

---

# RIA.com Documentation

*Выпуск 2.0*

RIA.com Documentation

нояб. 01, 2018



<b>1</b>	<b>AUTO.RIA</b>	<b>1</b>
1.1	AUTO.RIA б.у авто . . . . .	2
1.2	AUTO.RIA новые авто . . . . .	41
<b>2</b>	<b>DOM.RIA</b>	<b>53</b>
2.1	Параметры для работы с DOM.RIA API . . . . .	54
2.2	Список характеристик недвижимости . . . . .	54
2.3	Информация по id объявления . . . . .	72
2.4	<b>Подсчёт средней цены</b> . . . . .	76
2.5	Поиск объявлений . . . . .	82
2.6	Методы для работы с областями, городами, районами . . . . .	84
<b>3</b>	<b>RIA.com</b>	<b>99</b>
3.1	Перечень статусов заказов . . . . .	99
3.2	Перечень заказов . . . . .	100
3.3	Изменение статуса заказа . . . . .	101
<b>4</b>	<b>Общие Ошибки Веб-Службы</b>	<b>103</b>
<b>5</b>	<b>Использование API Key</b>	<b>105</b>
<b>6</b>	<b>Лимиты Веб-Сервисов</b>	<b>107</b>



### Лимиты Веб-Сервисов

Ограничения распространяются на количество запросов к веб-службам RIA.com, которые Вы можете создать, используя свой API ключ.

Часовой лимит: 1,000 запросов в час Эти ограничения применяются ко всем веб-службам RIA.com. Превышение этих лимитов приведет к временному блокированию Вашего API ключа. В зависимости от превышенного предела, блок будет автоматически отменен в течении следующего часа или календарного дня.

#### Часовой Лимит

Почасовые счетчики для API ключа аннулируются на постоянной основе.

Пример: Пример: Если Вы сделаете 500 запросов в 10:15 и 500 запросов в 10:25, Ваш API ключ будет временно заблокирован. Блок будет аннулирован в 11:15, после чего Вы сможете сделать еще 500 запросов.

#### Ограничение Запросов

Если Ваш API ключ превышает лимиты запросов, Вы получите ответ с HTTP статус-кодом 429 (слишком много запросов).

#### Нужно повысить лимиты?

Если Вы создаете приложение, которое требует более высоких лимитов, мы будем рады сотрудничать. Свяжитесь с нами для более подробной информации.

## 1.1 AUTO.RIA б.у авто

### 1.1.1 Методы для работы с Параметрами

#### Типы транспорта

Что бы получить список типов транспорта нужно отправив GET запрос на адрес [https://developers.ria.com/auto/categories/?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/?api_key=YOUR_API_KEY). Результат будет примерно следующим:

```
[
  { name: "Легковые", value: 1 },
  { name: "Мото", value: 2 },
  { name: "Водный транспорт", value: 3 },
  { name: "Спецтехника", value: 4 },
  { name: "Прицеп", value: 5 },
  { name: "Грузовик", value: 6 },
  { name: "Автобус", value: 7 },
  { name: "Автодом", value: 8 },
  { name: "Воздушный транспорт", value: 9 }
]
```

#### Типы кузова

Типы кузова зависят от типов транспорта. Поэтому для того, чтобы получить список типов кузова необходимо отправить GET запрос на адрес [https://developers.ria.com/auto/categories/:categoryId/bodystyles?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/:categoryId/bodystyles?api_key=YOUR_API_KEY), где *categoryId* - идентификатор типа транспорта, *api\_key*- Ваш ключ.

Например, для легковых автомобилей ([https://developers.ria.com/auto/categories/1/bodystyles?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/1/bodystyles?api_key=YOUR_API_KEY)) результат будет следующим:

```
[
  { name: "Седан", value: 3 },
  { name: "Внедорожник / Кроссовер", value: 5 },
  { name: "Минивэн", value: 8 },
  { name: "Хэтчбек", value: 4 },
  { name: "Универсал", value: 2 },
  { name: "Купе", value: 6 },
  { name: "Легковой фургон (до 1,5 т)", value: 254 },
  { name: "Кабриолет", value: 7 },
  { name: "Пикап", value: 9 },
  { name: "Лимузин", value: 252 },
  { name: "Другой", value: 28 }
]
```

Также типы кузова могут быть разделены на группы. Это актуально для спецтехники. Поэтому существует способ получить сгруппированные типы кузова отправив GET запрос по адресу [https://developers.ria.com/auto/categories/:categoryId/bodystyles/\\_group?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/:categoryId/bodystyles/_group?api_key=YOUR_API_KEY), где *categoryId* - идентификатор типа транспорта, *api\_key*- Ваш ключ.

Например, для мотоциклов ([https://developers.ria.com/auto/categories/2/bodystyles/\\_group?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/2/bodystyles/_group?api_key=YOUR_API_KEY)) результат будет следующим:

```
[
  [
    { name: "Мопеды", value: 58 },
    { name: "Скутер / Мотороллер", value: 11 },
    { name: "Макси-скутер", value: 12 }
  ],
  [
    { name: "Мотоциклы", value: 13 },
    { name: "Мотоцикл Без обтекателей (Naked bike)", value: 15 },
    { name: "Мотоцикл Внедорожный (Enduro)", value: 21 },
    { name: "Мотоцикл Кастом", value: 30 },
    { name: "Мотоцикл Классик", value: 14 },
    { name: "Мотоцикл Кросс", value: 19 },
    { name: "Мотоцикл Круизер", value: 24 },
    { name: "Мотоцикл Многоцелевой (All-round)", value: 25 },
    { name: "Мотоцикл с коляской", value: 29 },
    { name: "Спортбайк", value: 18 },
    { name: "Мотоцикл Спорт-туризм", value: 17 },
    { name: "Мотоцикл Супермото (Motard)", value: 22 },
    { name: "Мотоцикл Триал", value: 20 },
    { name: "Мотоцикл Туризм", value: 16 },
    { name: "Мотоцикл Чоппер", value: 23 }
  ],
  [
    { name: "Мини мотоциклы", value: 31 },
    { name: "Мини спорт", value: 32 },
    { name: "Мини крос (Питбайк)", value: 33 }
  ],
  { name: "Трицикл", value: 34 },
  { name: "Трайк", value: 57 },
  [
    { name: "Квадроциклы", value: 35 },
    { name: "Квадроцикл детский", value: 36 },
    { name: "Квадроцикл спортивный", value: 39 },
    { name: "Квадроцикл утилитарный", value: 41 },
    { name: "Мотовездеход", value: 42 },
    { name: "Вездеход-амфибия", value: 43 },
    { name: "Гольф-кар", value: 44 },
    { name: "Картинг", value: 45 }
  ],
  { name: "Снегоход", value: 46 },
  { name: "Другое", value: 56 }
]
```

Формат данных при этом отличается от обычного - это коллекция объектов, в которой есть другие коллекции. Группа типов кузовов всегда начинается с её названия. Например, в группу *Квадроциклы* входят типы кузовов *Квадроцикл детский*, *Квадроцикл спортивный*, *Квадроцикл утилитарный*, *Мотовездеход* и т.д.

