# Uphold Bot

## Assessment Challenge

Utilize the Uphold API to develop a proficient bot capable of providing timely alerts regarding price oscillations on a specific currency pair. Ensure that all code is securely stored in a private repository, which can be hosted on platforms such as GitHub, Bitbucket, or GitLab. Upon completion of the project, kindly grant access to the designated email addresses mentioned in the Contacts section below and include a direct link to the repository housing the project.

Our API documentation is available [here](https://uphold.com/en/developer/api/documentation/).

*Note: If you have the desire to deliver an enhanced bot, you have the option to tackle both the optional and bonus items. It is not mandatory to complete the optional phase in order to choose a bonus item, and you are also welcome to make any additional improvements that you deem necessary.*

## Requirements

You can use any [npm](https://www.npmjs.com/) package you see fit to solve this problem.

* *Language:* ECMAScript 2023
* *Node.js:* > v20

## 

## Phase 1 (mandatory)

* To ensure smooth execution of the bot, it is essential to create a README.md file in the project root, providing comprehensive instructions on how to run it. Please ensure the README.md includes all necessary setup and execution instructions, avoiding any implicit prerequisites. By doing so, anyone accessing the project will be able to understand the requirements and easily follow the steps to run the bot successfully.
* Your task is to establish a connection with the Uphold public ticker and retrieve the BTC-USD rate at regular intervals of 5 seconds. Upon each rate retrieval, the bot should compare it with the last rate that triggered an alert and determine whether to notify about an oscillation. For this exercise, the requirement is to generate an alert (a simple log is sufficient) when the price undergoes a change of 0.01 percent in either direction (i.e., when the price increases or decreases).

## 

## Phase 2 (optional)

* Enable the bot to handle multiple currency pairs concurrently, allowing it to monitor and analyze several pairs simultaneously.
* Accept all parameters, such as currency pairs, fetch interval, price oscillation percentage, etc., as arguments to enhance flexibility and customization.
* Implement a comprehensive test suite for your code, utilizing popular testing frameworks such as Jest, Vitest or Mocha, ensuring the reliability and accuracy of the bot's functionality.

## Phase 3 (bonus)

* Dockerize your application to facilitate easy deployment and ensure consistent execution across different environments.
* Implement a database, such as Postgres, to store all the alerts generated by the bot. This will enable persistent storage and easy retrieval of historical alert data.
* Store relevant information in the database for each alert, including timestamps, rates, and the bot configuration at the time of the alert. This allows for comprehensive tracking and analysis of the alerts over time.

## 

## 

## 

## Contacts

Feel free to reach out with any questions!

Ricardo Gonçalves - [ricardo.goncalves@uphold.com](mailto:ricardo.goncalves@uphold.com)

Sandro Machado - [sandro.machado@uphold.com](mailto:sandro.machado@uphold.com)