Интеграция с данными системы Wialon будет реализована в два этапа (связано с ограничениями по доступу к системе в режиме разработки).

1. Файловый обмен.
2. Web-сервисы.

# Файловый обмен

Данные импортируются из файла в csv формате. Файл формируется на основе событий въезда локомотива с GPS в гео-зону. Файл дописывается новыми значениями событий в режиме онлайн (лог). Поля файла:

1. Номер локомотива: тип локомотива + серийный номер.
2. Дата и время события (въезда).
3. Название гео-зоны.
4. Средняя скорость локомотива.
5. Пробег локомотива на момент события наростающим итогом по данным **Wialon**.

Данные загружаются в несколько этапов:

1. В таблицу показателей на базе **InFlux** (нужно проекпериментировать).
2. В таблицу регистра трекинга объектов по геозонам **GeoZoneTrackingRegistry**.
3. Проводится регистрация события нахождения объектов на локации.
4. Проводится выгрузка сообщений с приростом пробега в модуль ТОиР.

# Регистрация в InFlux

Регистрация проводится для историчности данных. Далее данные будут выгружаться в **ClickHouse** с прицелом на дальнейшее использование для машинного обучения.

Регистрация производится с использованием настроенных потоков сообщений в системе **Node-RED**.

В **InFlux** хранятся первичные данные, пришедшие от **Wialon** GPS.

Структура загрузки:

**bucket** = *tracking*

**measurement** = *geozone\_tracking*

**tags**:

* **serial\_number** = серийный номер локомотива
* **zone\_name** = название зоны (пробел как esc последовательность (`\ `))

**fields**:

* **speed** = скорость
* **mileage** – пробег

**timestamp** – unix time с точностью до секунды.

# Регистрация объектов в таблице GeoZoneTrackingRegistry

Регистрация в таблице проводится для сохранения истории пробегов в системе координат **Wialon**, которая отличается от используемых пробегов в системе АСКОУ, которые в свою очередь привязаны к бизнес-процессам обслуживания оборудования.

## Справочник гео-зон

Справочник предназначен для ведения гео-зон, созданных в системе **Wialon**.

Необходимо создать таблицу **Geozones** / Геозоны для ведения информации по гео-зонам со следующей структурой:

1. **Id** – int – синтетический ключ записи.
2. **Name** – string – название геозоны. Уникальные значения.
3. **LocationId** – ссылка на справочник **hr.Locations**.
4. **RouteNodeId** – ссылка на справочник tracking.**RouteNodes**.

У справочника должен быть стандартный интерфейс.

Справочник поместить в меню: Трекинг — Справочники — Геозоны.

# Регистр событий трекинга по гео-зонам — GeoZoneTrackingRegistry

Необходимо создать регистр/журнал **GeoZoneTrackingRegistry** со следующей структурой:

1. **Id** – int – синтетический ключ записи.
2. **GeoZoneId** – ссылка на справочник tracking.Geozones.
3. **AssetId** – ссылка на справочник inventory.Assets.
4. **EventDateTime** – DateTime – время события.
5. **EventType** – int = 1 – вход в зону (всегда 1 для файла АГМК).
6. **TrackingRegistryId** – ссылка на регистр **TrackingRegistry**.
7. **MileageValue** – double – значение пробега по GPS треку.

## Доработка регистра TrackingRegistry

Так как пробег теперь может рассчитываться по данным нескольких систем, то для облегчения/ускорения расчетов необходимо сохранять данные (пробега), связанные с событием регистрации объекта на локации.

Поэтому необходимо добавить дополнительное поле **LinkedData** в формате JSON.

## Загрузка данных в GeoZoneTrackingRegistry

Данные загружаются в массив строк. Обработка проводится от последней обработанной в предыдущем сеансе. Последняя строка регистрируется (пока) в справочнике настроек системы с кодом LAST\_GPS\_FILE\_LINE\_NO со значением Value:

{

“Line\_No”: NNN,

“Timestamp”: “yyyy-mm-dd hh:mm:ss”

}

**Line\_No** – номер последней обработанной строки в файле.

**Timestamp** – значение времени для последней строки для дополнительной проверки, что строка найдена правильно.

Строка разбирается по полям и сохраняется в таблице. Для поиска GPS зоны используется название зоны. Ссылка на **TrackingRegistry** будет заполняться на последующих этапах.

## Регистрация события пребывания на локации в TrackingRegistry

На этом шаге обработки определяется прирост пробега и вагоны, которые входят в состав.

### Определение прироста пробега

Находится последнее значение, когда локомотив был зарегистрирован в **TrackingRegistry**. Для этого события находится пробег из **GeoZoneTrackingRegistry** по ссылке. Находится разница между новым значением пробега и предыдущим. Если в **TrackingRegistry** есть записи после найденной (локомотив был зарегистрирован на промежуточных локациях), то их дистанции (в поле **LinkedData**) отнимаются от полученной дистанции. Полученное значение будет использовано для локомотива и вагонов состава.

### Определение вагонов состава

Вагоны состава определяются по таблице журнала составов **tracking.ComposedVehicleJournals**. Находится состав, в который входит локомотив (запись с **FinishedOn** == null и **AssetId** == локомотиву). Выбираются все позиции, которые, входят в этот состав и у которых **FinishedOn** == null).

### Формирование сообщений для ТОиР

Для локомотива и каждого вагона формируются записи в TrackingRegistry. При этом формируются сообщения для ТОиР с приростом пробега.

В **TrackingRegistry>LinkedData** также записывается информация для использования при обработке следующего события:

* 1. **Mileage­Delta** - прирост пробега.
  2. **GpsLink** - Ссылка на запись **GeoZoneTrackingRegistry**.