

Curriculum vitae

Personal data

Name: Azat N.

Technical Knowledge and Skills

Java, Spring, Git, CI/CD, Linux, Rest API, SQL, GraphQL



Azat is a Backend engineer with 9 years of IT experience and has built a platform which helps to build Back-end applications faster. Azat is a good manager, recruiter, can play in a team and solo, responsible for whatever is done, inspiring all team members. With great communication skills, she has worked in teams of all sizes. Also he is an experienced team leader and he is keen on agile processes.

Education

September 2010 —
July 2014

Kazakhstan
Software-Engineering, Kazakh-British Technical University

Work experience

January 2019 -
Present

IT Services and IT Consulting Company, Kazakhstan
Product Architect

Stack:

- Java,
- Spring,
- CI/CD,
- Gitlab,
- Kubernetes,
- NEXUS,

- Zeebe,
- MongoDB,
- ElasticSearch,
- MyBatis,
- Docker,
- AWS,
- PostgreSQL,
- Liquidbase,
- Flyway,
- JHipster,
- Angular.

Responsibilities:

- Team recruitment;
- Working close with clients;
- Problem statement;
- Task management;
- Architecture development;
- Microservice architecture development.

August 2017 -
January 2019

System integration Company
Software Development Team Lead

Stack:

- Java,
- SpringBoot,
- RabbitMQ,
- PostgreSQL,
- Oracle,
- Liquidbase,
- Hibernate,
- MongoDB,
- GIT,
- MicroServices,
- AWS,
- QueryDSL.

Responsibilities:

- Monolith refactoring and constructing of new microservice architecture;
- FinTech Development;
- Code review;

- Motivation and mentoring for junior programmers and other colleagues;
- Implemented mobile and internet banking in 3 major banks of KZ.

May 2013 -
January 2019

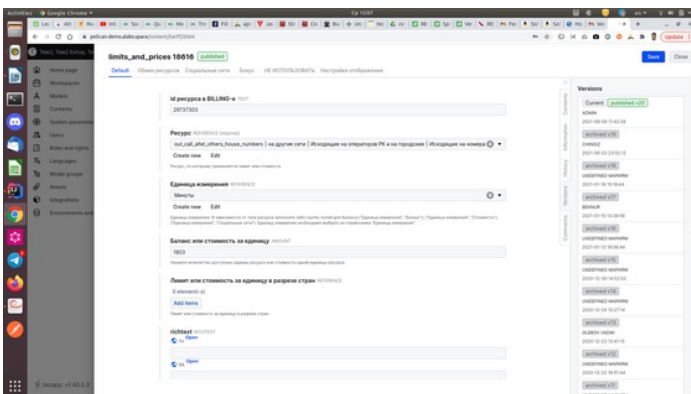
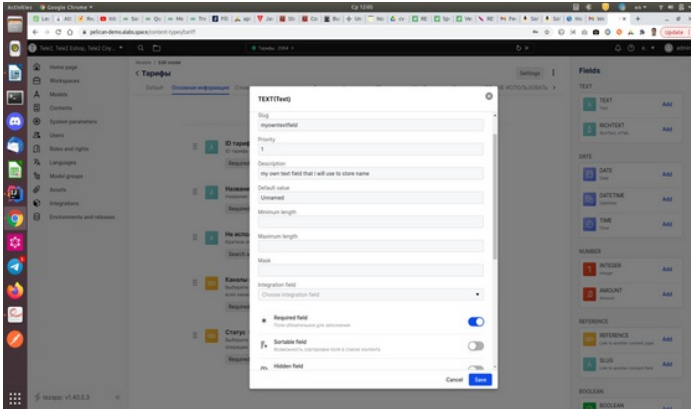
Research Services Company, Kazakhstan
Software Architect, Software Developer

Stack:

- Java,
- NodeJS,
- RabbitMQ,
- SOLR,
- PostgreSQL,
- Apache Storm,
- Spring,
- Hibernate,
- Aerospike,
- MongoDB,
- GIT,
- MicroServices,
- AWS,
- Apache Nutch,
- ExtJS,
- SVN.

Responsibilities:

- Code reviews for my team;
- Refactoring of code;
- Leading sprint activities;
- Building Architecture;
- Apache Storm integration(monolite data crawler);
- Scripts to automate backups.



No-code, low-code platform/Visual software development environments

January 2020 - May 2022

Description:

This project was created from scratch by me and one more frontend developer. For the past 2 years the team has grown to 15 people now. Initially I had the position of Software Architect, nowadays I am CTO of this product. I used to manage code reviews for my team, cross team management, architecture building, developed microservices using Java, research and development, recruiting, stuff training, participated on sales stages, made presentations and applied activity on international conferences, established partnerships, lead sprint activities, worked closely with business and owners.

History:

Our company initially had 2 different projects to finish at the same time with limited resources. But those projects had similar functionality as both intended to be used in the Telecom field. As I finished my previous project I was chosen to help with that situation. But where to get 20 developers and finish all projects in 6 months? So, we have decided to try a new approach, hCMS based architecture. As a result we have our own product, we have finished those projects(which are now easy to maintain) with just 4 developers.

