**Problem:**

BestDeals Bank wants to build a Returns calculator for its bankers by subtracting the outflows from the inflows from each deal, ignoring the timings of the cashflow in the first release. To start off, the returns calculator will allow the user to calculate the returns for the following kind of deals

Simple Compound Interest, Return



Annual Simple Interest, return

*I = P \* r \* N*

The returns should multiplied with the currency’s Fx Rate to USD to calculate the $ returns. The tool should be extensible so that it can calculate the returns of a number of different kinds of deals with different inputs. The tool will also allow users to dynamically calculate the returns for every client by accumulating the returns in dollars across all its deals.

Please design and deliver a persistent server-side Java application with appropriate RESTFul Webservices for such an application. The solution will be graded on the basis of its design extensibility, Unit Testing, code quality, performance and appropriate usage of latest java developments.

Please bring a running application, code and unit tests to the interview for consideration for the role.

**Solution:**

Thank you very much for considering for the problem solving interview. I have reviewed the problem and provided solution.

**Technologies Used:** Core Java, Spring Boot, Gradle, RESTful Services, IntelliJ IDE, jUnit test cases, Behaviour Driven Development approach, Cucumber BDD framework and JSON.

**How to test:**

**JSON Service Testing:**

1. **Start the Spring Boot service by running the below class**

com.java.bestbank.calculator.services.CalculatorApp

1. Once the service get started, open any RESTer services client. I have used Crome App Advanced RESTer client for my testing. Please check the screenshot.
2. <http://localhost:8080/compoundService>

**Request:** {"calculatorBean":{"principal":5400.0,"time":3,"interestrate":2.4,"numberofTime":4}}

**Response:**

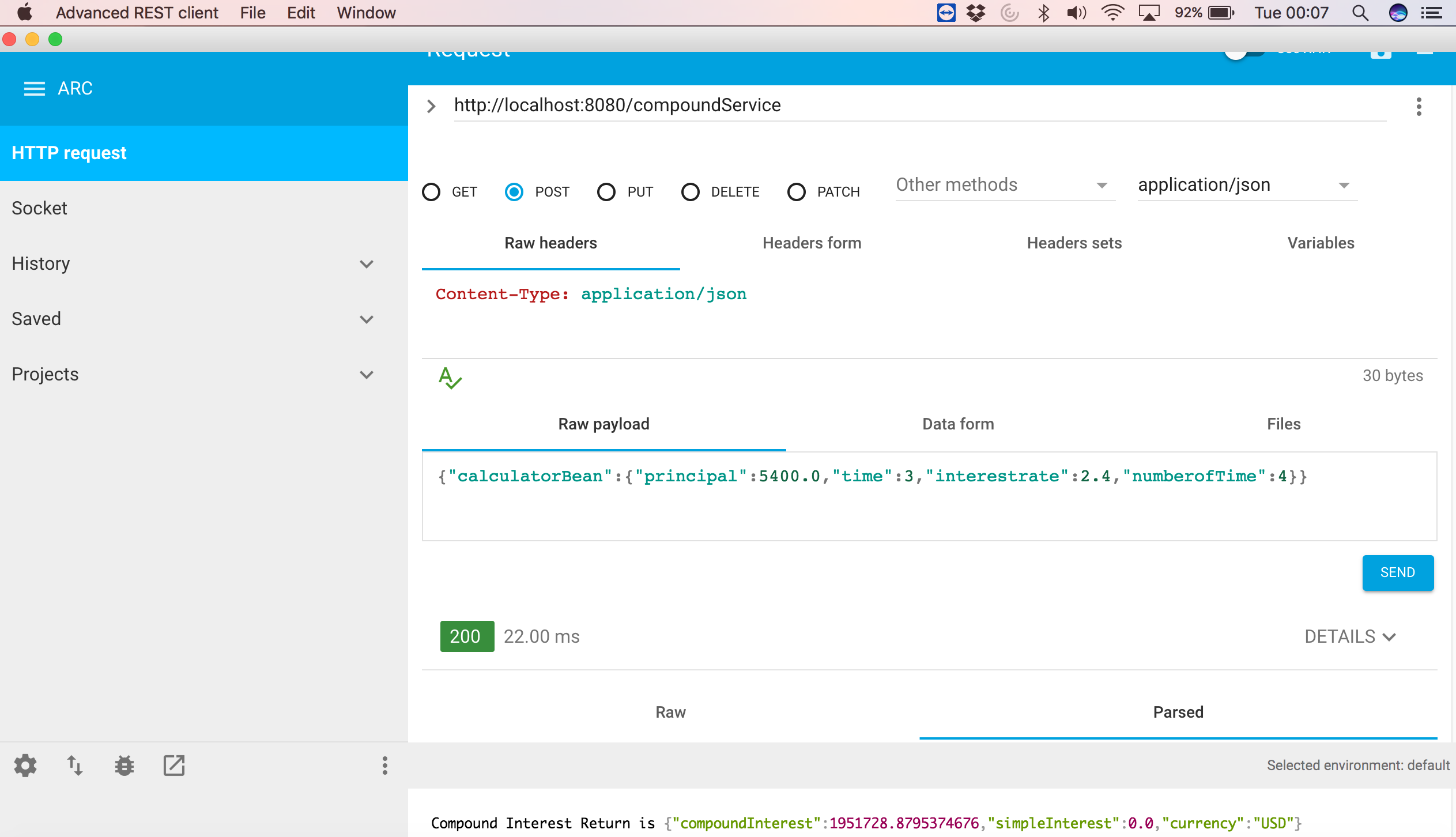
Compound Interest Return is {"compoundInterest":1951728.8795374676,"currency":"USD"}

