

Документация по серверу AGL

1. Необходимые модули для Api:

```
# npm install MD5
# npm install kladrapi
# npm install node-fetch
# npm install mongoose
# npm install redis
```

2. Необходимые модули для Agl Server:

```
# npm install cluster
# npm install minimist
# npm install net
```

3. Необходимые модули для Agl agent WS Server:

```
# npm install net
# npm install http
# npm install websocket
# npm install minimist
```

Параметры запуска для Agl Server:

```
# ./agl.js --redis=127.0.0.1:6379 --mongo=127.0.0.1:27017 --fork=127.0.0.1:4420 --rpass=password --rdb=12
```

или

```
# ./agl.js -r 127.0.0.1:6379 -m 127.0.0.1:27017 -f 127.0.0.1:4420 -p password -b 12
```

Параметры запуска для Agl agent WS Server:

```
# ./ws.js --redis=127.0.0.1:6379 --server=127.0.0.1:3320 --fork=127.0.0.1:4420 --rpass=password --rdb=12
```

или

```
# ./ws.js -r 127.0.0.1:6379 -s 127.0.0.1:3320 -f 127.0.0.1:4420 -p password -b 12
```

Данные параметры за исключением **Password Redis** и **DataBase Redis** прописаны по умолчанию и вводить их нет необходимости, параметры запуска указываются только если они отличаются от текущих.

Описание экшенов API

1. **parseAddress** - Преобразование строкового вида адреса в объект (субъекты адреса должны быть разделены запятой и иметь обозначение типа субъекта)

Параметры запроса:

address - Адрес для интерпретации данных (субъекты должны быть разделены запятыми)

*

запрос:

address = страна Российская Федерация, Ивановская область, Южский район, город Южа, улица Смирнова, дом 6, квартира 51

ответ:

```
{
  "country": {
    "name": "Российская федерация",
    "type": "Страна"
  },
  "region": {
    "name": "Ивановская",
    "type": "Область"
  },
  "district": {
    "name": "Южский",
    "type": "Район"
  },
  "city": {
    "name": "Южа",
    "type": "Город"
  },
  "street": {
    "name": "Смирнова",
    "type": "Улица"
  },
  "house": {
    "name": "6",
    "type": "Дом"
  },
  "apartment": {
    "name": "51",
    "type": "Квартира"
  },
  "address": "{country}, {region}, {district}, {city}, {street}, {house}, {apartment}",
  "fullAddress": "Российская федерация страна, Ивановская область, Южский район, Южа город, Смирнова улица, 6 дом, 51 квартира",
  "lightAddress": "Российская федерация, Ивановская, Южа, Смирнова, 6"
}
```

если в качестве параметра передать строку без разделителей запятых и явных обозначений субъекта то система будет считать что это страна и обернет объект в параметр **subject**

2. **getAddress** - Извлечение данных адреса из строки

Параметры запроса:

address - Адрес для интерпретации данных (разделение запятыми субъектов не обязательно) *

запрос:

address = Россия Ивановская область г.Иваново пр-т.Строителей

ответ:

```
{
  country: {
    name: 'Россия',
    type: 'Страна',
    typeShort: 'ст-а',
    contentType: 'country',
    nameFull: 'Российская Федерация',
    nameShort: 'РФ',
    id: '7',
    code: 'ru',
    lat: 61.52401,
    lng: 105.318756,
    gps: [105.318756, 61.52401],
    timezone: {
      dstOffset: 0,
      rawOffset: 25200,
      timeZoneId: 'Asia/Krasnoyarsk',
      timeZoneName: 'Красноярск, стандартное время'
    }
  },
  region: {
    name: 'Ивановская',
    zip: null,
    type: 'Область',
    typeShort: 'обл',
    okato: '240000000000',
    contentType: 'region',
    id: '37000000000000',
    code: 'ru',
    lat: 57.1056854,
    lng: 41.4830084,
    gps: [41.4830084, 57.1056854],
    timezone: {
      dstOffset: 0,
      rawOffset: 10800,
      timeZoneId: 'Europe/Moscow',
      timeZoneName: 'Москва, стандартное время'
    }
  },
  district: false,
  city: {
    timezone: {
```

```

        timeZoneName: 'Москва, стандартное время',
        timeZoneId: 'Europe/Moscow',
        rawOffset: 10800,
        dstOffset: 0
    },
    id: '370000001000000',
    name: 'Иваново',
    zip: 153015,
    type: 'Город',
    typeShort: 'г',
    okato: '244010000000',
    contentType: 'city',
    code: 'ru',
    lat: '56.9984452',
    lng: '40.9737394',
    regionId: '37000000000000',
    gps: [40.9737394, 56.9984452]
},
street: {
    name: 'Строителей',
    zip: 153038,
    type: 'Проспект',
    typeShort: 'пр-кт',
    okato: '244010000000',
    contentType: 'street',
    id: '370000001000098000',
    code: 'ru',
    lat: '56.9686335',
    lng: '41.0111737',
    gps: [41.0111737, 56.9686335],
    timezone: {
        dstOffset: 0,
        rawOffset: 10800,
        timeZoneId: 'Europe/Moscow',
        timeZoneName: 'Москва, стандартное время'
    },
    regionId: '37000000000000',
    cityId: '370000001000000'
}
}

```

данные извлекаются из локальной базы, и если каких-то субъектов в базе нет, то данный запрос результата не даст. В таком случае следует использовать метод **findAddress**

3. **findAddress** - Поиск данных адреса по строке

Параметры запроса:

address - Адрес для интерпретации данных (разделение запятыми субъектов обязательно)
*

запрос:

address = Россия Ивановская область г.Иваново пр-т.Строителей

ответ:

```
{
  country: {
    name: 'Россия',
    type: 'Страна',
    typeShort: 'ст-а',
    contentType: 'country',
    nameFull: 'Российская Федерация',
    nameShort: 'РФ',
    id: '7',
    code: 'ru',
    lat: 61.52401,
    lng: 105.318756,
    gps: [ 105.318756, 61.52401 ],
    timezone: {
      dstOffset: 0,
      rawOffset: 25200,
      timeZoneId: 'Asia/Krasnoyarsk',
      timeZoneName: 'Красноярск, стандартное время'
    }
  },
  region: {
    name: 'Ивановская',
    zip: null,
    type: 'Область',
    typeShort: 'обл',
    okato: '240000000000',
    contentType: 'region',
    id: '37000000000000',
    code: 'ru',
    lat: 57.1056854,
    lng: 41.4830084,
    gps: [41.4830084, 57.1056854],
    timezone: {
      dstOffset: 0,
      rawOffset: 10800,
      timeZoneId: 'Europe/Moscow',
      timeZoneName: 'Москва, стандартное время'
    }
  },
  district: false,
  city: {
    timezone: {
```

```

        timeZoneName: 'Москва, стандартное время',
        timeZoneId: 'Europe/Moscow',
        rawOffset: 10800,
        dstOffset: 0
    },
    id: '370000001000000',
    name: 'Иваново',
    zip: 153015,
    type: 'Город',
    typeShort: 'г',
    okato: '244010000000',
    contentType: 'city',
    code: 'ru',
    lat: '56.9984452',
    lng: '40.9737394',
    regionId: '37000000000000',
    gps: [40.9737394, 56.9984452]
},
street: {
    name: 'Строителей',
    zip: 153038,
    type: 'Проспект',
    typeShort: 'пр-кт',
    okato: '244010000000',
    contentType: 'street',
    id: '370000001000098000',
    code: 'ru',
    lat: '56.9686335',
    lng: '41.0111737',
    gps: [41.0111737, 56.9686335],
    timezone: {
        dstOffset: 0,
        rawOffset: 10800,
        timeZoneId: 'Europe/Moscow',
        timeZoneName: 'Москва, стандартное время'
    },
    regionId: '37000000000000',
    cityId: '370000001000000'
}
}

