From: **Thomas Luce** thomas.luce@highspot.com
Subject: Highspot coding exercise
Date: December 13, 2019 at 12:00 PM
To: halmueller@gmail.com

TL

Hi Hal,

Thanks for your time today; it was great speaking with you.

I'm excited to invite you to complete a brief coding exercise. In lieu of a technical phone screen, we offer candidates the opportunity to complete this exercise to show off their technical acumen. My team anonymously reviews all exercises, and we determine next steps in the interview process based on that review.

Let me know if you have any questions after reading through this.

Cheers,
Thomas

Highspot Coding Exercise

Welcome to the Highspot interview process! For this exercise, you will be creating a command-line batch application. Highspot's product depends on both a rich web service and on batch processing — for internal use, for our customers, and for partner integrations — so this exercise is relevant to real-life Highspot engineering work. During in-person interviews, we may use your code as a discussion point. This should take between one and two hours to complete.

Overview

In a programming language of your choice, create a console application that applies a batch of changes to an input file in order to create an output file.

In the instructions below, we provide you with the input JSON file, which is called `mixtape.json`, and we tell you the types of changes your application should support. You will design the structure of the changes file and you will write write the code that processes the specified changes and outputs a new file.

We'll expect to interact with your application like this:

```
$ application-name <input-file> <changes-file> <output-file>
```

For example:

```
$ killer-app mixtape.json changes.json output-file.json
```

Background

Why a take-home exercise?

In our experience take-home exercises, compared to technical phone screens, are more objective, and they facilitate a more equitable interview process.

Furthermore, in the normal course of software development, you have the freedom to mine your previous assignments, search Stack Exchange and Google, solicit advice from friends and relatives, and engage in quiet contemplation in order to solve problems. We do that too, and we'd like to see you exhibit a real-world-style work product.

How much time should I invest?

Highspot devs have completed this exercise ourselves, and we think that an hour or two of development time is reasonable for an experienced engineer. Spend more or less time on this, it's up to you, but please don't let perfect be the enemy of good — we'd rather see something that illustrates your talents than miss out on a chance to work with you.

We respect your time and we will accept and review any solution you submit, full or partial. If you can't invest the time to show us your best work, we're open to discussing alternatives.

Considerations

Logistics

- Keep it simple: focus on implementing the set of changes enumerated below. Your application doesn't need to handle any other operations.
- Assume that other people have to run, read, and support your code, that we're going to run tests against it, and that future assignments will build upon it.
- Use any language, tools, or solutions you see fit. For mobile positions, we like to see a mobile-appropriate language (Swift, Java, Obj-C, Kotlin, etc.), but the choice is yours.

OK, let's go!

Project Requirements

Here are the basic parameters for this exercise:

1. This input JSON file consists of a set of users, songs, and playlists that are part of a music service: **mixtape.json**.
2. Your application ingests `mixtape.json`.
3. Your application ingests a changes file, which can take whatever form you like (we use `changes.json` in our example, but you're free to make it text, YAML, CSV, or whatever). The changes file should include multiple changes in one file.
4. Your application outputs `output.json` in the same structure as `mixtape.json`, with the changes applied. The types of changes you need to support are ennumerated below and the application should process all changes in one pass.
5. Your solution includes a README that explains how to use your application.
6. Your README describes what changes you would need to make in order to scale this application to handle very large input files and/or very large changes files. Just describe the changes — you don't actually need to implement a scaled-up version of the application.

application.
7. Don't worry about creating a UI, DB, server, or deployment.
8. Your code should be executable on Mac or Linux.

The types of changes your application needs to support are:

1. Add an existing song to an existing playlist.
2. Add a new playlist for an existing user; the playlist should contain at least one existing song.
3. Remove an existing playlist.

Project Delivery

Send your working code and your README, which should also include instructions for how to run your project, including instructions about any dependencies.

What happens next?

After you submit your code, we will evaluate it, and contact you about next steps, which may involve further elaboration on your code submission.

Thanks again for your time and we look forward to receiving your reply.

The Highspot Engineering Team

--
THOMAS LUCE
Engineering Manager

E thomas.luce@highspot.com
P 206.535.2855
M 303.521.8627
W Highspot | LinkedIn

HIGHSPOT