



Sprawozdanie

Projekt REST API

Programowanie we frameworkach internetowych

Wykonał:

Imię i nazwisko Bartosz Wiśniewski

Nr indeksu 78331

Grupa L3

Rok Akademicki 2022/2023

Prowadzący

Mgr. Łukasz Szeremeta

KinoAPI

Aplikacja REST'owa udostępniająca API kina.

0. Wprowadzenie

Użytkownik aplikacji Kina, może odpytać o rozkład filmowy lokalnego Kina (CinemaDTO). Dodatkowo może w on korzystać z CRUD'a na oddzielnej bazie filmów oraz forum kina. Forum zabezpieczone jest za pomocą BasicAuth.

Konta wbudowane

-Admin

Login:admin

Hasło:admin

-User

Login:user

Hasło:user

Użytkownik może dowolnie operować na swojej bazie danych filmów.

Jeśli chodzi o forum to użytkownik może tworzyć fora i dodawać do nich posty, może edytować i usuwać posty stworzone przez siebie. Administrator ma pełną kontrolę.

1. Użyte technologie

- a. .NET Framework 6 WEB API - framework
 - i. Bogus 34.0.2 – dostawca fake danych
 - ii. Microsoft.AspNetCore.Authentication 2.2.0 - autentykacja
 - iii. Newtonsoft.Json 13.0.1 – operacje na Json
 - iv. StackExchange.Redis 2.6.70 – zarządzanie bazą Redis
 - v. Swashbuckle.AspNetCore 6.2.3 - dokumentacja
 - vi. Microsoft.Net.Http.Headers 2.2.0 – zarządzanie nagłówkami
 - vii. Microsoft.AspNetCore.StaticFiles 2.2.0 – operacje na plikach
- b. Docker 4.11.1 (84025) - konteneryzacja bazy
- c. Redis 7.0.1 Alpine – jako baza danych

2. Struktura bazy danych

Dane są serializowane i przechowywane w bazie pod postacią NoSQL, jako key-value.

Podział opiera się na modelach przechowywanych w bazie oraz na DTO, mapowane na obiekty bazodanowe.

Posty zawierają ID (Guid) forum do którego przynależą.

a. Movie

```
Movie v {
  id                string($uuid)
  name              string
  genre             string
  length            TimeSpan v {
    ticks            integer($int64)
    days             integer($int32)
    hours            integer($int32)
    milliseconds     integer($int32)
    minutes          integer($int32)
    seconds          integer($int32)
    totalDays        number($double)
    totalHours       number($double)
    totalMilliseconds number($double)
    totalMinutes     number($double)
    totalSeconds     number($double)
  }
  ticketPrice       number($float)
  premiumTicketPrice number($float)
  imageUrl          string
}
```

b. MovieDTO

```
MovieDTO v {
  name              string
  genre             string
  lengthInMinutes   integer($int32)
  ticketPrice       number($float)
  imageUrl          string
}
```

c. Post

```
Post v {
  id                string($uuid)
  author            string
  data              string($date-time)
  text              string
  forumId           string($uuid)
}
```

d. PostDTO

```
PostDTO {
  text string nullable: true
}
```

e. Forum

```
Forum {
  id string($uuid)
  author string nullable: true
  data string($date-time)
  title string nullable: true
  text string nullable: true
}
```

f. ForumDTO

```
ForumDTO {
  title string nullable: true
  text string nullable: true
}
```

g. Cinema– Najbardziej złożona struktura, api zwróci użytkownikowi pełną bazę najbliższych i minionych dni projekcji

Kino składa się ze swojej nazwy i dni projekcji

```
CinemaDTO {
  screeningDays > [...]
}
```

Dzień projekcji zawiera swoją datę, status otwarcia i kolejną listę – hale kinowe

```
ScreeningDay {
  id string($uuid)
  date string($date-time)
  isOpen boolean
  isExtraPaid boolean
  cinemaHalls > [...]
}
```

Hale kinowe – mają swoją nazwę , ilość miejsc i kolejną listę – sesje filmowe

```
CinemaHall {
  id          string($uuid)
  name        string
  seats        integer($int32)
  premiumSeats integer($int32)
  isOpen      boolean
  movieSessions > [...]
}
```

Sesja filmowa to przede wszystkim konkretna godzina i grany film - Movie

```
MovieSession {
  id          string($uuid)
  occupiedSeats integer($int32)
  occupiedPremiumSeats integer($int32)
  startDateTime string($date-time)
  moviePlayed  Movie > {...}
}
```

3. Endpointy – funkcjonalności

a. D

4. Dodane nagłówki

Klasyczny zwrot w przypadku tekstu np. **api/Redis/HealthCheck**

Response headers

```
content-type: text/plain; charset=utf-8
date: Sat,12 Nov 2022 14:01:54 GMT
server: Kestrel
strict-transport-security: max-age=31536000; includeSubDomains; preload
transfer-encoding: chunked
x-frame-options: DENY
```

Dodatkowy nagłówek podczas pobierania pliku tekstowego

```
content-disposition: attachment; filename=test.txt; filename*=UTF-8''test.txt
```