**Sprint 3. Сложные сущности в REST. Тесты производительности**

**Спецификация:**

К предыдущему заданию добавлены REST-сервисы: Summary, Report и GrouppedView, у которых есть только метод GET.

Все методы в качестве параметра получают сущность EventFilter , которая отвечает за фильтрацию и паджинацию данных.

Сервис Summary возвращает сущность Summary, отражающую статистику по событиям, попавшим в выборку. Если заполнено поле GroupBy, то метод должен возвращать ошибку InternalServerError.

В сущности Summary:

* поле Count – означает общее количество событий и интеракций в выборке (обратите внимание, что работает паджинация)
* поле InteractionCount – общее количество интеракций в выборке
* поле EventCount – общее количество событий в выборке
* MostPopularBroker – выдает имя самого встречающегося брокера среди записей в выборке
* AverageDuration – среднее значение поля Duration из записей в выборке.

Сервис GrouppedView группирует события, попавшие в выборку, по полю GroupBy и возвращает их.

Сервис Report возвращает все события, попавшие в выборку и генерирует CSV-файл. Если заполнено поле GroupBy, то метод должен возвращать ошибку InternalServerError.

Все методы REST-сервисов принимают в качестве одного из параметров SessionId, который можно получить с помощью сервиса Accounts. Если SessionId не предоставлен или предоставлен неверный, то должно выбрасываться исключение “Access is denied”.

Все новые и старые методы REST-сервисов должны выполняться не более чем 2 секунды.

**Адреса сервисов:**

Accounts – http://soap-att.deejayentertainment.ru/AccountsV3

Interactions – http://rest-att.deejayentertainment.ru/v3/Interaction

Events - <http://rest-att.deejayentertainment.ru/v3/Event>

Summary - http://rest-att.deejayentertainment.ru/v3/Summary

Report - http://rest-att.deejayentertainment.ru/v3/Report

GrouppedView - http://rest-att.deejayentertainment.ru/v3/GrouppedView

**Задания:**

Написать не менее 100 тестов на новую функциональность (всего должно быть не менее 230 тестов).

Найти не менее 15 багов (вместе со старым функционалом) и написать на них «красные» тесты.

PS не стоит забывать про регрессионное тестирование, потому что возможно что-то сломалось/починилось в новой версии API ☺

**Новые баги:**

* Interaction/Create записывает Title фейковым значением
* Interaction/GetAll пропускает 1-ую запись в списке
* Summary возвращает Ok с ошибкой вместо InternalServerError
* Summary возвращает EventCount больше на единицу
* Summary возвращает 0 в поле AverageDuration, если AverageDuration больше 10000
* Event/GetById превышает допустимое время ожидания ответа, если находит интеракцию вместо события
* Event/Update превышает допустимое время ожидания ответа в любом случае
* Interaction/Delete превышает допустимое время ожидания ответа в случае успешного удаления
* 15

**Не исправленные баги:**

* Login падает при длине больше 50 символов.
* Interaction/GetById выполняется даже если сессия не предоставлена
* Create не пропускает duration = 0
* Update не позволяет Title иметь меньше 24 символов вместо 12.
* Interaction/Delete возвращает InternalServerError вместо NotFound, если предоставлен «левый» ID
* Event/Update не записывает поле Sectors
* Event/Create всегда записывает поле InteractionType как Conference.

**Исправленные баги:**

* Logout не разлогинивает пользователя при верно введенной сессии
* Registrate возвращает null если пароль удовлетворяет всем требованиям сразу
* Event/GetById возвращает первое событие в списке вместо события по ID.
* Create может создать Interaction в будущем.
* Update не валидирует брокера
* Create игнорирует поле Location.City при валидации
* Interaction/Delete не удаляет интеракцию.
* Event/GetAll возвращает OK без данных, если предоставлен «левый» ID
* Event/Create возвращает неверный ID если ID % 5 ==0

**Вновь появившиеся баги:**

* Registrate пропускает 2 условия: цифры и знаки препинания
* Logout падает с FormatException если подсунули сессию которую невозможно конвертировать в Guid.