

# AlexWayfer's Site

Hello. My name is Alexander (Aleksandr) Popov.

I'm 25 years old, but I think person experience and opinions are more important than age.

I'm from Moscow (Russia), but I can relocate and make the same things in a lot of places.

## Education

I've finished a school in a class with advanced study of physics and mathematics with excellent grades.

Then I went to the university for "Programming Engineering" and learned some interesting things like PHP, JavaScript, MySQL.

I've not finished it because I don't like such disciplines like philosophy, history, quantum physics and computer chipsets from the '90s: they were required and took around 80% of time.

I think, I'm ready to study "Computer Science" in a good university (probably in another country).

## English language

My level of English around Intermediate. I do my best and I like it more than Russian, probably because of its laconism and simplicity.

And I want to learn it more, and I think it's easier when living (or communicating) with English speakers.

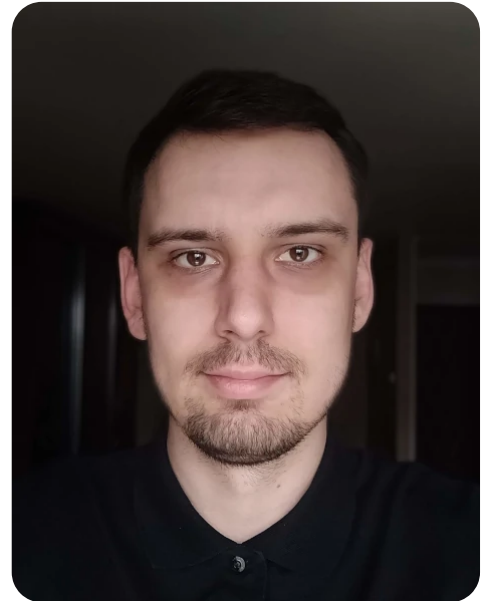
I use my gadgets (smartphone, PC, etc.) and all services (sites) in English, also I search information almost every time in English, and I use English for open-source discussions and all commits.

## Software development

### General

I love [Ruby Language](#) [↗](#).

I like [snake\\_case](#) [↗](#) and [OOP](#) [↗](#).



But I can write a code in any language. My experience includes C, C++, Pascal, Delphi, Go, PHP, Java, JavaScript. Especially JavaScript.


## Specialization

I'm Software Engineer. I was leading a team with up to 10 members. I can design application architecture. Also usually I communicate with other teams (like Mobile Development) and departments (like Marketing).

## Skills levels

[CodersRank profile](#)  .

## Open-source projects and activity

I love open-source too much, here is my [GitHub profile](#)  .

## Projects (experience)

- [Simple Player](#) 

**2011–2012**

### My personal project

Music player for Windows, plays local files or audios from VK.com. My main project on Delphi, I've learned a lot of things and was trying to make a good design (via FireMonkey). It was hard to make a good (in my taste) visualizer.

Technologies:

- Delphi (FireMonkey)
- VK API

- [Soundly](#) 

**2013–2014**

### My personal project

Reincarnation of Simple Player as web service (site). Plays local files (selected via browser) and integration with VK (again). Without visualizer, sadly. But it worked with VK Audios faster (especially search, global too), than VK itself. Until VK has closed their Audio API for

monetization. Single file of plain JavaScript with about 1500 LOC: I didn't think about architecture and third-party tools (libraries) at this time.

Technologies:

- JavaScript
- VK API

## • **BCleaner**

**2013**

**Single software developer**

System for detecting bots among Twitter followers.

In these times there were many bots in Twitter and popular people cheated followers count. But then declaring that you have no bots among your followers and you're honest became fashionable. So, there was a request to make an automated tool (service) for detecting bots among your followers and ban them (it was the force unfollow in this time, as I remember). Then you could declare that you're "clean" and "honest" too.

I was trying to make such system for money. Bots recognition algorithm was pretty good, but I have not finalized this project because I was fired up for fail in another parallel non-related project in the same company.

Technologies:

- PHP
- Twitter API

## • **[RDI](#)**

**2014–2015**

**Front-end developer**

Site of the development company (in Moscow region). My only experience of changing layout (HTML, CSS, PNG) by just a single JPEG file with target design.

Technologies:

- PHP
- 1C-Bitrix
- GIMP
- Basecamp

## • Icons of the World

**2014–2015**

**CTO, team of 2–4 developers**

Presentation site of the same named books. Home page contained almost full-screen Google Maps with custom markers, pop-ups and clusters. It was hard to make custom optimizations of requests to database bypassing WordPress engine.

Technologies:

- PHP
- WordPress
- MySQL
- CSS
- JavaScript
- Google Maps API
- Basecamp

## • [KEY Group](#)

**2014–2015**

**CTO, team of 2–3 developers and out-source**

It was a weird experience of making the layout with diagonal sections. Adaptive! I've made it via CSS `transform` finally, although there were many different tries with other approaches. Also it was my practice of making SVG icons via font characters. I've learned a lot of CSS with this project.

Technologies:

- PHP
- MySQL
- CSS
- JavaScript
- Fontello
- BitBucket
- Basecamp



## • GetWiFi

**2015**

## CTO, team of 2–4 developers

Dashboard for Wi-Fi captive portals among cafes, hotels, etc.

At that time I knew PHP at high level and saw its disadvantages for me. So, I wanted to learn any new programming language with more consistency and better potential for project files structure. Finally I was choosing between Python and Ruby, and I like (more consistent) OOP approach of Ruby, `snake_case` and its philosophy.

