

WELCOME TO MONEY JAR'S TAKE-HOME ASSIGNMENT

This assignment is designed to give you an idea of the work you'll be facing day-to-day in this role, while also providing you the opportunity to showcase your thought process and skill set. Your solution will never be used for any purposes beyond evaluation.

We hope you find this experience challenging but enjoyable. Before we begin, please read the following two points, and do not proceed unless you agree to them.

1. I understand that I am encouraged to use both online and offline resources to inform my solution, but I promise not to plagiarize or copy the work of others. I also promise not to discuss the details of this assignment with others, nor seek their assistance in completing it.
2. I understand that I own the IP rights to my solution and am granting Money Jar the license to use my solution for evaluation purposes. My work will never be shared with third parties.

If you agree to the above, please proceed with the assignment below.

A BRAND-NEW BANK

Your assignment is to build an internal API for a fake financial institution.

BRIEF

While modern banks have evolved to serve a plethora of functions, at their core, banks must provide certain basic features. Today, your task is to build the basic HTTP API for one of those banks! Imagine you are designing a backend API for bank employees. It could ultimately be consumed by multiple frontends (web, iOS, Android etc).

TASKS

- Implement assignment using:
 - Language: The language you're most comfortable with
 - Framework: Any framework
- There should be API routes that allow them to:
 - Create a new bank account for a customer, with an initial deposit amount.
 - A single customer may have multiple bank accounts.
 - Transfer amounts between any two accounts, including those owned by different customers.
 - Retrieve balances for a given account.
 - Retrieve transfer history for a given account.
- Write tests for your business logic.

Feel free to pre-populate your customers with the following:

```
[  
  {"id": 1, "name": "Arisha Barron"},  
  {"id": 2, "name": "Branden Gibson"},  
  {"id": 3, "name": "Rhonda Church"},  
  {"id": 4, "name": "Georgina Hazel"}  
]
```

You are expected to design any other required models and routes for your API.

EVALUATION CRITERIA

- Engineering best practices
- Correctness: does the functionality act in sensible, thought-out ways?
- Maintainability: is it written in a clean, maintainable way?
- Testing: is the system adequately tested?
- Documentation: is the API well-documented?

CODE SUBMIT

Please organize, design, test and document your code as if it were going into production – then create a zip or tar archive of the solution and send the archive on to Money Jar for evaluation, along with a paragraph describing your thought process and rationales behind any decisions you made while solving the problem.

All the best and happy coding,

The Money Jar Team