```

Разница между методом **findAddress** и **getAddress** заключается в том, что данный метод ищет данные не только в своей базе но и за пределами в интернете, следовательно выполняться он будет дольше и нагрузка на систему будет выше чем у getAddress

4. *findCountry* - Поиск страны

Параметры запроса:

str - Начальные символы названия страны *

limit - Количество запрашиваемых данных (максимум 100, по умолчанию 10)

запрос:

str = Россия

limit = 1

ответ:

```
[{
  name: 'Россия',
  type: 'Страна',
  typeShort: 'ст-а',
  contentType: 'country',
  nameFull: 'Российская Федерация',
  nameShort: 'РФ',
  id: '7',
  code: 'ru',
  lat: 61.52401,
  lng: 105.318756,
  gps: [105.318756, 61.52401],
  timezone: {
    dstOffset: 0,
    rawOffset: 25200,
    timeZoneId: 'Asia/Krasnoyarsk',
    timeZoneName: 'Красноярск, стандартное время'
  }
}]
```

поиск стран происходит как внутри локальной базы так и за её пределами в интернете
если в локальной базе данные не найдены

5. *findRegion* - Поиск региона

Параметры запроса:

str - Начальные символы названия региона *

limit - Количество запрашиваемых данных (максимум 100, по умолчанию 10)

запрос:

str = Ивановская

limit = 1

ответ:

```
[{
  name: 'Ивановская',
  zip: null,
  type: 'Область',
  typeShort: 'обл',
  okato: '240000000000',
  contentType: 'region',
  id: '37000000000000',
  code: 'ru',
  lat: 57.1056854,
  lng: 41.4830084,
  gps: [41.4830084, 57.1056854],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  }
}]
```

поиск регионов происходит как внутри локальной базы так и за её пределами в интернете
если в локальной базе данные не найдены

6. *findDistrict* - Поиск районов в регионе

Параметры запроса:

str - Начальные символы названия района *

regionId - Идентификатор региона

limit - Количество запрашиваемых данных (максимум 100, по умолчанию 10)

запрос:

str = Южский

regionId = 3700000000000

limit = 1

ответ:

```
[{
  name: 'Южский',
  zip: 155635,
  type: 'Район',
  typeShort: 'р-н',
  okato: '24235000000',
  contentType: 'district',
  id: '3702000000000',
  code: 'ru',
  lat: 56.6089536,
  lng: 42.0256303,
  gps: [42.0256303, 56.6089536],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '3700000000000'
}]
```

поиск районов происходит как внутри локальной базы так и за её пределами в интернете
если в локальной базе данные не найдены

7. *findCity* - Поиск городов в регионе или районе

Параметры запроса:

str - Начальные символы названия города *

regionId - Идентификатор региона

districtId - Идентификатор района

limit - Количество запрашиваемых данных (максимум 100, по умолчанию 10)

запрос:

str = Южа

regionId = 3700000000000

limit = 1

ответ:

```
[{
  id: '3702000100000',
  name: 'Южа',
  zip: 155630,
  type: 'Город',
  typeShort: 'Г',
  okato: '24235501000',
  contentType: 'city',
  code: 'ru',
  lat: '56.584042',
  lng: '42.010929',
  regionId: '3700000000000',
  districtId: '3702000000000',
  gps: [42.010929, 56.584042],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
}]
```

поиск городов происходит как внутри локальной базы так и за её пределами в интернете
если в локальной базе данные не найдены

8. *findStreet* - Поиск улиц в городе

Параметры запроса:

str - Начальные символы названия улицы *

cityId - Идентификатор города

limit - Количество запрашиваемых данных (максимум 100, по умолчанию 10)

запрос:

str = Строителей

cityId = 37000000100000

limit = 1

ответ:

```
[{
  name: 'Строителей',
  zip: 153038,
  type: 'Проспект',
  typeShort: 'пр-кт',
  okato: '24401000000',
  contentType: 'street',
  id: '3700000010000098000',
  code: 'ru',
  lat: '56.9686335',
  lng: '41.0111737',
  metro: [],
  gps: [41.0111737, 56.9686335],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '37000000000000',
  cityId: '37000000100000'
}]
```

поиск улиц происходит как внутри локальной базы так и за её пределами в интернете
если в локальной базе данные не найдены

9. *findHouse* - Поиск домов на улице

Параметры запроса:

str - Первые строки в названии дома *

streetId - Идентификатор улицы где производится поиск

limit - Лимит вариантов к выдаче (по умолчанию 10, максимум 100)

запрос:

str = 3

streetId = 37019001000027900

limit = 1

ответ:

```
[{
  name: '3',
  zip: 155900,
  type: 'дом',
  typeShort: 'д',
  okato: '24411000000',
  contentType: 'building',
  id: '3701900100002790010',
  code: 'ru',
  lat: '56.8400475',
  lng: '41.3907855',
  gps: [41.3907855, 56.8400475],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '3700000000000',
  districtId: '3701900000000',
  cityId: '3701900100000',
  streetId: '37019001000027900'
}]
```

поиск домов происходит как внутри локальной базы так и за её пределами в интернете
если в локальной базе данные не найдены

10. *findMetro* - Поиск метро

Параметры запроса:

str - Первые строки в названии станции *

cityId - Идентификатор города где находится станция

limit - Лимит вариантов к выдаче (по умолчанию 10, максимум 100)

lineId - Идентификатор линии где находится станция

lineName - Название линии где находится станция

lineColor - Цвет линии где находится станция

запрос:

str = Румянцево

cityId = 77000000000000

limit = 1

ответ:

```
{
  id: '975c23bb4d69e2efc343cf11',
  name: 'Румянцево',
  lat: '55.633',
  lng: '37.4419',
  order: 20,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '77000000000000',
  gps: [37.4419, 55.633]
}
```

11. **findNearStationsMetroByIds** - Поиск по массиву идентификаторов станций метро, ближайших к ним станций метро

Параметры запроса:

ids - Массив идентификаторов станций метро *

distance - Расстояние поиска в метрах (по умолчанию 5000)

