



# Картографические библиотеки Mfw

Дмитрий Лапшин

Старший разработчик

ООО “Открытая Мобильная Платформа”

# Схема работы приложения



# Основные требования и критерии к ПО

## Функциональные:

- Легкая интеграция в код приложений
- Поддержка векторных (сжатость и хорошая масштабируемость) и растровых карт
- Работа в режиме онлайн/оффлайн

## Нефункциональные:

- Хорошая документация и примеры
- Возможность коммерческого использования
- Свободное ПО
- Разрешительная лицензия

# Решения для использования карт в ОС Аврора



## **Qt Location + WebTiles**

Минусы: только растровые тайлы, ограниченный функционал

Документация: [https://developer.auroraos.ru/doc/software\\_development/guides/webtiles](https://developer.auroraos.ru/doc/software_development/guides/webtiles)

## **OSM Scout server**

Минусы: сложная интеграция с Аврора и структура, не предоставляет API для работы с тайлами

Документация: <https://rinigus.github.io/osmscout-server/en/>

## **Navitel**

Минусы: стороннее решение, не предоставляющее средства интеграции

Документация: <https://navitel.ru/ru>

## **Yandex**

Минусы: отсутствует API под ОС Аврора, платные

Ссылка: <https://yandex.ru/dev/mapkit/doc/ru/>

## **Mfw libraries**

<https://gitlab.com/omprussia/examples-extra/MfwMap>

# Компоненты Mfw



**Библиотеки**

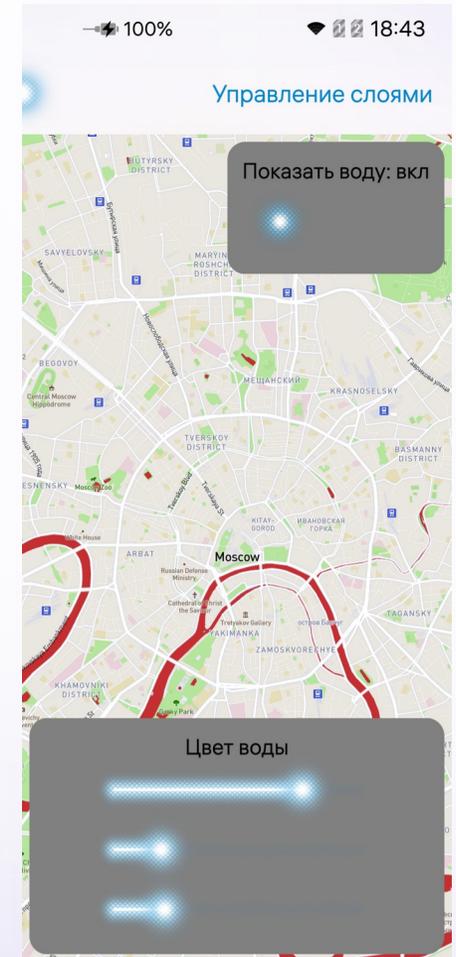
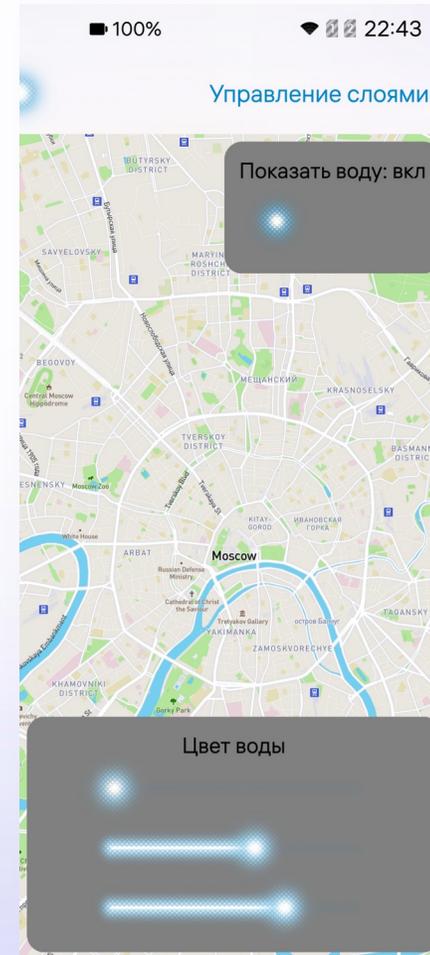
**Приложения-  
примеры**

