

# Организация трекинга доставки

## Оглавление

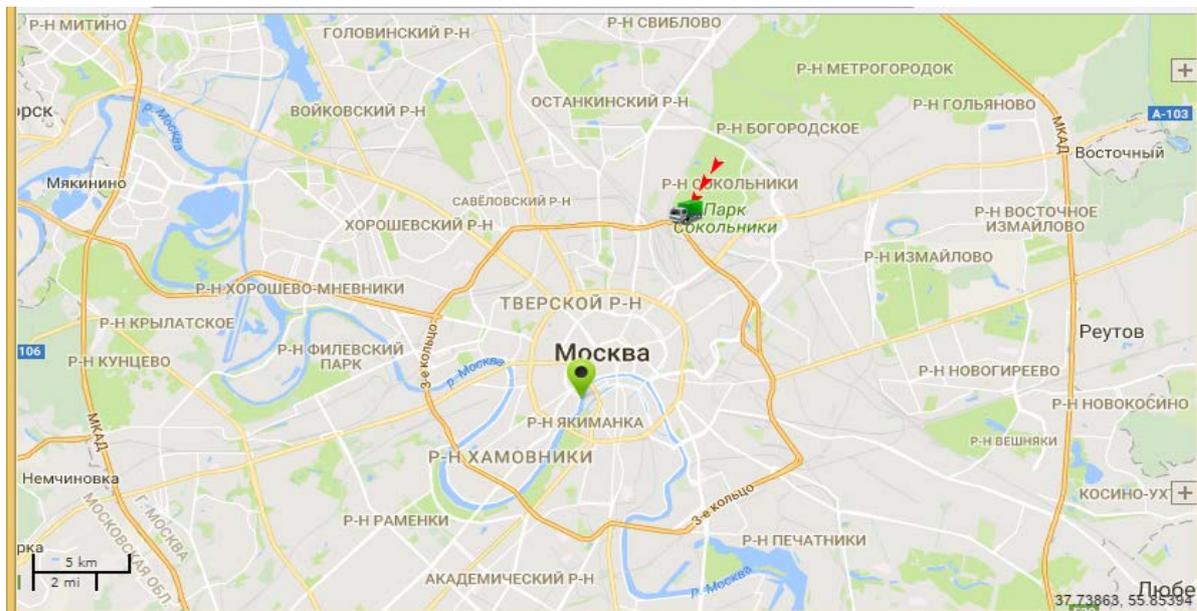
Общее описание решения .....	1
Организация решения со стороны web-приложения .....	2
Используемые технологии.....	3
Структура БД.....	3
Работа веб-сервиса .....	5
Команда на добавление трекинга .....	5
Команда на удаление трекинга .....	6
Получение актуальной геопозиции ТС .....	7

## Общее описание решения

Решение предназначено для предоставления информации по позиции автомобиля, который осуществляет доставку.

За час до планового прибытия автомобиля на адрес доставки – из внешней системы планирования уходит команда на веб-сервис «Команда на добавление трекинга». При этом на email адресата уходит электронное письмо с содержанием ссылки на страницу трекинга: <http://sitename.domain/tracking/track?orderid=3826d1fd-d16a-4774-8eec-2581774d6386>

Адресат при переходе по ссылке видит страницу следующего содержания:



На странице отображается адрес доставки (зеленая точка), местоположение ТС (иконка автомобиля), хвост движения автомобиля (красные стрелки около ТС). Каждые 15 секунд

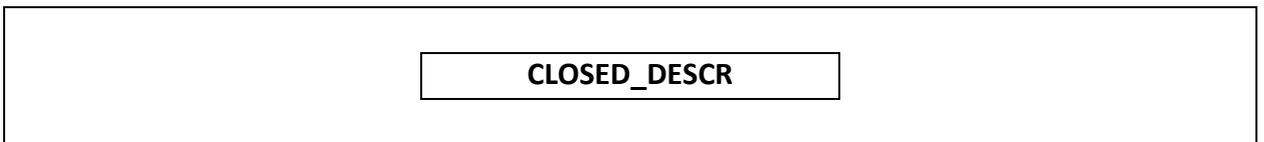


- Если запрос вернет 0 записей - показать статус 404 и описание ошибки Unknown order ID
- Если запрос вернет статус=0 – показать текст CLOSED\_DESCR в красивом виде
- Если запрос вернет статус=1 – запросить в таблицах TRACKING и CAR\_DEVICE\_POS детали по доставке, и страницу карты с адресом доставки и положением ТС

Для данного проекта – **требуется использовать Leaflet**, и отображать на нем карту Yandex (вот пример отображения карты yandex как слоя leaflet <http://webmap-blog.ru/examples/leaflet-examples/leaflet-10.html> )

Как только страница сгенерирована – JS каждые 30 секунд должен обращаться к методу «Получение актуальной геопозиции ТС» и обновлять местоположение иконки ТС и хвоста. Если метод «Получение актуальной геопозиции ТС» вернул status=0 – обновить страницу – при этом должны отобразиться детали закрытой заявки.

