



## Implementacja DDD .NET - Zadanie 6

Bottega IT Minds

# 1. Zadanie 6 - Agregat jako Dokument

## 1.1. Wprowadzenie

Persystencja Agregatu z wykorzystaniem ORM to tylko jeden ze sposobów. Kolejnym sposobem jest traktowanie Agregatu jako Dokumentu i zapisanie go w bazie w postaci zserializowanej.

W module *Scoring* powstał (**obok agregatu persystowanego przez ORM**) agregat **CustomerScoringDocument**, który jest persystowany jako dokument. Do tego celu została użyta biblioteka *Marten*, która umożliwia pracę z bazą relacyjną PostgreSQL jakby była bazą dokumentową.

Jako, że agregat jest w postaci dokumentowej, podczas odczytu możemy z niego skorzystać, jednak ważne, aby przekształcać go do snapshot'u (snapshot pattern) - nie chcemy agregatów używać bezpośrednio do odczytów.

```
public class CustomerScoringDocument : DocumentEntity
{
    public Guid CustomerId { get; private set; }

    [JsonProperty]
    private Money LoanLimit { get; set; }

    private CustomerScoringDocument()
    {
        // Only for Marten.
    }

    --- logika biznesowa

    public CustomerScoringDocumentSnapshot ToSnapshot()
    {
        return new CustomerScoringDocumentSnapshot(CustomerId, LoanLimit.Amount,
LoanLimit.CurrencyCode);
    }
}
```

## 1.2. Treść Zadania

Na bazie modułu *Scoring* dodaj w module *Availability* implementację przypadków użycia

1. Tworzenie Zasobu
2. Pobranie Zasobu
3. Zablokowanie Zasobu

tworząc kopię encji **Resource** i zapisując ją i odczytując jako Dokument.

Dopisz osobne testy integracyjne (jednostkowych nie musisz - zachowanie jest takie samo), które sprawdzą poprawne działanie persystencji.

Zastanów się nad różnicami w obu podejściach (ORM vs Dokument).