Также, при необходимости, можно получить просто весь список типов кузовов, послав GET запрос по адресу [https://developers.ria.com/auto/bodystyles?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/bodystyles?api_key=YOUR_API_KEY).

## Марки

Марки зависят от типов транспорта. Поэтому для того, чтобы получить список марок необходимо отправить GET запрос по адресу [https://developers.ria.com/auto/categories/:categoryId/marks?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/:categoryId/marks?api_key=YOUR_API_KEY), где *categoryId* - идентификатор типа транспорта, *api\_key* - Ваш ключ.

Например, для легковых автомобилей ([https://developers.ria.com/auto/categories/1/marks?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/1/marks?api_key=YOUR_API_KEY)) результат будет следующим:

```
[
  { name: "Acura", value: 98 },
  { name: "Adler", value: 2396 },
  { name: "Aixam", value: 2 },
  { name: "Alfa Romeo", value: 3 },
  { name: "Alpine", value: 100 },
  { name: "Altamarea", value: 3988 },
  { name: "Aro", value: 101 },
  { name: "Artega", value: 3105 },
  { name: "Asia", value: 4 },
  { name: "Aston Martin", value: 5 },
  { name: "Audi", value: 6 },
  { name: "Austin", value: 7 },
  { name: "Autobianchi", value: 102 }
  ...
]
```

## Модели

Модели зависят от типов транспорта и марок. Следовательно список марок можно получить по адресу [http://api.auto.ria.com/categories/:categoryId/marks/:markId/models?api\\_key=YOUR\\_API\\_KEY](http://api.auto.ria.com/categories/:categoryId/marks/:markId/models?api_key=YOUR_API_KEY), где *categoryId* - идентификатор типа транспорта а *markId* - идентификатор марки, *api\_key*- Ваш ключ.

Например, для мотоциклов BMW ([https://developers.ria.com/auto/categories/2/marks/9/models?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/2/marks/9/models?api_key=YOUR_API_KEY)) список моделей будет следующим:

```
[
  { name: "Adventure", value: 25290 },
  { name: "C", value: 25291 },
  { name: "CS", value: 25292 },
  { name: "DKW", value: 28318 },
  { name: "F", value: 25293 },
  { name: "G", value: 29468 },
  { name: "GS", value: 25295 },
  { name: "HP", value: 38148 },
  { name: "Independent", value: 25297 },
  { name: "K", value: 25298 },
  { name: "LT", value: 25299 },
  { name: "R", value: 25300 },
  { name: "RS", value: 32736 },
  { name: "RT", value: 25301 },
  { name: "S", value: 25302 },
  { name: "X", value: 42030 }
]
```

Модели, также как и типы кузовов, могут быть сгруппированы. Чтобы получить такой список, необходимо отправить запрос по адресу [https://developers.ria.com/auto/categories/:categoryId/marks/:markId/models/\\_group?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/:categoryId/marks/:markId/models/_group?api_key=YOUR_API_KEY), где *categoryId* - идентификатор типа транспорта а *markId* - идентификатор марки, *api\_key*- Ваш ключ.

Например, для легковых автомобилей BMW ([https://developers.ria.com/auto/categories/1/marks/9/models/\\_group?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/1/marks/9/models/_group?api_key=YOUR_API_KEY)) список моделей будет следующим:



```
[
  [
    { name: "1 Series (все)", value: 2161 },
    { name: "116", value: 34670 },
    { name: "118", value: 34671 },
    { name: "120", value: 34672 },
    { name: "123", value: 34673 },
    { name: "125", value: 34674 },
    { name: "130", value: 34675 },
    { name: "135", value: 34676 }
  ],
  [
    { name: "3 Series (все)", value: 3219 },
    { name: "3 Series GT", value: 43029 },
    { name: "315", value: 37454 },
    { name: "316", value: 30851 },
    { name: "318", value: 31612 },
    { name: "320", value: 31611 },
    { name: "321", value: 37389 },
    { name: "323", value: 34677 },
    { name: "324", value: 30687 },
    { name: "325", value: 29713 },
    { name: "326", value: 44061 },
    { name: "328", value: 31661 },
    { name: "330", value: 34678 },
    { name: "335", value: 34679 },
    { name: "340", value: 35568 }
  ],
  ...
  { name: "Alpina", value: 906 },
  { name: "Dixi", value: 33383 },
  { name: "I3", value: 44838 },
  { name: "I8", value: 44537 },
  { name: "Isetta", value: 32380 },
  { name: "Z1", value: 97 },
  { name: "Z3", value: 98 },
  { name: "Z4", value: 99 },
  { name: "Z8", value: 100 }
]
```

Формат данных такой же как и в случае с типа кузова - коллекция объектов, в которой могут быть другие объекты. Группа моделей всегда начинается с её названия.

Также, при необходимости, можно просто получить список всех моделей, отправив GET запрос по адресу [https://developers.ria.com/auto/models?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/models?api_key=YOUR_API_KEY).

## Области

Получить список областей можно отправив GET запрос по адресу [https://developers.ria.com/auto/states?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/states?api_key=YOUR_API_KEY).

