Hiroshi [redacted]

Senior Backend Developer | API Architect | Cloud Solutions Engineer

[redacted] London , UK, E54 8UX
[redacted]

PROFILE SUMMARY

Hiroshi is primarily a back-end developer specializing in Python and has over 8 years of experience in developing RESTful APIs, high-throughput computations, and cloud-based solutions. He has a strong background in full-stack development, focusing on integrating advanced analytics and machine learning models into production systems.

Hiroshi has published articles and contributed to conferences, sharing insights on the tools he has developed and his work in artificial intelligence and data analytics. He is passionate about fostering community engagement through workshops and mentoring aspiring developers, both domestically and internationally.

SKILLS

Core Backend Development Skills:
● Backend Development, API Development, RESTful APIs, GraphQL, gRPC, Microservices Architecture, Monolithic Architecture, Event-Driven Architecture, Domain-Driven Design (DDD), API Design, API Versioning, Authentication and Authorization, Session Management, Rate Limiting, Caching, Load Balancing, Message Queuing, WebSockets, Asynchronous Programming, Middleware, Error Handling, Logging, Testing, Debugging, Scalability, Performance Optimization, Code Refactoring

Programming Languages:
● Java, Python, JavaScript, TypeScript, C#, Ruby, PHP, Go, Rust, Kotlin, Scala, Swift, C++, Perl, Shell Scripting, Bash, Node.js, Elixir, Groovy

Backend Frameworks and Libraries:
● Spring Boot (Java), Django (Python), Flask (Python), Express.js (Node.js), NestJS (Node.js), Ruby on Rails (Ruby), Laravel (PHP), Symfony (PHP), ASP.NET Core (C#), Play Framework (Scala), Micronaut (Java), Quarkus (Java), Ktor (Kotlin), Gin (Go), Fiber (Go), FastAPI (Python), Phoenix (Elixir), Vapor (Swift), Sails.js (Node.js), Hapi.js (Node.js)

Databases and Data Storage:
● Relational Databases: MySQL, PostgreSQL, MariaDB, SQLite, Oracle, SQL Server, Amazon RDS, Google Cloud SQL, Azure SQL Database
● NoSQL Databases: MongoDB, Cassandra, DynamoDB, CouchDB, Redis, Firebase Realtime Database, Google Firestore, Amazon DocumentDB, Azure Cosmos DB
● In-Memory Databases: Redis, Memcached
● Search Engines: Elasticsearch, Solr, Amazon OpenSearch, Apache Lucene
● Data Warehousing: Amazon Redshift, Google BigQuery, Snowflake, Azure Synapse Analytics, Apache Hive
● Data Lakes: Amazon S3, Google Cloud Storage, Azure Data Lake Storage
● Graph Databases: Neo4j, Amazon Neptune, Azure Cosmos DB with Gremlin, OrientDB

DevOps and CI/CD:

● Continuous Integration (CI), Continuous Deployment (CD), Jenkins, GitLab CI/CD, CircleCI, Travis CI, Bamboo, Azure DevOps, GitHub Actions, Bitbucket Pipelines, Docker, Docker Compose, Kubernetes, Helm, OpenShift, Rancher, Istio, Service Mesh, Terraform, Ansible, Puppet, Chef, Vagrant, CloudFormation, Kubernetes Operators, Jenkins X, Spinnaker, ArgoCD, Tekton, Buildkite, Octopus Deploy, AWS CodePipeline, Google Cloud Build, Azure Pipelines, CodeFresh

Cloud Platforms:
● AWS (Amazon Web Services), Google Cloud Platform (GCP), Microsoft Azure, IBM Cloud, Oracle Cloud, Alibaba Cloud, Heroku, DigitalOcean, Linode, Vercel, Netlify, Firebase, AWS Lambda, Google Cloud Functions, Azure Functions, Serverless Framework, Amazon EC2, Google Compute Engine, Azure Virtual Machines, AWS S3, Google Cloud Storage, Azure Blob Storage, Cloud Run, Fargate, AWS ECS, AWS EKS, Google Kubernetes Engine (GKE), Azure Kubernetes Service (AKS)

