Pretty Version: https://invocatis.github.io/resume/target/resume.html

Luke Cook

Contact

Location: Ithaca, NY | United States of America

Email: lukecook.e0280@passmail.net

LinkedIn: https://www.linkedin.com/in/invocatis

GitHub: https://www.github.com/invocatis

Education

Degree: Bachelor’s

Field Of Study: Computer Science

Institution: Rochester Institute of Technology

Year: 2017

Experience

Datasembly

Roles

- **Title:** Senior Software Engineer

Duration: 2022-2024

- **Title:** Software Engineer

Duration: 2020-2022

Start Date: December 2020

End Date: July 2024

Location: Remote

Responsibilities

- Build & maintain web scrapers for grocery domain data
- Implement architectural advancements to web scraping framework
- Reverse engineer site APIs; adapt to scraping framework
- React to bot detection; develop and apply obfuscation techniques
- Work with Product and Customer teams to convert data to actionable insights
- Lead domain-specific dictionary initiative

Projects

Developed headless browsing framework in Scala using Playwright

- Proposed, by me, to replace the existing framework written in Puppeteer / Javascript
- Playwright on the JVM greatly reduced development and deployment fcomplexity; no longer needed to maintain a cluster of headless workers, or a remote execution DSL
- Enabled scraping of websites that required Javascript execution, had sufficiently complex login APIs, or employed encryption of API request information
- Framework was developed to easily allow different headless browser libraries to be used

Primary Developer who created & maintained scrapers for Amazon, Instacart, and Shipt

- Some of the most complex data sources we collected from
- Instacart was an easy avenue for collecting new banners, creating fast turn around for new customers
- Instituted new code paradigms that others could leverage to make scrapers more legible, and stable

Skills

- Scala
- Scala Cats
- Puppeteer / Playwright
- SQL
- Snowflake
- GCP
- HTTP Protocol
- Airflow
- Web Scraping

Hyperfiddle

Roles

- **Title:** Software Engineer Contractor

Start Date: March 2020

End Date: September 2020

Location: Remote

Responsibilities

- Development on Full Stack Clojure (script), Datomic CRUD application framework
- Setup CI/CD pipeline using CircleCI and AWS ECR and ECS, configured in Terraform

Projects

Formalized extensible algebra for Datomic statements

- Algebra supported simplification of negating statments (eg. 1 + -1 = 0)
- Frontend logic would submit new statements on change, and the state would self simplify
- Datomic statements were proved to form an Algebraic Group

Research and Development

- Reactive streams for rendering in Clojurescript
- Category theory-based programming in Clojure
- Pattern Matching algorithms

Skills

- Clojure(script)
- Datomic
- Reactive Streams
- Haxe
- AWS
- Terraform
- Docker

Cornell University

Roles

- **Title:** Application Programmer II

Start Date: June 2019

End Date: June 2020

Location: Ithaca New York

Responsibilities

Improved genomic CSV digester application

- Simplified instruction file format, reducing redundancy and improving readability (V2)
- Rewrote digester to be generic; now uses Aspect Files that describe the CSV to be loaded (V3)

Wrote file translator to convert instructions files between V1, V2, and V3; written in Clojure(script) both as command line utility, and web app

Projects

Worked on backend of genomic data ETL application

Created CSV file digest framework (See Digester Project Below)

- Provided a data file, and a descriptive aspect file, the application produces a stream tuples that would be committed to a database.
- Aspects would be used to generate streams of cells; these streams would be aligned based on properties of each aspect
- Written in Java; strong focus on an object-oriented design

Developed framework for automated end-to-end testing; focused on composability, ease of use, and self-cleanup (See Ernie Project Below)

- Developed domain-specific scripting language to simplify semantic layer of test suit in 3 layers: Arrange/Act, Assert, and Cleanup
- Was built in 3 separate components: A language backend (written in Clojure), a business logic middle layer (written in Java), and a testing logic script suite (written in custom syntax)
- Cleanup ensured automatic test independence and idempotency; tests were often run on deployed production environments]

Skills

- Java
- Clojure
- Docker
- PostgreSQL
- Object-Oriented Design

Skills

- Java
- Clojure(script)
- Scala
- Docker
- Google Cloud
- Aiflow
- Snowflake
- Airflow
- SQL

Projects

- **Name:** eldritch

Description: Algebraic Data Types & Pattern Matching

Link: https://www.github.com/invocatis/eldritch

- **Name:** motif

Description: Recursive Pattern Matching in Clojure

Link: https://www.github.com/invocatis/motif

- **Name:** ernie

Description: Testing Framework for Java written in Clojure

Link: https://www.github.com/invocatis/ernie

- **Name:** Scala 3 Algebraic Type Exploration

Description: Exploration of algebraic concepts defined in Scala 3

Link: https://github.com/Invocatis/scala3-algebraic-type-exploration

---

- **Name:** CSV Digester

Description: CSV ingestion and manipulation library

Link: https://bitbucket.org/gobiiproject/gobii.masticator/src/develop/