# Multi-Merchant Transactions Challenge



## Whoami


Kelvin Kamara
www.kelvinkamara.com
www.github.com/kkamara
kelvinkamara@protonmail.com


## Problem

We want to build a solution to reliably link transactions from various integrations for the transactions page.

## Solution

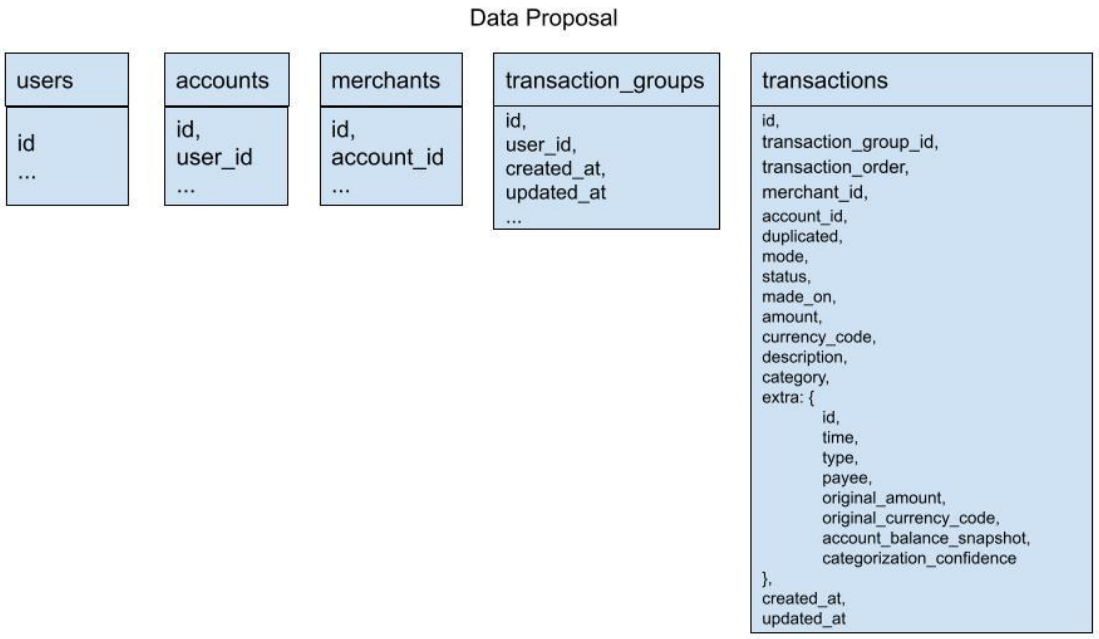
### Tldr

Each transaction data set to have transaction_group_id, transaction_order and merchant_id fields.

We will remove those hidden fields from displaying data to the end user. That's easily done by creating schemas in our code. When we pull the data from our database our code will omit those sensitive fields.

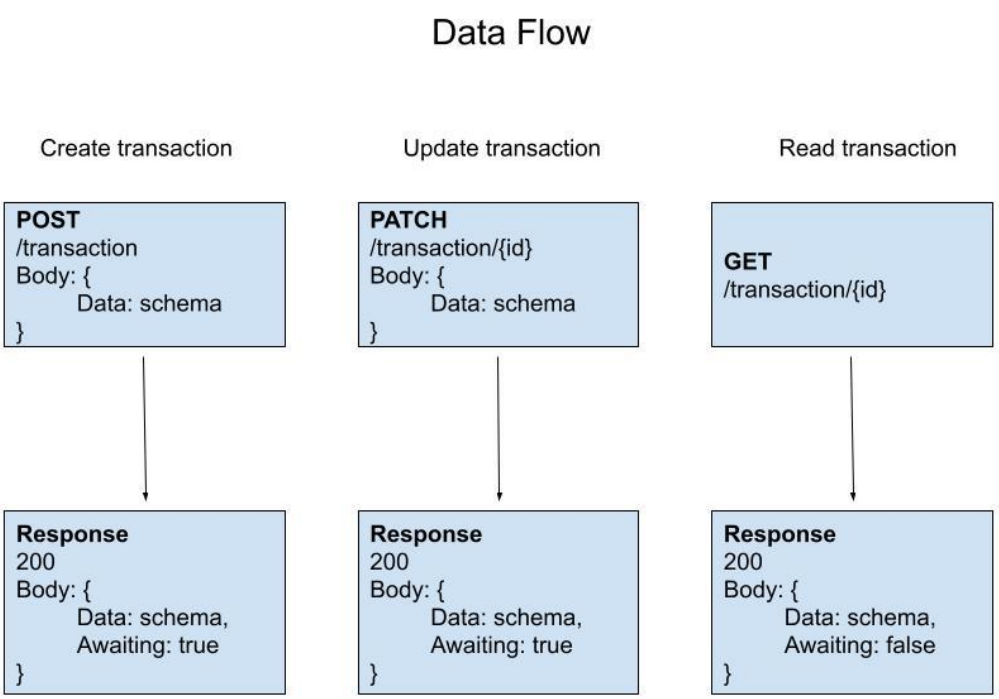
It makes sense for transaction schema to be tied to the merchant in which each transaction belongs. Meaning we can also write transaction schema for each merchant we work with.

# Data Proposal




From the transactions table we can see that the schema is quite large. We would omit various fields when storing and selecting data based on the merchant each transaction belongs to. That means some transactions fields will be optional.

## Data Flow




The above describes adhering to standard HTTP convention when building our RESTful api.

## How it works

Each transaction data set to have transaction_group_id, transaction_order and merchant_id fields. With transaction_group_id being a relationship object linking to our transaction_groups data table, transaction_order being the order in which the transaction took place and merchant_id relating to the merchants table.

We will remove those hidden fields from displaying data to the end user. That's easily done by creating schemas in our code. When we pull the data from our database our code will omit those sensitive fields.

When writing our code it makes sense for transaction schema to be tied to the merchant in which each transaction belongs. Meaning we can also write transaction schema for each merchant we work with. That means the data in our code will be the best representation of what is received from each merchant.

Each purchase will trigger a transaction_group item inserted to our database. Each order transaction will then belong to that transaction_group. That solves our problem to where we now have a solution we can bring to reality.