This project was a first serious try to make it with Ruby. I didn't like Ruby on Rails, mostly because of its approaches and "magic", so I've developed this project via [Sinatra](#) , but was a big mistake because of controllers architecture and it became the impulse to create [Flame framework](#) .

Although there was nice (in my opinion) horizontal-nested simple UI, and pretty good database architecture.

Technologies:

- MikroTik firmware and configs
- Wi-Fi Captive Portal
- Ruby
- Sinatra
- MySQL
- DataMapper (abandoned ORM)
- JavaScript
- BitBucket
- Basecamp

## • [GetTransfer](#)

**2015–2017**

## CTO, team of 2–10 developers, including remote

People transfers aggregator, between passengers and carriers. It's pretty popular and successful.

The first project written via [Flame](#) .

There was a lot of work with:

- SCSS architecture;
- JavaScript plugins and custom code;
- database architecture and optimizations;

This is a snapshot of the [alexwayfer.name](https://alexwayfer.name) dated August 24, 2020.

- form data validations;
- client-side and server-side tight integration with Google Maps API;
- integrations with a lot of payment systems, like PayPal or Platron;
- translations via Crowdin and Gengo;
- currencies exchanges;
- timezones and time convertations;
- comfortable dashboards for different types of users, from managers to client;
- aggregated data exports in different formats;
- forecasts algorithms for prices.

In all time of working on this project I was leading a team of up to 10 people, they all were different and many of them were cool. I've got a lot of experience in leading and talking with people, especially reviewing their code. 🙄 Now I'm more calm and respectful.

Honestly, I don't like what project became after my leave.

Technologies:

- Ruby
- Flame
- PostgreSQL
- Sequel
- ERB
- JavaScript
- SCSS
- Fontello
- Google Maps API
- GeoNames API
- PayPal API
- Platron API
- Crowdin
- Money gem
- Sentry
- GitLab
- AWS
- DigitalOcean
- Scaleway
- Basecamp

• [GetTransport](#) 

## 2017-2020

### CTO, team of 2–5 developers, including remote

Cargo transportation aggregator, between shippers and carriers. Another project written via [Flame](#) ↗ .

I've tried to make this project taking into account my architecture errors of GetTransfer. So, it's using a next version of Flame, 5, with a lot of improvements. Also we have started to use service objects instead of fat controllers and models. Then I service objects turned into [Formalism](#) ↗ forms.

This project was pretty similar to GetTransfer, so I was working with similar technologies, sometimes with a better or cleaner approach.

Technologies:

- Ruby
- Flame
- PostgreSQL
- Formalism
- ERB
- JavaScript
- SCSS
- IcoMoon
- Google Maps API
- GeoNames API
- PayPal API
- Platron API
- Crowdin
- Money gem
- Sentry
- GitLab
- Hetzner
- Basecamp

## • Stream Assistant

### 2018-2020

#### My personal project

Bot for assisting broadcaster and moderators in stream chats. It answers to frequently asked questions. Works via regular expressions. Right now only in Russian, but other languages in plans.

There is still no site for it, sadly, because of [Puma's issue](#) .

I've integrated it with YouTube at first, via [official Google's gem](#) , then I've integrated it with Twitch via gem for API and another gem for chat. They're pretty abandoned, there are PRs for them:

- [PR to twitch-rb](#)
- [PR to twitch-chat](#)

Also I've created a separated gem for authentication: [twitch\\_oauth2](#) .

Also it has integrations with third-party services, like Steam, I've created [a PR to the gem](#) with improvements.

For time difference I was trying many gems and chose [moments](#) , but [I've made some improvements for it](#) (it looks maintained, PRs were merged).

Technologies:

- Ruby
- PostgreSQL
- YouTube API
- Twitch API
- Twitch IRC
- Steam API
- GitLab
- Scaleway
- Hetzner

## • [Stream Timer](#)

**2020**

### **My personal project**

Simple application with configuration window (page) and a page with configured timer for streaming application, like [OBS](#) .

Useful for scenes of AFK (stream pauses), just configurable time countdown.

Currently it built only via Vue.js and static, but I have plans to make a complete web-application via Flame, database (configuration saving) and web-sockets. But there is still [the issue of Puma](#) .

Technologies:

- JavaScript
- Vue.js



- GitHub

## • Who2Follow

**2020**

### **My personal project**

A try to create my own algorithm of follows/friends/etc. suggestions. Platform-independent, but Twitch and Twitter are first in the plan (just because they're more important for me).

Core (algorithm in a Ruby gem) and CLI tool are ready, but there is no site version still because of [the Puma's issue](#) [↗](#).

Technologies:

- Ruby
- Clamp
- Twitch API
- GitLab



## Tastes


- Beautiful and laconic.
- Simple and understandable.
- Flexible and configurable.
- Fast and smooth.
- Consistent and rational.
- Reliable and maintainable.

## Contacts

- [Telegram](#) [↗](#)  
Direct messaging.
- [Email](#) [↗](#)  
Another (more official) direct messaging.
- [Twitter](#) [↗](#)  
Personal (micro) blog.

This is a snapshot of the [alexwayfer.name](https://alexwayfer.name) dated August 24, 2020.

- [Twitch](#)   
Programming and games streams.
- [Last.fm](#)   
What music I'm listening.

Compiled in August 2020. [Source code](#)  .