**Скрипты для генерации  
оффлайн данных**

**Документация**

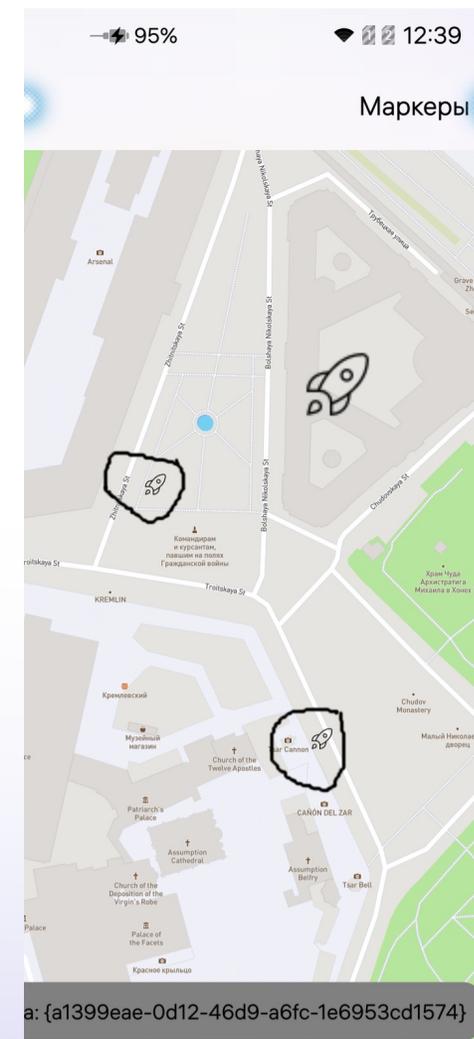
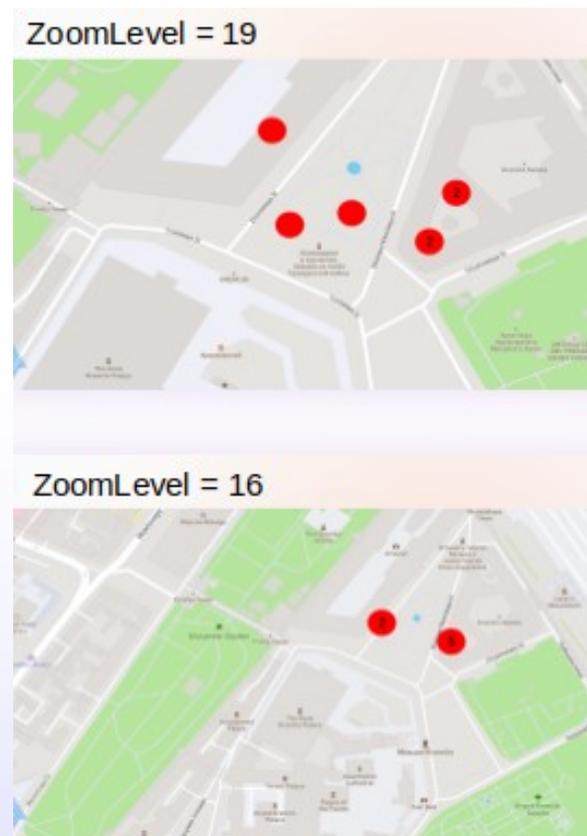
# Компонент отображения Карт: слои и стили

- Множество слоев (фон, заливка, линия, символ, растровое изображение, окружность) с возможностью управлять слоями
- Управление стилями (цвет заливки, фоновый узор и непрозрачность)
- Растровые изображения в качестве одного из слоев



# Компонент отображения Карт: объекты на карте

- Добавление группы объектов на карту с возможностью их кластеризации
- Маркер с местоположением устройства
- Метка с информацией об условиях предоставления картографических данных
- Маркеры (пины) с привязкой к координатам
- Маршруты на карте с подсказками для навигации и обновлением пройденного маршрута