Результат будет следующим:

```
[
  { name: "Винницкая", value: 1 },
  { name: "Волынская", value: 18 },
  { name: "Днепропетровская", value: 11 },
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{ name: "Донецкая", value: 13 },
{ name: "Житомирская", value: 2 },
{ name: "Закарпатская", value: 22 },
{ name: "Запорожская", value: 14 },
{ name: "Ивано-Франковская", value: 15 },
{ name: "Киевская", value: 10 },
{ name: "Кировоградская", value: 16 },
{ name: "Луганская", value: 17 },
{ name: "Львовская", value: 5 },
{ name: "Николаевская", value: 19 },
{ name: "Одесская", value: 12 },
{ name: "Полтавская", value: 20 },
{ name: "Республика Крым", value: 21 },
{ name: "Ровенская", value: 9 },
{ name: "Сумская", value: 8 },
{ name: "Тернопольская", value: 3 },
{ name: "Харьковская", value: 7 },
{ name: "Херсонская", value: 23 },
{ name: "Хмельницкая", value: 4 },
{ name: "Черкасская", value: 24 },
{ name: "Черниговская", value: 6 },
{ name: "Черновицкая", value: 25 }
]
```

## Города

Города зависят от областей, поэтому, чтобы получить их список, необходимо послать GET запрос по адресу [https://developers.ria.com/auto/states/:stateId/cities?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/states/:stateId/cities?api_key=YOUR_API_KEY), где *stateId* - идентификатор области, *api\_key* - Ваш ключ.

Например, для Винницкой области ([https://developers.ria.com/auto/states/1/cities?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/states/1/cities?api_key=YOUR_API_KEY)) список городов будет следующим:

```
[
  { name: "Винница", value: 1 },
  { name: "Жмеринка", value: 27 },
  { name: "Казатин", value: 30 },
  { name: "Крыжополь", value: 31 },
  { name: "Липовец", value: 32 },
  { name: "Литин", value: 33 },
  { name: "Могилев-Подольский", value: 34 },
  { name: "Мурованые Куриловцы", value: 35 },
  { name: "Немиров", value: 36 },
  { name: "Оратов", value: 37 },
  { name: "Песчанка", value: 38 },
  { name: "Погребище", value: 39 },
  { name: "Теплик", value: 40 },
  { name: "Тывров", value: 41 },
  { name: "Томашполь", value: 42 },
  { name: "Тростянец", value: 43 },
  { name: "Тульчин", value: 44 },
  { name: "Хмельник", value: 45 },
  { name: "Черновцы", value: 46 },
  { name: "Чечельник", value: 47 },
  { name: "Шаргород", value: 48 },
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[
  { name: "Ямполь", value: 49 },
  { name: "Бар", value: 597 },
  { name: "Бершадь", value: 599 },
  { name: "Гайсин", value: 602 },
  { name: "Ильинцы", value: 603 },
  { name: "Калиновка", value: 604 },
  { name: "Гнивань", value: 609 },
  { name: "Ладыжин", value: 644 }
]
```

## Типы привода

Типы привода также зависят от типа транспорта, поэтому, чтобы получить их список, необходимо послать GET запрос по адресу [https://developers.ria.com/auto/categories/:categoryId/driverTypes?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/:categoryId/driverTypes?api_key=YOUR_API_KEY), где *categoryId* - идентификатор типа транспорта, *api\_key* - Ваш ключ.

Например, список типов привода для мотоциклов ([https://developers.ria.com/auto/categories/2/driverTypes?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/2/driverTypes?api_key=YOUR_API_KEY)) выглядит следующим образом:

```
[
  { name: "Кардан", value: 4 },
  { name: "Ремень", value: 5 },
  { name: "Цепь", value: 6 }
]
```

## Типы топлива

Типы топлива можно получить отправив GET запрос по адресу [https://developers.ria.com/auto/type?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/type?api_key=YOUR_API_KEY). Ответ будет выглядеть так:

```
[
  { name: "Бензин", value: 1 },
  { name: "Дизель", value: 2 },
  { name: "Газ", value: 3 },
  { name: "Газ/бензин", value: 4 },
  { name: "Гибрид", value: 5 },
  { name: "Электро", value: 6 },
  { name: "Другое", value: 7 },
  { name: "Газ метан", value: 8 },
  { name: "Газ пропан-бутан", value: 9 }
]
```

## Коробки передач

Коробки передач зависят от типа транспорта, поэтому, чтобы получить их список, необходимо послать GET запрос по адресу [https://developers.ria.com/auto/categories/:categoryId/gearboxes?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/:categoryId/gearboxes?api_key=YOUR_API_KEY), где *categoryId* - идентификатор типа транспорта, *api\_key* - Ваш ключ.

Например, список коробок передач для мотоциклов ([https://developers.ria.com/auto/categories/2/gearboxes?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/2/gearboxes?api_key=YOUR_API_KEY)) будет выглядеть следующим образом:

```
[
  { name: "Ручная / Механика", value: 1 },
  { name: "Автомат", value: 2 },
  { name: "Типтроник", value: 3 },
  { name: "Адаптивная", value: 4 },
  { name: "Вариатор", value: 5 }
]
```

## Опции

Опции зависят от типа транспорта. Получить их список можно отправив GET запрос по адресу [https://developers.ria.com/auto/categories/:categoryId/options?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/:categoryId/options?api_key=YOUR_API_KEY), где *categoryId* - идентификатор типа транспорта, *api\_key*- Ваш ключ.

Например, список опций для легковых автомобилей ([https://developers.ria.com/auto/categories/1/options?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/1/options?api_key=YOUR_API_KEY)) будет выглядеть примерно так:

```
[
  { name: "ABD", value: 354 },
  { name: "ABS", value: 217 },
  { name: "ESP", value: 459 },
  { name: "Галогенные фары", value: 463 },
  { name: "Замок на КПП", value: 481 },
  { name: "Иммобилайзер", value: 225 },
  { name: "Пневмоподвеска", value: 442 },
  { name: "Подушка безопасности (Airbag)", value: 211 },
  { name: "Серворуль", value: 485 },
  { name: "Сигнализация", value: 303 },
  ...
]
```