запрос:

ids = 975c23bb4d69e2efc343cf11, c3900aa9021cec3fce26bb41

ответ:

```
[{
  metro: {
    id: '975c23bb4d69e2efc343cf11',
    name: 'Румянцево',
    lat: '55.633',
    lng: '37.4419',
    order: 20,
    line: 'Сокольническая',
    color: 'E42313',
    city: 'Москва',
    lineId: '31b83c4d182032e41ad85357',
    cityId: '7700000000000',
    gps: [37.4419, 55.633]
  },
  near: [{
    id: '07fad26e60a6a0378d211c87',
    name: 'Саларьево',
    lat: '55.6227',
    lng: '37.424',
    order: 21,
    line: 'Сокольническая',
    color: 'E42313',
    city: 'Москва',
    lineId: '31b83c4d182032e41ad85357',
    cityId: '7700000000000',
    gps: [37.424, 55.6227]
  }, {
    id: 'c4fca5a3388d09a6d04cdde2',
    name: 'Тропарево',
    lat: '55.6459',
    lng: '37.4725',
    order: 19,
    line: 'Сокольническая',
    color: 'E42313',
    city: 'Москва',
    lineId: '31b83c4d182032e41ad85357',
    cityId: '7700000000000',
    gps: [37.4725, 55.6459]
  }, {
    id: '943f7c7793a06fcf3809d6b1',
    name: 'Юго-Западная',
```

```

lat: '55.663146',
lng: '37.482852',
order: 18,
line: 'Сокольническая',
color: 'E42313',
city: 'Москва',
lineId: '31b83c4d182032e41ad85357',
cityId: '7700000000000',
gps: [37.482852, 55.663146]
}, {
  id: '1db098df0a53fbda33d52d8e',
  name: 'Теплый Стан',
  lat: '55.61873',
  lng: '37.505912',
  order: 21,
  line: 'Калужско-Рижская',
  color: 'F07E24',
  city: 'Москва',
  lineId: '84aa74bc75c408fdd995998f',
  cityId: '7700000000000',
  gps: [37.505912, 55.61873]
}, {
  id: 'da82d147d1db1a39bb48b0d1',
  name: 'Коньково',
  lat: '55.631857',
  lng: '37.519156',
  order: 20,
  line: 'Калужско-Рижская',
  color: 'F07E24',
  city: 'Москва',
  lineId: '84aa74bc75c408fdd995998f',
  cityId: '7700000000000',
  gps: [37.519156, 55.631857]
}]
}, {
  metro: {
    id: 'c3900aa9021сес3fсе26bb41',
    name: 'Горьковская',
    lat: '56.313933',
    lng: '43.99482',
    order: 0,
    line: 'Автозаводская',
    color: 'D80707',
    city: 'Нижний Новгород',
    lineId: '874a3d2cdebc6d1bc5db1d49',
    cityId: '5200000100000',
    gps: [43.99482, 56.313933]
  },
  near: [{
    id: 'f97005fe546a9d76ddb8542b',
    name: 'Московская',
    lat: '56.321097',
    lng: '43.945799',
    order: 1,
    line: 'Автозаводская',
    color: 'D80707',

```

```

    city: 'Нижний Новгород',
    lineId: '874a3d2cdebc6d1bc5db1d49',
    cityId: '5200000100000',
    gps: [43.945799, 56.321097]
  }, {
    id: '15718ec80fc1fa9c89f23e47',
    name: 'Московская 2',
    lat: '56.321097',
    lng: '43.945799',
    order: 0,
    line: 'Сормовская',
    color: '0071BC',
    city: 'Нижний Новгород',
    lineId: '7f8ebb791fff539ddf71d161',
    cityId: '5200000100000',
    gps: [43.945799, 56.321097]
  }, {
    id: '63f6e64c94533b06cd4f12d6',
    name: 'Чкаловская',
    lat: '56.310637',
    lng: '43.936933',
    order: 2,
    line: 'Автозаводская',
    color: 'D80707',
    city: 'Нижний Новгород',
    lineId: '874a3d2cdebc6d1bc5db1d49',
    cityId: '5200000100000',
    gps: [43.936933, 56.310637]
  }, {
    id: 'd28612dc4d850c65f3a0c9e0',
    name: 'Ленинская',
    lat: '56.297798',
    lng: '43.937328',
    order: 3,
    line: 'Автозаводская',
    color: 'D80707',
    city: 'Нижний Новгород',
    lineId: '874a3d2cdebc6d1bc5db1d49',
    cityId: '5200000100000',
    gps: [43.937328, 56.297798]
  }, {
    id: 'f9473daa97e585b561cab76',
    name: 'Канавинская',
    lat: '56.320273',
    lng: '43.927438',
    order: 1,
    line: 'Сормовская',
    color: '0071BC',
    city: 'Нижний Новгород',
    lineId: '7f8ebb791fff539ddf71d161',
    cityId: '5200000100000',
    gps: [43.927438, 56.320273]
  }
}

```


12. ***findMetroById*** - Поиск метро по идентификатору станции метро

Параметры запроса:

id - Идентификатор станции метро *

запрос:

id = 975c23bb4d69e2efc343cf11

ответ:

```
{
  id: '975c23bb4d69e2efc343cf11',
  name: 'Румянцево',
  lat: '55.633',
  lng: '37.4419',
  order: 20,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '7700000000000',
  gps: [37.4419, 55.633]
}
```

13. **findMetroByStreetId** - Поиск метро по идентификатору улицы

Параметры запроса:

id - Идентификатор улицы *

запрос:

id = 770000000000095300

ответ:

```
[{
  id: '2f3b45576732245a55297b2d',
  name: 'Проспект Вернадского',
  lat: '55.676549',
  lng: '37.504584',
  order: 17,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '77000000000000',
  gps: [37.504584, 55.676549]
}, {
  id: '943f7c7793a06fcf3809d6b1',
  name: 'Юго-Западная',
  lat: '55.663146',
  lng: '37.482852',
  order: 18,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '77000000000000',
  gps: [37.482852, 55.663146]
}]
```

14. **findMetroByHouseId** - Поиск метро по идентификатору дома

Параметры запроса:

id - Идентификатор дома *

запрос:

id = 77000000000009530044

ответ:

```
[{
  id: '2f3b45576732245a55297b2d',
  name: 'Проспект Вернадского',
  lat: '55.676549',
  lng: '37.504584',
  order: 17,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '77000000000000',
  gps: [37.504584, 55.676549]
}, {
  id: '943f7c7793a06fcf3809d6b1',
  name: 'Юго-Западная',
  lat: '55.663146',
  lng: '37.482852',
  order: 18,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '77000000000000',
  gps: [37.482852, 55.663146]
}]
```

15. *hintCountries* - Подсказки по странам

Параметры запроса:

str - Подстрока для поиска *

запрос:

str = Ро

ответ:

```
[{
  name: 'Россия',
  type: 'Страна',
  typeShort: 'ст-а',
  contentType: 'country',
  nameFull: 'Российская Федерация',
  nameShort: 'РФ',
  id: '7',
  code: 'ru',
  lat: 61.52401,
  lng: 105.318756,
  gps: [105.318756, 61.52401],
  timezone: {
    dstOffset: 0,
    rawOffset: 25200,
    timeZoneId: 'Asia/Krasnoyarsk',
    timeZoneName: 'Красноярск, стандартное время'
  }
}]
```

подсказки работают за счет кэширования данных, если в кэше данных нет то подсказки работать не будут и данные сначала нужно загрузить с помощью метода **findCountry**

16. *hintRegions* - Подсказки по регионам

Параметры запроса:

str - Подстрока для поиска *

запрос:

str = Ив

ответ:

```
[{
  name: 'Ивановская',
  zip: null,
  type: 'Область',
  typeShort: 'обл',
  okato: '240000000000',
  contentType: 'region',
  id: '37000000000000',
  code: 'ru',
  lat: 57.1056854,
  lng: 41.4830084,
  gps: [41.4830084, 57.1056854],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  }
}]
```

подсказки работают за счет кэширования данных, если в кэше данных нет то подсказки работать не будут и данные сначала нужно загрузить с помощью метода **findRegion**