# Компонент отображения Карт: управление

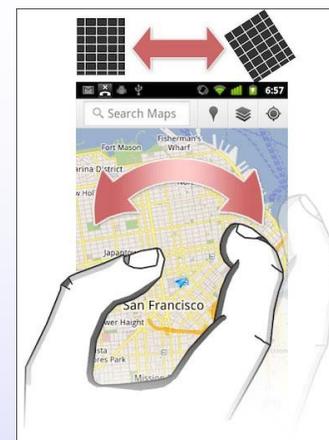
- Управление картой с помощью жестов,
- Управление картой с помощью API
- Отображение карты в статическом режиме без взаимодействия с пользователем



Перемещение



Приближение/  
Отдаление

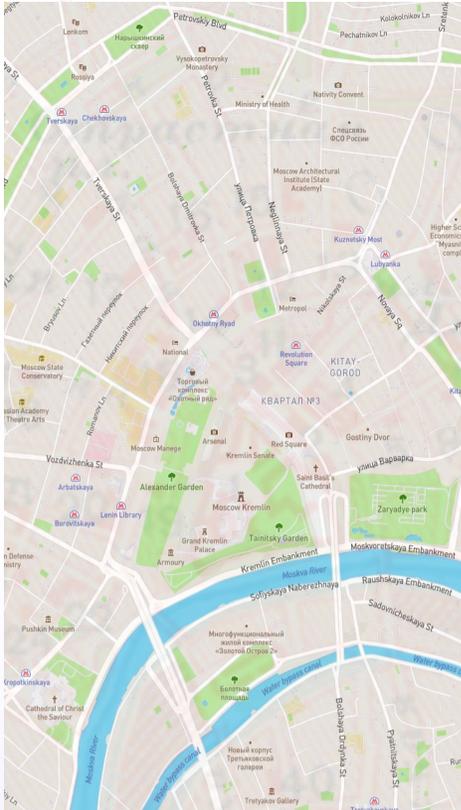


Вращение



Наклон

# Как карта получает тайлы



**Векторные  
тайлы:  
MVT PBF**

**Растровые  
тайлы:  
PNG  
JPG**

**Онлайн**

- [openstreetmap.org](https://openstreetmap.org),  
[maps.vk.com](https://maps.vk.com) и др.

- API Key

- accessToken

**Оффлайн**

- одна или несколько  
БД SQLite (MBTiles)