## Цвета

Получить список всех цветов можно, если отправить GET запрос по адресу [https://developers.ria.com/auto/colors?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/colors?api_key=YOUR_API_KEY). Результат будет следующим:

```
[
  { name: "Бежевый", value: 1 },
  { name: "Черный", value: 2 },
  { name: "Синий", value: 3 },
  { name: "Бронзовый", value: 4 },
  { name: "Коричневый", value: 5 },
  { name: "Золотой", value: 6 },
  { name: "Зеленый", value: 7 },
  { name: "Серый", value: 8 },
  { name: "Апельсин", value: 9 },
  { name: "Магнолии", value: 10 },
  { name: "Розовый", value: 11 },
  { name: "Фиолетовый", value: 12 },
  { name: "Красный", value: 13 },
  { name: "Серебряный", value: 14 },
  { name: "Белый", value: 15 },
  { name: "Желтый", value: 16 },
  { name: "Голубой", value: 17 },
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{ name: "Вишнёвый", value: 18 },
{ name: "Сафари", value: 19 },
{ name: "Гранатовый", value: 20 },
{ name: "Асфальт", value: 21 }
]
```

### Страна производитель

Получить список стран производителей можно, если отправить GET запрос по адресу [https://developers.ria.com/auto/countries?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/countries?api_key=YOUR_API_KEY). Результат будет следующим:

```
[
  {
    "name": "Австрия", "value": 40 },
  {
    "name": "Англия", "value": 826 },
  {
    "name": "Аргентина", "value": 32 },
  {
    "name": "Беларусь", "value": 112 },
  {
    "name": "Бельгия", "value": 56 },
  {
    "name": "Болгария", "value": 100 },
  {
    "name": "Бразилия", "value": 76 },
  {
    "name": "Венгрия", "value": 348 },
  {
    "name": "Германия", "value": 276},
  {
    "name": "Дания", "value": 208 }
    .....
]
```

### Остальные параметры

#### Растаможка

Параметр растаможки может принимать только два значения: 1 - нерастаможенный и 0 - растаможенный.

#### После ДТП

Данный параметр принимает следующие значения: 1 - объявления после ДТП 0 - остальные объявления

### **Взято в кредит**

Данный параметр может принимать следующие значения: 1 - объявления взятые в кредит 0 - остальные объявления

### **Конфискат**

Данный параметр может принимать следующие значения: 1 - только конфискованные объявления 0 - только неконфискованные объявления

### **Не на ходу**

Данный параметр может принимать следующие значения: 1 - только объявления не на ходу 0 - только объявления на ходу

### **Цена**

Данный параметр может принимать значения которые Вы зададите:

price\_ot= - цена от

price\_do= - цена до

### **Валюта**

Данный параметр может принимать следующие значения:

1 - доллары США

2 - евро

3 - гривна

### **Торг**

Данный параметр может принимать следующие значения:

0 - не включать в запрос поиска

1 - включить в запрос поиска

### **Обмен на недвижимость**

Данный параметр может принимать следующие значения:

0 - не включать в запрос поиска

1 - включить в запрос поиска

### Обмен на автомобиль

Данный параметр может принимать следующие значения:

0 - не включать в запрос поиска

1 - включить в запрос поиска

Фильтр обмена

`exchange_filter[marka_id]=0` - Марка автомобиля любая

`exchange_filter[model_id]=0` - Модель автомобиля любая

### Авто не в Украине

Данный параметр может принимать следующие значения:

0 - по умолчанию показать все

1 - включить в запрос поиска

2 - не включать в запрос поиска

### Не на ходу

Данный параметр может принимать следующие значения:

1 - отобразить

2 - не отображать

### Продавец

Данный параметр может принимать следующие значения:

0 - по умолчанию все

1 - частное лицо

2 - компания

### Пригнан из

Для данного параметра можно использовать элементы полученный с помощью сервиса **Страна производитель**

`matched_country=` - Пригнан из

### Объем двигателя

Данный параметр может принимать значения которые Вы зададите:

`engineVolumeFrom=` - объем от

`engineVolumeTo=` - объем до

### Расход топлива

Данный параметр может принимать значения которые Вы зададите:

fuelRateFrom= - расход от

fuelRateTo= - расход до

Цикл

fuelRatesType может быть равен трем значениям

city - городской цикл

combine - шоссе

route - смешанный

### Мощность двигателя

Данный параметр может принимать значения которые Вы зададите:

powerFrom= - мощность от

powerTo= - мощность до

Единица измерения мощности

1 - лошадиные силы

2 - кВт

### Сортировка

Данный параметр может принимать следующие значения:

0 - по умолчанию обычная сортировка

1 - от дешевых к дорогим

2 - от дорогих к дешевым

### Период подачи

Данный параметр может принимать следующие значения:

0 - по умолчанию за все время

1 - за час

2 - за сегодня

3 - за три дня

4 - за неделю

5 - за месяц

6 - за 3 месяца

8 - за 3 часа

9 - за 6 часов



- 10 - за 2 дня
- 11 - за сутки
- 14 - за 12 часов

### **Актуальность**

Данный параметр может принимать следующие значения:

- 0 - все автомобили
- 1 - показывать только проданные авто
- 2 - не отображать проданные автомобили

### **VIN код**

Данный параметр может принимать следующие значения:

- 0 - не включать в запрос поиска
- 1 - включить в запрос поиска

### **Колесная формула**

Данный параметр может принимать следующие значения:

- 0 - по умолчанию
- 1 - 4x2
- 2 - 4x4
- 3 - 6x2
- 4 - 6x4
- 5 - 6x6
- 7 - 8x2
- 8 - 8x4
- 9 - 8x6
- 10 - 8x8
- 11 - 10x4
- 12 - 10x8

### **Количество осей**

Данный параметр может принимать следующие значения:

- 0 - по умолчанию
- 1 - одна ось
- 2 - две оси

- 3 - три оси
- 4 - четыре оси
- 5 - больше четырех осей

### Только с Фото

Данный параметр может принимать следующие значения:

- 0 - не включать в запрос поиска
- 1 - включить в запрос поиска

### Только с Видео

Данный параметр может принимать следующие значения:

- 0 - не включать в запрос поиска
- 1 - включить в запрос поиска

### Количество отображаемых id

Данный параметр может принимать следующие значения от 1 до '100

aa

## 1.1.2 API Поиска

Чтобы обратиться к RIA API search, Вам необходимо выполнить GET запрос такого вида:

`https://developers.ria.com/auto/search?api_key=YOUR_API_KEY&PARAMETERS`

или `curl -i -g -X GET "https://developers.ria.com/auto/search?api_key=YOUR_API_KEY&PARAMETERS"`

Он состоит из:

- **SEARCH** — название метода API, к которому Вы хотите обратиться.
- **API\_KEY** — ключ доступа. Для получения ключа доступа нужно зарегистрироваться на портале dev.ria.com
- **PARAMETERS** — входные параметры, последовательность пар name=value, разделенных амперсандом. Список параметров указан *здесь*

В ответ на такой запрос Вы получите ответ в формате JSON:

```
[
{
  "additional_params": {
    "lang_id": 2,           // Русский язык
    "page": 0,             // Порядковый номер страницы
    "view_type_id": 0,
    "target": "search",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"section": "auto", // Поиск по авто
"catalog_name": "",
"elastica": true,
"nodejs": true
},
"result": { // Результат поиска
  "search_result": {
    "ids": [ // id объявлений
      ....
      ....
      ....
    ],
    "count": 11, // Количество id объявлений доступных
    ↪ по заданным параметрам
    "last_id": 0,
    "qs": {
      "fields": [
        "_id"
      ],
      "size": 50, // Количество отображаемых id
    ↪ объявлений
    "from": 0,
  ]
}

```

## Пример

Допустим Вы ищете:

- Легковые машины (category\_id=1)
- Кузов *Седан, Универсал* (bodystyle[0]=3&bodystyle[4]=2)
- Япония (brandOrigin[0]=276)
  - Toyota (marka\_id[0]=79)
  - Все модели (model\_id[0]=0)
  - Год выпуска от 2010 по 2017г. (s\_yers[0]=2010&po\_yers[0]=2017)
- Германия (brandOrigin[1]=392)
  - Volkswagen (marka\_id[1]=84)
  - Все модели (model\_id[1]=0)
  - Год выпуска от 2012 по 2016г. (s\_yers[1]=2012&po\_yers[1]=2016)
- Цена от 1000 до 60000 (price\_ot=1000&price\_do=60000)
  - Цена указана в доларах США (currency=1)
- Возможен торг (auctionPossible=1)
- Возможен обмен на недвижимость (with\_real\_exchange=1)
- Возможен обмен на автомобиль (with\_exchange=1)
  - Марка автомобиля любая (exchange\_filter[marka\_id]=0)
  - Модель автомобиля любая (exchange\_filter[model\_id]=0)

- Область
  - Винницкая - поиск по всем городам этой области (`state[0]=1&city[0]=0`)
  - Житомирская - поиск по всем городам этой области (`state[1]=2&city[1]=0`)
  - Киевская - поиск по всем городам этой области (`state[2]=10&city[2]=0`)
- Не отображать авто которые находятся не в Украине (`abroad=2`)
- Не отображать нерастаможенные авто (`custom=1`)
- Гаражное хранение (`auto_options[477]=477`)
- Топливо
  - Бензин (`type[0]=1`)
  - Дизель (`type[1]=2`)
  - Газ/бензин (`type[3]=4`)
  - Электро (`type[5]:6`)
- КПП
  - Ручная / Механика (`gearbox[0]=1`)
  - Автомат (`gearbox[1]=2`)
  - Типтроник (`gearbox[2]=3`)
- Объем 1.4 - 3.2 л. (`engineVolumeFrom=1.4&engineVolumeTo=3.2`)
- Мощность 90 - 250 (`powerFrom=90&powerTo=250`)
  - Единица измерения мощности - л.с (`power_name=1`)
- Только с фото (`with_photo=1`)

В итоге мы получаем запрос такого [вида]:

[https://developers.ria.com/auto/search?api\\_key=YOUR\\_API\\_KEY&category\\_id=1&bodystyle\[0\]=3&...](https://developers.ria.com/auto/search?api_key=YOUR_API_KEY&category_id=1&bodystyle[0]=3&...)

В случае успешного выполнения запроса по указанным параметрам результат будет со статусом **200 ОК**.

Пример успешного ответа:

```
[
  {
    "additional_params": {
      "lang_id": 2,           // Русский язык
      "page": 0,             // Порядковый номер страницы
      "view_type_id": 0,
      "target": "search",
      "section": "auto",    // Поиск по авто
      "catalog_name": "",
      "elastica": true,
      "nodejs": true
    },
    "result": {              // Результат поиска
      "search_result": {
        "ids": [
          "19519211",
          "19684763",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "19493489",
    "19714833",
    "19393702",
    "19692238", // id объявлений
    "19574398",
    "18154136",
    "19391327",
    "18693600",
    "19431563",
    "18047892"
  ],
  "count": 1, // Количество id объявлений доступных по
  ↪ заданным параметрам
  "last_id": 0,
  "qs": {
    "fields": [
      "_id"
    ],
    "size": 50, // Количество отображаемых id объявлений
    "from": 0,
  }
}
]

```

**Важно**

Максимальное количество отображаемых id объявлений за один запрос равно 100(*countpage*). В случае если результат поиска превышает это значение можно добавить параметр *page* (порядковый номер страницы) с помощью которого можно просматреть все результаты поиска.

```

"additional_params": {
  "lang_id": 2,
  "page": "1", // Порядковый номер страницы
  "view_type_id": 0,
  "target": "search",
  "section": "auto",
  "catalog_name": "",
  "elastica": true,
  "nodejs": true
},
"result": {
  "search_result": {
    "ids": [
      "19885907",
      "19885290",
      "19847856",
      "19876230",
      "19662019",
      "18493054",
      "19440597",
      "19865852",
      "18814906",
      "19862470",
      "19738583",
      . . . . .
    ]
  }
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "19860637",
    "19747721",
    "18050784",
    "19810589",
    "19746765",
    "18412097",
    "18545537"
  ],
  "count": 1578,           // Количество id по результатам поиска
  "last_id": 0,
  "qs": {
    "fields": [
      "_id"
    ],
    "size": 100,
    "from": 100,
  },

```

Параметры поиска на сайте AUTO.RIA отличаются от параметров указанных ниже. Для использования параметров поиска с сайта AUTO.RIA нужно обратиться к сервису конфигурации новых параметров в старые.

#### Пример

Допустим Вам нужен список автомобилей по параметрам указанных ниже:

[https://auto.ria.com/search/?categories.main.id=1&brand.id\[0\]...](https://auto.ria.com/search/?categories.main.id=1&brand.id[0]...)

Используем сервис конфигурации

[https://developers.ria.com/new\\_to\\_old?api\\_key=YOUR\\_API\\_KEY&categories.main.id=1&brand.id...](https://developers.ria.com/new_to_old?api_key=YOUR_API_KEY&categories.main.id=1&brand.id...)

