

James Cooke

Senior Software Engineer

✉ hi@jamescooke.info

🌐 jamescooke.info

🐙 [jamescooke](https://github.com/jamescooke)

Brighton UK, UTC/BST

As a software engineer with 17 years' experience, I have worked on web-related development with Python for the last 8 years. Software behaviours, especially related to consistency and reliability, are the focus of my programming practice, which employs a blend of test driven development, unit-testing and integrated testing.

I am looking to work with a company that cares about software quality; is iterating its products and development processes over a long-term life cycle; and delivers measurable value to customers.

Primary Technical Skills

API design and documentation, Bash, CI design and implementation (CircleCI, GitLab, Jenkins, Travis), GNU/Linux, Git, Golang, **Python**, **Test Driven Development** (including teaching and tooling), Vim.

Recent Experience

A full CV with all previous work is available on request.

Senior Software Engineer

Lucid Holdings LLC

Mar 2019 → Current

Using: Docker, Golang, Java 8, Jenkins CI, Jira, OpenAPI v3

(9 months)

Lucid's Marketplace provides survey respondents to market research companies and generated over \$100M in revenue this year. Primarily my work at Lucid has been within the team redesigning and building a new Marketplace to use Golang microservices.

My primary responsibilities in this team have been:

- To lead the API design of three new services, described in OpenAPI v3 documents. The use of OpenAPI to describe all Lucid services is a new policy and I worked: to design a linting process for OpenAPI documents wired into each project's CI; to train other Lucid dev team members in OpenAPI and our processes; and to problem solve specific OpenAPI issues and design questions, especially related to how OpenAPI documents were integrated into the SDET team's testing processes.
- To develop in Golang one of these new services, along with an abstract DSL to describe the logic between answers to adjust subsequent question flow. I had responsibility for this language's definition and developed its Golang library providing syntax validation.
- To build an integration test suite in Java for another of these new services. The suite compiles a Java client API based on service's OpenAPI document, which it then exercises to test the service.

Lead Developer

Funding Options Limited

Dec 2018 → Feb 2019

Using: Django Rest Framework, Postgres, Test Driven Development

(3 months)

Funding Options scans the business loan market to find client businesses the best options. I lead the team of frontend and backend engineers of various levels of experience. My responsibilities were:

- To design and implement with the team a real-time loan approvals flow for self-service customers.
- To guide continued upgrades to the existing codebase by dividing "the monolith" into microservices.
- To train and coach team members in Test Driven Development, Django best practices and to work with the head of tech to improve agile rituals.

Backend Developer.....

Construct Technology

Oct 2017 → Nov 2018

Using: Ansible, Couchbase Sync Gateway, Django Rest Framework, Grafana, Swift (1 year, 1 month)

Construct was an iOS project management app for the construction industry which was acquired at the end of 2018. As the sole backend developer, I focused on improving the stability and predictability of the API and hosted services, while continuing to develop functionality. My first task as a permanent employee was to build out a Grafana / Graphite / statsd stack to collect data from Construct's API servers, queue workers and Couchbase nodes to help us better identify and manage system performance and bottlenecks.

Previously, from October 2014 to September 2017, I worked with Construct as a contractor to build their original API server, push notifications and Dropbox and Couchbase integrations.

Python Development Contractor.....

Coding Cooke Ltd

Dec 2013 → Sep 2017

(3 years, 10 months)

At Coding Cooke I was able to boost my experience by working with a wide variety of clients in different technical environments. Coding Cooke's USP was high quality code covered by tests that could be maintained by the client's team after the contract end. Coding Cooke had thirteen clients, three are below.

Glint Pay

Apr 2017 → Jun 2017

Using: Cucumber, Django Rest Framework, Docker Compose, Docker, MySQL (3 months)

Glint provided a gold-backed prepaid Mastercard and mobile app using a microservice architecture comprising approximately fifty containers. At Glint I developed and integrated new admin service containers, backfilled behaviour-driven tests against existing transaction services and refactored existing Django code.

Department for International Trade

Jun 2016 → Aug 2016

Using: Django, MS Azure, MS Dynamics CRM 2011 (3 months)

The DIT were developing an internal tool called Data Hub to enable them and their partners to better understand, manage and record their relationships with businesses. I worked on building an integration test suite for the Python Microsoft Dynamics 2011 library, as well as conducting some research into the existing data and technical needs of the project.

Huq Industries

Nov 2015 → Apr 2016

Using: Ansible, Django Rest Framework, PostgreSQL (6 months)

Huq is an analytics provider for mobile app developers. As the sole backend developer on this project, I built out the collection and resolution pipeline and completed the reporting API to make the product ready for launch. As part of that work, I backfilled end to end tests and eliminated existing critical bugs.

Education

Imperial College, London

Oct 1998 → Jun 2002

MEng Computing (hons)

Selected Personal Projects

Flake8 AAA

Feb 2018 → Current

github.com/jamescooke/flake8-aaa (1 year, 10 months)

A linter for Python tests. This Flake8 plugin ensures that tests are formatted to the Arrange Act Assert pattern to help make tests simpler, more consistent and easier to understand.

Factory Djoy

May 2016 → Current

github.com/jamescooke/factory_djoy (3 years, 8 months)

Ensures that Django model instances created by Factory Boy are always validated before saving to the database for safer and more reliable testing.