

Test task from Innocode

Завдання:

Потрібно написати Rails сервіс для отримання і зберігання даних по Route.

За основу візьмемо той факт, що об'єкт Route вже існує в базі.

В якості респонсу може бути використаний нище наведений приклад. Вихідні умови додаються.

Виконане завдання потрібно надати у вигляді .rb файлів

Умова:

Гугл сервіс, який надає дані по руху транспорту

api url - '<https://routes.googleapis.com/directions/v2/computeRoutes>'
ключ доступу - зберігається у ENV змінній 'X-Goog-API-Key'

Запити формуються на основі моделі Route, яка для спрощення має наступну схему

```
t.bigint "id"  
t.string "title"  
t.string "origin" # у форматі "latitude:longitude"  
t.string "destination" # у форматі "latitude:longitude"  
t.timestamp "last_updated_at"
```

API приймає параметри у наведеному нижче вигляді, POST методом у JSON форматі. Для авторизації запиту необхідно передавати у хедері X-Goog-API-Key відповідне значення ENV змінної

```
'origin' => {  
  'origin' => {  
    'latLng' => {  
      'latitude' => string,  
      'longitude' => string  
    }  
  }  
},  
'destination' => {  
  'location' => {  
    'latLng' => {  
      'latitude' => string,  
      'longitude' => string  
    }  
  }  
}
```

```
    'longitude' => string
  }
}
```

зразок JSON респонсу

```
{
  "routes": [
    {
      "duration": 240, # seconds
      "distanceMeters": 1190, # meters
      "staticDuration": 180 # seconds
    }
  ]
}
```

Вимоги до сервісу:

HTTP клієнт будь який - Faraday, Excon, httparty etc

Сервіс має логувати помилку(через стандартний Rails logger) при будь-якому респонсі клієнта крім 200

При 200 респонсі клієнта об'єкт Route має оновити last_updated_at

Результат має зберігатись у Redis з можливістю майбутнього отримання звіди(окремий store клас, який для цього імплементує інстанс методи #set та #get)

Для унікальності записів Redis store використовує композитний ключ, який складається з назви класу та id об'єкта Route

Перед збереженням даних має бути провалідована наявність значення duration та distance.

У разі невалідних даних збереження не потрібне має бути залогована відповідна помилка

Формат збереження даних наступний:

```
{
  updated_at: timestamp,
  duration: int,
  static_duration: int,
  distance: int,
  speed: int #km/h,
  traffic_state: string
}
```

updated_at - поточний час збереження даних

для розрахунку speed використовується значення duration

traffic_state розраховується наступним чином

jam => speed <= 10 km/h

normal => 10 km/h < speed <= 40 km/h

freeway => speed > 40 km/h

У кого можна уточнити завдання і задати питання?

Питання можна задавати Максиму (він був на співбесіді), його пошта:
maksym.dubinin@innocode.no

Успіхів!