В результате мы получим параметры которые сможем применять в API Search:

```

{
  "unrecognized": {
    "countpage": "100"
  },
  "converted": {
    "category_id": "1",
    "marka_id[0]": "9",
    "model_id[0]": "0",
    "s_yers[0]": "2011",
    "po_yers[0]": "2016",
    "custom": "1",
    "type[5]": "6",
    "gearbox[1]": "2",
    "gearbox[2]": "3",
    "countpage": "10\"
  },
  "string": "category_id=1&marka_id%5B0%5D=9&model_id%5B0%5D=0&s_yers%5B0%5D=2011&po_yers%5B0%5D=2016&custom=1&type%5B5%5D=6&gearbox%5B1%5D=2&gearbox%5B2%5D=3&countpage=10%22"
}

```

Этот запрос можно производить и в обратном порядке:

Используем сервис конфигурации

[https://developers.ria.com/old\\_to\\_new?api\\_key=YOUR\\_API\\_KEY&category\\_id=1&marka\\_id\[0\]=9...](https://developers.ria.com/old_to_new?api_key=YOUR_API_KEY&category_id=1&marka_id[0]=9...)

В результате мы получим новые параметры которые можем использовать на сайте AUTO.RIA

```

{
  "unrecognized": {
  },
  "converted": {
    "categories.main.id": "1",
    "brand.id[0]": "9",
    "year[0].gte": "2011",
    "year[0].lte": "2016",
    "custom.not": "1",
    "fuel.id[5]": "6",
    "gearbox.id[1]": "2",
    "gearbox.id[2]": "3",
    "size": "100"
  },
  "string": "categories.main.id=1&brand.id%5B0%5D=9&year%5B0%5D.gte=2011&year%5B0%5D.lte=2016&
↳ custom.not=1&fuel.id%5B5%5D=6&gearbox.id%5B1%5D=2&gearbox.id%5B2%5D=3&size=100"
}

```

### 1.1.3 Подсчёт средней цены

“Подсчёт средней цены” — первый сервис, основанный на актуальных статистических данных **AUTO.RIA** (ежемесячно 10 тыс. опубликованных объявлений о продаже 7 800 марок авто, которые ежедневно собирают 8 млн. просмотров).

Теперь вы можете:

- узнавать актуальные средние цены автомобилей разных марок и моделей;
- следить за изменениями цен в кратко- и долгосрочном периодах;
- анализировать и прогнозировать изменения цен и спроса на автомобили;
- размещать полученную информацию на вашем сайте.

Ознакомьтесь с технической документацией, чтобы получить доступ и экспортировать необходимую информацию в программу вашей компании.

#### Формат данных в запросе

Все параметры описанные в таблице поддерживаемых параметров должны передаваться в виде чисел. Исключениями являются только параметры - *год выпуска*, *пробег*, *опции* и *коробка передач*.

Если передать массив в параметре *коробка передач*, то это будет интерпретироваться как поиск коробок передач с логическим оператором *ИЛИ*. Т.е. [https://developers.ria.com/auto/average\\_price?api\\_key=YOUR\\_API\\_KEY&marka\\_id=9&model\\_id=31612&gear\\_id=1&gear\\_id=2](https://developers.ria.com/auto/average_price?api_key=YOUR_API_KEY&marka_id=9&model_id=31612&gear_id=1&gear_id=2)

- выберет для подсчета все **BMW 318** с автоматическими и ручными коробками передач. Аналогично и с параметром *тип топлива*, например: [https://developers.ria.com/auto/average\\_price?api\\_key=YOUR\\_API\\_KEY&marka\\_id=9&model\\_id=31887&years=2014&fuel\\_id=1&fuel\\_id=2](https://developers.ria.com/auto/average_price?api_key=YOUR_API_KEY&marka_id=9&model_id=31887&years=2014&fuel_id=1&fuel_id=2)
- выберет для подсчета все **BMW 520 2014 года** с дизельными и бензиновыми двигателями.

Если передать массив в параметре *год выпуска* или *пробег* это будет интерпретироваться как диапазон значений. Например, [https://developers.ria.com/auto/average\\_price?api\\_key=YOUR\\_API\\_KEY&raceInt=10&raceInt=100](https://developers.ria.com/auto/average_price?api_key=YOUR_API_KEY&raceInt=10&raceInt=100) - выберет для подсчета средней цены все объявления с пробегом от 10 до 100 тыс. км.

Если передать массив значений в параметре *options* это будет интерпретироваться как поиск опций с логическим оператором «И». Т.е. [https://developers.ria.com/auto/average\\_price?api\\_key=YOUR\\_API\\_KEY&options=217&options=463](https://developers.ria.com/auto/average_price?api_key=YOUR_API_KEY&options=217&options=463) выберет для подсчета все объявления, у которых есть опция *ABS И Галогенные фары*.

### Формат данных в ответе

В случае успешного подсчета средней цены по указанным параметрам результат будет со статусом **200 ОК**.

Пример успешного ответа:

```
{
  total: 17,
  arithmeticMean: 16305.882352941177,
  interQuartileMean: 8483.333333333334,
  percentiles: {
    1.0: 1944,
    5.0: 2520,
    25.0: 3500,
    50.0: 8000,
    75.0: 23500,
    95.0: 53539.999999999985,
    99.0: 64868
  },
  prices: [
    67700,
    27000,
    3000,
    23500,
    3500,
    8100,
    10000,
    3500,
    2700,
    8000,
    11000,
    45800,
    50000,
    1800,
    4350,
    4400,
    2850
  ],
  classifieds: [
    14663610,
    14226353,
    14138132,
    13969588,
    14697569,
    13386778,
    13279188,
    14555863,
    14754932,
    14816842,
    14664706,
    13873344,

```

(continues on next page)



(продолжение с предыдущей страницы)

```

    14681607,
    14772056,
    14059841,
    14290096,
    14890250
  ]
}

```

Расшифровка параметров:

- *total* - общее количество объявлений, участвующих в подсчете.
- *arithmeticMean* - среднее арифметическое.
- *interQuartileMean* - среднее арифметическое из значений, находящихся между первым и четвертым квантилем. Грубо говоря, это среднее арифметическое без учета 25% самых маленьких и самых больших значений.
- *percentiles* - значения **процентилей**. Т.е. для данного примера 25% всех объявлений имеют цену ниже \$3500.
- *prices* - список цен, которые участвовали в подсчете средней цены. Размер ограничен 1000 элементов.
- *classifieds* - идентификаторы объявлений, к которым принадлежат цены соответственно. Размер ограничен 1000 элементов.