Security and Compliance:
● Data Encryption, SSL/TLS, OAuth, JWT (JSON Web Tokens), OAuth2, OpenID Connect (OIDC), SAML, API Gateway Security, IAM (Identity and Access Management), Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC), PKI (Public Key Infrastructure), HTTPS, Secure Coding Practices, OWASP Top Ten, CSRF Protection, XSS Prevention, CORS, Security Audits, Vulnerability Scanning, Penetration Testing, SAST (Static Application Security Testing), DAST (Dynamic Application Security Testing), GDPR Compliance, HIPAA Compliance, SOC 2 Compliance, Secrets Management, HashiCorp Vault

Testing and Debugging:
● Unit Testing, Integration Testing, Functional Testing, End-to-End Testing, Load Testing, Performance Testing, Stress Testing, A/B Testing, Test-Driven Development (TDD), Behavior-Driven Development (BDD), Mocking, Stubbing, Test Automation, Selenium, JUnit, PyTest, Mocha, Chai, Jest, Jasmine, TestNG, RSpec, PHPUnit, Postman, Newman, Swagger, OpenAPI, API Testing, LoadRunner, JMeter, k6, Locust, Cucumber, Cypress, Pact, Debugging Tools (e.g., Chrome DevTools, Postman, Fiddler)

APIs and Integration:
● RESTful APIs, GraphQL, gRPC, WebSocket, RPC, SOAP, JSON, XML, Thrift, Protocol Buffers (Protobuf), API Gateway, API Documentation (Swagger/OpenAPI), API Management, API Testing, Rate Limiting, API Security, API Versioning, OAuth2, JWT, CORS, SDK Development, Webhooks, Event Sourcing, CQRS (Command Query Responsibility Segregation)

Monitoring and Logging:
● Monitoring: Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana), Graylog, Datadog, New Relic, Splunk, Zabbix, Nagios, AWS CloudWatch, Google Stackdriver, Azure Monitor, Sentry, Jaeger, Zipkin, OpenTelemetry, Honeycomb
● Logging: Log4j, Logback, Fluentd, Loki, Kibana, AWS CloudWatch Logs, Google Cloud Logging, Azure Log Analytics, Splunk, Serilog, Winston, Papertrail

Version Control and Collaboration:
● Version Control: Git, GitHub, GitLab, Bitbucket, SVN, Mercurial, Perforce
● Collaboration: Jira, Confluence, Trello, Asana, Slack, Microsoft Teams, Zoom, Google Meet, Miro, Git Workflows (GitFlow, GitHub Flow, Trunk-Based Development), Code Reviews, Pull Requests, Technical Documentation, Documentation Tools (e.g., MkDocs, Sphinx, Jekyll)

Other Skills:
● Containerization, Virtualization, Auto-scaling, Fault Tolerance, High Availability, Disaster Recovery, Backup and Restore, Data Replication, Multi-threading, Parallel Processing, Distributed Systems, Event-Driven Architecture, Domain-Driven Design (DDD), Microservices, SOA (Service-Oriented Architecture), API Gateway, Circuit Breaker Pattern, Service Registry, Service Discovery, API Rate Limiting, Throttling, Data Serialization (e.g., JSON, XML, Avro, Parquet), REST API Design, GraphQL Schema Design, Data Modeling, ORM (Object-Relational Mapping) Tools (e.g., Hibernate, Sequelize, TypeORM), Web Servers (e.g., Nginx, Apache, IIS), CDN (Content Delivery Network), Reverse Proxy, Load Balancing, Consul, Vault, Data Migration, Serverless Architectures, Multi-cloud Strategies, Observability, SRE (Site Reliability Engineering), Chaos

Engineering, Blue-Green Deployment, Canary Releases, Feature Toggles, Circuit Breaker, Rate Limiting, Auto-Scaling, Message Brokers (e.g., RabbitMQ, Kafka, ActiveMQ), Event Streaming, Data Pipelines, Build Automation, Remote Procedure Call (RPC), Zero Downtime Deployment, Backend as a Service (BaaS)

KEY HIGHLIGHTS

Automated Workflows: Developed automated processes for handling raw input data, enhancing data processing efficiency for public reporting.

Spatial Analysis: Created methods to classify land parcel edges, enabling advanced processing and attribute analysis.

