

# Bartosz P.

## Software Developer

✉ [ritu@ritu.dev](mailto:ritu@ritu.dev)

### WORK EXPERIENCE

---

#### ▼ Co-founder & Software Developer | Notify+

December 2016 - Present

<https://notifyplus.io>

Sole developer at Notify+, the original notification service made for the enthusiasts of limited-edition sneakers (e.g. Yeezy), streetwear apparel (e.g. Supreme) and other difficult to acquire collectibles. My Python codebase is 24,508 lines of code, not including config files and the deprecated code.

- Web scraping nearly 350 online stores
- Filtered notifications are quickly posted via Slack ([example](#)), Discord, Twitter and SMS (Twilio/Bandwidth) API when desirable products get released or get restocked
- All networking and parsing done from scratch without using any scraping frameworks
- Steady 2500 HTTP requests per second as shown in this [Datadog chart](#) makes the notification service very fast
- Developed my own networking library (listed in Projects) because there is no other Python library with HTTP/2 support that isn't currently work in progress
- Administrate 2 bare metal dedicated servers with 64 cores and 256GB RAM combined, along with Linode, Vultr and other instances for the website, databases etc.
- Help manage 250-500 users who pay the \$15-\$40 subscription fee

`python` `javascript` `redis` `mariadb` `mongodb` `zeromq` `squid`

#### ▼ Software Developer | Agentboost

Made an application that extracted real estate agents from Zillow to then automatically build a WordPress website with all the agent's info (all listed/sold properties etc.) and advertise it for sale to the agent.

Did the same for the sister company advertising personal websites to LinkedIn users by scraping LinkedIn, automatically creating a personal website and offering it as a subscription service.

- Created a lightweight WordPress Multisite with redis caching
- Circumvented the bot protection of Zillow and LinkedIn
- Created software that recorded (using ffmpeg and Windows macros) personalized videos that looked as if human was controlling the mouse and showcasing the website, with a pre-recorded voiceover. Videos were then embedded on personalized landing pages

`python` `javascript` `php` `redis` `mariadb` `nginx` `ffmpeg` `imagemagick`

#### ▼ Other Experience

- Created an application that automatically adjusted bids for thousands of keywords on PPC and PPV advertising campaigns
- Created a DMV reservation system for a NYC driving school
- Freelancing & Consulting: Completed dozens of different projects and earned a "Top Rated" status on one platform
- SEO: Ranked a page on top of Google for a seasonal sports event keyword with around 12M exact match searches according to the Keyword Planner

- ▼ **CurlRequests: HTTP/2 Curl Requests for Humans** 2019  
<https://gitlab.com/ritudev/curl-requests>
- CurlRequests is an open-source Python HTTP library that uses PycURL (a high performance interface to the C-based libcurl) while remaining user friendly and easy to use like the Requests library.
- CurlRequests implements a number of methods from Requests and in many scenarios can be considered a high-performance drop-in replacement for Requests.
- The main motivation behind the making of CurlRequests was the need for multiplexing HTTP/2 requests which is what curl is great for, but what is difficult to implement in PycURL.
- As of August 2019, there are is no other Python library with HTTP/2 support that isn't currently work in progress.
- Significantly reduced the amount of open TCP connections in Notify+ multithreading thanks to multiplexing connections with HTTP/2
  - Reusing the interfaces of the Requests library in CurlRequests has made it a 100% drop-in replacement in Notify+ and my other projects with no code changes
- python
- ▼ **reCAPTCHA solving API** 2019
- Created a farm of very quick reCAPTCHA solving instances with an API.
- The API solves reCAPTCHA for any website, on any URL. It does so by passing the job to the RabbitMQ message queue which then passes it to a non-busy instance for solving which then returns a successful reCAPTCHA answer token back to the client once the CAPTCHA is solved.
- This is the same process and the same result as if a human user clicked on the "I'm not a robot" except, ironically, it is a robot. It takes less than 3 seconds since the API request is submitted to complete the challenge and send back the success token.
- I modified and compiled my own version of Chromium to make the CAPTCHA solve as quick as possible and undetectable.
- Disclaimer: it's meant to be a proof of concept, it's not public, it's not sold and it's not used for commercial purposes.
- python javascript flask rabbitmq
- ▼ **Tumblr Archive** 2018
- On December 3, 2018, Tumblr announced that a full NSFW ban would go into effect on December 17.
- Under the 2-week deadline, I made a backup of nearly all Tumblr blogs flagged as NSFW by making a local copy of all API responses. It amounted to ~10TB worth of uncompressed text.
- Did the same for some other sites, such archiving the IMDB message boards before they got removed.
- python mongodb
- ▼ **Scanr** 2016
- Created Scanr, a perceptual image recognition Reddit bot used for finding reposts of the same image/gif/webm/mp4 across Reddit, as well for finding similar images.
- Scanr uses perceptual image hashing developed according to specifications outlined here:  
<http://www.hackerfactor.com/blog/?/archives/432-Looks-Like-It.html>  
<http://www.hackerfactor.com/blog/?/archives/529-Kind-of-Like-That.html>
- First, I made a web scraper for all the shared direct links to the media files and also for the platforms like Imgur, TinyPic, ImageShack, Gfycat, Instagram, Photobucket, Pinterest, DeviantArt or Flickr.

