

Technologia: Java 8, Spring (boot, REST, JPA/Hibernate) , serwer SQL: PostgreSQL.

Aplikacja REST wyliczająca zyski z lokaty

Kapitalizacja odsetek w okresach: 1m-c, 3m-ce, 6m-cy, 12m-cy

Dwa algorytmy obliczania odsetek: - na koniec okresu lokaty / na dzień kalkulacji

Dla uproszczenia miesiąc ma zawsze 30 dni, rok 360.

Parametry lokaty oraz wynik kalkulacji przechowywane w bazie danych

Model danych lokaty: - nazwa, oprocentowanie, okres kapitalizacji, okres ważności (od - do)

Model danych kalkulacji: - kwota, data kalkulacji, lokata, algorytm wyliczania odsetek, zysk

Endpointy REST:

- GET api/investments – lista dostępnych lokat
 - wejście: brak
 - wyjście: header: 200, body: lista lokat z danymi zawężonym do id oraz nazwa
- POST api/investments – dodaje nową lokatę
 - wejście: zgodnie z modelem
 - wyjście: header: 204, body: id, nazwa, oprocentowanie, czas trwania w dniach
- POST api/investments/{id}/calculations – wylicza lokatę
 - wejście: kwota, algorytm odsetek
 - wyjście: header: 204, body: kwota, data kalkulacji, lokata, algorytm, zysk
- GET api/investments/{id}/calculations – informacje o kalkulacjach historycznych
 - wejście: brak
 - wyjście: header: 200, body: dane lokaty i lista kalkulacji (kwota, data kalkulacji, algorytm, zysk)

Testy jednostkowe dla napisanego kodu: logika + endpointy

Nie wymagamy warstwy widoku.

Dodatkowe punktowane funkcjonalności

- kalkulacji dla lokaty z przyszłości (liczona na podstawie daty okresu ważności lokaty) powinna kończyć się wyrzuceniem dedykowanego wyjątku oraz zwróceniem kodu odpowiedzi HTTP o wartości 400
- podczas dodawanie nowej lokaty wykonać sprawdzenia poprawności podanych data dla okresu ważności lokaty za pomocą adnotacji @Valid
- za pomocą programowania aspektowego w logu aplikacji powinny zostać zapisane informacje o fakcie wywołania REST endpointów zawierającego: typ metody wywołania (GET, PUT, ...), ścieżkę endpointa oraz kodu HTTP odpowiedzi
- poprawne wykonanie kalkulacji powinno inkrementować licznik wykonanych kalkulacji przechowywany jako dedykowane pole w tabeli lokaty. Inkrementację licznika zabezpiecz za pomocą transakcji Spring
- odczyt danych na potrzeby endpointa GET api/investments nie powinien zaczytywać powiązanych kalkulacji

Jako rozwiązanie proszę dostarczyć projekt z kodem źródłowym aplikacji.