

# CV

Name	Gleb Sinyavsky
Occupation	Backend engineer
Location	Kazan, Russia. Looking for relocation to Europe.
Age	27
Expectations	\$50000-\$70000
Email	<a href="mailto:zhulik.gleb@gmail.com">zhulik.gleb@gmail.com</a>
LinkedIn	<a href="#">Gleb Sinyavsky</a>

## Professional Experience

- 2011-2013 UZPS, Ufa, Russia - embedded linux developer

### *Accomplishments:*

- GUI and low-level integrations for IPTV STB of own manufacture
  - JS-C++ bridge for Qt's Web and QML engines
  - Side project - an Android tool for remote control of IPTV STB
  - YouTube, IVI and Twitter apps for IPTV STB
  - Z-Wave home automation client for IPTV STB
- 2013-2015 Racoons Group, Kazan - full stack developer

### *Accomplishments:*

- Integration between private enterprise Ruby application and Active Directory
  - ERP application based on [Bonita BPM](#) for architecture companies
  - Integrations with various payment gateways for many customers
- 2014-present RoadAR, Kazan - backend developer(part-time: 2-3 hours per week)

### *Accomplishments:*

- Moving server infrastructure from Azure to Digital Ocean
  - Administrating and monitoring of highly loaded Sidekiq(150-200k jobs per day)
  - DB denormalization and SQL optimization
  - [Mobile app](#) API and optimization
  - Administration interface based on AngularJS
  - Various backends for side-projects
  - WEB app for manual recognition and dataset collection for road signs and car licence plates based on AngularJS and React
  - Build system for car plate recognition project based on CMake
  - Building RPMs and self-contained bundles for car plate recognition system
- 2017-present Mechanizm, Kazan - backend developer

### *Accomplishments:*

- Microservice backend for [MrShoebbox](#)

# Education

- [USATU](#) - bachelor
- [UKSIVT](#) - technician

# Technologies

	I'm good at	I'm familiar
I want to keep working with	<ul style="list-style-type: none"><li>• Ruby/Rails/Grape and so on</li><li>• Go</li><li>• Linux</li><li>• Docker</li><li>• Ansible</li><li>• PostgreSQL</li><li>• PostGIS</li><li>• QGIS</li><li>• nginx</li><li>• git</li><li>• bash/zsh</li><li>• Redis</li><li>• Vagrant</li></ul>	Everything below
I want to improve	Everything above	<ul style="list-style-type: none"><li>• MongoDB</li><li>• Graphite</li><li>• Elixir/Erlang and functional programming</li><li>• Kafka</li><li>• RabbitMQ</li><li>• Java</li><li>• ML and DL</li><li>• Messaging bots for Telegram/Facebook</li><li>• Android programming</li><li>• Swagger</li></ul>
I don't want to work with	<ul style="list-style-type: none"><li>• C++</li><li>• Qt</li><li>• QML</li><li>• BPMN</li></ul>	<ul style="list-style-type: none"><li>• Python</li><li>• JavaScript</li></ul>

Skills	Open source and contributions
Strong OOP	<a href="#">margelet</a>
Strong ruby metaprogramming	<a href="#">pinball</a>
Strong refactoring	<a href="#">bubing</a>
Unit and integration testing, automatization	<a href="#">go-telegram-bot-api</a>
Linux administration	Multiple issues in various projects
CI/CD	
Microservices	
SQL optimization	
Crosscompilation	
DDD	
TDD	
DI	

## Additional info

I prefer to write unit and integration tests for almost everything, sometimes before writing the code. It's very helpful for refactoring and makes me more confident in my work. Also I prefer to use functional programming patterns even in OO languages: if something can be done with pure functions - I will do it in this way, but without fanaticism and breaking the rest architecture.

Ruby is a very magical language and allows developer to make one task in different ways, so I usually write my code in a very obvious manner, but when the task requires to write boilerplate code, I use metaprogramming. If something can be done without metaprogramming and other magic it should be done without metaprogramming and magic. I prefer explicit interfaces in my code, it simplifies reading and understanding.

I like Ruby because it helps me to write very expressive code very fast. Ruby is a good choice for writing tricky business logic if it shouldn't be very fast. Code can be tested in a very easy way and programmer can focus on business requirements instead of platform limitations.

I like Go because it is the most nonmagical language I've ever known, it allows to write very transparent and understandable concurrent code with high performance and without additional runtime. It's a good choice for very fast and small concurrent programs for networks and tasks that require low latency.

I don't like Python and JS because of API inconsistency, I can't enter into the flow state when I work with them.

## Hobbies

- Music (spent 4 years playing drums)
- Bicycle
- Mountain Skiing
- Video games
- SF books and movies