Then, I used the Reddit's BigQuery SQL database to find every single image/gif/webm/mp4 that has been ever been posted on Reddit, as well as short YouTube videos.

Web scraping and downloading took ~30TB of bandwidth and I computed a 16, 64 and 128 bit dHash and pHash using a hashing library and NumPy. For media files that had more than one frame (gif, webm, mp4), the middle frame was used for computing the hash.

Scanr also scanned all new Reddit submissions in realtime.

Scanr computed a Hamming distance of two 64 bit pHash hashes (hash of the source file and hash of the file to compare) so it found duplicates of the same media even if it had been slightly altered, which is something that wouldn't be possible using hashes such as md5.

Due to the large scale of the project, I experimented with Elasticsearch and MariaDB for storing the hashes and the Reddit data.

Scanr worked like TinEye or Google's reverse image search. It was used to find:

- Reposts of the same image/gif/webm/mp4
- Versions of image with better quality or higher resolution
- Colored versions of black and white pictures ([example](#))
- Non-watermarked version of the same image
- Edits of the same meme macro or comic panel e.g. with different speech bubbles
- Discover subreddits that share the same or similar content

`python` `numpy` `elasticsearch` `mariadb` `imagemagick` `ffmpeg`

#### ▼ Chrome extension

2016

Created Anti-scamaz, an open-source Chrome extension for Twitch streamers used for detecting "fake" .jpg images posted on Imgur which turn out to be animated instead of static, sometimes causing streamers to get banned by unwillingly showing inappropriate content to the viewers when an innocent looking .jpg image suddenly starts to animate and turns into, for example, a NSFW gif.

Whenever a user opens a .jpg Imgur link, the Chrome extension submits the same link it to the server for analysis, the server then quickly downloads it and checks if it's a "real" .jpg image or if it's a .gif in disguise or any other animation sneakily renamed to .jpg to trick people.

The backend API uses redis to cache the results so that when users open the same link (or if the same user opens it again), the cached result is immediately sent back to the extension.

`python` `javascript` `flask` `redis`

#### ▼ EVE Online Trading Application

2014

Automatically traded in the EVE Online MMORPG video game making an equivalent of \$20/day in the in-game currency.

Created a Google Sheets spreadsheet that automatically updated itself with the prices of thousands of items using the public EVE Online API's and was then used for automatic trading on the in-game market.

#### ▼ Cryptocurrency Mining Pool

2014

Setup, optimized and operated a Dogecoin cryptocurrency mining pool with a friend for a small number of people.

### INTERNSHIP

---

#### ▼ Asseco

<https://asseco.com>

Deployed hundreds of corporate employees with laptops and did other tasks (Excel etc.) at Asseco, one of the largest corporations in the technology sector.

### EDUCATION

---

Computer Science Dropout | Polish-Japanese Academy of Information Technology

2010 - 2010