

Задание «Сервер RESTful API»

Исходные данные

Участникам предлагается создать сервер RESTful API для высоконагруженного Web-приложения. Особенностью этого приложения является большое количество простых запросов. Цель - минимизировать использование ресурсов под каждый запрос. Второй особенностью является тот факт, что MySQL почти всегда будет узким местом системы. Отсюда следует, что все запросы должны быть максимально простыми. Возможно, какие-то данные придется кэшировать и реализовать ограничение на количество запросов к серверу.

Сервер должен уметь

1. Хранить для каждого пользователя следующие данные:
 - user_id - идентификатор пользователя - строка
 - auth_key - ключ доступа пользователя к серверу API - строка
 - дату и время последнего запроса
 - количество дней, которые составляют непрерывную последовательность ежедневных запусков приложения пользователем
 - общее количество дней, когда пользователь запускал приложение
 - некий набор объектов (объект представляется в виде пары "имя" - "данные")
2. Сохранять по требованию клиента объекты с данными. Возможно объединение нескольких объектов для передачи соответствующих данных через один запрос.
3. Выдавать клиенту по требованию:
 - дату и время последнего запроса
 - количество дней, которые составляют непрерывную последовательности ежедневных запусков приложения пользователем
 - общее количество дней, когда пользователь запускал приложение
 - объект (по имени)
4. Возможно объединение выдачи нескольких данных через один запрос.

В каждом запросе есть два обязательных параметра - это user_id и auth_key, которые каждый раз должны проверяться. Алгоритм проверки auth_key произвольный. Сохранение и выдача данных должны быть доступны только для авторизованных пользователей. Для определенности необходимо использовать JSON в качестве формата всех запросов и ответов.

Оформление результата работы:

1. Весь разработанный код включая SQL дампы базы данных необходимо загрузить в публичный репозиторий на Github.com
2. В описании репозитория должна быть подробная инструкция по установке проекта, подразумевающая клонирование репозитория на сервер проверяющего и простой запуск.

Базовая задача (обязательная)

Реализовать два типа запроса: сохранение и получение данных пользователя.

Максимальное количество баллов: 300

Дополнительные задачи

1. Реализовать запросы для создания нового пользователя и удаления существующего.
Максимальное количество баллов: 75
2. Реализовать кэширование с помощью Memcached. Необходимо следить за актуальностью данных в кэше.

Максимальное количество баллов: 100

3. Сделать частотные ограничения для запросов к серверу API. Настройки ограничений должны быть индивидуальными для каждого пользователя. При превышении количества запросов в заданную единицу времени сервер должен возвращать ошибку "Too many requests per second/minute/hour/day"

Максимальное количество баллов: 125