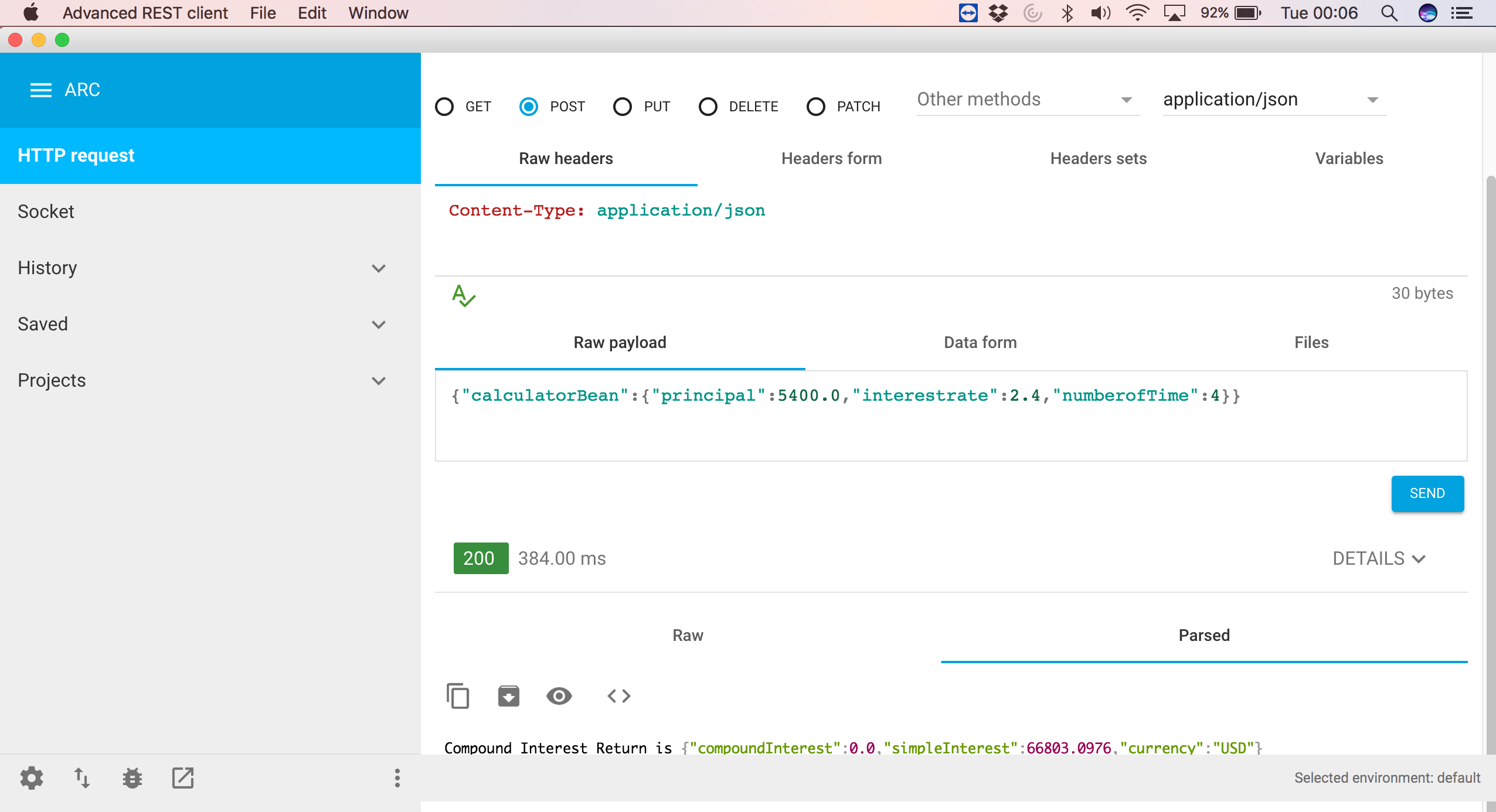
1. http://localhost:8080/simpleInterestService

**Request:** {"calculatorBean":{"principal":5400.0,"interestrate":2.4,"numberofTime":4}}

**Response:**

Compound Interest Return is {"simpleInterest":66803.0976,"currency":"USD"}





1. Stand-alone Gradle build and Test on Intellij

Run the below java class to run all the Cucumber BDD features.

com.java.bestbank.calculator.core.tests.ReturnsCalculatorRunner