

## Завдання

Контент є важливою частиною навчальної платформи, а гнучкість його організації часто звучить як вимога від багатьох клієнтів.

В даному випадку мінімальною одиницею контенту будемо вважати **відео (video)**, в якому може бути розкритий один з аспектів теми, яка вивчається, а для того, щоб надати повну картину, по якійсь темі необхідна можливість об'єднувати декілька відео в **навчальний шлях (flow)**.

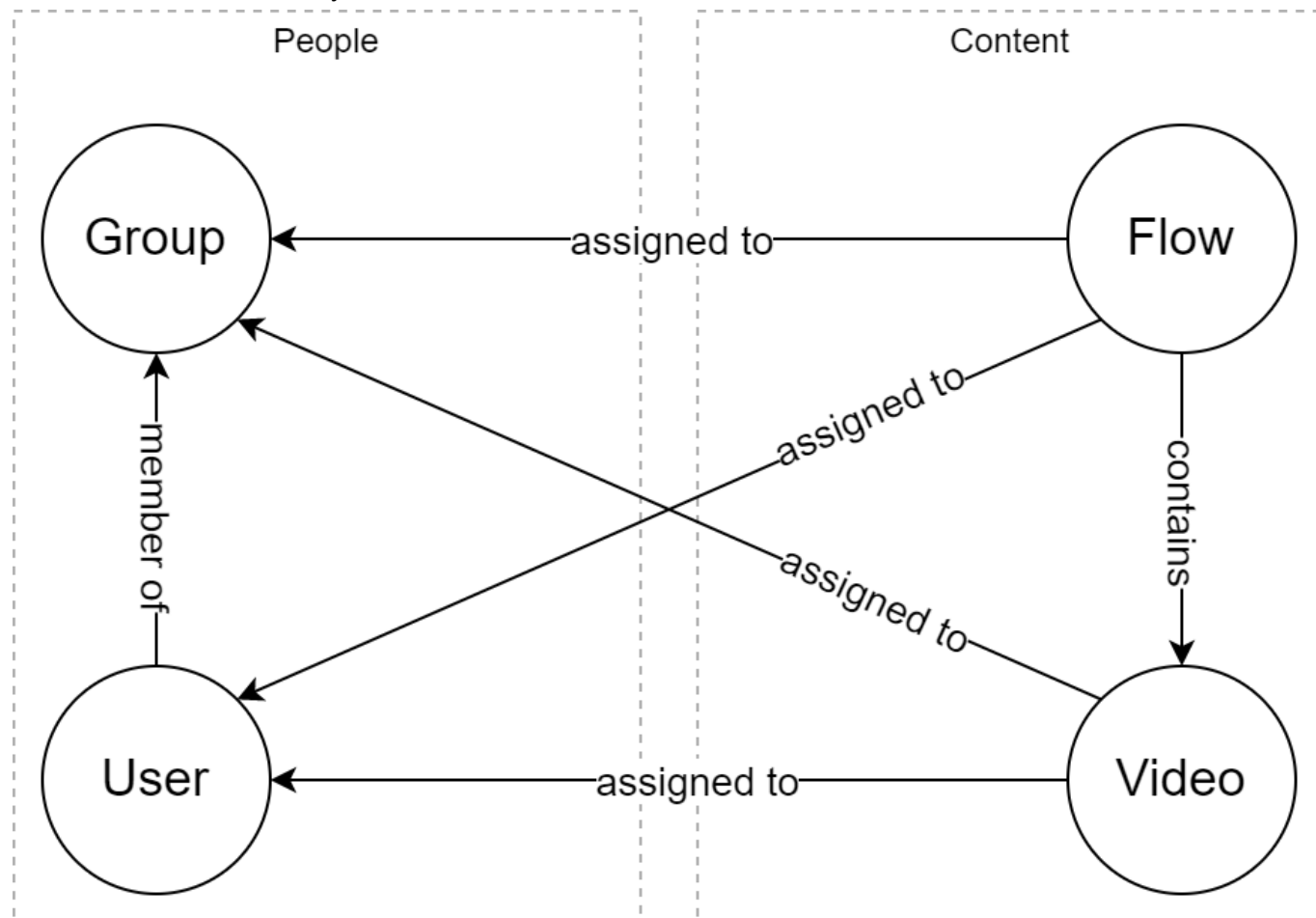
Іноколи виникає ситуація, при якій **працівнику (user)** або **групі працівників (group)** не вистачає певних знань по якійсь з тем, тому виникає сенс мати можливість призначати **відео** або цілий **навчальний шлях** для заповнення пробілів.

