Berlin, Germany
luca@tabone.io

Luca Tabone

Senior Backend Engineer

Employment History

Backend Engineer (Freelance) at Klarna, Berlin

September 2024 — Present

- Built an internal AI driven Chatbot that checks employee knowledge around global bank regulations through an LLM driven Q&A chat where the employee answers predefined questions via free text or multiple choice/multi select
- The language model validates the correctness of free text answers and is able to provide additional information around the topic of a specific question
- Python, SQL, FastAPI, Pydantic, Airflow, OpenAI Completion API, Slack Messaging API, Kubernetes

Backend Engineer (Freelance) at Mirror Health, Berlin

December 2023 — September 2024

- Leading the development of the Backend and Data-Pipelines of a German stealth-startup
- Backend- & Data-Engineering with Python
- Daily work includes technologies like Python, Docker, Kubernetes, Kafka, AWS, PostgreSQL, MongoDB, FastAPI, Pydantic, REST, Airflow, CI/CD, Multi-Processing, Multi-Threading, Async Programming
- Working cross-functionally with the Product-, Design-, Frontend- and DevOps Team to ensure consistent alignment across the projects lifetime

Engineering Lead at Earlybird Venture Capital, Berlin

July 2022 — November 2023

- Built a multitude of Data-Pipelines using Airflow, Cloud Composer and Kafka
- Built a general web-scraping framework to parse Website data to structured JSON.
- Built a web-crawling and scraping system that enables us to scrape social platforms using mobile proxies.
- Designed the local-storage-architecture of our Next.js EagleEye app using Zustand.
- Drove the effort to change the Tech-Stack of our EagleEye web-app from (multi-repo, Node.js, JavaScript, GraohQL and React) to a monorepo based t3-stack (Next.js in the backend and frontend, TailwindCSS, tRPC as a type-safe API layer, Prisma ORM). This lead to a vast increase in iteration cycles during the development of EagleEye and fewer engineers needed to build and maintain the app.
- Implemented the tRPC API backend for our web-app.
- Lead the hiring of 7 engineers.
- Lead the bi-weekly sprint plannings and quarterly OKR plannings.
- Stakeholder Management. Consolidating feedback and ideas from our general Partners and making sure to deliver on deadlines.
- Designed dozens of database models for our PostgresDB.
- Driving UX decisions based on user feedback.
- Integrated all software projects within the Earlybird technology department into a monorepo.

- Lead the development of a programming-language-agnostic code-library within our monorepo.
- Python, FastAPI, Pydantic, Airflow, GCP

Engineering Lead at OCEANSAPART, Berlin

February 2021 — June 2022

- Built a total of 22 Data-Pipelines using BigQuery, Airflow and Kafka
- Created a React & Node.js based webapp to enable the inbound logistics team to handle and update product stock across various online shops.
- Built a Python pipeline that automatically handles exchange cases and saves OCEANSAPART more than 1.1 million € each year.
- Python pipeline that automatically handles refund cases. This task was previously also handled by a third party involving human agents to process refund cases. The pipeline I built for handling > 50 % of all refund cases automatically helped to save > 400k € each year.
- Python pipeline to keep the product pictures of our online shops in sync with the Google Drive folder of the Design team.
- Python pipeline to handle automatic emails to keep customers up-to-date on shipping information regarding their orders.
- Python pipeline to programmatically apply discounts to our products.
- Python pipelines to aggregate information on our ad-spent across the platforms (Instagram, Snapchat, TikTok, Facebook) and sync this data with a Google Sheet used by the C-Levels of the company which they used to assess our marketing performance.
- Python, FastAPI, Pydantic

Founder at Loopa, Berlin

January 2020 — January 2021

- Built a mobile app in React-Native that enables coffee shops and restaurants to accept credit card payments and reward their customers loyalty with a digital stamp-card system.
- Built a second mobile app in React Native for the customers of the shops which allowed them to view their digital stamp-card wallet and load their loyalty cards with money using Apple- or Google-Pay.
- Built the REST based serverless Python API with Flask for the two mobile apps. The API was deployed to Google Cloud Run.
- Built the Postgres Database on Google Cloud SQL.
- Our user (the customer of a coffee shop) would earn a stamp by getting their QR code scanned in the shop. In order to provide the user with feedback on the stamp they just earned, I integrated the Realtime Database of Google Firebase into our backend and into both mobile apps.

Consultant at Netlight, Berlin

July 2019 — December 2019

- During my 6 months at Netlight I consulted the finance department of one of our clients in the automotive industry on data visualisation using Looker.
- Hands-On supported the engineering teams on further developing the Java based backend that processes card payments on our clients site.

Machine Learning Engineer at Innoloft, Aachen

October 2017 — June 2019

- Given a hierarchical category tree where each node represents an industry, *e.g. mobility or energy*, I developed a net of 142 interconnected Multi-Layer-Perceptrons to solve hierarchical text categorization. I achieved a classification accuracy of 96.7 %.
- The text data used to train the supervised model mentioned above, was scraped from 10 different sources. I implemented a headless Selenium scraper deployed on Google Compute Engine to continuously scrape these sites.
- Built various Data Engineering Pipelines for our SQL database
- Python, Flask, Pydantic, Selenium, GCP

Education

M.Sc. Computer Science, RWTH Aachen University

October 2015 — September 2017

Links

Website

LinkedIn

Skills

●	Python	●	TypeScript
●	Javascript	◕	Kafka
◕	Docker	●	PostgreSQL
●	MongoDB	●	REST
●	CI/CD	●	Serverless
●	FastAPI	●	Pandas
●	Pydantic	●	PyTest
◕	Protobuf	●	Poetry
●	Google Cloud	◕	AWS
●	tRPC	●	Next.js
●	React	●	Node.js
◕	Terraform	●	Web Crawling
●	Web Scraping	●	Google Firebase
◕	Grafana	●	Pydantic

Languages

●	**German** Native speaker	●	**English** Native speaker