

## Ćwiczenie 2 instrukcja

1. Ponownie ściągamy projekt z pierwszego ćwiczenia (*git clone <https://github.com/anicos/quickstart.git>*).
2. Przełączamy się na branch 'cw2'.
3. Do projektu został dodany nowy service `AverageExchangeRatesService`.
4. Serwis ten zawiera metodę `getCurrencyWithTheHighestCurrentRate` która, powinna pobierać walutę która aktualnie ma z najwyższy kurs średni w tym celu używamy zapytania GET <http://api.nbp.pl/api/exchangerates/tables/a?format=json>.
5. Serwis ten(`AverageExchangeRatesService`) powinien być zbudowany analogicznie do `GoldPriceService` czyli powinien być kompozycją `AverageExchangeHttpClient` i `AverageExchangeCalculator`.
6. W myśl zasady Single Responsibility `AverageExchangeHttpClient` jest odpowiedzialny za pobranie danych z serwera a `AverageExchangeCalculator` za wyliczenie waluty z najwyższym kursem średnim.
7. Każda klasa (`AverageExchangeCalculator`, `AverageExchangeHttpClient`, `AverageExchangeRatesService`) powinna posiadać testy analogicznie do serwisów związanych z ceną złota.
8. W myśl zasady TDD należy zacząć pracę od testu integracyjnego dla serwisu `AverageExchangeRatesService` następnie gdy już napiszemy test oraz implementację możemy przejść do refraktoru dodając nowe serwisy `AverageExchangeCalculator`, `AverageExchangeHttpClient`.

**Uwaga! Zadanie to będzie podstawą zaliczanie przedmiotu na ocenę pozytywną.**