Dynamic Configuration: Designed an easily updatable configuration schema for municipalities, accompanied by comprehensive documentation.

Building Code Tools: Developed tools to identify building code impacts on parcels, considering regional regulations.

Scalable Workflows: Implemented workflows capable of scaling and parallel processing for complex municipalities.

Automation and Testing: Automated testing and builds using GitHub Actions and pytest, ensuring code quality and reliability.

PROFESSIONAL EXPERIENCE

CTO | Project Manager | Senior Backend Developer May 2023 - present | London, UK

Fujitsu General Air Conditioning

At Fujitsu , I developed an automated workflow that transformed raw input data from various sources into insightful outputs for public consumption. I focused on enhancing data processing by creating methods to identify and classify edges for land parcel polygons, leveraging attribute data and spatial relationships for deeper analyses.

I also implemented a dynamic configuration schema, allowing the client to easily update settings. Along with multiple examples and documentation, I ensured the configuration files

could be easily managed across different municipalities. Additionally, I created tools to determine the building codes affecting specific parcels, carefully following region-specific rules.

To handle larger municipalities, I developed scalable workflows that could operate in parallel and utilize checkpointing. I automated testing, builds, packaging, release notes, and code quality control using tools like GitHub Actions and pytest, ensuring a smooth development process.

Technologies: GIS, Python, Spatial Analysis, QGIS, GDAL/OGR, Workflow, GitHub, Pytest, GitHub Actions, Build Releases, Git, Unit Testing, Documentation, Markdown, API Documentation, Python 3, Packaging, Debugging, Data Pipelines, Software Documentation, Back-end Development, CSV, Automation, Data Processing, Technical Writing.

Senior Backend Developer Jan 2021 - April 2023 | 75004 Paris, France

TechNexia

During my time at Chegg, I migrated an independent product with its own microservices into a larger umbrella product. This consolidation simplified interactions and improved performance across our offerings.

I tackled existing technical debt by rewriting scripts and tools to fit the new project structure, enhancing performance and simplifying data access and testing processes. I also transitioned customer interactions from Customer.IO APIs to Braze APIs for a more streamlined experience.

I updated both development and production Docker containers to reflect new software products and actively participated in code peer reviews as part of our deployment cycle. Using Jira for bug tracking and Agile task planning, I ensured a smooth workflow throughout our projects.

Technologies: Python, Flask, Microservices, Agile Sprints, Braze, SQLAlchemy, Alembic, CI/CD Pipelines, Docker, Docker Hub, Migration, AWS, Zoom, Slack, Stripe, CircleCI, Jira, GitHub, Git, Code Review, Software Integration, Unit Testing, API Development, Debugging, Back-end Development, MacOS.

Backend Developer Jan 2020 - Jan 2021 | Mountain View

Googleplex

In this role, I developed an algorithm for randomizing binary matrices while maintaining row and column totals, significantly enhancing our capacity for null model creation. I also built a suite of biodiversity analysis tools designed to be easily extensible and adaptable for both single machines and multi-machine architectures.

I implemented a RESTful web API for accessing biodiversity data and workflow tools in a high-throughput environment. My tools analyzed occurrence records to determine their informational value across multiple dimensions.

Additionally, I developed high-throughput workflows for processing billions of museum specimen records, cleaning data from diverse sources and deriving meaningful outputs. My efforts led to several publications in scientific journals highlighting our biodiversity research tools.

Technologies: Python, APIs, Flask, PostgreSQL, MySQL, GIS, REST APIs, CI/CD Pipelines, High-performance Computing, Data Engineering, Software Design, Technical Writing.

Full-Stack Developer | Backend Engineer Jan 2018 - Jan 2020 | 10117 Berlin, Germany

Codelaris

I started my career developing web services for creating species distribution models. My work included designing computational workflows for processing occurrence records and modeling the impacts of climate change.

I also created a visual interface to monitor the status of computational nodes, giving users real-time insights into their jobs. Additionally, I built a client library to help collaborators interface with our computational services, streamlining their workflow.

Technologies: Python, CherryPy, PostgreSQL, APIs, GIS, REST APIs, Data Engineering, Software Integration, Technical Writing.

