, log4j, EJBs, JPA, JUnit, Jenkins.

Saab EDS, Gothenburg
Role: System Developer
2014 March – 2014 August

Evaluated the possibility to introduce visual GUI testing of the radar systems that were being developed by Saab. The system presents the operator with a map view upon which moving targets are projected. The operator can interact with the targets, classify them and issue orders to other operators. Although extensive tests were already conducted, there is a great value in testing the system on a GUI level, which was not possible at the time. Using image recognition algorithms, scripts with regression tests can automatically be executed. The result Johan produced was far above Saab's expectations, despite the great demand for robustness due to the moving targets. The conclusion was that the test frequency of the system could be substantially raised, bugs and errors could be found in a very early stage and the cost of system testing could be drastically reduced.

Technologies: Sikuli, JAutomate, Python, git

RUAG Space, Gothenburg
Role: System Developer
2013 May – 2013 August

Johan wrote a program for generating electronic schematic symbols for printed circuit board design. The program allowed the user to describe a symbol in a very concise manner and drew a graphical representation. Input to the program came from other tools, and the program produced files that were loaded into software from Mentor Graphics. This format was proprietary and largely undocumented, so effort was needed to reverse engineer and understand the file format that was to be produced.

Technologies: Tcl/Tk, Reverse engineering, Domain specific language design, parsing, svn, Linux

RUAG Space, Gothenburg
Role: System Developer
2012 June – 2012 August; 2013 January – 2013 February

Developed a tool to automatically generate hardware specifications. The user could tell the program the specifications of a hardware board, and in what ways components and interfaces should be connected. The user input was short and concise, and it was fed into the TCL parser which powered the application. As should always be the case when it comes to user input, much emphasis was placed on automatically checking if the given data is safe, sound and without conflicts. As development progressed much faster than anticipated, new requirements and ideas were added multiple times.

Technologies: Tcl/Tk, Domain specific language design, parsing, svn, Linux

Voluntary assignments

Chairman of the Board
Sällskapet Vikingatida Skepp
March 2019 – ongoing

Chairman of the board of an association that has built and is maintaining and sailing a Viking ship. The association consists of around 175 people. In his role as chairman, Johan oversaw the day-to-day as well as longterm activities of the association. He made sure the maintenance was done, he helped to do recruitments, worked with the goals and visions of the association, as well as planning logistics and trips. Johan was involved in several tough situations, such as handling interpersonal issues, downscaling and selling one of the ships in which many members had much invested feelings, talking to the media, and planning the purchase of a new sail as well as encouraging members to donate funds for it.

Member of the Board
Sällskapet Vikingatida Skepp
March 2015 – March 2018

Member of an association that has built and is maintaining and sailing a Viking ship. Johan made sure the maintenance was done, he helped to do recruitments, worked with the goals and visions of the association, as well as planning logistics and trips.

Chairman of the Board
Computer science's Student Division Board, Chalmers University of Technology
May 2012 – May 2013

Chairman of the computer science student division and its board. Represented about 700 students and was ultimately responsible for a turnover of above half a million SEK. The role involved a lot of administrative work, leading board meetings, leading longer meetings for the whole student division (around 80 participants) and being involved in the shaping of the future of the education.

Member of the Board
Computer science's Student Division Board, Chalmers University of Technology
May 2011 – May 2012

Member of the computer science student division board. Participated in decision-making at board meetings each week and was involved in administrative work.

President
Computer science's rustmästeri, Chalmers University of Technology
May 2011 – May 2012

President of the committee responsible for the maintenance of the computer science student division premises. Apart from the premises, the committee also had responsibility over machines and tools, and arranged a large variety of activities for computer science students.

Purchasing Manager
Computer science's pub committee, Chalmers University of Technology
May 2010 – May 2011

Purchasing manager at the pub of the computer science student division. Was responsible for a varied range of about 140 kinds of beer. Held a large number of events during the year, especially during the Chalmers reception weeks. Johan used his social skills to welcome new students, and was crucial in creating a friendly and welcoming environment where people felt at home. Ten years after his work in the pub committee, students he had never met still knew his name.