17. **hintDistricts** - Подсказки по районам

Параметры запроса:

str - Подстрока для поиска *

regionId - Идентификатор региона

запрос:

str = Шы

regionId = 3700000000000

ответ:

```
[{
  name: 'Шуйский',
  zip: 155903,
  type: 'Район',
  typeShort: 'р-н',
  okato: '24233000000',
  contentType: 'district',
  id: '3701900000000',
  code: 'ru',
  lat: 56.7586237,
  lng: 41.3915336,
  gps: [41.3915336, 56.7586237],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '3700000000000'
}]
```

подсказки работают за счет кэширования данных, если в кэше данных нет то подсказки работать не будут и данные сначала нужно загрузить с помощью метода **findDistrict**

18. *hintCities* - Подсказки по городам

Параметры запроса:

str - Подстрока для поиска *

regionId - Идентификатор региона

districtId - Идентификатор района

запрос:

str = Шу

regionId = 3700000000000

ответ:

```
[{
  name: 'Шуя',
  zip: 155905,
  type: 'Город',
  typeShort: 'г',
  okato: '24411000000',
  contentType: 'city',
  id: '3701900100000',
  code: 'ru',
  lat: '56.8441',
  lng: '41.3796',
  gps: [41.3796, 56.8441],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '3700000000000',
  districtId: '3701900000000'
}]
```

подсказки работают за счет кэширования данных, если в кэше данных нет то подсказки работать не будут и данные сначала нужно загрузить с помощью метода **findCity**

19. *hintStreets* - Подсказки по улицам

Параметры запроса:

str - Подстрока для поиска *

cityId - Идентификатор города

запрос:

str = Св

cityId = 3701900100000

ответ:

```
[{
  name: 'Свердлова',
  zip: 155908,
  type: 'Улица',
  typeShort: 'ул',
  okato: '24411000000',
  contentType: 'street',
  id: '37019001000027900',
  code: 'ru',
  lat: '56.8400475',
  lng: '41.3907855',
  gps: [41.3907855, 56.8400475],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '3700000000000',
  districtId: '3701900000000',
  cityId: '3701900100000'
}]
```

подсказки работают за счет кэширования данных, если в кэше данных нет то подсказки работать не будут и данные сначала нужно загрузить с помощью метода **findStreet**

20. *hintHouses* - Подсказки по домам

Параметры запроса:

str - Подстрока для поиска *

streetId - Идентификатор улицы

запрос:

str = 30

streetId = 37019001000027900

ответ:

```
[{
  name: '30',
  zip: 155900,
  type: 'дом',
  typeShort: 'д',
  okato: '24411000000',
  contentType: 'building',
  id: '3701900100002790010',
  code: 'ru',
  lat: '56.8400475',
  lng: '41.3907855',
  gps: [41.3907855, 56.8400475],
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '3700000000000',
  districtId: '3701900000000',
  cityId: '3701900100000',
  streetId: '37019001000027900'
}]
```

подсказки работают за счет кэширования данных, если в кэше данных нет то подсказки работать не будут и данные сначала нужно загрузить с помощью метода **findHouse**

21. *hintMetro* - Подсказки по метро

Параметры запроса:

str - Подстрока для поиска *

streetId - Идентификатор улицы

houseId - Идентификатор дома

запрос:

str = Пр

streetId = 770000000000095300

ответ:

```
{
  id: '2f3b45576732245a55297b2d',
  name: 'Проспект Вернадского',
  lat: '55.676549',
  lng: '37.504584',
  order: 17,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '77000000000000',
  gps: [37.504584, 55.676549]
}
```

22. **getAddressByGPS** - Геокодирование по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

запрос:

lat = 55.676549

lng = 37.504584

ответ:

```
{
  id: '9c74954dc1d76371ca89f6d9',
  lat: '55.676514',
  lng: '37.504633',
  gps: [37.504633, 55.676514],
  boundingbox: ['55.676414', '55.676614', '37.504533', '37.504733'],
  description: '303, проспект Вернадского, Проспект Вернадского, район Проспект
Вернадского, Западный административный округ, Москва, ЦФО, 119415, Россия',
  address: {
    zip: 119415,
    city: 'район Проспект Вернадского',
    code: 'ru',
    street: 'проспект Вернадского',
    region: 'Москва',
    country: 'Россия',
    district: 'Западный административный округ'
  }
}
```

данный метод является системным и использовать его не рекомендуется, так как может нарушиться работоспособность всей системы на сутки и более

23. **getAddressByString** - Геокодирование по адресу

Параметры запроса:

address - Адрес запроса *

запрос:

address = 303, проспект Вернадского, Проспект Вернадского, район Проспект Вернадского, Западный административный округ, Москва, ЦФО, 119415, Россия

ответ:

```
{
  id: '9c74954dc1d76371ca89f6d9',
  lat: '55.676514',
  lng: '37.504633',
  gps: [37.504633, 55.676514],
  boundingbox: ['55.676414', '55.676614', '37.504533', '37.504733'],
  description: '303, проспект Вернадского, Проспект Вернадского, район Проспект
Вернадского, Западный административный округ, Москва, ЦФО, 119415, Россия',
  address: {
    zip: 119415,
    city: 'район Проспект Вернадского',
    code: 'ru',
    street: 'проспект Вернадского',
    region: 'Москва',
    country: 'Россия',
    district: 'Западный административный округ'
  }
}
```

данный метод является системным и использовать его не рекомендуется, так как может нарушиться работоспособность всей системы на сутки и более

24. *getCountries* - Получение данных стран

Параметры запроса:

page - Номер страницы начиная с 1 (по умолчанию 1)

limit - Количество загружаемых документов, максимально 100 (по умолчанию 10)

запрос:

page = 1
limit = 10

ответ:

```
{
  page: 1,
  limit: 10,
  count: 1,
  data: [{
    name: 'Россия',
    type: 'Страна',
    typeShort: 'ст-а',
    contentType: 'country',
    nameFull: 'Российская Федерация',
    nameShort: 'РФ',
    id: '7',
    code: 'ru',
    lat: 61.52401,
    lng: 105.318756,
    gps: [105.318756, 61.52401],
    timezone: {
      dstOffset: 0,
      rawOffset: 25200,
      timeZoneId: 'Asia/Krasnoyarsk',
      timeZoneName: 'Красноярск, стандартное время'
    }
  }]
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findCountry**

25. *getRegions* - Получение данных регионов

Параметры запроса:

type - Тип искомого значения (область, край, республика, автономный округ)

page - Номер страницы начиная с 1 (по умолчанию 1)

limit - Количество загружаемых документов, максимально 100 (по умолчанию 10)

запрос:

page = 1

limit = 2

ответ:

```
{
  page: 1,
  limit: 2,
  count: 42,
  data: [{
    name: 'Татарстан',
    zip: '420000',
    type: 'Республика',
    typeShort: 'Респ',
    okato: '920000000000',
    contentType: 'region',
    id: '1600000000000',
    code: 'ru',
    lat: 55.1802364,
    lng: 50.7263945,
    gps: [50.7263945, 55.1802364],
    timezone: {
      "timeZoneName" : "Москва, стандартное время",
      "timeZoneld" : "Europe/Moscow",
      "rawOffset" : 10800,
      "dstOffset" : 0
    }
  }, {
    name: 'Калмыкия',
    zip: null,
    type: 'Республика',
    typeShort: 'Респ',
    okato: '850000000000',
    contentType: 'region',
    id: '0800000000000',
    code: 'ru',
    lat: 46.5676845,
    lng: 45.7731614,
    gps: [45.7731614, 46.5676845],
    timezone: {
      "timeZoneName" : "Москва, стандартное время",
      "timeZoneld" : "Europe/Moscow",
      "rawOffset" : 10800,
      "dstOffset" : 0
    }
  }
}]
```

}

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findRegion**

26. *getDistricts* - Получение данных районов

Параметры запроса:

regionId - Идентификатор региона *

type - Тип искомого значения (район, округ)

page - Номер страницы начиная с 1 (по умолчанию 1)

limit - Количество загружаемых документов, максимально 100 (по умолчанию 10)

запрос:

```
page = 1
limit = 2
regionId = 1600000000000
```

ответ:

```
{
  page: 1,
  limit: 2,
  count: 21,
  data: [{
    name: 'Буинский',
    zip: 422404,
    type: 'Район',
    typeShort: 'р-н',
    okato: '922180000000',
    contentType: 'district',
    id: '1601500000000',
    code: 'ru',
    lat: 54.9642608,
    lng: 48.2035894,
    gps: [48.2035894, 54.9642608],
    timezone: {
      "timeZoneName" : "Москва, стандартное время",
      "timeZoneld" : "Europe/Moscow",
      "rawOffset" : 10800,
      "dstOffset" : 0
    },
    regionId: '1600000000000'
  }, {
    name: 'Тюлячинский',
    zip: 422080,
    type: 'Район',
    typeShort: 'р-н',
    okato: '922560000000',
    contentType: 'district',
    id: '1604100000000',
    code: 'ru',
    lat: 55.8868459,
    lng: 50.2265391,
    gps: [50.2265391, 55.8868459],
    timezone: {
      "timeZoneName" : "Москва, стандартное время",
      "timeZoneld" : "Europe/Moscow",
```



```
        "rawOffset" : 10800,  
        "dstOffset" : 0  
    },  
    regionId: '16000000000000'  
}  
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findDistrict**

27. *getCities* - Получение данных городов

Параметры запроса:

regionId - Идентификатор региона

districtId - Идентификатор района

type - Тип искомого значения (деревня, село, город)

page - Номер страницы начиная с 1 (по умолчанию 1)

limit - Количество загружаемых документов, максимально 100 (по умолчанию 10)

запрос:

page = 1

limit = 2

regionId = 1600000000000

ответ:

```
{
  page: 1,
  limit: 2,
  count: 24,
  data: [{
    id: '160000001000000',
    name: 'Казань',
    zip: 420032,
    type: 'Город',
    typeShort: 'г',
    okato: '924010000000',
    contentType: 'city',
    code: 'ru',
    lat: '55.76041925',
    lng: '49.1902944737668',
    timezone: {
      "timeZoneName" : "Москва, стандартное время",
      "timeZoneId" : "Europe/Moscow",
      "rawOffset" : 10800,
      "dstOffset" : 0
    },
    regionId: '16000000000000',
    metro: [
      "c89658dfa3d5e8402ff6bb32",
      "74e101c88320d010a2813d95",
      "a470f1203404b84eb63029bf",
      "662d205f4abf3878a8053d77",
      "659ecb47e92e377d5e7a8a4f",
      "6b4a188e47c68d9fc4d9109d",
      "d1116ec169c6dbd45a75ee24",
      "f00cbdc3c0340c011d73fb7f",
      "3d3d22992a6c15237762b067",
      "425b27c712f4d205997eab1a"
    ],
    gps: [49.1902944737668, 55.76041925]
  }, {
    id: '160000002000000',
    name: 'Набережные Челны',
```

```
    zip: 423812,  
    type: 'Город',  
    typeShort: 'г',  
    okato: '924300000000',  
    contentType: 'city',  
    code: 'ru',  
    lat: '55.7420117',  
    lng: '52.3992173',  
    timezone: {  
        "dstOffset" : 0,  
        "rawOffset" : 10800,  
        "timeZoneId" : "Europe/Moscow",  
        "timeZoneName" : "Москва, стандартное время"  
    },  
    regionId: '16000000000000',  
    metro: [],  
    gps: [52.3992173, 55.7420117]  
    }  
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findCity**

28. **getStreets** - Получение данных улиц

Параметры запроса:

cityId - Идентификатор города

type - Тип искомого значения (улица, площадь, проспект)

page - Номер страницы начиная с 1 (по умолчанию 1)

limit - Количество загружаемых документов, максимально 100 (по умолчанию 10)

запрос:

page = 1

limit = 2

cityId = 7700000000000

ответ:

```
{
  page: 1,
  limit: 2,
  count: 1,
  data: [{
    id: '770000000000095300',
    name: 'Вернадского',
    zip: 119415,
    type: 'Проспект',
    typeShort: 'пр-кт',
    okato: null,
    contentType: 'street',
    code: 'ru',
    lat: '55.6752212',
    lng: '37.495215',
    timezone: {
      "timeZoneName" : "Москва, стандартное время",
      "timeZoneId" : "Europe/Moscow",
      "rawOffset" : 10800,
      "dstOffset" : 0
    },
    regionId: '7700000000000',
    cityId: '7700000000000',
    metro: [
      "2f3b45576732245a55297b2d",
      "943f7c7793a06fcf3809d6b1"
    ],
    gps: [37.495215, 55.6752212]
  }]
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findStreet**

29. *getHouses* - Получение данных домов

Параметры запроса:

streetId - Идентификатор улицы

type - Тип искомого значения (тип постройки)

page - Номер страницы начиная с 1 (по умолчанию 1)

limit - Количество загружаемых документов, максимально 100 (по умолчанию 10)

запрос:

page = 1

limit = 2

streetId = 77000000000095300

ответ:

```
{
  page: 1,
  limit: 2,
  count: 2,
  data: [{
    id: '7700000000009530044',
    name: '105к1',
    zip: 119526,
    type: 'дом',
    typeShort: 'д',
    okato: '45268592000',
    contentType: 'building',
    code: 'ru',
    lat: '55.6752212',
    lng: '37.495215',
    timezone: {
      "dstOffset" : 0,
      "rawOffset" : 10800,
      "timeZoneId" : "Europe/Moscow",
      "timeZoneName" : "Москва, стандартное время"
    },
    regionId: '77000000000000',
    cityId: '77000000000000',
    streetId: '77000000000095300',
    metro: [
      "2f3b45576732245a55297b2d",
      "943f7c7793a06fcf3809d6b1"
    ],
    gps: [37.495215, 55.6752212]
  }, {
    id: '7700000000009530045',
    name: '105к3стр1',
    zip: 119526,
    type: 'дом',
    typeShort: 'д',
    okato: '45268592000',
    contentType: 'building',
    code: 'ru',
    lat: '55.6752212',
```

```
    lng: '37.495215',
    timezone: {
      "dstOffset" : 0,
      "rawOffset" : 10800,
      "timeZoneId" : "Europe/Moscow",
      "timeZoneName" : "Москва, стандартное время"
    },
    regionId: '77000000000000',
    cityId: '77000000000000',
    streetId: '770000000000095300',
    metro: [
      "2f3b45576732245a55297b2d",
      "943f7c7793a06fcf3809d6b1"
    ],
    gps: [37.495215, 55.6752212]
  }
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findHouse**

30. *getMetro* - Получение данных метро

Параметры запроса:

cityId - Идентификатор города

lineId - Идентификатор линии метро

page - Номер страницы начиная с 1 (по умолчанию 1)

limit - Количество загружаемых документов, максимально 100 (по умолчанию 10)

запрос:

page = 1

limit = 2

cityId = 7700000000000

ответ:

```
{
  page: 1,
  limit: 2,
  count: 100,
  data: [{
    id: 'd680731df1cf225d20d1f71b',
    name: 'Арбатская',
    lat: '55.752122',
    lng: '37.601553',
    order: 9,
    line: 'Филевская',
    color: '1EBCEF',
    city: 'Москва',
    lineId: '88abe571337b0547723be907',
    cityId: '7700000000000',
    gps: [37.601553, 55.752122]
  }, {
    id: '8169e9d83c8b1b0cc80ae2d4',
    name: 'Перово',
    lat: '55.75098',
    lng: '37.78422',
    order: 2,
    line: 'Калининская',
    color: 'FFCD1C',
    city: 'Москва',
    lineId: '13079b440bd88586a6265525',
    cityId: '7700000000000',
    gps: [37.78422, 55.75098]
  }]
}
```

31. ***getCountryById*** - Получение данных страны по идентификатору

Параметры запроса:

id - Идентификатор страны *

запрос:

id = 7

ответ:

```
{
  id: '7',
  name: 'Россия',
  type: 'Страна',
  typeShort: 'ст-а',
  contentType: 'country',
  nameFull: 'Российская Федерация',
  nameShort: 'РФ',
  code: 'ru',
  lat: '61.52401',
  lng: '105.318756',
  gps: [105.318756, 61.52401]
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findCountry**

32. **getRegionById** - Получение данных региона по идентификатору

Параметры запроса:

id - Идентификатор региона *

запрос:

id = 01000000000000

ответ:

```
{
  id: '01000000000000',
  name: 'Адыгея',
  zip: 385000,
  type: 'Республика',
  typeShort: 'Респ',
  okato: '790000000000',
  contentType: 'region',
  code: 'ru',
  lat: '44.8229155',
  lng: '40.1754463',
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  gps: [40.1754463, 44.8229155]
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findRegion**

33. *getDistrictByld* - Получение данных района по идентификатору

Параметры запроса:

id - Идентификатор района *

запрос:

id = 01008000000000

ответ:

```
{
  id: '01008000000000',
  name: 'Гиагинский',
  zip: 385600,
  type: 'Район',
  typeShort: 'р-н',
  okato: '792050000000',
  contentType: 'district',
  code: 'ru',
  lat: '44.8692223',
  lng: '40.170299',
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '01000000000000',
  gps: [40.170299, 44.8692223]
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findDistrict**

34. **getCityById** - Получение данных города по идентификатору

Параметры запроса:

id - Идентификатор города *

запрос:

id = 01000000200000

ответ:

```
{
  id: '01000000200000',
  name: 'Адыгейск',
  zip: 385200,
  type: 'Город',
  typeShort: 'Г',
  okato: '79403000000',
  contentType: 'city',
  code: 'ru',
  lat: '44.8833775',
  lng: '39.1909625',
  regionId: '0100000000000',
  timezone: {
    timeZoneName: 'Москва, стандартное время',
    timeZoneId: 'Europe/Moscow',
    rawOffset: 10800,
    dstOffset: 0
  },
  gps: [39.1909625, 44.8833775]
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findCity**

35. **getStreetById** - Получение данных улицы по идентификатору

Параметры запроса:

id - Идентификатор улицы *

запрос:

id = 370000001000098000

ответ:

```
{
  id: '370000001000098000',
  name: 'Строителей',
  zip: 153038,
  type: 'Проспект',
  typeShort: 'пр-кт',
  okato: '24401000000',
  contentType: 'street',
  code: 'ru',
  lat: '56.9686335',
  lng: '41.0111737',
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '37000000000000',
  cityId: '37000000100000',
  metro: [],
  gps: [41.0111737, 56.9686335]
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findStreet**

36. *getHouseById* - Получение данных домов по идентификатору

Параметры запроса:

id - Идентификатор дома *

запрос:

id = 3701900100002790010

ответ:

```
{
  id: '3701900100002790010',
  name: '31',
  zip: 155900,
  type: 'дом',
  typeShort: 'д',
  okato: '24411000000',
  contentType: 'building',
  code: 'ru',
  lat: '56.8400475',
  lng: '41.3907855',
  timezone: {
    timeZoneName: 'Москва, стандартное время',
    timeZoneId: 'Europe/Moscow',
    rawOffset: 10800,
    dstOffset: 0
  },
  regionId: '37000000000000',
  districtId: '37019000000000',
  cityId: '37019001000000',
  streetId: '37019001000027900',
  metro: [],
  gps: [41.3907855, 56.8400475]
}
```

данный метод загружает данные только с локальной базы и если какие-то данные не загружены их нужно сначала загрузить с помощью метода **findHouse**

37. ***getMetroStationById*** - Получение данных станций метро по идентификатору

Параметры запроса:

id - Идентификатор станции метро *

запрос:

id = c3900aa9021cec3fce26bb41

ответ:

```
{
  id: 'c3900aa9021cec3fce26bb41',
  lat: '56.313933',
  lng: '43.99482',
  order: 0,
  name: 'Горьковская',
  cityId: '5200000100000',
  lineId: '874a3d2cdebc6d1bc5db1d49',
  gps: [43.99482, 56.313933]
}
```

38. ***getMetroLineById*** - Получение данных линий метро по идентификатору

Параметры запроса:

id - Идентификатор линий метро *

запрос:

id = 874a3d2cdebc6d1bc5db1d49

ответ:

```
{
  id: '874a3d2cdebc6d1bc5db1d49',
  name: 'Автозаводская',
  cityId: '52000000100000',
  color: 'D80707',
  stationsIds: [
    'c3900aa9021cec3fce26bb41',
    'f97005fe546a9d76ddb8542b',
    '63f6e64c94533b06cd4f12d6',
    'd28612dc4d850c65f3a0c9e0',
    '5c588cf81b88da827840b66a',
    '564c7fac5fffd3f314605dce',
    '966c4073e8d794620124a05c',
    'c02e8cd4d720f47f7c59df3a',
    '620fcd6362eb5cce33f43b25',
    'cb7718cf6f5f27ef139dac6a',
    'cd78298e188704bdbc1a405c'
  ]
}
```

39. ***getMetroCityByld*** - Получение данных городов метро по идентификатору

Параметры запроса:

id - Идентификатор города где есть метро *

запрос:

id = 77000000000000

ответ:

```
{
  id: '77000000000000',
  name: 'Москва',
  linesIds: [
    '13079b440bd88586a6265525',
    'f5acad4420711409231aaea8',
    '84aa74bc75c408fdd995998f',
    '31b83c4d182032e41ad85357',
    '3e274fd59654448f2a0cf31e',
    '88abe571337b0547723be907',
    '5e2b77b73e754efcb0d90833',
    '097c6f41dd3c5880cd3cc38b',
    '4f88b91807f6557ca67b1557',
    'f3270eac2a6d2e5acb29aa5d',
    'df5c912873338af59540ff52',
    'a91cedd8b9976cf5a0183390']
}
```


40. ***getTimezoneByGPS*** - Получение временной зоны по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

запрос:

lat = 55.755826

lng = 37.6173

ответ:

```
{
  dstOffset: 0,
  rawOffset: 10800,
  timeZoneId: 'Europe/Moscow',
  timeZoneName: 'Москва, стандартное время'
}
```

данный метод является системным и использовать его не рекомендуется, так как может нарушиться работоспособность всей системы на сутки и более