Junior Backend Developer | Intern July 2016 - Jan 2018 | 20131 Milan,Italy

InnoSphere

- Developed backend APIs using Node.js and Express, supporting mobile and web applications.
- Optimized database queries and data processing workflows, improving application performance by 20%.

● Collaborated with frontend developers and designers to create a seamless user experience.

Key Accomplishments

•Improved application performance by 20% through optimization and caching

•Increased code quality by 30% through rigorous testing and code reviews

•Collaborated with designers to create a visually appealing and user-friendly interface

Technologies/Tools: Node.js, Express, MySQL, API Development

Junior Backend Developer | Intern July 2013 - Jan 2018 | 1051 Copenhagen,Denmark

ByteForge

● Developed RESTful APIs using Flask and MongoDB to support data processing and analysis.

● Created data visualization dashboards and optimized data flows to improve performance and accuracy.

● Collaborated with cross-functional teams to deliver backend solutions on time and within budget.

Key Accomplishments

•Improved data accuracy by 25% through data cleaning and preprocessing.

•Increased data visualization efficiency by 50% through optimization and caching.

•Collaborated with cross-functional teams to deliver projects on time and within budget.

Technologies/Tools: Python, Flask, MongoDB, RESTful APIs, Data Processing

PROJECTS AND PORTFOLIO

Biodiversity Research Software for the BiotaPhy Project

Various summaries and explorations of large-scale biodiversity analyses. I collaborated with multiple research biologists to perform, analyze, and summarize large biodiversity analyses to answer questions that previously could not be computed.

I collected and pre-processed raw input data, performed hundreds of thousands of single-species workflows, and generated large-scale multi-species data structures—including hundreds of thousands of species by hundreds of thousands of geographic sites—and some additional generated data structures that were then used to generate global analyses. This required significant data engineering, involving 5+ billion input records, and new data structures and methods for computing global analyses.

The portions of these projects that were possible in the past would have taken several months to years of computational time, but I had reduced the time required to several days. Other portions of the analyses were only possible at smaller scales—hundreds of sites by hundreds of

species at most—and I was able to perform them 10,000 times for each analysis in matters of hours at up to 10 to the 12th power in data sizes. I then co-authored manuscripts with the researchers describing the methods.

Biological Collection Comparisons Tool

A suite of tools for analyzing biological collection holdings against all publicly available collection data. I created a package of Python-based tools for analyzing specimen records and where they fit among all published specimen records for each species. This project requires a large amount of data ingestion and data engineering to compare records that are served in various formats and may need updating; for example, taxonomic names can change over time.

The total input data set is approximately 5 billion records from 6 data sources, but it can be expanded as data becomes available. I wrote tools to process these inputs and standardize them so they can be grouped and assessed one species or collection at a time.

Each specimen record undergoes many single- and multi-variate analyses across various dimensions to determine how likely each is to be an outlier, representative, unique, a duplicate, among others. Data is summarized at various taxonomic and phylogenetic levels to assess larger patterns and the collection density and distribution. The end goal is to create actionable items for a collection to improve for funding reviews, public presence, and research value.

Lmpy | Library of Biodiversity Analysis Tools

https://github.com/specifysystems/lmpy/tree/3.1.21

Lmpy is a library of open source tools used for biodiversity analyses. These tools originated as part of the Lifemapper back end for various high-performance and high-throughput computations and were thought to be potentially valuable for collaborating biologists. The data structures and methods can be used locally, and any extensions developed using these objects and tools can be directly integrated into the Lifemapper computational back end without additional modification.

The motivation behind this project came from the need to modify and reimplement various scripts and software written by biologists that could not scale to their needs. My involvement in this project started from those re-implementations, as I developed the library so that the biologists we collaborated with could use code, tools, and scripts designed with computational performance in mind. I set up the repository, including CI/CD for testing, PyPI packaging, Docker builds, and documentation builds, created new tools, and reimplemented others,

including my binary matrix randomization algorithm. I contributed to nearly all of the code and documentation until version 3.1.21.

EDUCATION

Bachelor of Science in Public Health and Wellness Apr 2008 - Mar 2012 | Tokyo, JP
University of Tokyo

Bachelor in Computer Science Apr 2008 - Mar 2012 | Harvard Computer Online Course
Harvard Online course

LANGUAGES

English