Вам пропонується розробити сервіс для роботи з асайментами контенту. Для реалізації даного сервісу необхідною умовою є використання платформи .Net та не in-memory бази даних. Вибір бази даних для виконання завдання лягає на плечі учасника, також цей вибір буде впливати на підсумкову оцінку.

### Вхідні дані

*README.md* який може бути використаний як шаблон опису рішення.

*Data.json* в якому описані зв'язки між користувачами або групами користувачів та контентом, який може бути відео або навчальний шлях. Дані з цього файлу повинні бути використані для наповнення бази даних, з якої потім буде здійснюватись читання відповідно до вимог. Ілюстрація зв'язків між основними сутностями.



Зв'язок **assigned to** - це той, який містить інформацію про пріоритет призначеного контенту, який може мати наступні значення: low, medium, high, critical.

## Вимоги

- Відповідь на запит `GET /users/{userId}/videos` не повинна містити дублікати контенту, а тільки той, пріоритет якого найвищий.

## Додаткові вимоги

- Кінцеві точки сервісу повинні підтримувати сортування за пріоритетом, використовуючи параметри запиту.

Приклад:

`GET /users/{userId}/videos?priority=asc`

`GET /users/{userId}/videos/{videoId}/paths?priority=desc`

## Очікуваний результат

Ієрархія контенту для користувача з id 5:

```
User 5
├── Video 5 medium
├── Video 6 high
├── Video 7 low
├── Flow 4 medium
│   ├── Video 5
│   └── Video 7
└── Group 5
    ├── Video 8 medium
    └── Flow 4 high
        ├── Video 5
        └── Video 7
```

Серверний API, який реалізує наступну функціональність.

- Можливість отримати весь контент, до якого юзер має доступ**

`GET /users/5/videos?priority=desc`

```
[
  { "video": "Video 5", "priority": "high" },
  { "video": "Video 6", "priority": "high" },
  { "video": "Video 7", "priority": "high" },
  { "video": "Video 8", "priority": "medium" }
]
```

- Можливість отримати інформацію про те, яким чином юзер має доступ до контенту**

`GET /users/5/videos/7/paths?priority=asc`

```
[
  {
    "path": "users/5/videos/7",
    "priority": "low"
  },
  {
    "path": "users/5/flows/4/videos/7",
    "priority": "medium"
  },
]
```

```
{
  "path": "users/5/groups/5/flows/4/videos/7",
  "priority": "high"
},
]
```

Extra:

- Наповнення черги RabbitMQ повідомленнями, які містять інформацію в форматі JSON наступного вигляду:

```
{
  "url": "/users/5/videos/7/paths?priority=asc",
  "itemsCount": 4,
  "elapsedTime": 30
}
```

де url - це адреса, за якою запит успішно виконано, itemsCount - це кількість елементів в результаті запиту, а elapsedTime - це час виконання запиту в мілісекундах.

## Формат презентації результатів

1. Рішення повинно бути опубліковано в окремому публічному репозиторії на GitHub.
2. Рішення має бути надано у вигляді серверної частини. Підніматись в контексті віртуального оточення Docker. Для старту вашого додатка має бути необхідним запустити єдину команду: `docker-compose up`.
3. README файл, в якому обов'язково вказуйте:
  - Короткий опис використаних тех рішень;
  - Кроки для старту сервісу;
  - Кроки для запуску тестів (unit, integration, component);
  - Речі, на які ви б хотіли звернути увагу.
4. Поділіться лінком на репозиторій з рішенням за допомогою [google forms](#).

Організатори та ментори залишають за собою право дискваліфікувати роботу учасника, якщо робота:

- Виконана у форматі, відмінному від того, що вказаний у завданні;
- Виконана за допомогою сторонніх осіб, а не учасником особисто.

## Обмеження та формат оцінювання

Категорія	Критерії	Бали
<b>GET /users/{userId}/videos</b>		
Medium	Повертається коректний контент	50
Hard	Повертається коректний пріоритет контенту	60
Easy	Підтримується сортування за пріоритетом	15
<b>GET /users/{userId}/videos/{videoId}/paths</b>		
Medium	Повертається коректний набір шляхів	50
Hard	Повертається коректний пріоритет контенту за шляхом	60
Easy	Підтримується сортування за пріоритетом	15

<b>Extra</b>		
Medium	Черга RabbitMQ наповнюється коректними повідомленнями	50
<b>Other</b>		
Easy	Вибір бази даних для зберігання та вичитки даних	50
Easy	Дотримання код стайлу, структурованість, простота коду та чистота історії комітів	25
Bonus	Наявність Unit тестів	15
Bonus	Наявність Integration тестів	30

## Контакти

Питання та уточнення, щодо змісту та вимог завдання:

[Join Slack channel](#)

Ментори будуть ігнорувати питання, які не стосуються завдання.