Записи в поля «classifieds» и «prices» идут в одинаковом порядке.

Если по каким-либо причинам не удалось подсчитать среднюю цену, ответ будет иметь статус **400 Bad Request**, а тело ответа будет содержать следующее:

```
{ "message": "Not Enough Data" }
```

## Примеры

Средняя цена по BMW X5 с пробегом от 10 до 100 тыс. км. - [https://developers.ria.com/auto/average\\_price?api\\_key=YOUR\\_API\\_KEY&marka\\_id=9&model\\_id=96&raceInt=10&range=100000](https://developers.ria.com/auto/average_price?api_key=YOUR_API_KEY&marka_id=9&model_id=96&raceInt=10&range=100000)

Средняя цена для Honda Accord в Киеве - [https://developers.ria.com/auto/average\\_price?api\\_key=YOUR\\_API\\_KEY&marka\\_id=10&model\\_id=100&city=Kyiv](https://developers.ria.com/auto/average_price?api_key=YOUR_API_KEY&marka_id=10&model_id=100&city=Kyiv)

### 1.1.4 AUTO info Информация по id объявления

Чтобы обратиться к RIA API info, Вам необходимо выполнить GET запрос такого вида:

[https://developers.ria.com/auto/info?api\\_key=YOUR\\_API\\_KEY&auto\\_id=id\\_объявления](https://developers.ria.com/auto/info?api_key=YOUR_API_KEY&auto_id=id_объявления)

или

```
curl -i -X GET "https://developers.ria.com/auto/info?api_key=YOUR_API_KEY&auto_id=id_объявления"
```

Он состоит из:

- **info** — название метода API, к которому Вы хотите обратиться.
- **API\_KEY** — ключ доступа. Для получения ключа доступа нужно зарегистрироваться на портале [developers.ria.com](https://developers.ria.com)
- **auto\_id=** — номер объявления по которому нужна информация

## Пример

Допустим Вам нужна информация по *19050985* id.

Данный запрос будет выглядеть так:

```
https://developers.ria.com/auto/info?api_key=YOUR_API_KEY&auto_id=19050985
```

или

```
curl -i -X GET "https://developers.ria.com/auto/info?api_key=YOUR_API_KEY&auto_id=19050985"
```

В случае успешного выполнения запроса результат будет со статусом **200 ОК**.

Пример успешного ответа:

```
[
  {
    "userId": 489269, // id
    ↪ пользователя
    "chipsCount": 0,
    "locationCityName": "Днепропетровск", // Город
    "auctionPossible": false, // Торг
    "exchangePossible": false, // Обмен
    ↪
    "realtyExchange": false, // Обмен
    ↪ на недвижимость
    "exchangeType": "Любой", // Тип
    ↪ обмена
    "exchangeTypeId": 0, //
    ↪ Идентификатор типа обмена
    "addDate": "2017-04-24 23:02:06", // Дата
    ↪ и время добавления
    "updateDate": "2017-04-24 23:02:06", // Дата
    ↪ и время обновления
    "expireDate": "2017-07-24 23:02:06", // Дата
    ↪ и время истечения
    "userHideADSStatus": false, //
    ↪ Пользователь скрыл статус
    "userPhoneData": {
      "phoneId": "3336492", // id
    ↪ телефона пользователя
      "phone": "(067) 522-60-77" //
    ↪ Телефон пользователя
    },
    "USD": 33500, //
    ↪ Стоимость в долларах США
    "UAH": 880380, //
    ↪ Стоимость в гривнах
    "EUR": 30815, //
    ↪ Стоимость в евро
    "isAutoAddedByPartner": false,
    "partnerId": 0,
    "levelData": {
      "level": 10, //
    ↪ Уровень топа
      "label": 3, // Метка
      "hotType": "срочно", // Метка
    ↪ "ТОР" или "Срочно"
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "expireDate": "2017-05-04 23:02:07" // Дата
↪ и время истечения метки "top" "Срочно"
  },
  "autoData": { //
↪ Характеристики машины
    "description": "Итак. Пришло время в срочном порядке продать свой любимый автомобиль. " +
    "\r\nПокупал его в августе 2013 года. \r\nПолностью обслужен на о", //
↪ Описание

    "version": "",
    "onModeration": false,
    "year": 2013, // Год
↪ выпуска
    "autoId": 19050985, // id
↪ объявления
    "statusId": 0, // -
    "withVideo": false, //
↪ Наличие видео
    "race": "60 тыс. км", // Пробег
    "raceInt": 60, // -
    "fuelName": "Дизель", // Тип
↪ топлива
    "gearboxName": "Автомат", // Тип
↪ привода
    "isSold": false, // Этом
↪ параметр показывает или машина продана или нет
    "mainCurrency": "USD", //
↪ Основная валюта
    "fromArchive": false,
    "categoryId": 1, // Тип
↪ транспорта
    "custom": 0
  },
  "markName": "Volvo", // Марка
↪ автомобиля
  "markId": 85, // id
↪ марки автомобиля
  "modelName": "XC90", //
↪ Модель автомобиля
  "modelId": 824, // id
↪ Модели автомобиля
  "photoData": { //
↪ Фотографии
    "count": 15, //
↪ Количество фотографий
    "seoLinkM": "https://cdn.riastatic.com/photosnew/auto/photo/volvo_xc90__181949196m.jpg",
    "seoLinkSX": "https://cdn.riastatic.com/photosnew/auto/photo/volvo_xc90__181949196sx.jpg",
    "seoLinkB": "https://cdn.riastatic.com/photosnew/auto/photo/volvo_xc90__181949196b.jpg",
    "seoLinkF": "https://cdn.riastatic.com/photosnew/auto/photo/volvo_xc90__181949196f.jpg"
  },
  "linkToView": "/auto_volvo_xc90_19050985.html", // Линк
↪ на объявление
  "title": "Volvo XC90", //
↪ Название
  "stateData": {
    "name": "Днепропетровск", // Город

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "regionName": "Днепропетровская", //␣
↪ Область
    "linkToCatalog": "/city/dnepropetrovsk/",
    "title": "Поиск объявлений по городу Днепропетровск",
    "stateId": 11 // id␣
↪ Города
  },
  "oldTop": {
    "isActive": true,
    "expireDate": ""
  },
  "canSetSpecificPhoneToAdvert": false,
  "dontComment": 0,
  "sendComments": 0,
  "badges": [

],
  "checkedVin": { //␣
↪ Проверка vin-кода
    "isShow": false, //␣
↪ Отображать vin-код
    "linkToReport": "/vin-check/auto/19050985/",
    "hasRestrictions": false, //␣
↪ Ограничения
    "checkDate": "04.05.2017",
    "isChecked": false // vin-
↪ код проверен
  },
  "isLeasing": 0, // Лизинг
  "dealer": {
    "link": "",
    "logo": "",
    "type": "",
    "id": 0,
    "name": "",
    "packageId": 0,
    "typeId": 0
  },
  "withInfoBar": false, //␣
↪ Информационная панель
  "infoBarText": "",
  "optionStyles": [

]
]

