# Тестовое задание Full-Stack .Net Core Dev

Требуется спроектировать и реализовать систему, эмулирующую трекинг запросов от рекламной сети.

Система должна состоять из двух сервисов:

* Tracking - сервис принимающий входящие HTTP запросы от внешней рекламной сети и записывающий их в SQL
* Admin Desk - web сервис позволяющий просмотреть отчет по полученным данным

## Tracking

Требуется HTTP эндпоинт который будет принимать запросы от внешней рекламной сети. Пример HTTP GET запроса:

[http://tracking-service.address/feed?token=secret-qwerty&client\_ip=1.1.1.1&domain=example.com&os=Android&client\_id=23c84ca5-67c4-4de0-a918-c2ba1d31558e](http://tracking-service.address/feed?token=secret-token&ip=1.1.1.1&domain=example.com&os=Android&clientid=23c84ca5-67c4-4de0-a918-c2ba1d31558e)

* token - сервис должен принимать запросы только с токеном secret-qwerty
* client\_ip - всегда ipv4 адрес клиента запросившего рекламу
* domain - домен сайта где показывается реклама, всегда второго уровня
* os - операционная система клиента, может быть Android, iOS, Windows, MacOS, Linux
* client\_id - уникальный id клиента, всегда guid/uuid
* в HTTP запросе есть заголовок ‘Site-IP’ в котором передается IPv4 адрес сайта где показывается реклама
* все параметры являются обязательными

От сервиса требуется принять запрос, обработать, и сохранить в SQL в той структуре, которая лучше всего подойдет для того что требуется отобразить в отчете Admin Desk.

В обработку запроса входит:

* Валидация (валидность токена, наличие всех необходимых параметров и соответствие их ожидаемым форматам и тд). Если токен неправильный, выдавать HTTP 401. Если запрос неправильный, выдавать HTTP 400.
* Определение страны на основе IPv4
* Проверка на уникальность client\_id, если запрос от такого клиента уже поступал, нужно вернуть HTTP 200 “Duplicate” и не записывать в базу

Admin Desk

Требуется web сервис, в котором можно будет проанализировать данные которые сохраняет Tracking.

Нужно два отчета:

* By Request
* By Country

Отчет By Request выводит все сохраненные записи для каждого рекламного запроса, в виде таблицы:

|Date|Client ID|Client IP|Client Country|OS|Domain|Site Country|

* Date - дата и время приема запроса в виде 2020-01-01 20:14:09
* Client Country - страна клиента на основе ip-адреса из параметра client\_ip, в виде двухбуквенного кода страны, например US
* Site Country - страна сайта на основе ip-адреса из заголовка ‘Site-IP’, в виде двухбуквенного кода страны, например DE
* остальные поля соответствуют параметрам запроса

Над таблицей нужен фильтр по OS, где можно будет выбрать либо все записи, либо отфильтрованные по одной из пяти операционных систем.

Отчет By Country группирует записи в базе по стране, таблица:

|Client Country|Requests|

* Client Country - страна клиента, в виде двухбуквенного кода страны, например US
* Requests - количество запросов с такой страной

## Реализация

Просьба использовать:

* .NET Core 3.1, 5, 6
* VUE JS + Bootstrap 5 (или любой другой современный JS фреймворк)
* MS SQL Server Express

Конечным результатом должна быть ссылка на репозиторий с проектом состоящим из двух приложений и одной базы. Проект должен собираться в Visual Studio.

Будет замечательно (но не обязательно) если все это будет упаковано в Docker контейнеры. Еще лучше, если сервис будет где-нибудь развернут, и его можно будет протестировать без сборки и деплоя с нашей стороны.

## Примечания и пожелания

* Для определения страны по IP, лучше всего использовать бесплатную версию базы <https://dev.maxmind.com/geoip/geoip2/geolite2/>
* Примерная тестовая нагрузка на сервис трекинга будет составлять 10-20 запросов в секунду
* Тестовые запросы не всегда можно будет корректно обработать и записать, приветствуются ваши варианты обработки таких событий.