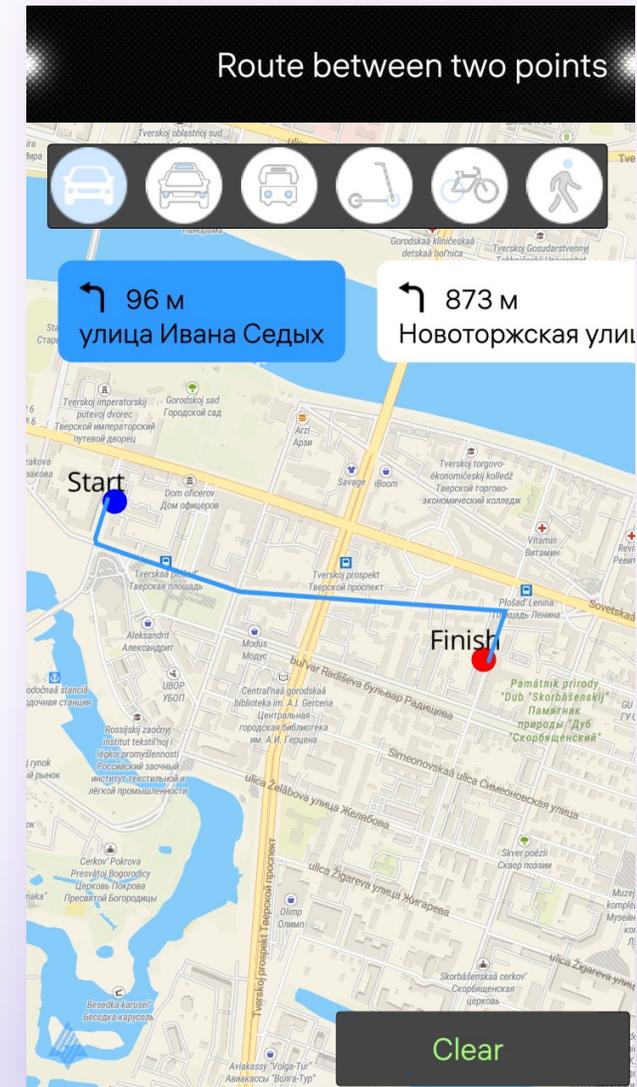
- набор файлов в  
директориях

**Сервер,  
предоста-  
вляющий  
OSM PBF**

**Скрипты для  
получения  
оффлайн  
данных**

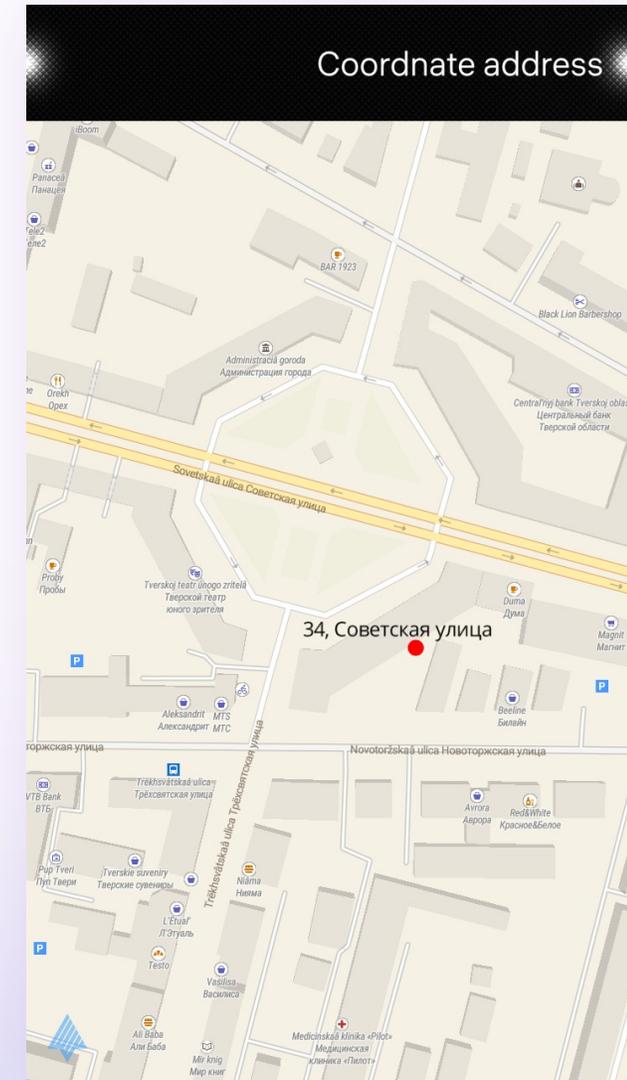
# Компонент построения маршрутов

- Не требуется доступ к сети
- Посторение маршрута как по двум точкам, так и по нескольким
- Выбор типа транспорта
- Подсказки при движении по маршруту