About project:

While using traditional CMS architecture has been the standard, companies are now looking for increased flexibility and scalability provided by headless CMS solutions. A headless CMS is any

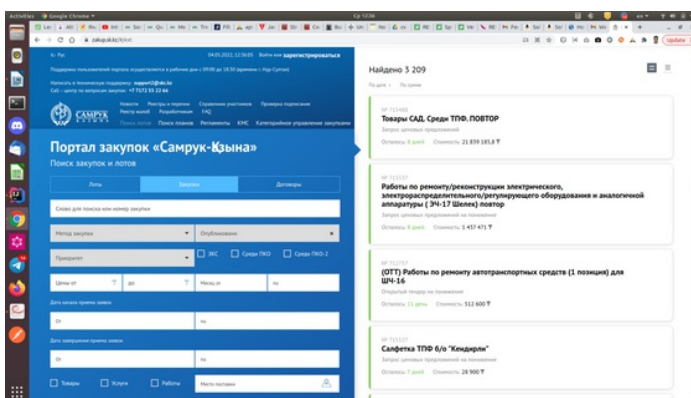
type of back-end content management system where the content repository “body” is separated or decoupled from the presentation layer “head.” Content that is housed in a headless CMS is delivered via APIs for seamless display across different devices. Headless architecture is a similar concept to a headless CMS, as it is a multichannel solution for effectively publishing dynamic content across a variety of platforms and devices. Content stored in a headless architecture is raw and unformatted, and its final presentation isn’t limited by a front-end system. So, we created a homegrown low-code/no-code platform - product and service. After 2 years of development it is used in 8 biggest corporations of Kazakhstan, decreases development time(time to market) by 50-60% and is involved in development of 12 projects. Our headless CMS integrated with Zeebe, so that users can create data models(including one to one, one to many, many to many relationships), fulfill content and manipulate it, after all data is available to be retrieved via generated GraphQL API. Data can be of types: double, integer, references, color, texts, rich texts(html), dates, boolean, documents. Also text fields can be marked as multilingual, so that field should be filled in many languages in order to solve localization problems. All data is distributed through development environments and can be migrated from one to one(from dev to prod). Data can be published with CRON and also migrated through other APIs into the system. All data is versioned. Creating and managing custom roles and user access rules is also possible. Asset galley stores digital data of different types: images, videos, audios, documents. All user activities are logged, stored and are auditable. Implemented webhooks and notifications. All data workflows and entity life cycles are declared through the system and can be seen and manipulated via the admin panel.

Technologies used:

Java 11, Spring Boot, Gradle, ReactJS, GraphQL, REST API, GraphQL API, MongoDB, PostgreSQL, Elasticsearch, MyBatis, Flyway, Redis, RabbitMQ, API Gateway, Apollo Gateway CI/CD, Docker, Kubernetes, Zeebe.

Technologies used:

Java, Spring, React, GraphQL, MongoDB, CI/CD, PostgreSQL, RabbitMQ, Rest API



Procurement system/Gov-tech
January 2019 - January 2020

Description:

This project was about the development of 4 new modules for the existing ecosystem. Was in the position of Software Architect. Used to manage code reviews for my team, cross team management, architecture building, developed microservices using Java, research and development, recruiting, stuff training, participated on sales stages, led sprint activities, worked closely with business and owners. There was a team of 5 people – 2 java devs, 1 Angular dev, 1 analyst and me as software architect.

History:

One of the first gov-tech projects in the company, so that was very important to make good relationships with them. Small prices but big opportunities.

Total profit: \$50 000. Collected a team of 9 people, including developers, analysts, managers. Made a deal to start 3 more long term new projects(currently developing). Economy for clients is around 1 billion USD yearly. 60 000 end users.

About project:

Supplier exchange, is the process of requisitioning, ordering and purchasing goods and services online. It is a business-to-business process. Unlike e-commerce, e-procurement utilizes a supplier's closed system and is only available to registered users. It was about automating business processes for procurements for all Quasi-Governmental Organizations - around 350 companies. 20000 suppliers registered to maintain the needs of 350 buyers. There were many types of procurement: one source, tendering, open tendering, closed tendering, multi round tendering and others. We developed an ESB, so that all buyers(350 companies) which are very big companies with their own IT ecosystems could integrate with our platform, and all data is transparent in order to make it all very honest, so API was used not only by the government but by other platforms. Also we developed a module that allows us to unify many procurements into one big, so that the amount of items or price is very big and it is optimal to buy those items from the factory at a big discount. Which means a big economy for the government. Platform allows users to plan procurement calendars, buy items, make auctions, sign contracts, everything is online.

Technologies used:

Java 8, Spring Boot, Gralde, AngularJS, REST API, MongoDB, PostgreSQL, ElasticSearch, JPA, Flyway, RabbitMQ, Jhipster, Gitlab, CI/CD, Docker.

