Bartosz Sławecki

Python Developer

✉ bartoszpiotrslawecki@gmail.com 📞 +48 535 950 567 📍 Wrocław, Poland ⌓ bswck
🎧 bswck in bswck 👤 he/him

PROFILE

Passionate expert with deep knowledge of Python (3 years of experience) and code quality assurance. Life-long learner inspired by inventive approaches to solving problems. Appreciated teacher, endorsed for his ability to convey specialist knowledge to beginners. Active, motivated and creative open-source software contributor. Supported and trusted by some prominent figures of the IT industry, including <u>Poetry</u> 🔗 .

EXPERTISE

Domains: Automation and scripting, Web services, Collaboration, Education, Code Quality Assurance, Developer tools, Testing
Python: 3.x, FastAPI, Flask, Django, SQLAlchemy, Pydantic
Containerization: Docker, Kubernetes (basics)
Cloud: AWS (basics), GCP (basics)
CI/CD: GitHub Actions
Databases: SQL, PostgreSQL, SQLite, Redis, MongoDB
Web: HTML, CSS, Ajax, JSON, jQuery, XML, XPath
Other Languages: Bash, JavaScript, TypeScript, Dart, SQL
Testing: Unit testing, Coverage measurement
Operating Systems: Linux, Windows, FreeBSD

PROFESSIONAL EXPERIENCE

Freelance Python Developer   03/2023 – present
Artur Byczkowski MATEMATYKA
- Began a long-lasting cooperation with a popular YouTuber @apocomitamatma 🔗 with 100k+ subscribers
- Designed, created, and estimated costs & delivery time of a full-fledged Discord bot moderating an online community of over 2.7k+ people
- Created and deployed a Flask website (within <u>Phusion Passenger</u> 🔗) for hosting files and short URLs associated with their brand.
- Designed and estimated costs of a modern website, currently being built with FastAPI and Next.js
- Recruited the moderation team for their online community and taught them the required technical skills
- Created numerous YouTube API integrations, including a livestream-chat bot
- Implemented and deployed a subscriber adhering to <u>the Google PubSubHubbub 4.0 protocol</u> 🔗
- Developed a tool for bulk substitutions with regular expressions in YouTube video titles & descriptions
- Automated the creation of YouTube search tags for over 500 videos using NLP technologies, including nltk, TensorFlow, Morfeusz2, ConcraftPL and TermoPL

Technical Trainer (Python Bootcamp)   12/2023 – present
infoShare Academy
- Illustrated and led 4 presentations on the Python Object-Oriented Paradigm, including the basics of: objects, classes, types, classes vs types (gradual typing, <u>PEP 483</u> 🔗), attributes, methods, descriptors (methods, properties, class methods, static methods, custom descriptors —referenced <u>SQLAlchemy hybrid attributes</u> 🔗), data model, inheritance (C3 linearization and MRO), duck typing & protocols, type hinting (including relevant PEPs, e.g. <u>585</u> 🔗 , <u>604</u> 🔗 , <u>612</u> 🔗 , <u>613</u> 🔗 , <u>695</u> 🔗), abstract classes and the collections.abc module

- Conducted Q&A sessions regarding the aforementioned presentations, providing real-life analogies of programming concepts and live coding explanations

Multimedia Inserts Developer   08/2023 – 11/2023
TVP (Polish Television)
- Created and deployed an offline in-browser fake OS PWA built with OS.js
- Implemented and installed a fake phone call application in Flutter. The application used the phone's proximity sensor to turn off the screen when the phone was held to the ear
- Designed and created 6 different static websites with Bootstrap
- Estimated delivery time & costs of the project

