

## Overview

---

Elysian Insurance Services is developing an "AI Claim Conductor" system to streamline and automate various aspects of the insurance claims handling process. The Claim Conductor is responsible for ingesting and processing claim-related data, maintaining a comprehensive understanding of each claim's state, and determining the appropriate next steps or actions to be taken. The system relies upon a claims knowledge graph: A dataset of interconnected information about active and historical claims, providing context for processing and decision-making.

Your task for this take-home assignment is to **build and deploy** a version of the Claim Conductor in miniature. Specifically, you will design, test, and deploy a system that is capable of accepting webhooks with updates to a person's name from a simulated phone book provider, which then must be capable of processing the webhooks and updating the system's internal state of understanding about the person.

## Requirements

---

Your service must have HTTP endpoints that conform to the specifications provided in our OpenAPI documentation, which can be found at <https://elysian-cto-takehome.vercel.app/> (with the YAML spec available at <https://gist.github.com/jovialis/3b017e7b5149780272e7689324a28b19>). The endpoints will include:

1. **accept\_webhook:** This endpoint should accept incoming webhook payloads containing NameAdded, NameRemoved, and NameUpdated objects (see API spec). Your system should process these payloads and update an internal data store to reflect the current state of each person's name.
2. **get\_name:** This endpoint should accept a person's ID as a query string parameter and return the person's most up-to-date name (or null if the person doesn't exist or has been removed).

We will provide you with a utility script to generate random webhook payloads for testing purposes, which can be found at <https://gist.github.com/jovialis/7837ff48507fb74dfc01430e19991f3d>.

## Additional Challenge (Time Permitting)

---

If you have additional time, you should integrate a frontend that gives the ability for a user to interact with your service using natural language, answering questions such as:

- "What is the current name of person ID: 05bf9399-f9ec-416e-822d-35ec8d2085a9?"
- "What are all the previous names of person ID: 05bf9399-f9ec-416e-822d-35ec8d2085a9?"

## Time Limit

---

We expect this assessment to be challenging, but we don't want you to spend more than 4 hours working on it. We care more about learning how you approach and think through problems than whether you deliver a fully completed solution.

# Submission

---

Please provide the following as part of your submission:

1. A link to your deployed service.
2. Your code, organized and documented in a way that makes it easy for us to understand and run.
3. A **brief** write-up (e.g. paragraph) discussing your approach, any assumptions you made, challenges you encountered, and trade-offs you considered.
4. If applicable, an explanation of your thought process and planned approach for any remaining work if you reached the time limit.

Given that this is meant to simulate a real task if you were to join Elysian, be prepared to chat on how you would approach improving and scaling the system over the next 30, 60, and 90 days. We would love to brainstorm on this with you in the next round of the interview.

Thank you for taking the time to complete this assessment. We look forward to reviewing your submission and discussing it further during your debrief interview!