41. **getCountryByGPS** - Получение страны по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

запрос:

lat = 55.755826

lng = 37.6173

ответ:

```
{
  country: {
    id: '7',
    name: 'Россия',
    type: 'Страна',
    typeShort: 'ст-а',
    contentType: 'country',
    nameFull: 'Российская Федерация',
    nameShort: 'РФ',
    code: 'ru',
    lat: '61.52401',
    lng: '105.318756',
    gps: [105.318756, 61.52401]
  }
}
```

данный метод загружает данные как из локальной базы так и из интернета, если в локальной базе данных не найдено

42. *getRegionByGPS* - Получение региона по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

запрос:

lat = 55.755826

lng = 37.6173

ответ:

```
{
  country: {
    id: '7',
    name: 'Россия',
    type: 'Страна',
    typeShort: 'ст-а',
    contentType: 'country',
    nameFull: 'Российская Федерация',
    nameShort: 'РФ',
    code: 'ru',
    lat: '61.52401',
    lng: '105.318756',
    gps: [105.318756, 61.52401]
  },
  region: {
    name: 'Москва',
    zip: null,
    type: 'Город',
    typeShort: 'г',
    okato: '450000000000',
    contentType: 'region',
    id: '77000000000000',
    code: 'ru',
    lat: 55.755826,
    lng: 37.6173,
    gps: [37.6173, 55.755826],
    timezone: {
      dstOffset: 0,
      rawOffset: 10800,
      timeZoneId: 'Europe/Moscow',
      timeZoneName: 'Москва, стандартное время'
    }
  }
}
```

данный метод загружает данные как из локальной базы так и из интернета, если в локальной базе данных не найдено

43. *getDistrictByGPS* - Получение района по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

запрос:

lat = 56.7586237

lng = 41.3915336

ответ:

```
{
  country: {
    id: '7',
    name: 'Россия',
    type: 'Страна',
    typeShort: 'ст-а',
    contentType: 'country',
    nameFull: 'Российская Федерация',
    nameShort: 'РФ',
    code: 'ru',
    lat: '61.52401',
    lng: '105.318756',
    gps: [105.318756, 61.52401]
  },
  region: {
    name: 'Ивановская',
    zip: null,
    type: 'Область',
    typeShort: 'обл',
    okato: '240000000000',
    contentType: 'region',
    id: '3700000000000',
    code: 'ru',
    lat: 57.1056854,
    lng: 41.4830084,
    gps: [41.4830084, 57.1056854],
    timezone: {
      dstOffset: 0,
      rawOffset: 10800,
      timeZoneId: 'Europe/Moscow',
      timeZoneName: 'Москва, стандартное время'
    }
  },
  district: {
    name: 'Шуйский',
    zip: 155903,
    type: 'Район',
    typeShort: 'р-н',
    okato: '242330000000',
    contentType: 'district',
    id: '3701900000000',
```

```
code: 'ru',  
lat: 56.7586237,  
lng: 41.3915336,  
gps: [41.3915336, 56.7586237],  
timezone: {  
  dstOffset: 0,  
  rawOffset: 10800,  
  timeZoneId: 'Europe/Moscow',  
  timeZoneName: 'Москва, стандартное время'  
},  
regionId: '37000000000000'  
}
```

данный метод загружает данные как из локальной базы так и из интернета, если в локальной базе данных не найдено

44. *getCityByGPS* - Получение города по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

запрос:

lat = 56.8441

lng = 41.3796

ответ:

```
{
  country: {
    id: '7',
    name: 'Россия',
    type: 'Страна',
    typeShort: 'ст-а',
    contentType: 'country',
    nameFull: 'Российская Федерация',
    nameShort: 'РФ',
    code: 'ru',
    lat: '61.52401',
    lng: '105.318756',
    gps: [105.318756, 61.52401]
  },
  region: {
    name: 'Ивановская',
    zip: null,
    type: 'Область',
    typeShort: 'обл',
    okato: '240000000000',
    contentType: 'region',
    id: '3700000000000',
    code: 'ru',
    lat: 57.1056854,
    lng: 41.4830084,
    gps: [41.4830084, 57.1056854],
    timezone: {
      dstOffset: 0,
      rawOffset: 10800,
      timeZoneId: 'Europe/Moscow',
      timeZoneName: 'Москва, стандартное время'
    }
  },
  city: {
    name: 'Шуя',
    zip: 155905,
    type: 'Город',
    typeShort: 'г',
    okato: '24411000000',
    contentType: 'city',
    id: '3701900100000',
```

```
code: 'ru',  
lat: '56.8441',  
lng: '41.3796',  
gps: [41.3796, 56.8441],  
regionId: '3700000000000',  
districtId: '3701900000000'  
}  
}
```

данный метод загружает данные как из локальной базы так и из интернета, если в локальной базе данных не найдено

45. *getStreetByGPS* - Получение улицы по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

запрос:

lat = 56.8515719

lng = 41.370719

ответ:

```
{
  country: {
    id: '7',
    name: 'Россия',
    type: 'Страна',
    typeShort: 'ст-а',
    contentType: 'country',
    nameFull: 'Российская Федерация',
    nameShort: 'РФ',
    code: 'ru',
    lat: '61.52401',
    lng: '105.318756',
    gps: [105.318756, 61.52401]
  },
  region: {
    name: 'Ивановская',
    zip: null,
    type: 'Область',
    typeShort: 'обл',
    okato: '240000000000',
    contentType: 'region',
    id: '3700000000000',
    code: 'ru',
    lat: 57.1056854,
    lng: 41.4830084,
    gps: [41.4830084, 57.1056854],
    timezone: {
      dstOffset: 0,
      rawOffset: 10800,
      timeZoneId: 'Europe/Moscow',
      timeZoneName: 'Москва, стандартное время'
    }
  },
  city: {
    name: 'Шуя',
    zip: 155905,
    type: 'Город',
    typeShort: 'г',
    okato: '24411000000',
    contentType: 'city',
    id: '3701900100000',
```



```

        code: 'ru',
        lat: '56.8441',
        lng: '41.3796',
        gps: [41.3796, 56.8441],
        timezone: {
            dstOffset: 0,
            rawOffset: 10800,
            timeZoneId: 'Europe/Moscow',
            timeZoneName: 'Москва, стандартное время'
        },
        regionId: '3700000000000',
        districtId: '3701900000000'
    },
    street: {
        name: 'Комсомольская',
        zip: 155900,
        type: 'Площадь',
        typeShort: 'пл',
        okato: '24411000000',
        contentType: 'street',
        id: '37019001000010900',
        code: 'ru',
        lat: '56.8515719',
        lng: '41.370719',
        gps: [41.370719, 56.8515719],
        timezone: {
            dstOffset: 0,
            rawOffset: 10800,
            timeZoneId: 'Europe/Moscow',
            timeZoneName: 'Москва, стандартное время'
        },
        regionId: '3700000000000',
        districtId: '3701900000000',
        cityId: '3701900100000'
    }
}

```

данный метод загружает данные как из локальной базы так и из интернета, если в локальной базе данных не найдено

46. *getStreetsByGPS* - Получение улиц по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

distance - Расстояние в метрах (по умолчанию 3000)