Если при запросе страницы трекинга – status=0, или мы получили status=0 при обращении к методу «Получение актуальной геопозиции ТС» при отображении страницы – требуется отобразить страницу вида:



CLOSED\_DESCR берется из таблицы БД TRACKING

## Используемые технологии

Генерация контента: PHP

Web-сервер: Apache

База данных: MS SQL 2012

Отображение карты: Leaflet (leafletjs.com)

Формат обмена данными: JSON

## Структура БД

База данных MS SQL 2012

Таблица API\_KEYS

Имя поля	Тип	Описание
KEY	UNIQUEIDENTIFIER, Primay key, NOT NULL	API ключ

DESCR	VARCHAR(100), NULL	Описание
F_ENABLE	BIT, NOT NULL	Флаг активности, 0 – не активен, 1 - активен

Индексы таблицы

PK\_API\_KEYS – по полю KEY уникальный

Таблица **TRACKING**

Имя поля	Тип	Описание
ID	UNIQUEIDENTIFIER, Primary key, NOT NULL	ID заявки
ORDER_LAT	NUMERIC(10,7), NOT NULL	Широта WGS84 точки доставки
ORDER_LON	NUMERIC(10,7), NOT NULL	Долгота WGS84 точки доставки
ORDER_DESCR	VARCHAR(500), NOT NULL	Описание всплывающего окна в точке доставки
CAR_DESCR	VARCHAR(500), NOT NULL	Описание всплывающего окна в ТС
CLOSED_DESCR	VARCHAR(500), NOT NULL	Описание текста при закрытой заявке
CAR_DEVICE	INT, NOT NULL	Идентификатор GPS-трекера
DELIVERY_PERIOD	DATETIME, NULL	Плановый период прибытия, UTC0
CLOSING_PERIOD	DATETIME, NOT NULL	Дата время безусловного закрытия трекинга, UTC0
CHANGED	DATETIME, NOT NULL	Дата и время изменения записи, UTC0
API_KEY	UNIQUEIDENTIFIER, NOT NULL	API ключ, под которым меняли запись

Индексы таблицы

PK\_TRACKING – по полю ID уникальный

Таблица **TRACKING\_STATUS**

Имя поля	Тип	Описание
ID	UNIQUEIDENTIFIER, Primary key, NOT NULL	ID записи, уникальный идентификатор, генерируется новый на каждую запись
TRACKING_ID	UNIQUEIDENTIFIER, NOT NULL	ID трекинга таблицы TRACKING
STATUS	TINYINT, NOT NULL	Статус трекинга, 1 – активен, 0 - закрыт

CHANGED	DATETIME, NOT NULL	Дата и время изменения записи, UTC0
---------	--------------------	-------------------------------------

Индексы таблицы

PK\_TRACKING\_STATUS – по полю ID уникальный

IDX\_TRACKING\_ID\_CHANGED – по полям TRACKING\_ID и CHANGED

### Таблица **CAR\_DEVICE\_POS**

Данная таблица будет обновляться из внешней базы данных, с периодичностью 30 сек

Имя поля	Тип	Описание
ID	UNIQUEIDENTIFIER, Primary key, NOT NULL	ID поля
CAR_DEVICE	INT, NOT NULL	Идентификатор GPS-трекера
PERIOD	DATETIME, NOT NULL	Актуальность данных, UTC0
LAT	NUMERIC(10,7), NOT NULL	Широта WGS84 точки доставки
LON	NUMERIC(10,7), NOT NULL	Долгота WGS84 точки доставки
SPEED	NUMERIC(3,0), NOT NULL	Скорость
COURSE	NUMERIC(3,0), NOT NULL	Путевой угол
TAIL	VARCHAR(250), NULL	Хвост маршрута, 3 точки трека перед точкой актуальности, несколько позиций вида <lat>55.2</lat> <lon>37.6</lon> <course>268</course> ...

Индексы таблицы

PK\_CAR\_DEVICE\_POS – по полю ID уникальный

IDX\_CAR\_DEVICE – по полю CAR\_DEVICE

## Работа веб-сервиса

### *Команда на добавление т рекинга*

На вебсервис от внешней системы поступает следующий запрос:

<http://sitename.domain/tracking/add>

GET параметры запроса:

Параметр	Тип	Описание	Обязательный
KEY	UNIQUEIDENTIFIER	Ключ API	Да

POST параметры запроса

Параметр	Тип	Описание	Обязательный
ORDERID	UNIQUEIDENTIFIER	Идентификатор заявки	Да
ORDER_LAT	NUMERIC(10,7)	Широта WGS84 точки доставки	Да

ORDER_LON	NUMERIC(10,7)	Долгота WGS84 точки доставки	Да
ORDER_DESCR	Строка	Описание всплывающего окна в точке доставки	Да
CAR_DESCR	Строка	Описание всплывающего окна в ТС	Да
CAR_DEVICE	INT	Идентификатор GPS-трекера	Да
DELIVERY_PERIOD	UNIXTIME	Плановый период прибытия, UTC0	Нет
CLOSING_PERIOD	UNIXTIME	Дата время безусловного закрытия трекинга, UTC0	Да

