**Retail Bank Corp -> Smart-Save TDD Exercise (1 hour coding + 20 min Q&A)**

Retail Bank Corp is building a Smart-Save feature to their bank account app that allows customer to deposit and withdraw funds to a Smart-Save bank account at the till. This allows our customers to “Save as they go”, and for our customers to access cash back from their savings at the till if needed.

Your task as the lead developer in the team is to implement a full stack API solution using Java with SpringBoot2 and Reactor for the new Smart Save account.

**Smart-Save API Requirements**

**This functional sprint**  
Account : create, deposit, withdraw, balance

**Further requirements**

* Event sourcing
* Online statement query (date, transaction type, transaction amount, rolling balance)

**Our Developer Rules**

* Must use the TDD process
* Must not use abbreviations
* Must use Object oriented programming
* Must keep all objects small, highly cohesive and isolated
* Should use a functional/lamda programming style using Reactor framework where possible
* Should use dependency injection using Spring
* Could implement an event loop
* Could implement an event source

**Proposed strawman software architecture**

****

**Starter Project**

A starter project has been prepared for you, with the necessary Spring Boot and Reactor dependencies and some example code and tests.

**API Acceptance tests:**

**Given** a valid SmartSave account creation request with a valid customerId (and Authorization-X header containing a valid “customer-auth-token”) **When** the new account is created **Then** the *accountId* of the new account must be returned.

**Given** a newly created SmartSave account **When** the opening balance is queried **Then** it should return 0.00GBP

**Given** a newly created SmartSave account **When** a deposit of 120.22 GBP is made **Then** the balance should return 120.22GBP

**Given** a newly created SmartSave account **When** a withdrawal of 20.22 GBP is requested **Then** a BadRequest exception should be returned stating “Insufficient funds available”

**Given** a newly created SmartSave account with a deposit of 20.22 GBP and a subsequent withdrawl of 10.00GBP **When** the balance is queried **Then** it should return 10.22 GBP

**Given** a newly created SmartSave account **When** multiple deposits and withdrawals are made in random order **Then** the account balance should be returned correctly

**Given** a newly created SmartSave account, when a second request to create the account is received for the same customer it should return the existing SmartSave accountId.