**Citrine Technical Challenge**

**Introduction**

Software engineers at Citrine develop services that manage materials data as well as the machine learning infrastructure for learning on that data. The following technical challenge is meant to be representative of the type of work that you would be doing as an employee at Citrine. There is no time limit, but the problem is designed to take approximately 5-10 hours so please do not spend much more time on it than that!

## **1) Writing**

Communication to both technical and non-technical coworkers is critical. For this part of the challenge, pretend that you are living in the **Hypothetical World** described below:

You are an engineer working on the data Ingestion pipeline. You see developers struggle to release their code, as they sometimes produce bugs or mess up the deployment scripts. You believe that a continuous integration and deployment system would dramatically improve the quality and cadence of engineering work, but the CTO is skeptical when you bring it up, citing the following reasons:

* We have too many customers to configure CI/CD for - at Citrine every customer gets their own AWS account
* We should release our code at the end of every sprint
* Each team uses different technologies
* We can’t afford to take engineers off of product work to work on it

*Write an e-mail to our CTO to convince them that implementing a CI/CD workflow would be a worthy investment.*

**2) Software**

Each row of the CSV that is attached to this email represents a single material and several of its measured properties. Please implement a web service that lets consumers search and add new records to the data with appropriate GET and POST requests.

In your README, in addition to documenting your API, please provide example requests that express the following:

1. Add a new record (compound + properties/values)
2. Return all records where the band gap is between 0 and 3.
3. Return all records where the compound contains Gallium.
4. Return all records where the compound contains Gallium and the band gap is between 0 and 3.

The solution to the challenge should include a readme or other write-up that describes how the code is configured and how it can be setup and run. With respect to frameworks and languages – we’re pretty flexible, but please use one of **ruby, python, java, or scala**. No Microsoft dependencies.

**Grading**: The challenge is graded based on your expression of software engineering fundamentals (defensive programming, code readability, documentation) and solution quality (API design, search functionality).