Coding Challenge

Version: 2.0

Introduction

Seen has provided you with an API to retrieve a set of sample transactions. Using this data, we want you to build a simple Node.js application that enriches this data.

- The application should be written in Typescript and use Node.js.

- Any framework can be used (i.e. Koa, Express, Nest) but your choice of framework should be appropriate to the task below. You can assume that the complexity and scope of the task will not significantly grow in future.
- Your submission should include tests
- Your submission should work. `npm start` should work.
- No frontend or database is required.

Your task

Seen has provided you with an API that contains sample transactions:

```
GET https://cdn.seen.com/challenge/transactions-v2.json
```

Using this data, build a Node application that serves the following two APIs:

API 1: Customer transactions

Inputs

A customer ID

Used by

The Seen mobile app

Objective

Individual transactions should be aggregated such that the latest status is shown. For example, a single purchase at Amazon may result in 2 transactions: one transaction for the *processing* phase and another once it is *settled* (see the reference below). Your API should only return 1 record here but include an array representing the timeline (see example below). The `relatedTransactionId` attribute or `authorizationCode` attributes will be helpful.

Returns

```
{
    "transactions": [
            {
                "createdAt": "2022-09-01T11:46:42+00:00", // bas
                "updatedAt": "2022-09-03T15:41:42+00:00", // bas
            "transactionId": 1, // use the first transaction
                "authorizationCode": "F10000",
                "status": "SETTLED",
            "description": "Amazon.com",
                "transactionType": "POS",
                "metadata": {},
                "timeline": [
                    {
                        "createdAt": "2022-09-01T11:46:42+00:00"
                        "status": "PROCESSING",
                        "amount": 5000.00
                    },
                    {
                        "createdAt": "2022-09-03T15:41:42+00:00"
                        "status": "SETTLED",
                        "amount": 5000.00
                    }
                ]
            },
            {} // other transactions
        ]
}
```

API 2: Related Customers

Inputs

A customer ID

Used by

An internal fraud monitoring tool

Objective

Customers may be related by device or by sending funds to each other.

Returns

```
{
    "relatedCustomers": [
        {
            "relatedCustomerId": 3,
            "relationType": "P2P_SEND"
        },
        {
            "relatedCustomerId": 5,
            "relationType": "P2P_RECEIVE"
        },
        {
            "relatedCustomerId": 3,
            "relationType": "DEVICE"
        }
    ]
}
```

How the challenge is evaluated

- Did you demonstrate a good understanding of the requirements?

- Did you solve the problem? Did you enumerate multiple different solutions and evaluate the merits of each?
- Did you use the relevant Typescript language features?
- How did you structure the APIs?
- How readable and maintainable is your code?
- How robust is your code to bad data and edge cases?
- Does your code have sufficient test coverage? Is it testing the right things?
- How well did you present your solution?
- Does your code have a README with instructions for getting it running?

Reference

The following information will be useful in understanding the data and building your solution.

Transaction Types

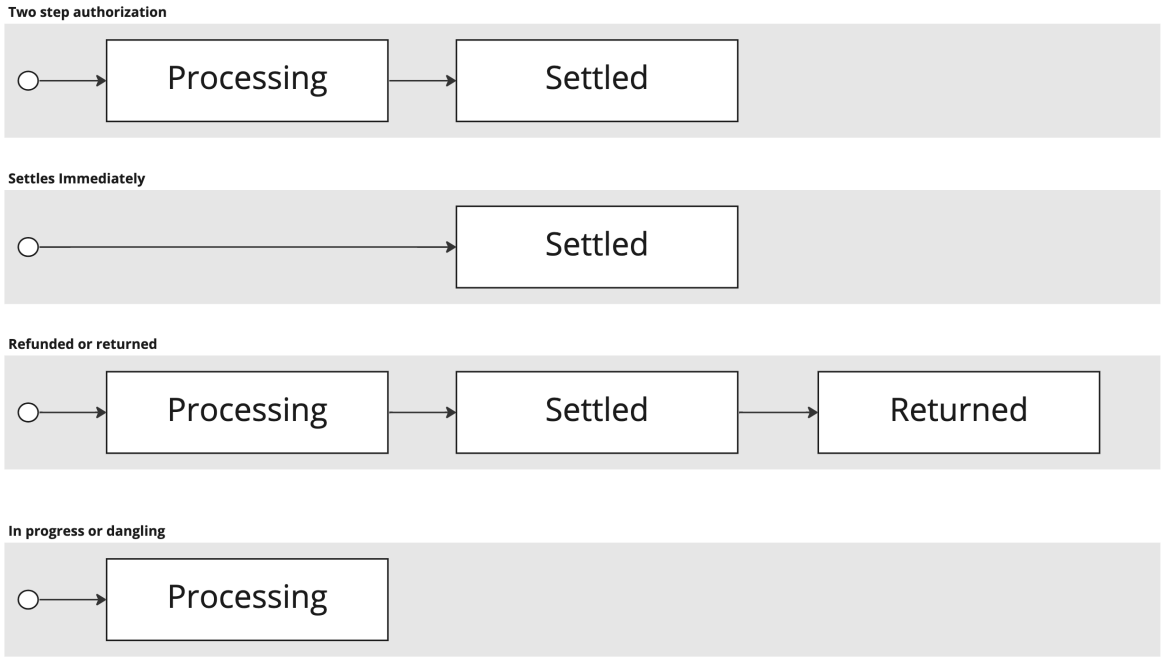
Every transaction has a `transactionType` attribute.

Transaction Type	Description	Originates in App
ACH_INCOMING	An ACH is a bank-to-bank transfer (similar to a Wire or a SEPA transfer). This specific transaction represents an ACH that is received by a customer (i.e. it is sent from another bank and received in their Seen bank account). Incoming ACH's can take up to 3 days to fully settle.	❌
ACH_OUTGOING	An ACH is a bank-to-bank transfer (similar to a Wire or a SEPA transfer). This	✅

Transaction Type	Description	Originates in App
	specific transaction represents an ACH that is sent by a customer	
WIRE_INCOMING	A Wire is another type of bank-to-bank transfer. This type represents a wire that is received by a customer. It settles immediately in most cases.	❌
WIRE_OUTGOING	A Wire that is sent from a Seen customer to another bank	✅
POS	A Point of Sale transaction using a card (either in person or online). POS transactions typically have two legs: an authorization phase and a final settlement phase. Sometimes the transaction can also be returned.	❌
P2P_SEND / P2P_RECEIVE	Customers can make instant transfers to other Seen members using the app	✅ (SEND)
FEE	Certain transactions incur an additional fee	Varies

Transaction Lifecycle

Transactions can enter different states as they are processed. The lifecycle varies between transaction types.



Fraud Signals

From past experience, the fraud team has determined the behaviour to be indicative of fraud:

- Money laundering: a customer receives funds and then immediately sends the funds out again
- Fraud rings: multiple customers perform transactions using the same devices
- Fraud rings or identity theft: customers receive money in multiple accounts. They then aggregate this money into a single account and then withdraw it.
- Account takeover: a customer makes transactions that differ from their regular behaviour
- Money laundering: customers make transactions at risky merchants (i.e. Crypto) or only use payment apps (Venmo, Zel, Paypal) without making any point of sale transactions.