Technologies used:

Java, Spring, Angular, MongoDB, Git, CI/CD, RabbitMQ, PostgreSQL, Rest API

Internet Banking platform/Fin-tech

July 2017 - December 2018

Description:

This project was a legacy from an old team that dropped off. During the period I worked there, the company has grown from 5 people to 90 people. Initially I had the position of Senior Software Developer, but ended with a Software Architect position. Used to manage code reviews for my team, cross team management, architecture building, developed microservices using Java, research and development, recruiting, stuff training, participated on sales stages, established partnerships, led sprint activities, worked closely with business and owners. Refactoring project from monolith to microservices.

History:

When I came to the company, there were 4 people on our side and an outsourcing company in Ukraine. I did not want to be involved in product development, but I took another one. After 4 months(3 days after New Year), the Ukraine team has dropped off - no communication, no calls, just closed their office and disappeared. From that moment I asked for carte blanche in order to finish current projects and close all current commitments. As this was an online banking platform company already had commitment with 3 banks to exploit the system. But there was no product, but good analysis, and just 2 months to finish it all.

About project:

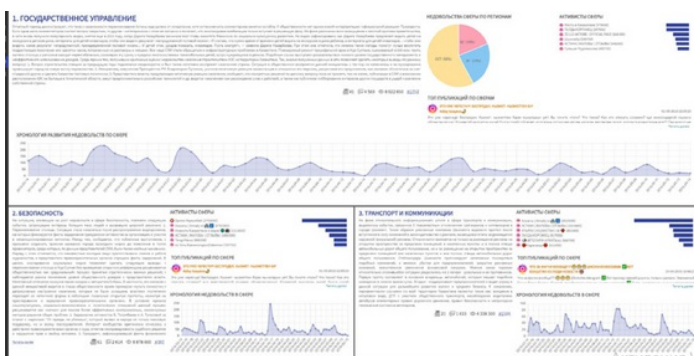
Internet banking platform(constructor) that allowed banks to easily maintain systems. Banking for both - individuals and legal entities, so that all can manage their accounts(opening, removing, blocking, freezing), transfer money from one account to another, from bank to bank, from bank to bank in another country and others. Individuals were enabled to pay their bills, utilities, pension, medical insurance and other payments through web and mobile applications. Plugin architecture allows easy integration with bank ESB in order to make some important operations such as opening accounts. There was customizable multi factor authentication and authorization.

Technologies used:

Java 8, Spring Boot, Gralde, ReactJS, REST API, MongoDB, PostgreSQL, QueryDSL, RabbitMQ, Jenkins CI/CD, Docker, Bitbucket, Redis.

Technologies used:

Java, Spring, React, Rest API, MongoDB, PostgreSQL, RabbitMQ, CI/CD



Web and social media data scraper/Social Media

June 2013 - July 2017

Description:

This project was absolute research and development. I have been involved in developing all 3 versions from scratch. In the first version I developed the frontend, in the second I developed the backend on Java. The 3rd version was fully architected by me and developed by a team of 20 people. Initially I had the position of Software Engineer, at the moment of completion of cooperation I was Software Architect. Used to manage code reviews for my team, cross team management, architecture building, developed microservices using Java, research and development, recruiting, stuff training, participated on sales stages, made presentations and applied activity on international conferences, led sprint activities, worked closely with business and owners.

History:

Our company was basically making gov-tech projects. The CEO decided to move further and go to the international market with service. Company was known for deploying a multitude of software solutions based on AI, but this was a new challenge, as there was no big data experience. There were 3 attempts to make it work, there was time the CEO decided to quit, fortunately we persuaded to not.

Total profit: \$4 mln USD. collected a team of 20 people, including developers, analysts, managers, QA, DevOps. 130 clients. Still working on a project.

About project:

International online service that provides round clock monitoring of news and mentions about the customer based on his/her keywords. It collects, extracts, and analyzes the right information in the media and social networks. Quickly monitors statements of customers, opinion leaders, government officials, competitors. This helps in time to see and react to negative messages or incorrectly filed information. Solved problems such as sentiment analysis, classterization, classification of big data.

Thus, customers can determine the degree of media interest in the events around his/her company. The results can be used to correct the information policy and the comparative analysis. System mostly used by PR agencies and marketers, but also used by individuals. System was developed using a micro services approach, providing safe and fast calculated analytics by means of SOLR. Analytics include sentiment analysis, calculated by neural network, relevancy etc. Project has customers around the world.

Technologies used:

Java 7, Maven,NodeJS, ExtJS,REST API, MongoDB, Aerospike, PostgreSQL, SOLR, PostgreSQL, JPA, Redis, RabbitMQ, Hadoop, Apache Storm, Bitbucket, Jenkins CI/CD, Docker, EC2, SQS, SNS, S3.

Technologies used:

Java, Node.js, Rest API, MongoDB, PostgreSQL, RabbitMQ, CI/CD

