FORM3
FINANCIAL CLOUD

Coding Exercise – Back-end Developer

This exercise consists of two parts: design and implementation. It is designed to take a few hours, but there is no hard time limit. Please do not feel rushed. Questions are welcomed, so feel free to ask.

Design

Provide a design for part of a payments API. A list of payments might look like http://mockbin.org/bin/41ca3269-d8c4-4063-9fd5-f306814ff03f.

Design requirements:

- API should be RESTFUL
- API should be able to:
    - Fetch a payment resource
    - Create, update and delete a payment resource
    - List a collection of payment resources
    - Persist resource state (e.g. to a database)

Example use case:

As an example, a JavaScript client should be able to call the API and list a collection of payments (building a client is outside the scope of this exercise however).

Please submit your design as a PDF.

Implementation

Implement the above design. Java is preferred, but if you are unfamiliar with Java, another object-oriented language such as C#, Ruby or Go may be used.

Implementation requirements:

- You should use best practice, for example TDD/BDD, with a focus on full-stack testing
- Prioritize correctness, robustness, and extensibility over extra features and optimizations.
- Write your code with the quality bar you would use for production code.
- Try to simplify your code by using open source frameworks and libraries where possible

Submitting your exercise

Please commit and share your solution using a hosted Git version control system, such as http://github.com or https://gitlab.com