Igor Gabaydulin

Curriculum Vitae

Weeks of coding can save you hours of planning

Experience

Vocational

2018–Present **Lead Software Developer**, *Evotor*, Moscow, Cloud Management Department.

Lead the team of three backend developers (including me). The number of microservices has exceeded 30, there are several scripts that automate pre-release processes and testing. Different kinds of tests now automatically run when we create pull requests. We have migrated our services from Java 8 to Java 12 (skipping 11 which is LTS version) and noticed improvements in memory consumption thanks by new GC called G1. We have updated all tools and libraries we use (we migrated to Spring 5.*, Spring Boot 2.*, Junit Jupiter, Gradle 5.* and etc.).

Less and less engaged in business tasks and spend a lot of time on technical tasks. To implement kind-of-atomicity when Kafka and PostgreSQL are used, we started to use Debezium and Kafka Connect. The second one was buggy and I implemented our own solution based on Debezium and Zookeeper (to coordinate several application instances between each other). I also have found a problem in Debezium which lead to a problem when PostgreSQL could not shutdown because Debezium would never end replication process. I have implemented a test written in Kotlin, integrated testwith GitHub Actions, reported the bug and lately contributed a fix for it. Step by step integrating multi-module architecture into our projects.

- ○ Languages: Java 12
- ○ Build Tool: Gradle 5.6.*
- ○ Frameworks: Spring 5.*, Spring Boot 2.1.*, Spring Cloud
- ○ Servlet Containers: Undertow
- ○ Testing Frameworks: JUnit 5, Testcontainers
- ○ Databases: PostgreSQL, ClickHouse
- ○ Containers: Docker, Kubernetes, helm
- ○ Caching: Redis, caffeine
- ○ Logging: Logback, Logstash, ElasticSearch, Kibana
- ○ Monitoring: Javamelody, Zabbix, Prometheus, Grafana
- ○ Kafka, Zookeeper, Debezium
- ○ Version-Control System: git, gitflow

Moscow, Russia

 +7 (962) 935 3115 • igabaydulin@gmail.com • igabaydulin

 igabaydulin • igabaydulin

2017–2018 **Senior Software Developer**, *Evotor*, Moscow, Cloud Management Department.
Migrated services to Kubernetes cluster. Created Kubernetes configurations for test and production environments. Lately migrated configurations to use helm. Made caching optimizations and migrated caching from in-memory to use remote cache (Redis). Fixed some memory problems (using standard Java utils and Eclipse Analyzer to find them) and found memory leaks when Logback was using async appender to write file logs. Added service metrics and alerts using Prometheus and Grafana. Created Jenkins pipelines to automate release process. Helped testers to create a new base for theirs automated tests.

- Languages: Java 8
- Build Tool: Gradle
- Frameworks: Spring 4.*, Spring Boot 1.*, Spring Cloud
- Servlet Containers: Undertow
- Testing Frameworks: JUnit 4, Mockito
- Databases: PostgreSQL
- Containers: Docker, Kubernetes, helm
- Caching: Redis, caffeine
- Logging: Logback, Logstash, ElasticSearch, Kibana
- Monitoring: Javamelody, Zabbix, Prometheus, Grafana
- Documentation: ReDoc, Asciidoc
- Kafka, Zookeeper
- Version-Control System: git, gitflow

Moscow, Russia
+7 (962) 935 3115 • ✉ *igabaydulin@gmail.com* • *igabaydulin*
in *igabaydulin* • *igabaydulin*

2017–2018 **Software Developer**, *Evotor*, Moscow, Cloud Management Department.

Evotor produces cash registers based on Android OS and has its own marketplace for applications which can be installed by customers and provides the platform for developers to create and publish application. Marketplace is based on microservice architecture and actively uses containerization. Back then we would package microservice in Docker image, publish to local Docker registry and then manually deploy it by docker-compose (connecting to server by ssh). I maintained about 10 microservices and designed a couple of new microservices (app rating service, targeting service). Reduced a lot of code duplication creating starter lib. Have fixed a lot of bugs (including the problems race condititions). No CI/CD has been used back then. Insisted to use Gradle Wrapper.

