Background

One of the modules that Quadient Accounts Payable (referred to as QAP in the rest of the document) offers is called the **Invoice Module**, and one of the functions it allows is for users to email invoices into our system.

As a working example, let's say an accounting assistant for a given company, the restaurant "So Fictitious Pub", might email the pub's invoices for Shaw (Cable TV service) and Telus (Phone service) into QAP. They would then use the workflow processing features in QAP to request that this invoice be approved by the pub's financial manager.

In the accounting world, Shaw is the "Vendor", and to be able to pay the vendor, one of the steps the financial manager must take is to determine what "Account" to pay the vendor from (e.g. should this come from the budget for Operating Expenses, or for Travel Expenses, etc.)

Now let's say that QAP sales team has been building its customer profile for the restaurant industry, looking at the tools that they use in their workflow, and has found out that they use a SaaS accounting solution called Xero.

The QAP engineering team was asked to look into this SaaS, and it has been found that Xero has a well documented API, and that the data points that should allow us to be able to integrate with their service seem to exist. A decision was made to build this integration, and the pub has decided to go ahead and sign up for QAP to use our AP Automation with their accounting platform, as soon as the integration is ready.

Exercise

As mentioned in the background, two mandatory data points needed for any **Invoice** to be processed are the **Vendor** (who to pay), and the **Account** (where to pay from).

Create a free Xero developer account (https://developer.xero.com/) and load it with their sample data/create a sample company.

Then design a solution to extract both the list of **Accounts**, and the list of **Vendors**, from the sample company, and store them on disk.

Guidelines:

- You are expected to write code using the technical stack mentioned in our job description (React, PHP)
- Your solution should be an effort in being production ready. This could mean any of:
    - well tested
    - useful logging
    - Documented

Please include any:

- business requirements you may have developed while scoping your solution
- system design documents and/or diagrams you may have created while designing your solution
- instructions needed to get your code up and running on a production server (this is really important - your code will be handed off to an engineer who has never interacted with you)

As part of the implementation, our expectation would be to have a simple Graphical user interfaces (web) + API.

This is not an exercise of what you do alone, but rather the holistic approach of how you did it, what your design considerations were, whether the approach was an approach that would be accepted in production-ready code, etc.