

# Challenge: Teller API Sandbox

---

## Scenario

Developers would like to quickly test the Teller API without connecting a real bank account.

Having a sandbox that returns account and transaction data mimicking the same schema as that in production would be very useful.

## Task requirements

Implement a server that returns JSON encoded accounts and transactions (see sample resources below). The API should return data that matches the Teller API schema (<https://teller.io/docs>).

## Core requirements

Your application must respond to the following routes:

```
GET /accounts
GET /accounts/:account_id
GET /accounts/:account_id/details
GET /accounts/:account_id/balances
GET /accounts/:account_id/transactions
GET /accounts/:account_id/transactions/:transaction_id
```

Your application must require authentication using an API token supplied using HTTP Basic Auth in the username field. The password field should be left empty. Token must be an opaque, URL safe string with a prefix of "test\_". The format of the API token is left to you, i.e. you are free to encode any state inside the token that you need.

Given the same API token your server returns the same data each time a request is made, meaning the same account(s) with exactly the same account information, and exactly the same feed of transactions, even after application restarts. See below for account names and institutions.

Your application may return one or more accounts for a given API token.

The transactions endpoint should return a list of transactions going back 90 days. There should be 1 transaction per day.

Each transaction should have the `type` "card\_payment", a `description` of one of the merchants names provided below, and a negative `amount`, i.e. money debited from the account.

Each transaction should have a running balance, i.e. the sum of transactions up to the given transaction applied to the opening balance.

The links in the JSON responses should map to your application's base URL.

Your application must not use a database/genserver/agent/file system or any persistence. It must be completely stateless and all data must be procedurally generated/chosen.

Your application should have an automated test suite.

You should include a readme informing the reviewer on how to use the applicaion.

## Optional requirements

A simple dashboard using Phoenix Live View to display some sort of metric associated with your API. For example the number of requests made per account.

Instead of 1 transaction per day, there should be between 0 and 5 transactions on the ledger per day.

As time progresses your application should return new transactions for each calendar day that passes. As above, between 0 and 5 per day.

Transaction pagination controls. `count` - The ability to limit the maximum number of transactions returned in the API response. `from_id` - The transaction from where to start the page. The first transaction in the API response will be the one immediately before the transaction in the ledger with this id.

Pagination controls are given as query params to the request URL.

```
GET /accounts/:account_id/transactions?count=5&from_id=txn_id
```

## Example Resources

### Account Resource

```
{
  "currency": "USD",
  "enrollment_id": "enr_nmf3f7758gpc7b5cd6000",
  "id": "acc_nmfff743stmo5n80t4000",
  "institution": {
    "id": "citibank",
    "name": "Citibank"
  },
  "last_four": "3836",
  "links": {
    "balances": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000/balances",
    "details": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000/details",
    "self": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000",
    "transactions": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000/transactions"
  },
  "name": "My Checking",
  "subtype": "checking",
  "type": "depository"
}
```

## Account Details Resource

```
{
  "account_id": "acc_nmfff743stmo5n80t4000",
  "account_number": "891824333836",
  "links": {
    "account": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000",
    "self": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000/details"
  },
  "routing_numbers": {
    "ach": "581559698"
  }
}
```

## Account Balances Resource

```
{
  "account_id": "acc_nmfff743stmo5n80t4000",
  "available": "33648.09",
  "ledger": "33803.48",
  "links": {
    "account": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000",
    "self": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000/balances"
  }
}
```

## Transaction Resource

```
{
  "account_id": "acc_nmfff743stmo5n80t4000",
  "amount": "90.54",
  "date": "2021-08-12",
  "description": "In-N-Out Burger",
  "details": {
    "category": "dining",
    "counterparty": {
      "name": "IN N OUT BURGER",
      "type": "organization"
    },
  },
  "processing_status": "complete"
},
{id": "txn_nmfo2gtnstmo5n80t4004",
"links": {
  "account": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000",
  "self": "https://api.teller.io/accounts/acc_nmfff743stmo5n80t4000/transactions/txn_nmfo2gtnstmo5n80t4004",
},
"running_balance": "33648.09",
"status": "posted",
"type": "ach"
}
```

## Merchants

```
[
  "Uber", "Uber Eats", "Lyft", "Five Guys", "In-N-Out Burger", "Chick-fil-A",
  "Apple", "Amazon", "Walmart", "Target", "Hotel Tonight", "Misson Ceviche",
  "Caltrain", "Wingstop", "Slim Chickens", "CVS", "Duane Reade", "Walgreens",
  "McDonald's", "Burger King", "KFC", "Popeye's", "Shake Shack", "Lowe's",
  "Costco", "Kroger", "iTunes", "Spotify", "Best Buy", "TJ Maxx", "Aldi",
  "Macy's", "H.E. Butt", "Dollar Tree", "Verizon Wireless", "Sprint PCS",
  "Starbucks", "7-Eleven", "AT&T Wireless", "Rite Aid", "Nordstrom", "Nordstrom Rack",
  "Bed, Bath & Beyond", "J.C. Penney", "Subway", "O'Reilly", "Wendy's",
  "PetSmart", "Dick's Sporting Goods", "Sears", "Staples", "Domino's Pizza",
  "Papa John's", "IKEA", "Office Depot", "Foot Locker", "Lids", "GameStop",
  "Panera", "Williams-Sonoma", "Saks Fifth Avenue", "Chipotle Mexican Grill",
  "Neiman Marcus", "Jack In The Box", "Sonic", "Shell",
]
```

## Merchant categories

```
[
  "accommodation", "advertising", "bar", "charity", "clothing", "dining",
  "entertainment", "fuel", "groceries", "health", "home", "income", "insurance",
  "office", "phone", "service", "shopping", "software", "sport", "tax",
  "utilities",
]
```

## Account names

```
["My Checking", "Jimmy Carter", "Ronald Reagan", "George H. W. Bush",
"Bill Clinton", "George W. Bush", "Barack Obama", "Donald Trump"]
```

## Institutions

The following is a list of institution names. Create the institution ID by snake casing the name.

```
["Chase", "Bank of America", "Wells Fargo", "Citibank", "Capital One"]
```