

# Luke Todd

☎ (512) 555-0186

✉ luke.todd.dev@example.com

🌐 linkedin.com/example/luketodd

📍 Austin, Texas 78701

## SUMMARY

Python Developer with seven years of experience building secure automation and data-extraction platforms for financial, healthcare, legal, insurance, and enterprise portals. Delivers practical solutions with Python, Playwright, Selenium, Scrapy, BeautifulSoup, requests, **asynchronous programming**, **headless browsers**, and JavaScript-rendered applications. Builds scalable AWS workloads with Lambda, **Step Functions**, S3, IAM, CloudWatch, and **Secrets Manager**, then supports them with structured logging, retries, fault tolerance, and clear runbooks. Brings hands-on experience with MFA flows, CAPTCHA-aware workflows, cookies, headers, sessions, tokens, HTTP, HTTPS, APIs, JSON, XML, and network inspection. Motivated by work that turns difficult portal processes into dependable services customers and operations teams can trust.

## EXPERIENCE

### Senior Python Automation Engineer

March 2025 - Present

Crescent Harbor Analytics

*Austin, Texas*

Owns asynchronous Python extraction services for secure financial and healthcare portals, guides architecture decisions, and supports production operations. Leads authentication design, AWS orchestration, testing, observability, and developer reviews so teams can deliver dependable data services with controlled security and recovery.

- Owned asynchronous Python extraction services for secure financial and **healthcare portals**, translating changing browser behavior into maintainable **Playwright** workflows.
- Reworked MFA, session, cookie, and token handling into isolated authentication components with Secrets Manager integration, improving recovery consistency across scheduled runs.
- Orchestrated Lambda and Step Functions workloads that distributed high-volume extraction across portal accounts with retry, timeout, and rate-control safeguards.
- Established **CloudWatch** dashboards, **structured logging**, and alert thresholds that gave operations teams visibility into failed navigation, CAPTCHA interruptions, and incomplete records.
- Introduced Docker-based **integration tests** with pytest for headless browser flows, API responses, JSON/XML transformations, and regression coverage across critical portal paths.
- Guided three developers through design reviews involving IAM boundaries, Terraform modules, **CI/CD** releases, and standards for secure, fault-tolerant scraping.

### Python Developer

July 2022 - February 2025

Blue Mesa Data Works

*Salt Lake City, Utah*

Built and maintained extraction pipelines for enterprise claims and document portals, combining browser automation, API retrieval, AWS deployment, testing, and downstream data delivery. Partnered with data engineering and support teams to keep portal workflows traceable, repeatable, and useful for customer operations.

- Built Scrapy and Playwright pipelines for enterprise claims and document portals, combining **browser automation** with requests-based API retrieval after network inspection identified stable endpoints.
- Implemented JavaScript-rendered page handling, asynchronous queues, pagination controls, and response validation for datasets spanning multiple customer accounts.
- Mapped multi-step login sequences involving **MFA** prompts, rotating **cookies**, CSRF tokens, and session expiration, then documented approved recovery paths for support teams.
- Deployed containerized development workflows and **AWS Lambda** jobs through GitHub Actions, using CloudFormation templates for repeatable environments.
- Added unit and integration tests for parsers, authentication utilities, retry policies, and JSON/XML normalization, reducing release risk during portal changes.
- Partnered with **data engineering** stakeholders to load validated outputs into **S3** and downstream ETL processes while preserving source metadata and audit fields.

### Automation Engineer

September 2020 - June 2022

Ironwood Systems Group

*Raleigh, North Carolina*

Delivered Python automation for legal and insurance portals, supporting browser workflows, API integration, scheduled extraction, AWS operations, and analyst validation. Focused on reliable records, clear failure handling, and secure delivery of information used by internal service teams.

- Delivered **Python** automation for legal and insurance portals using Selenium, **BeautifulSoup**, requests, and Playwright prototypes to extract structured case and account information.
- Investigated HTTP **headers**, redirects, cookies, and browser **network calls** to resolve authentication failures and distinguish rendering defects from source-system errors.
- Created fault-tolerant task runners with exponential backoff, checkpointing, duplicate detection, and actionable logs for recurring extraction schedules.