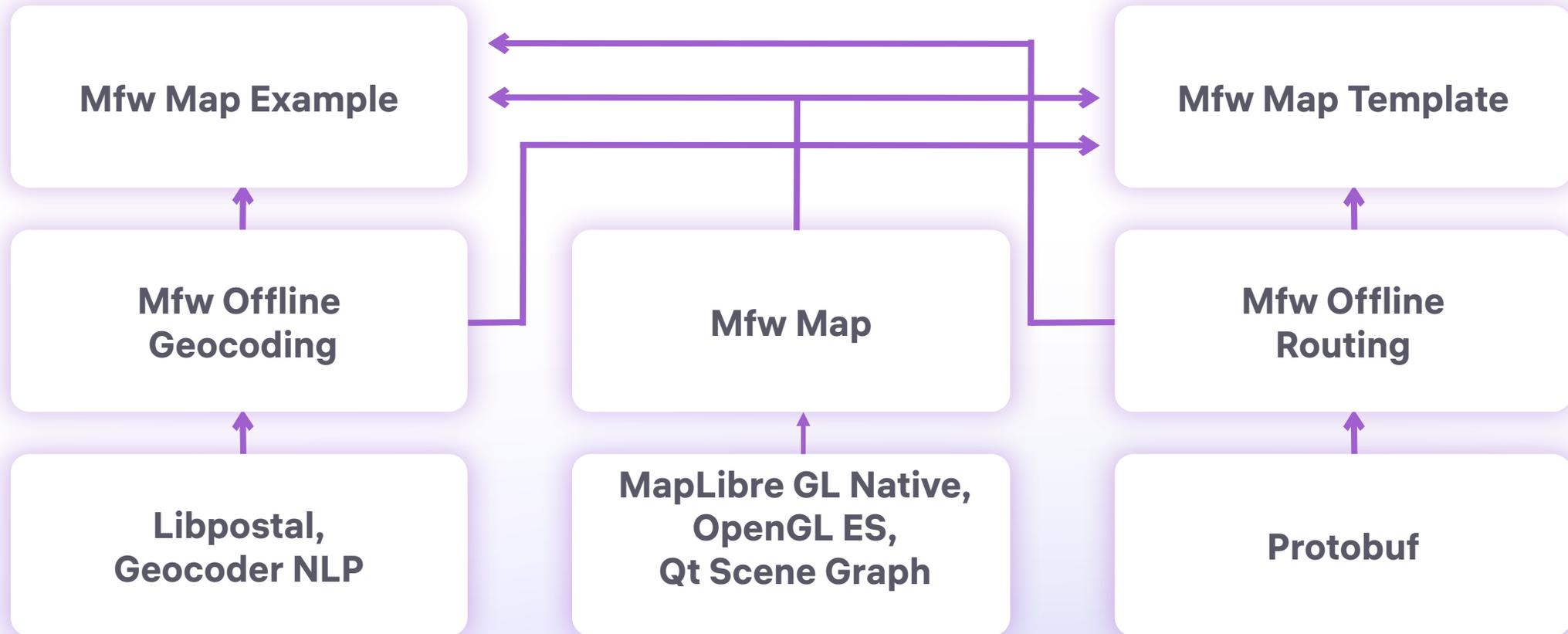


# Компонент геокодинга и поиска адресов

- Работает в режиме оффлайн
- Поиск по базе адресов с автоподсказкой и привязкой к текущему региону
- Определение координат для введенного в текстовом формате адреса
- Определение адреса и точек интереса по координатам



# Структура проекта



# Подключение классов плагинов в QtI-коде



```
import QtPositioning 5.2
import MfwMap 1.0
import MfwOfflineRouting 1.0
import MfwOfflineGeocoding 1.0

Page {

    MfwMap {
        id: map

        width: parent.width
        height: parent.height - header.height
        anchors.top: header.bottom

        center: QtPositioning.coordinate(51.659681, 39.204152) // Voronezh
        zoomLevel: 12.0
        metersPerPixelTolerance: 0.1

        accessToken: "pk.eyJ1Ijoic2xhdmFjaGVybmlrb2ZmIiwiaSI6ImNsZDBlemo0ejAxZnUzd3Fxc242Y2g2dHAifQ.YptKmH0xY8nfDfZnhdPiFg"
        styleUrl: "mapbox://styles/mapbox/streets-v10"
    }

    MfwOfflineRouting {
        id: router

        offlinePath: selectedFile ? selectedFile : ""
    }
}
//...
```

# Некоторые технические особенности компонента отображения карт



- Получение координат маркера с местоположением устройства через Qt Positioning
- Реализован API конвертации геометрических координат относительно компонента карты в географические и обратно
- Кластеризация основана на библиотеке MapLibre GL

