

VectorShift

Integrations Technical Assessment Instructions

Thank you for taking the time to interview with us at VectorShift! As part of the interview process, we would like you to complete an integrations technical assessment. You can find all files necessary for the assignment in the `/frontend/src` and `/backend` folders. Feel free to make any changes to the provided files, including adding new files, deleting existing files, installing new packages, and modifying any provided code. Please use JavaScript/React for the frontend and Python/FastAPI for the backend.

The assessment consists of two parts, which are detailed below. You are encouraged to read all parts of the assessment before starting so that you can plan your approach. You can run the frontend code by navigating to the `/frontend` directory and running (1) `npm i` and (2) `npm run start`. You can run the backend code by navigating to the `/backend` directory and running `uvicorn main:app --reload`. Don't forget to also spin up a redis instance using `redis-server`.

Feel free to reach out to recruiting@vectorshift.ai if you have any questions.

Part 1: HubSpot OAuth Integration

In the `/backend` directory, you will find a folder called `integrations`. This folder contains two completed files (`airtable.py` and `notion.py`) and one file that needs to be completed (`hubspot.py`). Use the structure of the completed integrations and any relevant documentation from HubSpot to finish the `authorize_hubspot`, `oauth2callback_hubspot`, and `get_hubspot_credentials` functions in `hubspot.py`.

In `/frontend/src`, you will find a folder called `integrations`. This folder contains JavaScript files for two existing integrations (`airtable.js` and `notion.js`) as well as an empty file for the HubSpot integration (`hubspot.js`). Please finish writing `hubspot.js`, and add the HubSpot integration to the relevant places in other files so that the integration is accessible in the UI.

The AirTable and Notion integrations will not work when you test them because the client information has been redacted. Please create a client id and secret to test your new HubSpot integration. For testing purposes, you can optionally also make your own Notion and AirTable app credentials.

Part 2: Loading HubSpot Items

After retrieving the credentials above, complete the `get_items_hubspot` function in `/backend/integrations/hubspot.py`. This function should query HubSpot's endpoints using the credentials returned from the OAuth flow and return a list of

IntegrationItem objects. Integration items have a list of parameters that may or may not be filled depending on which integration we are working with. It is up to you to decide which fields and endpoints are important for HubSpot. You can reference how Notion and AirTable are set up.

Similar to in Part 1, the AirTable and Notion integrations will not work when you test them because the client information has been redacted. Please create a client id and secret to test your new HubSpot integration. For testing purposes, you can optionally also make your own Notion and AirTable app credentials.

It's up to you how you want to display the resulting list of Integration Items. Printing the final list to the console is the suggested approach.