Ask2.ai Assignment

We want to learn how you design and code solutions to the type of problems we're solving.

The goal of this assignment is not the completion of a predefined project. Instead we want you to come up with a design and prototype of an API for a relatively open ended problem. At Ask2.ai, you will be dealing with a similar backend design.

Our goal is that, after the onsite, we'd feel confident that you could fix issues in our API and implement new features.

Formalities

- We expect you to return this challenge in ~48 hours after we send it to you.
- We expect you to spend at most 3-4 hours working on it
- You must use NodeJS with typescript.
- Use GraphQL schema driven approach to implement the backend. We use https://github.com/ardatan/graphql-tools. We do not use code driven approach to build our schema.
- **The submission should include:**
  - Actual working code shared on Github, showing a part of the final API as a proof of concept.
  - A brief writeup in README.md explaining the most important design decisions and remaining open questions
- We will discuss your submission as a team during your onsite interview

The Challenge

A core part of Ask's product is the ability to execute long running python scripts that are triggered by our GraphQL API. First part of this assignment is to come with a GraphQL API that allows client to run a linear regression model that takes about a minute to execute. Our http requests time out in 30 secs so we will have to implement

an asynchronous execution of this API call. The client can poll the API (although subscription is a better alternative) to check when the execution is finished and get the result back.

You can mock the Python script execution as a javascript promise that resolves in one minute. The users passes three parameters for linear regression:

1. List of (x,y) data points to run regression on. x and y are both floats.
2. Choice of regression model, which could be one of the following: Linear, Lasso, or Ridge
3. For linear regression, there's no additional parameter required. For Lasso and Ridge, there's an optional parameter "alpha" which must be positive float.

The result is a fitted regression model which can be represented as a "Model" graphql object. The client can execute this model to get the prediction for a new x. In other words, the client can specify the fitted model ID, and a float x, and the result is a float y. For simplicity, since we are not fitting an actual regression model, you can always return "x + 1" as the result.

Come up with a GraphQL schema and implement it in NodeJS. You are free to use any persistence you want. We are a big proponent of code generation and type safety so we'd love to see these in your code but it's not a requirement.

Goal

Once again; the goal is not completion. Instead we hope you come to the onsite with an interesting proof of concept + documentation that allows us to discuss the tradeoffs of your approach.