PROJECTS

skeleton (author) ☍   11/2023 – present
Implemented a Python project scaffolding with various SCM and CI/CD automations.
Inspired by <u>the skeleton project of Jason R. Coombs</u> ☍, to which I also contributed.
- Created several Jinja templates (with custom extensions and filters), including:
  - Useful GitHub Actions workflows for testing (pytest), type checking (mypy), measuring and deploying coverage reports (coverage.py, smokeshow), publishing to PyPI via <u>Trusted Publishing</u> ☍
  - Pre-commit and dependabot configurations that ensure supply chain security and Poetry package lock integrity with the project configuration and dependency specification
  - Default tooling settings & dependency specification used for ensuring any skeleton downstream codebase follows the best practices. Tooling included Ruff, mypy, towncrier and pre-commit
- Automated the build & release process with a template script and a GHA workflow
- Included handy ReadTheDocs and MkDocs configurations with automatic module discovery for generating API references from pre-existing docstrings with mkdocstrings
- Developed a Bash script Jinja template (linted with <u>Shellcheck</u> ☍) that allowed mechanized patches in skeleton downstream projects with features such as:
  - Automated pre-commit installation
  - Automated GHA environment provisioning
- Contributed <u>several new ideas</u> ☍ to <u>copier</u> ☍, the engine behind skeleton
- Adopted the mentioned skeleton tool to automate CI/CD management in numerous projects, including configzen, autorefine, proxyvars, runtime_generics, class_singledispatch and many more

configzen (author) ☍   04/2023 – present
Invented and created a library to validate and edit configurations with Pydantic. Available on PyPI as `configzen`.
- Designed a library for reading and editing configurations within Pydantic models
- Created built-in drivers for processing of JSON, Plist, YAML and TOML (last two in a style-preserving manner)
- Introduced a layer of abstraction allowing to operate on file streams both synchronously and asynchronously

cleo (contributor, maintainer since January 2024) ☍   08/2023 – present
Invited to maintain a CLI framework that powers Poetry—a famous and opinionated project dependency manager, build backend and much more.
- <u>Co-refactored the entire codebase</u> ☍
- Recently assigned to create a new CLI parser that will power the Poetry CLI

autorefine (author) ☍   12/2023 – present
NOT IMPLEMENTED: Invented a tool for automating the refactoring process at scale.
- Wrote down <u>the minimal requirements for the tool</u> ☍

- [Received positive feedback from a PyCQA member](#) 🔗

pip-run (contributor) and mempip (author) 🔗
12/2023 – present
- Contributed a few enhancements and ideas
- Assigned to introduce in-memory installation feature
- [Wrote down in-memory installation feature ideas for collaborators to comment on](#) 🔗 . The write-up included numerous important thoughts on usability, feature stability and unsolved problems

proxyvars (author) 🔗
07/2023 – present
Proxy objects inspired by Werkzeug local proxies. Available on PyPI as `proxyvars`.
- Invented and implemented the underlying mechanism and tests
- Adopted the library in a Discord bot for the aforementioned YouTuber

runtime_generics (author) 🔗
10/2023 – present
Invented and implemented a library that facilitates reusing type arguments for generic types at runtime. Available on PyPI as `runtime-generics`.
- Adopted the mentioned skeleton tool to automate the project CI/CD management
- Implemented and documented the underlying idea, wrote tests
- Practiced TDD for covariance and contravariance runtime checks
- Adopted the library in the previously mentioned project `configzen`

gd.py (maintainer) 🔗
12/2023 – present
- Contributed several CI/CD improvements
- Helped with developing a new major version

class_singledispatch (author, inspired by someone else's idea) 🔗
12/2023
Available on PyPI as `class-singledispatch`.
- Implemented `functools.singledispatch()` alternative for arguments that are classes
- Designed an exhaustive test suite resulting in 100% coverage

EDUCATION

Computer Science (Cloud Applications and Systems Engineer)
2023 – 2027
DSW University of Lower Silesia
Wrocław, Poland

LANGUAGES

English
B2/C1

Polish
Native

CREDENTIALS

Invitation to the Finals of the National Programming Competition "Gigathon" 🔗
15/12/2023
Giganci Programowania Sp. z o.o.
Qualified for the finals for remarkably high performance:
- 100% points for the first stage of the competition,
- 100% points for the first phase of the second stage,
- Qualified for the finals after submitting the projects in the other phase of the second stage.

The finals are scheduled to take place on the 3rd of February, 2024.

CONSENT