- Integrated REST endpoints and **XML** feeds with browser-derived records, applying schema validation and transformation rules before delivery to internal data services.
- Supported **AWS** deployments using Lambda, S3, IAM, and CloudWatch, following least-privilege practices and documenting operational runbooks for scheduled jobs.
- Converted manually reviewed workflows into repeatable test scenarios, giving analysts a dependable process for validating extracted records and exception queues.

**Junior Python Developer**

June 2019 - August 2020

Pine Ridge Software Labs

Columbus, Ohio

Developed supervised Python utilities for web data collection, reconciliation, parsing, testing, containerized setup, and an AWS proof of concept. Built early production habits around clear exceptions, repeatable environments, and records analysts could review with confidence.

- Developed Python utilities for web data collection and reconciliation, beginning with **requests** and BeautifulSoup before extending selected workflows into **Selenium** automation.
- Parsed HTML, **JSON**, and XML responses into normalized records used by analysts evaluating vendor and customer portal data.
- Added cookie persistence, header configuration, timeout handling, and clear exception messages to stabilize extraction tools against inconsistent web responses.
- Wrote pytest **unit tests** for parsers and transformation functions, then incorporated test execution into the team's initial CI pipeline.
- Packaged repeatable local environments with **Docker** and maintained setup documentation for developers testing headless browser behavior.
- Assisted with an AWS proof of concept using S3 and Lambda, presenting findings on execution limits, logging needs, and practical scaling considerations.

**PROJECTS**

**Secure Portal Extraction Mesh** 📅 2026

Created a serverless reference architecture combining Playwright workers, Lambda, Step Functions, S3, CloudWatch, IAM, and Secrets Manager for authenticated portal extraction. Added retry states, structured logs, session controls, and JSON/XML normalization for dependable downstream delivery.

**Authentication Flow Test Harness** 📅 2025

Built a Dockerized pytest harness for MFA-aware browser journeys, session and cookie persistence, token expiry, CAPTCHA interruption handling, API responses, and JavaScript-rendered pages. Produced repeatable regression evidence for portal changes.

**LEADERSHIP & AWARDS**

- Dean's List, The Ohio State University, 2017-2019
- Outstanding Capstone Software Project, Department of Computer Science, 2019
- Peer Mentor Recognition, Python Developers Group, 2023

**EDUCATION**

**Bachelor of Science in Computer Science**

2019

The Ohio State University GPA: 3.7

Columbus, Ohio

*Coursework: Python programming, Data structures, Database systems, Software engineering, Web technologies*

**CERTIFICATIONS**

- AWS Certified Developer - Associate 📅 2024
- HashiCorp Certified: Terraform Associate 📅 2025

**TECHNICAL SKILLS**

- **Python Automation:** Python, asyncio, pytest
- **Browser Automation:** Playwright, Selenium, headless browsers
- **Web Scraping:** Scrapy, BeautifulSoup, requests
- **AWS Serverless:** Lambda, Step Functions, S3
- **AWS Security:** IAM, Secrets Manager, least privilege
- **AWS Observability:** CloudWatch, structured logging, alerting
- **Authentication:** MFA, CAPTCHA, cookies
- **Web Protocols:** HTTP, HTTPS, headers
- **Data Formats:** JSON, XML, HTML
- **API Integration:** REST APIs, network inspection, tokens

- **Infrastructure as Code:** Terraform, CloudFormation, AWS resources
- **DevOps Delivery:** Docker, GitHub Actions, CI/CD
- **Data Engineering:** ETL, data transformation, S3 pipelines
- **Reliability Engineering:** Retries, backoff, checkpointing
- **Testing Practices:** Unit testing, integration testing, regression testing

SKILLS

---

- Python
- MFA flows
- Headless browsers
- Session management
- Playwright
- Selenium
- REST APIs
- CI/CD pipelines
- AWS Lambda
- Scrapy
- JSON and XML
- Data transformation
- Web scraping
- Async programming
- HTTP and HTTPS

PROFESSIONAL AFFILIATIONS

---

- Python Developers Group, Technical Workshop Mentor, 2021-2023
- Cloud Engineering Community, Secure Automation Study Lead, 2024-Present

LANGUAGES

---

- English (Native)
- Spanish (Intermediate)

ADDITIONAL INFORMATION

---

**Work Status** : Authorized to work in United States. No sponsorship required.

REFERENCES

---

AVAILABLE ON REQUEST