При выполнении данного запроса необходимо:

- 1) Проверить API\_KEY на правильность их ввода по маске символов (защита от SQL Injection)
- 2) в таблице API\_KEYS проверить ключ KEY  
SELECT F\_ENABLE FROM API\_KEYS WHERE KEY=@KEY
  - Если запрос не вернет ни одной записи: вернуть структуру result=0, errordesc=Invalid API key
  - Если запрос вернет запись F\_ENABLE=0: вернуть структуру result=0, errordesc= Inactive API key
  - Если запрос вернет запись F\_ENABLE=1 – дальше продолжать обработку
- 3) Проверить что указаны обязательные поля POST запроса, правильность их ввода (по маске символов, защита от SQL Injection). Если поля указаны не все, или не соответствуют маске – сформировать детальное описание ошибки на английском, и вернуть структуру result=0, errordesc= Детальное описание ошибки
- 4) В случае если по составу полей и заполнению все ОК – проверяем наличие записи с ORDERID в таблице TRACKING, если запись есть – апдейтим ее, если нет – добавляем в таблицу.
- 5) Далее добавляем запись в таблицу TRACKING\_STATUS:  
ID=NEWID(),TRACKING\_ID=@ORDERID, STATUS=1, CHANGED=GETUTCDATE()
- 6) Далее возвращаем структуру result=1

### *Команда на удаление т трекинга*

На вебсервис от внешней системы поступает следующий запрос:

<http://sitename.domain/tracking/delete>

GET параметры запроса:

Параметр	Тип	Описание	Обязательный
KEY	UNIQUEIDENTIFIER	Ключ API	Да

## POST параметры запроса

Параметр	Тип	Описание	Обязательный
ORDERID	UNIQUEIDENTIFIER	Идентификатор заявки	Да

При выполнении данного запроса необходимо:

- 1) Проверить API\_KEY на правильность их ввода по маске символов (защита от SQL Injection)
- 2) в таблице API\_KEYS проверить ключ KEY  
SELECT F\_ENABLE FROM API\_KEYS WHERE KEY=@KEY
  - Если запрос не вернет ни одной записи: вернуть структуру result=0, errordesc=Invalid API key
  - Если запрос вернет запись F\_ENABLE=0: вернуть структуры result=0, errordesc= Inactive API key
  - Если запрос вернет запись F\_ENABLE=1 – дальше продолжать обработку
- 3) Проверить что указаны обязательные поля POST запроса, правильность их ввода (по маске символов, защита от SQL Injection). Если поля указаны не все, или не соответствуют маске – сформировать детальное описание ошибки на английском, и вернуть структуру result=0, errordesc= Детальное описание ошибки
- 4) В случае если по составу полей и заполнению все ОК –добавляем запись в таблицу TRACKING\_STATUS: ID=NEWID(),TRACKING\_ID=@ORDERID, STATUS=0, CHANGED=GETUTCDATE()
- 5) Далее возвращаем структуру result=1

## *Получение акт уальной геопозиции ТС*

На вебсервис от веб-страницы поступает следующий запрос:

<http://sitename.domain/tracking/getpos>

Это вспомогательный метод, который возвращает крайние географические координаты трекинга и его статус, порядок работы с ним предлагаю разработать самостоятельно

Параметры запроса:

Параметр	Тип	Описание	Обязательный
ORDERID	UNIQUEIDENTIFIER	Идентификатор заявки	Да

Порядок обработки запроса:

- 3) Проверить ORDERID на правильность их ввода по маске символов (защита от SQL Injection). В случае ошибки вернуть структуру result=0,errordesc=Описание ошибки по английски.
- 4) Выполнить запрос к БД типа  
SELECT TOP 1 CASE WHEN TRACKING.CLOSING\_PERIOD <= GETUTCDATE() THEN 0 ELSE TS.STATUS END AS STATUS,  
TRACKING.CAR\_DEVICE AS CAR\_DEVICE

```
FROM TRACKING_STATUS TS  
LEFT JOIN TRACKING AS TRACKING ON TS.TRACKING_ID = TRACKING.ID  
WHERE TRACKING_ID=@ORDERID
```

- 5) Если запрос вернет 0 записей - вернуть структуру result=0,errordesc=Unknown order ID
- 6) Если запрос вернет статус=0 - вернуть структуру result=1,status=0
- 7) В случае статуса=1 – обращаемся к таблице CAR\_DEVICE\_POS с отбором по ранее полученному полю CAR\_DEVICE, и возвращаем result=1,status=1 и координаты позиции ТС и хвост трека ТС. Т.к. таблица CAR\_DEVICE\_POS будет постоянно обновляться из внешней системы, во избежании блокировок использовать уровень изоляции транзакций при запросе – «Грязное чтение».