

Feature 16 unikatowy pesel:

Wprowadzamy ograniczenie co do ilości kont, które można założyć na jeden pesel.

Jeżeli otrzymamy POST request o stworzenie konta z peselem, który został już użyty zwracamy kod **409** i odpowiednią wiadomość.

Jeżeli dopisaliśmy jakiś kod w folderze **app** aplikacji proszę pokryć go unit testami.

Dopisz testy API które sprawdzą, czy ograniczenie działa.

Feature 17 przelewy przez API

Wprowadzamy możliwość dokonywania przelewów wychodzących i przychodzących przez API.

Nasz endpoint POST powinien wyglądać w następujący sposób:

/api/accounts/<pesel>/transfer

przykład body:

```
{
  amount: 500,
  type: "incoming"
}
```

- Jeżeli konto o podanym peselu nie istnieje zwracamy 404.
- Obsługujemy "incoming", "outgoing" oraz "express". Zastanów się jaki kod zwrócić gdy dostaniemy nieznaną "type".
- Jeżeli konto istnieje wywołujemy odpowiednią metodę i zwracamy 200 z informacją "Zlecenie przyjęte do realizacji" o ile przelew się udał
- Dla przelewów wychodzących - status 422 gdy transakcja się nie udała (skąd wiemy czy przelew się udał?).

Napisz testy API do tej funkcjonalności.

Jakie przypadki testowe widzisz? (przynajmniej 5)

Co powinny sprawdzać nasze testy?

Podpowiedź: Do organizacji testów możesz użyć metod setUp (tworzenie konta) oraz tearDown (usuwanie konta)

Przemyśl:

- które testy na przelewy były łatwiejsze do napisania (unit czy api)?
- jak można uprościć (lub rozbić) endpoint transfer?