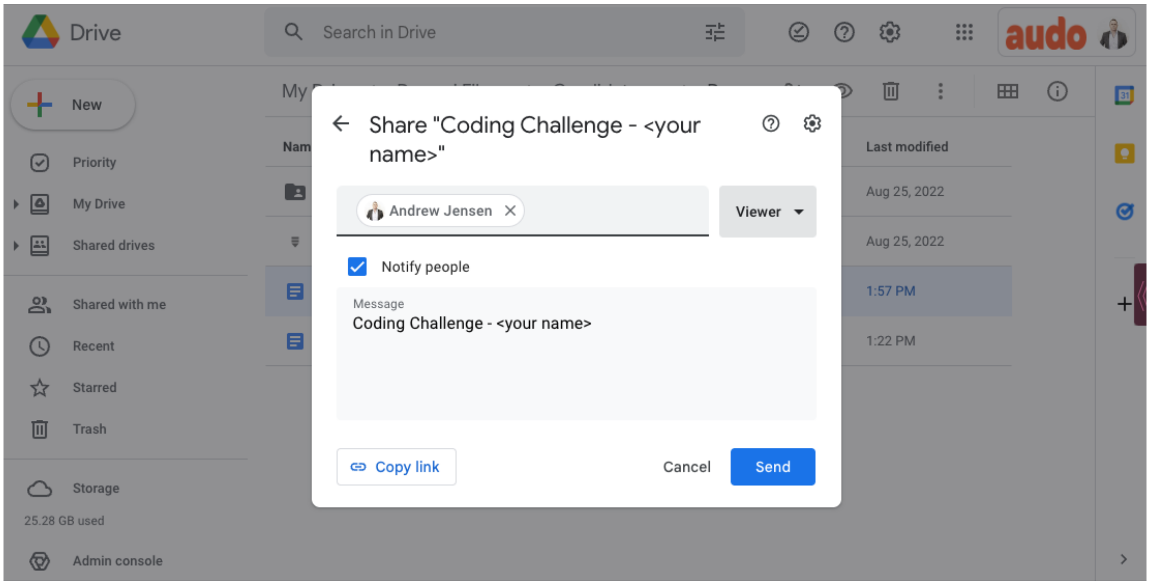
audo

Data Engineer Interview

The following exercise is meant to determine your ability to code realistic applications that are comparable to those that you would be expected to do in your day-to-day job at Audo. Please take a moment to read through this document in its entirety before starting and make sure to email your Audo contact at any time during the test if you have any questions.

- The total allowed time for this is four (4) hours from when you first are handed the document. You may take as little or as much of that time as you want.
- You CANNOT use any code that you found on the internet or have developed previously unless you properly reference your source(s).
- FAILURE TO PROPERLY DISCLOSE ALL SOURCES OR SEARCHES MAY LEAD TO DISQUALIFICATION.
- You are building a backend application and no UI is required, input can be provided using a configuration file or command line.
- You may use any existing framework, library, or packages that you desire so long as you document them.
- It is up to you to determine to what level of depth you would like to respond to the specifications requested in this coding challenge.
- If a product specification is not clear, use your best judgment to determine that product specification.
- Once completed, please place your code challenge files within a Google Drive folder and then share that folder with the following email address, andrew@audo.com, ensure that the “Notify people” checkbox is checked, add “Coding Challenge - <your name>” in the provided message textbox, and then click the “Send” button (see below image).



Share "Coding Challenge - <your name>"

Andrew Jensen

Viewer

☑ Notify people

Message
Coding Challenge - <your name>

Feel free to add on and be creative.

High Level Project Synopsis:

Create an information crawler that collects, cleanses, and stores data from a popular online information website, then makes that data accessible via an API.

Product Specification #1 – Information Crawler and Cleanse

Required:

- Using an open-source crawler framework (e.g., Scrapy, Node-crawler, PySpider, Apache Nutch, Heritrix), develop an application to crawl a popular online information website, such as Wikipedia, WebMD, CNN, BBC, CNBC, IMDB, Tripadvisor, Yelp, etc.
- Cleanse the crawled data to remove any superfluous and non-pertinent information by using a framework such as Readability.

Bonus:

- Categorize the crawled data into relevant fields.
    - Examples:
        - Headline
        - Article text
        - Author
        - URL

Product Specification #2 – Store Data

Required:

- Store the cleansed data into the database of your choice.

Bonus:

- Categorize the data into columns/fields/parameters that correspond with their respective data fields, as set in the bonus of Product Specification #1 above.
    - Examples:
        - Headline – headline

Feel free to add on and be creative.

        - Article text – article_text
        - Author – author

Product Specification #3 – API

Required:

- Develop an API that makes the stored data accessible.
- Provide API endpoints for retrieving the data.

Bonus:

- Allow the data to be searchable via an API endpoint using full text search.
    - Example:
        - Value = “New York” returns all pages with New York contained anywhere within the page.

BIG BONUS:

- **Allow the data to be searchable by way of the category fields used above in the bonus section of Product Specification #1.**
    - **Example:**
        - **Author search: Key = author and Value = “John Doe” returns all articles written by author John Doe**
        - **Headline search: Key = headline and Value = “North Carolina” returns all articles with North Carolina in the headline**

Feel free to add on and be creative.