

ManageSocial Tech Task

Time: Approx. 1 - 2 hours

Note: We value the aesthetics of the code that applicants write. Consistency of style, code and tests that are as succinct as can be. Use variable names and comments that are appropriately descriptive.

User Story

As a user I would like to make request to an API that creates a customer in a Stripe.com account, applies charges to this customer and can return a list of charges that have been applied.

Acceptance Criteria

- Given that I have made a json request to `POST /customers` endpoint I should receive a response with my Stripe customer details
- Given that I have made a json request to `POST /charges` endpoint I should receive a response with the charge details
- Given that I have made a request to `GET /charges?customer_id=<id>` endpoint I should receive a response with the list of charges for the supplied customer

Requirements

- Written using ES6 and demonstrating a knowledge of promises and using `async/await` or generators for using the asynchronous requests in a synchronous control flow
- The application reads the Stripe API key from `.env` in the root of the repository containing `STRIPE_KEY=sk_test_useyourkey`
- Create a Stripe account to test with, we will use our own test account when examining the submission
- Must be run with `npm start` and print address and port information to the console
- Tests must be written using `mocha` and run with the `npm test` command
- The test run should also check code is linted properly according to the definition which you are to provide in the repository
- The application only exposes the above 3 endpoints and uses `express` or `koa`
- Perform some validation of the input and provide documentation in a readme for the required and optional data that can be sent to the endpoints
- A series of integration tests are required which performs the following:
 - creates a user with id in the format `test_user_1488776945864` where the timestamp is `now()`
 - creates 3 charges against this user that total \$50.00 USD
 - retrieves a list of charges for this user and verifies the amounts

Submission

The application should be committed to a public repository on GitHub or BitBucket with the repository being the MD5 hash of the string `<your github/bb name>-billing-task` and simply send us the link to the repository. E.g. if `github.com/twitter` was doing this task their string would be `twitter-billing-task`.

NB: Please be verbose with your commit messages.