OAuth Backend Mini Project (Bank Account)

The goal of this mini project is to write a simple web site to mimic a “Banking Portal”. Through this web site, one can query about the balance, deposit money, and withdraw money. Just like any Bank, there are restrictions on how many transactions/amounts it can handle.

For authentication and user identification, the web site will leverage Facebook’s OAuth service. Once the user logins in, all transactions will be recorded as this login user.

The architecture guideline is to have a very light frontend (simple HTML page with jquery) which uses AJAX to call the web service on the backend. The web service will also handle the OAuth processing with Facebook. Given the timeline for this project, any Open Source libraries to help with your implementation are encouraged.

The details are described below for the web service portion of the project:

- Write a simple “Bank Account” web service using REST API design principles. You can use either the Scala/Play or Java/Spring/Tomcat framework.
    - Program should have 3 REST API endpoints: Balance, Deposit, and Withdrawal
    - No requirement for the backend store - you can store it in a file or database (your decision)
    - Balance endpoint - this will return the outstanding balance
    - Deposit endpoint - credits the account with the specified amount
        - Max deposit for the day = $150K
        - Max deposit per transaction = $40K
        - Max deposit frequency = 4 transactions/day
    - Withdrawal endpoint - deducts the account with the specified amount
        - Max withdrawal for the day = $50K
        - Max withdrawal per transaction = $20K
        - Max withdrawal frequency = 3 transactions/day
        - Cannot withdraw when balance is less than withdrawal amount
- The service should handle all the error cases and return the appropriate error HTTP status code and error message (Eg. If an attempt is to withdraw greater than $20k in a single transaction, the error message should say “Exceeded Maximum Withdrawal Per Transaction”).
- Write tests against your web service. (**Bonus:** implement a code coverage tool and show code coverage numbers for your tests)
- Make sure your code is readable and can be run.
- Check in your code to github and write instructions on readme on how to run.

Share with us the github and we will review your project from there.