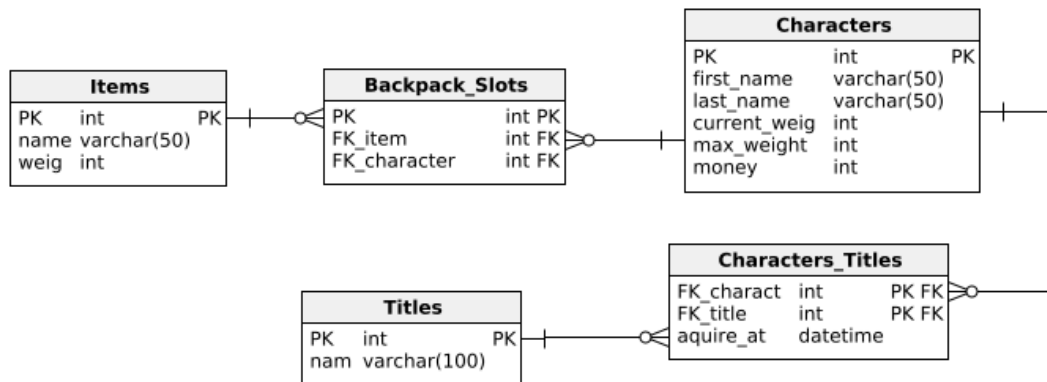


Kolowkium 2

Baza danych

Poniżej zaprezentowany jest diagram bazy danych wykorzystywanej w poniższych zadaniach.



Końcówki

Get

Zaprojektuj końcówkę, która będzie zwracać informacje o postaci o danym id.

Końcówka powinna reagować na zapytanie pod adres `api/characters/{characterId}` np.
HTTP GET `http://localhost:5000/api/characters/1`

Przykładowe dane jakie zostaną zwrócone:

```
{
  "firstName": "Siwy",
  "lastName": "Gerald",
  "currentWeight": 100,
  "maxWeight": 300,
  "money": 100,
  "backpackSlots": [
    {
      "slotId": 1,
      "itemName": "Miecz na potwory",
      "itemWeight": 50
    },
    {
      "slotId": 2,
      "itemName": "Miecz na ludzi",
      "itemWeight": 50
    }
  ]
}
```

```

],
"titles": [
  {
    "title": "Biały Wilk",
    "acquireAt": "2023-06-03T09:37:52.0951587"
  },
  {
    "title": "Gwynbleidd",
    "acquireAt": "2023-06-03T09:37:52.0951587"
  }
]
}

```

Post

Zaprojektuj końcówkę, za pośrednictwem której będzie można dodawać do ekwipunku danej postaci nowe przedmioty. Przedmioty do dodania przekazujemy jako lista id, które odpowiadają przedmiotom w bazie.

Kończówka powinna reagować na zapytanie pod adres `api/characters/{characterId}/backpackslots` np.

HTTP POST `http://localhost:5000/api/characters/1/backpackslots`

1. Sprawdzamy, czy wszystkie przedmioty istnieją w bazie danych.
2. Sprawdzamy, czy postać ma dostateczną ilość wolnego udźwigu, aby dodać wszystkie nowe przedmioty.
3. Dodajemy przedmioty do ekwipunku postaci oraz aktualizujemy aktualną wagę przedmiotów postaci.

Przykładowe dane, które przyjmuje końcówka:

```

[
  1,
  2,
]

```

Przykładowe dane, jakie zwróci końcówka:

```

[
  {
    "slotId": 1,
    "itemId": 1,
    "characterId": 1
  },
  {
    "slotId": 2,
    "itemId": 2,
    "characterId": 1
  }
]

```

Punktacja

Każda końcówka jest wyliczana na **10 punktów**. Czyli za całość można zdobyć **20 punktów**.

Punkty mogą być odjęte za:

- Niepoprawne lub nieczytelne nazwy zmiennych: **-2 pkt**
- Niepoprawne kody HTTP: **-4 pkt**
- Brak wydzielenia logiki biznesowej do oddzielnej warstwy: **-6 pkt**
- Brak wstrzykiwania zależności: **-8 pkt**
- Brak walidacji request modelu (jeżeli jest ona wskazana): **-2 pkt**
- Nieodpowiednie odwzorowanie bazy danych: **-4 pkt**

Uwagi

Kod należy pushować do prywatnego repozytorium na githubie oraz udostępnić je prowadzącemu. Rozwiązanie powinno zostać umieszczone na platformie **do końca zajęć**. Projekty umieszczone po czasie nie będą brane pod uwagę podczas sprawdzania (!).

- Kolokwium powinno być rozwiązane przy użyciu podejścia **Code-First**
- Oddanie niekompilującego się kolokwium - **0 pkt**
- Wykrycie plagiatu - **2 z przedmiotu** bez możliwości poprawy