Mobile Exercise

General Rules

- You have up to 2 weeks to complete the assignment.
- The project should be developed in a **native** language of the platform you are applying for.
- The project should be available in a **private** repository.
    - Add the user *nubank-code-reviewer* as a contributor (available on github, bitbucket and gitlab)
- You are free to use third-party open-source libraries.

Overview

As a payment company, we know that purchases can lead to problems. That's why being able to chargeback those purchases in an intuitive way is a great deal for our customers.

The goal of this exercise is to implement a minimalist chargeback flow (**that does not cover all real cases**).

The Exercise

Take some time to look at the UI specification:

https://www.dropbox.com/sh/rb0o5a8wfodqi2t/AABWhfHcMjvwc_zuI-tLCTFya?dl=0

The UI is fully defined by the provided images, but there are some important things to talk about:

1. Names (will be important to understand the API)

1.1 First Screen is called **Notice**

1.2 Second Screen is called **Chargeback**

2. API

2.1 There is just one static endpoint in this exercise, which is a hypermedia API: https://nu-mobile-hiring.herokuapp.com

2.2 From that endpoint, you'll **discover** all the other endpoints. Take some time to click the link above and start navigating through the endpoints to get the feeling of how this works.

2.3 All **actions** are of type **POST**. To make it easier for you, blocking/unblocking the card and submitting a chargeback are the only actions in this exercise.

2.4 All other requests should be called with a **GET** request

3. Blocking and Unblocking the card

3.1 You can assume that when the Chargeback screen opens, the card is **unblocked**.

3.2 If the field **autoblock** provided by the chargeback endpoint is true, then you **must** make a **POST** to **block** the card (that is: don't wait for the user to block it).

3.3 The user can block and unblock the card at will, by clicking on the padlock cell.

3.4 The texts of card blocked and card unblocked are static.

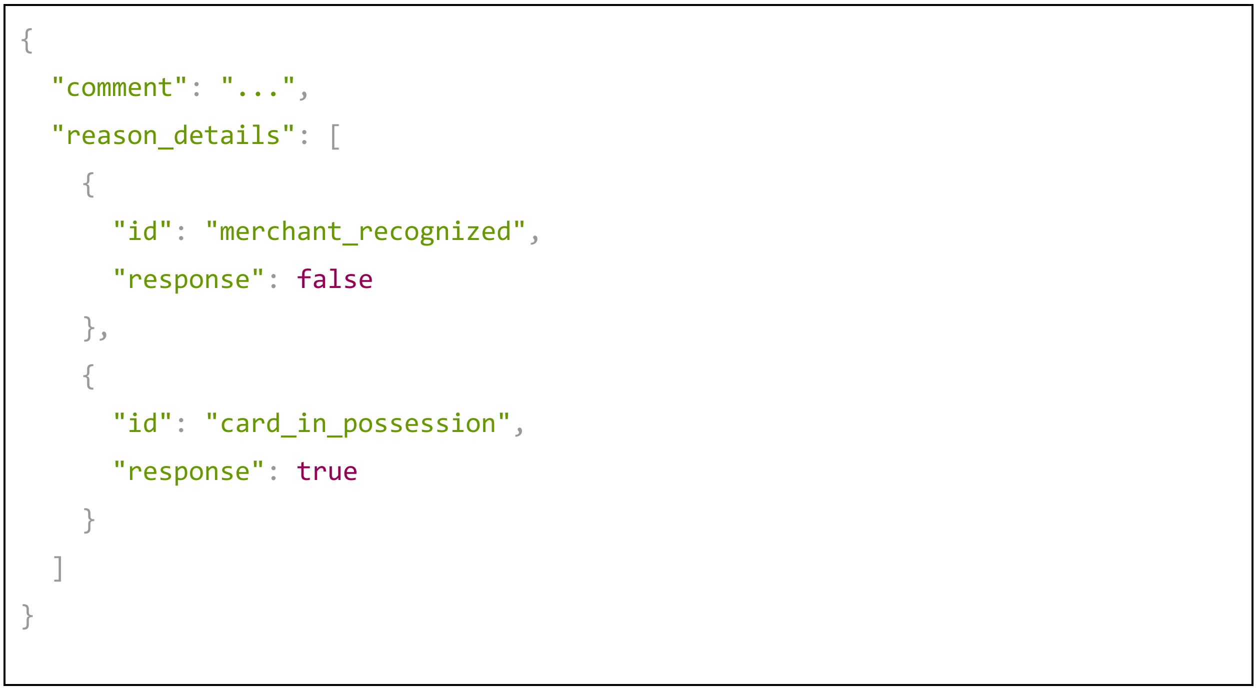
3.5 You don't have to send any parameters to block or unblock the card.

4. Submitting the Chargeback

4.1 Take a look at the chargeback json structure. Understanding it is a requirement for the next steps.

4.2 As you probably noticed by now, **no data on the first *two* screens are static, it's all filled by the API**. To submit the chargeback you should pass a json object to the endpoint described at "self".

4.3 The chargeback post request json structure consists of two fields: **comment** and **reason_details**



```
{
  "comment": "...",
  "reason_details": [
    {
      "id": "merchant_recognized",
      "response": false
    },
    {
      "id": "card_in_possession",
      "response": true
    }
  ]
}
```

5. Requirements

- Your code should be 100% **crash free.**
- You **must** include tests.
- You **must** include guidelines to setup, compile and run your code in your Readme.md file.

6. Tips

6.1 It’s a good practice to alert the user whenever the application is processing or fetching data.

6.2 APIs can return errors sometimes, it’s a good thing to be ready to handle them.

Take time to digest everything and plan how you'll do things before you start. A **good project** is made from **good planning**.

If you have any questions, please do not hesitate to contact us, we'll be more than happy to help you!