



# JOE TILSED

SENIOR SOFTWARE ENGINEER

## PROFILE

Helping lead the way with Python API, Software Development Lifecycle Standards, Cloud Adoption, Automation, Continuous Integration and Delivery, Test Driven Development and other Modern Engineering Practices, just to name a few.

## CONTACT

+44 (0) 770 407 0157

JOE@TILSED.COM

JOETILSED.COM

## SKILLS

SOFTWARE DEVELOPMENT LIFECYCLE



PYTHON REST APIs



FULL-STACK DEVELOPMENT



CLOUD COMPUTING



## CAREER HIGHLIGHTS

### FIRST-CLASS HONOURS BACHELOR'S DEGREE

October 2020

I achieved 100% in two of my modules, along with 8 achieving 80% or above.

Producing a standardized Python logging package brought the opportunity for enhanced querying and reporting of telemetry which was achieved by using a JSON format output.

Having a better understanding of the end user helped application teams reduce unnecessary logging, allowing for a higher value-add, at a reduced cost to resources and time. I attained a staggering 96% for this report, which is now being used as the university's "gold-standard" example.

### IBM TECHU PRAGUE

October 2019

Myself and a few others in my team were invited and flown out to attend IBM's TechU hosted in Prague. With world-class techies giving us detailed sessions on topics such as Artificial Intelligence, Machine Learning, Big Data, Cloud Architecture and many more.

All of the speakers were and are best in class, world experts, with the likes of IBM's Wolfgang Bosch who is the Business Development Executive for Watson and AI Innovation. The sessions were very personal small audiences, sub 20 people, so we could really dive deep into the topics with specific questions and answers, with many 1:1 follow up sessions later into the evenings.

### MY AUTHENTICATION PACKAGE

June - July 2019

The firm was rolling out a new SSO (Single Sign On), password-less authentication system, which meant application developers needed to migrate their current authentication methods over to this new system. This obviously was no small task, and to correctly verify, authenticate and validate users using the new system was vastly different to the previous access-control tool that was in place. I saw that with countless applications needing to make this change (near 1,500 lines of code per application), it would cause issues.

So I created a Python package which is an intermediary between the application and the authentication tool. This standardised the authentication process for application teams, which only took them 5 lines of code while using my package, increasing security with above industry standard authentication and decreasing technical debt. It quickly became the firm's strategic standard for Python authentication to use my package.

## FIREBASE PYTHON PACKAGE

FEBRUARY 2019

I wrote the Python package 'firebase' which is a Python interface to the Google's NoSQL Firebase REST API offering.

Currently (as of March 2021) the package is downloaded over 30,000 times a month, with a total of over a quarter million downloads. The package converts a user's NoSQL database into a Python readable object, with simple data manipulation, it can also handle complex queries such as ordering (even though its inherently a nonstructured database schema).

## OS BUILD AUTOMATION

2018 - 2019

There was a business need to improve the time to market on the provisioning of new operating systems for an accelerated computing offering. At the time the system that was in place could take weeks and in some cases months to complete, with a success rate of almost zero since every order would require manual intervention.

I wrote a suite of APIs which communicated with our estate management nodes, the firm's compute ordering tool and the firm's build pipeline orchestration tool. This new solution has been operational since, with almost daily production releases, with new features released to our clients with our continuous innovation, improving our customer's user-experience. The provisioning times went down from weeks/months to just hours, with it being fully automated end-to-end. Removing the need for manual intervention as a negligible number of builds require attention, and those that do are automatically notified to the team without the client even knowing there was an issue.

## JPMC'S FIRST PYTHON END-TO-END SDLC PIPELINE

October 2018

I had noticed that a fully automated Jenkins pipeline for Java was implemented internally, however there was no fully complete end-to-end Python solution. I was writing a suite of Python REST APIs at the time, releasing new features into production on a some-what regular basis. However, the major blocker was from lengthy wait times from other teams testing, scanning and approval of my code. I took the opportunity to write the firm's first fully end-to-end Python SDLC pipeline, it was written in Groovy and ran on Jenkins, just like the Java one to keep the tools standardised.

It automated the build, testing, scanning and deployment of an application, and its subsequent new features into a production environment, automatically meeting the firm's and regulatory requirements via the automated scans and associated "toll-gates". All of which would happen automatically from the point a developer was to push their code into a SCM (Source Code Management System, for example GIT), which they would have to be doing anyway. For a firm with over 40,000 technologists, this huge time-save was a big deal. I was flown over to New York and New Jersey and gave presentations to 100s of the firm's most senior developers. By showing them my work and implementation strategy, they too could utilise my pipeline for their applications. Saving them and the firm time, increasing value-add throughput, lowering code smell, all while increasing their applications security and improving their time to market.

## CLIENT SENTIMENT ANALYSIS TOOL

September 2017

I was part of a team which developed a system and method for implementing a client sentiment analysis tool. We did the initial development within just a 12 hour stretch in a Hackathon. This later lead to me becoming a recognised inventor by the US Patent & Trademark Office in July 2018 (20190220777).