запрос:

lat = 56.8515719

lng = 41.370719

ответ:

```
[{
  id: '37019001000010900',
  name: 'Комсомольская',
  zip: 155900,
  type: 'Площадь',
  typeShort: 'пл',
  okato: '24411000000',
  contentType: 'street',
  code: 'ru',
  lat: '56.8515719',
  lng: '41.370719',
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '3700000000000',
  districtId: '3701900000000',
  cityId: '3701900100000',
  metro: [],
  gps: [41.370719, 56.8515719]
}, {
  id: '37019001000027900',
  name: 'Свердлова',
  zip: 155908,
  type: 'Улица',
  typeShort: 'ул',
  okato: '24411000000',
  contentType: 'street',
  code: 'ru',
  lat: '56.8400475',
  lng: '41.3907855',
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
  regionId: '3700000000000',
```

```
districtId: '3701900000000',  
cityId: '3701900100000',  
metro: [],  
gps: [41.3907855, 56.8400475]  
}]
```

данный метод загружает данные только из локальной базы

47. *getHousesByGPS* - Получение домов по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

distance - Расстояние в метрах (по умолчанию 3000)

запрос:

lat = 56.8515719

lng = 41.370719

ответ:

```
[{
  id: '3701900100002790010',
  name: '31',
  zip: 155900,
  type: 'дом',
  typeShort: 'д',
  okato: '24411000000',
  contentType: 'building',
  code: 'ru',
  lat: '56.8400475',
  lng: '41.3907855',
  timezone: {
    timeZoneName: 'Москва, стандартное время',
    timeZoneId: 'Europe/Moscow',
    rawOffset: 10800,
    dstOffset: 0
  },
  regionId: '37000000000000',
  districtId: '37019000000000',
  cityId: '37019001000000',
  streetId: '37019001000027900',
  metro: [],
  gps: [41.3907855, 56.8400475]
}, {
  id: '3701900100002790005',
  name: '38',
  zip: 155912,
  type: 'дом',
  typeShort: 'д',
  okato: '24411000000',
  contentType: 'building',
  code: 'ru',
  lat: '56.8400475',
  lng: '41.3907855',
  timezone: {
    dstOffset: 0,
    rawOffset: 10800,
    timeZoneId: 'Europe/Moscow',
    timeZoneName: 'Москва, стандартное время'
  },
}
```

```

    regionId: '3700000000000',
    districtId: '3701900000000',
    cityId: '3701900100000',
    streetId: '37019001000027900',
    metro: [],
    gps: [41.3907855, 56.8400475]
  }, {
    id: '3701900100002790011',
    name: '39',
    zip: 155900,
    type: 'дом',
    typeShort: 'д',
    okato: '24411000000',
    contentType: 'building',
    code: 'ru',
    lat: '56.8400475',
    lng: '41.3907855',
    timezone: {
      timeZoneName: 'Москва, стандартное время',
      timeZoneId: 'Europe/Moscow',
      rawOffset: 10800,
      dstOffset: 0
    },
    regionId: '3700000000000',
    districtId: '3701900000000',
    cityId: '3701900100000',
    streetId: '37019001000027900',
    metro: [],
    gps: [41.3907855, 56.8400475]
  }, {
    id: '3701900100002790002',
    name: '36',
    zip: 155908,
    type: 'дом',
    typeShort: 'д',
    okato: '24411000000',
    contentType: 'building',
    code: 'ru',
    lat: '56.8400475',
    lng: '41.3907855',
    timezone: {
      timeZoneName: 'Москва, стандартное время',
      timeZoneId: 'Europe/Moscow',
      rawOffset: 10800,
      dstOffset: 0
    },
    regionId: '3700000000000',
    districtId: '3701900000000',
    cityId: '3701900100000',
    streetId: '37019001000027900',
    metro: [],
    gps: [41.3907855, 56.8400475]
  }
}

```

48. **getMetroByGPS** - Получение станций метро по GPS координатам

Параметры запроса:

lat - Широта *

lng - Долгота *

distance - Расстояние в метрах (по умолчанию 3000)

запрос:

lat = 55.633

lng = 37.4419

ответ:

```
[{
  id: '975c23bb4d69e2efc343cf11',
  name: 'Румянцево',
  lat: '55.633',
  lng: '37.4419',
  order: 20,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '7700000000000',
  gps: [37.4419, 55.633]
}, {
  id: '07fad26e60a6a0378d211c87',
  name: 'Саларьево',
  lat: '55.6227',
  lng: '37.424',
  order: 21,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '7700000000000',
  gps: [37.424, 55.6227]
}, {
  id: 'c4fca5a3388d09a6d04cdde2',
  name: 'Тропарево',
  lat: '55.6459',
  lng: '37.4725',
  order: 19,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '7700000000000',
  gps: [37.4725, 55.6459]
}]
```

49. ***updateTimeZones*** - Обновление временных зон у субъектов где они не установлены

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод является системным и использовать его не рекомендуется, так как может нарушится работоспособность всей системы на сутки и более

50. **updateCountries** - Обновление всей базы данных стран

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод перед обновлением, удаляет из базы все данные связанные со странами, его работа может занять длительное время

51. ***updateRegions*** - Обновление всей базы данных регионов

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод перед обновлением, удаляет из базы все данные связанные с регионами, его работа может занять длительное время

52. **updateDistricts** - Обновление всей базы данных районов

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод перед обновлением, удаляет из базы все данные связанные с районами, его работа может занять длительное время

53. **updateCities** - Обновление всей базы данных городов

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод перед обновлением, удаляет из базы все данные связанные с городами, его работа может занять несколько суток

54. **updateStreets** - Обновление всей базы данных улиц

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод перед обновлением, удаляет из базы все данные связанные с улицами, его работа может занять несколько суток

55. **updateHouses** - Обновление всей базы данных домов

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод перед обновлением, удаляет из базы все данные связанные с домами, его работа может занять несколько суток

56. **updateMetro** - Обновление всей базы данных метро

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод перед обновлением, удаляет из базы все данные связанные с метро, его работа может занять длительное время

57. ***initEmptyDatabases*** - Инициализация всей структуры базы данных на чистом сервере

Параметры запроса:

updateKey - Ключ администратора (устанавливается в конфигурационном файле) *

запрос:

updateKey = passkey

ответ:

true или false

данный метод перед обновлением, удаляет из базы абсолютно все данные, его работа может занять несколько суток

Агенты сервера

1. Агент WebSocket

пример подключения:

```
var WebSocket = require('ws'); // Только для Node.js в браузере эта строчка не нужна

var socket = new WebSocket("ws://mysite.com");

socket.onopen = function(){
    console.log("Соединение установлено.");
};

socket.onclose = function(event){
    if(event.wasClean) console.log("Соединение закрыто чисто");
    // например, "убит" процесс сервера
    else console.log("Обрыв соединения");
    console.log('Код:', event.code, 'причина:', event.reason);
};

socket.onmessage = function(event){
    console.log("Получены данные", JSON.parse(event.data).query);
};

socket.onerror = function(error){
    console.log("Ошибка", error.message);
};

// Выполняем запрос к серверу на получении станции метро Румянцево

socket.send(JSON.stringify({"action": "findMetro", "query": {"str": "Рум", "cityId": "7700000000000",
"lineName": "Сокольническая"}}));

> Получены данные [ { id: '975c23bb4d69e2efc343cf11',
  name: 'Румянцево',
  lat: '55.633',
  lng: '37.4419',
  order: 20,
  line: 'Сокольническая',
  color: 'E42313',
  city: 'Москва',
  lineId: '31b83c4d182032e41ad85357',
  cityId: '7700000000000',
  gps: [ 37.4419, 55.633 ] } ]
```