Wompi

Wompi Back End Dev Test

This test is intended to evaluate various aspects of back end development, like code clarity, API / endpoints definition, design simplicity, test coverage or correct use of the HTTP protocol, in a real-life like scenario.
You are in charge of:

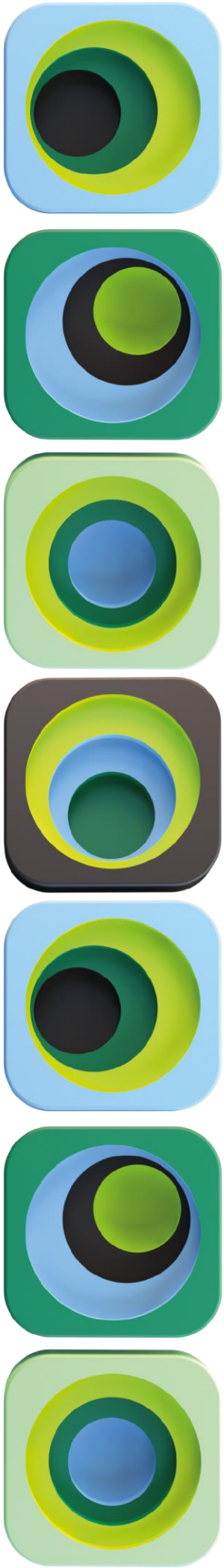
- The overall **design** of the API and information architecture (DB schema, folder structure, etc)
- Deciding what data should be **requested/responded** on each endpoint
- Deciding what **validations** each endpoint should perform, thinking of real-life situations that could arise
- Safely handling **sensitive data**.

The story

- In this test you must build a JSON RESTful API, for a small ride-hailing service, that will use the Wompi API (https://docs.wompi.co/docs/colombia/inicio-rapido/) for monetary transactions. For starters, you must:

- Create an account at https://comercios.wompi.co • You can use your own personal information
- Make sure you use the **Sandbox** environment. That means, there's NO need for any real money transactions, in the context of this test.

The API you create must have 2 different types of users, **rider** and **driver**. They must perform different types of requests, listed below:



Wompi

Rider:

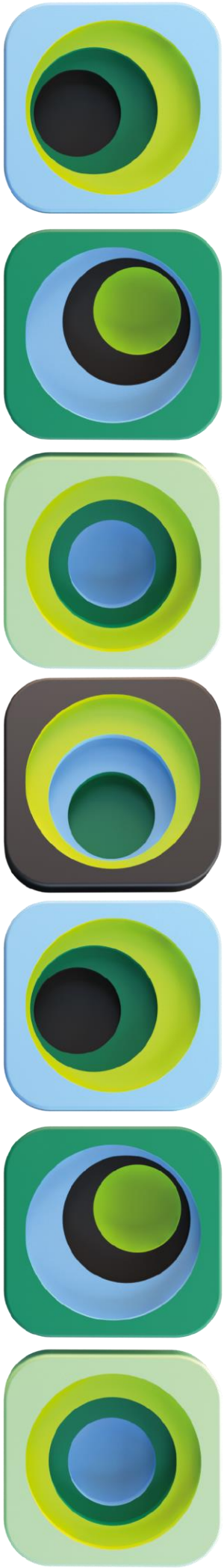
- **Create a payment method for a rider:** Using a PREVIOUSLY tokenized card (something out of the scope of this test; you can tokenize one manually beforehand), and thus creating a **Payment Source** (https://docs.wompi.co/docs/colombia/fuentes-de-pago) in order to make **multiple transactions** in the future, so that the rider doesn't have to input credit card data every time (like in real-world apps where you input card data just once). Please NEVER request credit card information in your API, this is a risky and potentially illegal thing to do.
- **Request a ride:** Sending the user's current location (latitude, longitude), among other data you consider necessary. The API must immediately:

  1. Assign a driver (unlike real-world situations where the driver is assigned moments later)
  2. Start a **ride**

Driver:

- **Finish a ride:** Given its ID and final location (assume the ride is a straight line in the map). It must immediately:
  - Calculate the total amount to be paid, given these fees:
    COP $1000 for each km
    COP $200 for every minute elapsed
    COP $3500 base fee, added always
  - Create a transaction, using the Wompi API and charging the user the total amount

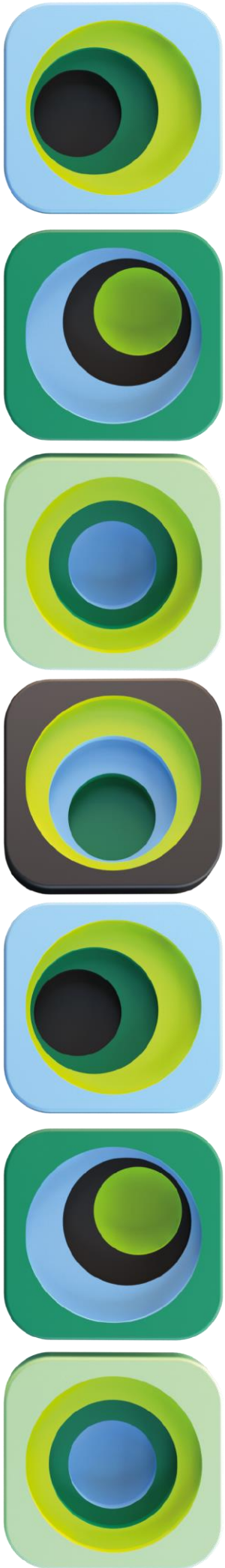
Medellín- Colombia
01 8000 129 999
ayuda@wompi.co



Wompi

General considerations:

- You **CANNOT USE** Ruby on Rails. We recommend using other frameworks like **Grape** or **Sinatra**.
- **Business logic** must NOT be handled within the routing / controller layer, but on a separate layer
- You **must use** the Dry Validation gem (https://dry-rb.org/gems/dryvalidation/1.4/) for performing validations on all endpoints. Hint: Do NOT validate data on the routing/transport layer of the API (Grape validations are a **no go**, for example)
- You can use **any type of database**. We recommend using PostgreSQL, as Heroku supports it.
- You can use **any type of ORM** and serialization library. We recommend using **Sequel.**
- The database must be **seeded** with dummy **Drivers**, with their current location (so that there are drivers to choose from whenever a rider requests a ride), and **Riders**. There's **NO need** for an endpoint to create new drivers or riders.
- Your code must have **unit and integration tests**, written in your library of choice (RSpec, minitest, etc.)
- You must push your code to **Github as a public repository**. Other services such as BitBucket, etc. are NOT allowed. Please do NOT use words like **"Wompi"** (company name), in your repository.
- You must write a **README** file (in english) with instructions for:
    - Running the code locally



Wompi

- Running the tests
- You must deploy your app.

Expected time of completion:

We expect you to dedicate a **maximum** of 25 working hours to complete this test.

Medellín- Colombia

01 8000 129 999

ayuda@wompi.co