- Languages: Java 8
- Build Tool: Gradle, Gradle Wrapper
- Frameworks: Spring 4.*, Spring Boot 1.*, Spring Cloud
- Servlet Containers: Embedded Tomcat
- Testing Frameworks: JUnit 4, Mockito
- Databases: PostgreSQL
- Containers: Docker, docker-compose, docker registry
- Caching: guava
- Logging: Logback, Logstash, ElasticSearch, Kibana
- Monitoring: Javamelody, Zabbix
- Kafka, Zookeeper
- Version-Control System: git, gitflow

2016–2017 **Backend Developer**, *VibrantFire*, Moscow, Startup Project.

VibrantFire is a startup project whose employees were then-students of Moscow State University. We were developing custom software (mobile applications usually). This was a first time when I developed web-server from scratch using Spring Framework and Tomcat (lately migrated to Spring Boot and Embedded Tomcat). It was a monolithic application with a layered architecture. No CI/CD has been used, instead we would connect directly to server by ssh, pull changes from GitHub, build and deploy jar manually. As a proxy server we would use HAProxy with minimal custom configuration.

- Languages: Java 7
- Build tool: Maven
- Frameworks: Spring (JPA, mvc, security and others), Spring Boot, Hibernate
- Testing Frameworks: JUnit 4, Mockito
- Servlet Container: Embedded Tomcat
- Database: PostgreSQL
- Version-Control System: git

Moscow, Russia

 +7 (962) 935 3115 • ✉ *igabaydulin@gmail.com* • ✪ *igabaydulin*
in *igabaydulin* • 🐦 *igabaydulin*

2014–2015 **Junior Developer**, *Ivannikov Institute for System Programming of the RAS*, Moscow.

At ISP RAS I was testing text analysis tool based on Wikipedia. One of the things it was able to do is to create relations between terms and I have been writing tests to build a tree of relations, excluding situations like when child term becomes a parent of its parent term. The library was written in Java, neither popular libraries nor tools have been used.

Languages

Russian	Native
English	Intermediate *Can read books and technical literature, watch movies, know grammar. Can speak some English, but don't have much practice.*

Computer skills

Programming	Java > C++ > Haskell > Kotlin	Databases	PostgreSQL
OS	Ubuntu, Windows	IDE	IntelliJ, DataGrip, PostMan
Containers	Docker, Kubernetes, helm	Other	bash, terminal, git, markdown

Interests

Programming I started to write code when I was about 14 years old. The first programming language I learned was Pascal and then Delphi 7. I thought they were the best languages until I learned about C# and then Java in Higher School of Economics in Bachelor's Programme 'Software Engineering'. There I also learned about disciplines such as computer vision and computational geometry. I really liked them, my first coursework described one of the image contouring algorithms and contained an implementation written in C#. When I was studying at Moscow State University I met sports programming. I didn't do very well, but I have learned a lot about algorithms and data structures.

Music I like to play the piano and guitar. I studied at a music school for seven years and now I can't imagine my life without some music.

Chess I like to play some chess, watch main chess tournaments and some players. But I wouldn't say I play chess well.

Extra

- I don't like Spring Framework very much. Yes, it may speed up a project to be ready for first release, but using Spring you sign a verdict and lately the migration to another framework will be very difficult. I don't want to see my code to be filled with framework's code. But nowadays it seems to be common to weave application functionality with framework's functionality.

Moscow, Russia

+7 (962) 935 3115 • igabaydulin@gmail.com • igabaydulin

igabaydulin • igabaydulin

- I don't like when everyone understands a term in their own term. I don't like what happened to OOP. I don't think inheritance must be a part of OOP. I don't think Java is OOP and Java EE made it even worse approving anemic objects. I see procedural code, not object-oriented. Then I read about "Elegant Objects" and it contained ideas what I felt OOP should contain. But the more I thought about OOP, the more I understand that what I think of is actually has been implemented in FP. In my free time I like to read about functional programming, category theory and Haskell (can't wait to read "Haskell in depth").
- I like to read books. Real printed books.

Moscow, Russia

📱 *+7 (962) 935 3115* • ✉ *igabaydulin@gmail.com* • ⦿ *igabaydulin*
in *igabaydulin* • 🐦 *igabaydulin*