DEV
CHALLENGE
BACKEND

Node.js Task | Online Round

Level: **Standard / Pro**
Duration = **6 hours**
Deadline: **24.03.2019**

STAR GATEWAY
Реалізувати центр управління флоту спротиву.

1. Завдання:

Центр управління має інформацію про функціонуючі та не захопленні імперією врата, а також рівень небезпеки маршрутів. Будь якій зореліт може запитати можливий шлях до певного сектору галактики. Командний центр може отримати інформацію про останній запит зорельота або список кораблів.

Вхідні дані: gates.txt - це список усіх доступних шляхів, кожний рядок - це рівень небезпеки. (перший рядок - перший рівень небезпеки...)

Приклад:
Якщо зореліт запитав про сектор 56 та у файлі gates.txt у 3ому рядку є підмасів [16, 19, 21] то це є шлях 3ого рівня небезпеки. Після знаходження першого шляху на рівни або якщо у рядку шлях не знайдено, потрібно шукати на подальших рівнях.

gates.txt:

```
1 7 13 17 22 34 56 78
2 5 9 12 13 15 16 25 33 47
1 7 16 19 21 34 56 78
2 3 7 11 16 17 20 25 33 47
5 9 17 19 20 25 33 47
```

Input:

```
56
```

Output:

```
[
  { securityLevel: 1, gates: [22, 34] }
  { securityLevel: 2, gates: [2, 5, 9, 12, 13, 15] },
  { securityLevel: 3, gates: [16, 19, 21] },
  { securityLevel: 4, gates: [2, 3, 7, 11, 16, 17] },
  { securityLevel: 5, gates: [17, 19, 20] }
]
```

Вимоги:

- можливість отримати усі можливі шляхи та їх рівні небезпеки до обраного сектору у вигляді JSON, шляхи повинні бути відсортовані починаючи з найменшого рівня небезпеки;
- можливість отримати список всіх зорельотів, які запитували шлях та їх останній сектор до якого вони прямують;
- можливість отримати список усіх шляхів для зорельота у хронологічному порядку;

Очікуваний результат:

- Ендпоїнт для отримання шляху для зорельота. Номер зорельота та сектору, до якого він прямує, мають передаватися в урлі.
- Ендпоїнти для командного центру. Перший для отримання списку зорельотів та їх останній сектор. Другій для отримання інформації про один зореліт та його шляхи.

DEV
CHALLENGE
BACKEND

2. Формат представлення результатів

1. Рішення оформлюється за вказаними критеріями
- **1.1.** Рішення зберігається в окремій папці
- **1.2.** Назва папки це назва номінації
- **1.3.** Усі папки рішень за Номінаціями архівуються в єдиний архів з назвою у форматі **Ім'я_Прізвище.zip**
- **1.4.** Архів завантажується на сайті в особистому кабінеті

2. Рішення має бути надано у вигляді серверної частини. Підніматись в контексті віртуального оточення Docker або Vagrant..
Для старту вашого додатка має бути необхідним запустити єдину команду: **docker-compose up**, або **vagrant up**.

3. README файл, в якому обов'язково вказуйте:
- **3.1.** Кроки для старту сервісу
- **3.2.** Кроки для запуску тестів
- **3.3.** Опис API
- **3.4.** Речі, на які ви б хотіли звернути увагу або наступні кроки для вдосконалення вашого сервісу

Зверніть увагу, що назва архіву - єдине місце, де ви вказуєте свої персональні дані.
Назви файлів всередині архіву не мають містити вашого ім'я чи прізвища.
Розмір архіву з рішенням не має перевищувати 10 MB.

4. Організатори та судді залишають за собою право дискваліфікувати роботу учасника, якщо робота:
- **4.1.** Містить будь-яку вказівку на ім'я, прізвище, електронну пошту, компанію, адресу чи інші персональні дані учасника
- **4.2.** Виконана у іншому форматі, ніж вказано у завданні
- **4.3.** Виконана за допомогою сторонніх осіб, а не учасником особисто

3. Обмеження та критерії оцінювання

Категорія	Критерії	Бали
STANDARD	Працездатний API	100
STANDARD	Кодстайл	40
STANDARD	Вибір та архітектура бази даних	40
BONUS	Наявність юніт тестів	12
BONUS	Наявність інтеграційних тестів	12
BONUS	Використання тільки базових модулів Node.js для обробки файла	20
BONUS	Урли відповідають стандарту REST	20
PRO	Складність алгоритму пошуку шляху повинна бути O(n)	36
PRO	Експорт запитів командного центра у CSV	20
PRO	Кешування запитів від зорельота на 5 хвилин	20