# Способы создания объектов на карте



- Объекты, описанные с помощью GeoJSON, включая методы управления ими
- Библиотечные классы и методы Qt Quick (произвольные линии, полигоны, окружности, надписи, маркеры и др.).
- Изменение свойств объектов автоматически приводит к обновлению соответствующих объектов на карте.
- Динамическое перемещение объектов.

```
mainPage.cluster = {
  "type": "geojson",
  "data": {
    "type": "FeatureCollection",
    "properties": {},
    "features": [
      {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [37.618072336, 55.7529749653]
        }
      },
      {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [37.618437116, 55.7529387382]
        }
      }
    ]
  },
  "cluster": true,
  "clusterMaxZoom": 16,
  "clusterRadius": 50
}
```

# Файл настройки стилизации

- Стили в формате MapLibre GL Style JSON с возможностью динамической смены стилей

```
1 {
2   "version": 8,
3   "name": "Dark Matter",
4
5   "sources": {
6     "openmaptiles": {
7       "type": "vector",
8       "url": "mbtiles://{TILES_PATH}/output.mbtiles"
9     }
10  },
11  "glyphs": "file://{TILES_PATH}/glyphs/{fontstack}/{range}.pbf",
12  "layers": [
13    {
14      "id": "background",
15      "type": "background",
16      "paint": {"background-color": "#162639"}
17    },
18    {
19      "id": "water",
20      "type": "fill",
21      "source": "openmaptiles",
22      "source-layer": "water",
23      "filter": [
24        "all",
25        ["=", "$type", "Polygon"],
26        ["!=", "brunnel", "tunnel"]
27      ],
28      "layout": {"visibility": "visible"},
29      "paint": {"fill-antialias": false, "fill-color": "#041326"}
30    },
```

# Некоторые технические особенности компонентов маршрута и геокодинга



- Компонент построения маршрута - невизуальный, основан на библиотеке Valhalla, построение маршрута с учетом выбранного типа транспорта — так же предоставляется Valhalla
- Компонент геокодинга - невизуальный, основан на библиотеках Geocoder-NLP/libpostal
- Компонент геокодинга использует нейросеть для автоподсказок адреса российского региона

# Скрипты генерации оффлайн данных



# Где скачать?



Пример Mfw Map  
example:

<https://gitlab.com/omp/russia/examples-extra/MfwMap>



Пример Mfw Map  
Template:

<https://gitlab.com/omp/russia/demos/MapTemplate>



Плагин Mfw Map:

<https://gitlab.com/omp/russia/libraries/mfw-map>



Плагин Mfw Offline  
Routing:

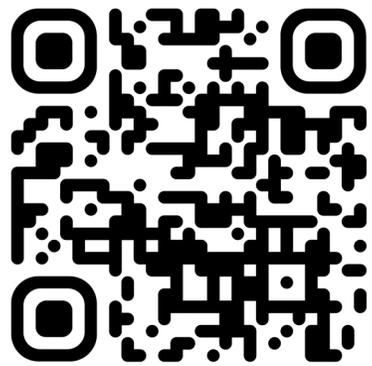
<https://gitlab.com/omp/russia/libraries/mfw-offline-routing>



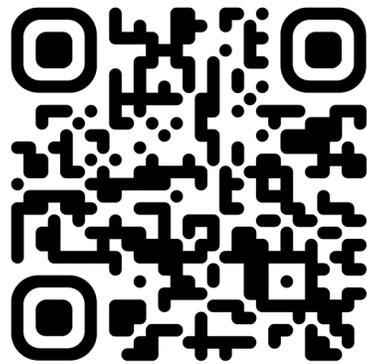
Плагин Mfw Offline  
Geocoding:

<https://gitlab.com/omp/russia/libraries/mfw-offline-geocoding>

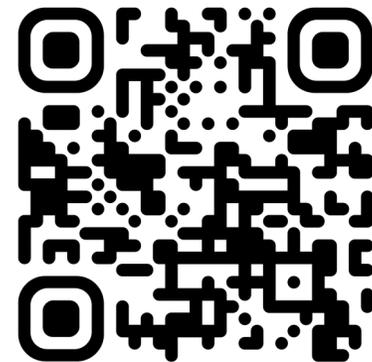
# Спасибо!



Сообщество  
во Вконтакте



Сайт  
auroraos.ru



Канал в Telegram