# Transactions page challenge

### Goal

Build a page showing a user's Brex Card transactions. We want to get a sense for how you:

- Understand a problem and architect a solution
- Prioritize work and make trade-offs with time constraints
- · Collaborate technically with others

# Agenda

### Part 1: Coding

You will be coding with three frontend engineers for 45 minutes each. We'll be focusing on the following areas over the course of the interview:

- JS, CSS, and HTML
- Architecture
- Data manipulation

#### Part 2: Demo, Q&A

This is your chance to meet the team and ask questions. We will:

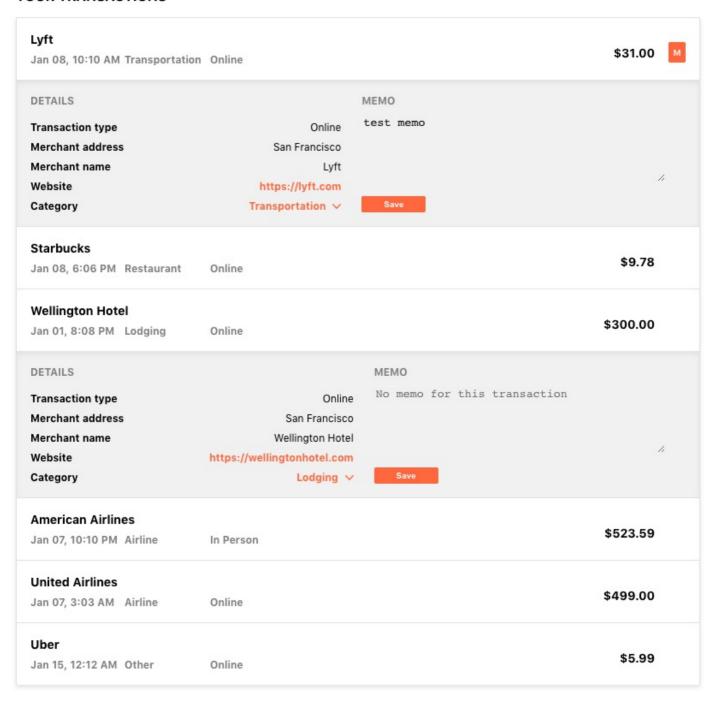
- Walk through what you've built
- Do a question and answer session about your code
- Discuss any questions you might have about Brex or the team

# Requirements

- Build a list of transactions from data received from the provided API
- · Each transaction must:
  - Be styled according to the included design specification
  - Display an orange M icon when it has a memo
  - Display the transaction date in the proper format
    - External library use is OK
  - Display the transaction amount with 2 decimal places
  - Toggle a drawer that displays more transaction information when it is clicked
    - Multiple drawers can be open at the same time
  - Have a clickable link to the transaction merchant's website
  - Display a dropdown of all available categories with the current category selected
    - The standard <select> element is OK
  - Have a memo input that accepts multiline text
- When a user changes the value in the transaction category dropdown, it must be updated using the setTransactionCategory API
- When a user changes the transaction memo and clicks on the Save button, it must be updated using the setTransactionMemo API

# Design

#### YOUR TRANSACTIONS



### Data API

- We will provide a TransactionAPI.js file that exports the necessary APIs.
- There are 4 API functions: getTransactions, getCategories, setTransactionCategory, and setTransactionMemo. Their types and the types of the data they return are listed below.
- All data is persisted to localStorage on each call to setTransactionCategory and setTransactionMemo.
   Subsequent calls to getTransactions and getCategories will show the latest changes, even on page refresh.

```
type Transaction = {
 id: string;
 categoryld: string;
 accrualDate: string;
 amountInCents: number;
 merchant: {
  name: string;
  website: string;
  address: string;
 };
 memo: string I null;
 captureMethod: "Online" I "In Person";
};
type Transactions = { [id: string]: Transaction };
type Category = {
 id: string;
 name: string;
};
type Categories = { [id: string]: Category };
function getTransactions(): Promise<Transactions>;
function getCategories(): Promise<Categories>;
function setTransactionCategory(
 transactionId: string,
 categoryld: string,
): Promise<Transaction>;
function setTransactionMemo(
 transactionId: string,
 memo: string,
): Promise<Transaction>;
```