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

**Спецификация:**

К предыдущему заданию добавлены REST-сервисы: Summary, GrouppedView и Report. Summary и GrouppedView имеют только метод GET, Report – GET и POST.

Все GET - 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. Метод должен игнорировать поле FileFilter.

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. Поле GroupBy является обязательным. Метод должен игнорировать поля FileFormat, Page и PageSize. Для событий с пустым полем, по которому идет группировка, должна создаваться отдельная группа.

Сервис Report работает с файлами. GET возвращает все события, попавшие в выборку и генерирует файл формата, указанного в FileFormat-параметре. Если заполнено поле GroupBy, то метод должен возвращать ошибку InternalServerError.

Report/POST-метод загружает файл определенного формата (создает новые события для заданного sessionId). Примеры файлов находятся в папке с заданием. Обратите внимание, что параметры EventId, Created и Updated игнорируются, а в CSV-файле их нет (обратите внимание, что для CSV-формата файл полученный через GET не подходит для загрузки через POST). Метод выполняет ту же самую валидацию, что и POST-метод в Event/Interaction сервисах. Дополнительно, CSV-файл обязан иметь все поля.

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, методы сервиса Reports – 7 секунд.

Описание полей фильтра:

* DateFrom – фильтрация по Date (дата не ранее)
* DateTo – фильтрация по Date (дата не позднее)
* Broker – фильтрация по полю Broker (поиск по подстроке)
* InteractionType – фильтрация по полю InteractionType (полное совпадение)
* MeetingTypes – фильтрация по полю MeetingType (совпадение хотя бы 1 значения в списке фльтра с 1 значением в списке сущности)
* Title – фильтрация по полю Broker (поиск по подстроке)
* Location – фильтрация по полям Location.City и Location.Country (поиск по подстроке), подстрока должна присутствовать хотя бы в 1 из полей.
* Company – фильтрация по полю Company (поиск по подстроке)
* AddressType – фильтрация по полю AddressType (полное совпадение)
* Page – номер страницы (начиная с 0)
* PageSize – количество записей на 1 странице
* IsInteraction – если задано true – присылает только события с Date меньше текущей даты, false – с Date больше текущей даты. Если не задано – возвращает все записи.

**Адреса сервисов:**

Accounts – http://soap-att.dio.red/AccountsV3

Interactions – http://rest-att.dio.red/v3/Interaction

Events - http://rest-att.dio.red/v3/Event

Summary - http://rest-att.dio.red/v3/Summary

Report - http://rest-att.dio.red/v3/Report

GrouppedView - http://rest-att.dio.red/v3/GrouppedView

**Задания:**

Написать не менее 150 тестов на новую функциональность (всего должно быть не менее 280 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
* GrouppedView превышает допустимое время ожидания ответа в случае если указан параметр фильтрации Company
* GrouppedView. Если поле PageSize = 0, группировка происходит только для 10 записей.
* GrouppedView. Группировка по AddressType возвращает только 1 запись в каждой группе
* GrouppedView. Группировка по Location не создает группу для событий с пустым Location.
* GrouppedView. Группировка по Broker возвращает все записи в группе.
* GrouppedView. При указании невалидного значения в GroupBy происходит группировка по InteractionType
* Report/GET – При запросе XML-формата отдается JSON и наоборот (они перепутаны)
* Report/GET – При запросе CSV – репорта дата отображается на 3 дня назад
* Report/GET – При запросе JSON приходит неверный Duration
* Report/GET – При запросе JSON IsInteraction = true данные приходят задублированными.
* Report/GET – Если указан фильтр по брокеру – данные не возвращаются
* Report/POST – Валидация не работает для событий CSV-файла c InteractionType = Conference
* Report/POST – XML: Высталяет город по умолчанию, вместо того, что загрузили
* Report/POST – XML: Если загружен EventId > 0, то MeetingTypes не проставляется
* Report/POST – JSON: Высталяет страну по умолчанию, вместо того, что загрузили
* Report/POST – JSON: Если загружен EventId == 0, то Sectors не проставляется

**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.