```

### 1.1.5 Работа с объявлениями

Зарегистрируйтесь или авторизируйтесь на Developers.RIA.com <<https://developers.ria.com/>>‘\_’, чтобы использовать персонализированные API. По умолчанию все права на доступ к сервисам не активны, что бы активировать нужный Вам сервис перейдите во вкладку «Мои разрешения» <<https://developers.ria.com/account/credentials>>‘\_’

Полное описание сервиса «Работа с объявления» описаний с помощью стандарта **DeFacto swagger 2.0** \_‘здесь <<http://swagger.ria.com/ui/?api=auto/advertisements#/>>‘\_’

## Структура параметров

Параметры которые должны обязательно присутствовать в запросе

Дополнительные параметры

## Добавление объявления

Чтобы добавить объявления, Вам необходимо выполнить POST запрос такого вида:

```
curl -X POST "https://developers.ria.com/auto/used/autos/?user_id=Ваш_ID&api_key=YOUR_API_KEY" -H
↪ "accept: application/json" -H "content-type: application/json" -d "{ \ \ }"
```

, где *user\_id* - Ваш ID в системе RIA.com, *-d "{ \ \ }"* - здесь Вы указываете массив параметров которые хотите добавить, *api\_key* - Ваш ключ.

Объявления по умолчанию добавляются в черновики, чтобы опубликовать их, Вам нужно зайти в свой личный кабинет на AUTO.RIA

Также стоит заметить, что существуют параметры обязательны для заполнения. Детально с параметрами можно ознакомиться [здесь](#).

## Пример запроса

```
curl -X POST "https://developers.ria.com/auto/used/autos/?user_id=7069830&api_key=YOUR_API_KEY" -H
↪ "accept: application/json" -H "content-type: application/json" -d "{ \"damage\": false, \"custom\
↪ \": false, \"year\": 2016, \"price\": { \"value\": 80000, \"currency\": { \"id\": 1 } }, \
↪ \"categories\": { \"main\": { \"id\": 1 } }, \"brand\": { \"id\": 5 }, \"model\": { \"id\": 963 },
↪ \"modification\": \"V12 AMG B-turbo\", \"body\": { \"id\": 6 }, \"mileage\": 32, \"region\": { \
↪ \"id\": 10 }, \"city\": { \"id\": 10 }, \"VIN\": \"JKBVNCB164A662141\", \"gearbox\": { \"id\": 1 }
↪ \", \"drive\": { \"id\": 2 }, \"fuel\": { \"id\": 2, \"consumption\": { \"city\": 10, \"route\": 6,
↪ \", \"combine\": 8 } }, \"engine\": { \"volume\": { \"liters\": 3.5 } }, \"power\": { \"hp\": 585, \
↪ \"kW\": 430 }, \"color\": { \"id\": 10, \"metallic\": true }, \"post\": { \"auctions\": true, \
↪ \"comments\": { \"allowed\": true, \"check\": false }, \"exchanges\": { \"payment\": { \"id\": 2 }
↪ \", \"type\": { \"id\": 1 } } }, \"video\": { \"key\": \"lLEiT9PeHSg\" }, \"description\": { \"ru\
↪ \": \"Авто в идеальном состоянии, вложений не требует\", \"uk\": \"Авто в ідеальному стані,
↪ \", \"spareParts\": false}"
```

Развернутый пример

```
curl -X POST "https://developers.ria.com/auto/used/autos/?user_id=7069830&api_key=YOUR_API_KEY" -H
↪ "accept: application/json" -H "content-type: application/json" -d
"{
  \"damage\":false,
  \"custom\":false,
  \"year\":2016,
  \"price\":{
    \"value\":80000,
    \"currency\":{
      \"id\":1
    }
  },
  \"categories\":{
    \"main\":{
      \"id\":1
    }
  },
  \"spareParts\":false}"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
\ "brand\" : {
  \ "id\" : 5
},
\ "model\" : {
  \ "id\" : 963
},
\ "modification\" : \ "V12 AMG B-turbo\" ,
\ "body\" : {
  \ "id\" : 6
},
\ "mileage\" : 32,
\ "region\" : {
  \ "id\" : 10
},
\ "city\" : {
  \ "id\" : 10
},
\ "VIN\" : \ "JKBVNCB164A662141\" ,
\ "gearbox\" : {
  \ "id\" : 1
},
\ "drive\" : {
  \ "id\" : 2
},
\ "fuel\" : {
  \ "id\" : 2,
  \ "consumption\" : {
    \ "city\" : 10,
    \ "route\" : 6,
    \ "combine\" : 8
  }
},
\ "engine\" : {
  \ "volume\" : {
    \ "liters\" : 3.5
  }
},
\ "power\" : {
  \ "hp\" : 585,
  \ "kW\" : 430
},
\ "color\" : {
  \ "id\" : 10,
  \ "metallic\" : true
},
\ "post\" : {
  \ "auctions\" : true,
  \ "comments\" : {
    \ "allowed\" : true,
    \ "check\" : false
  },
  \ "exchanges\" : {
    \ "payment\" : {
      \ "id\" : 2
    },
    \ "type\" : {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        \id\:1
      }
    },
    \video\:{
      \key\:\LEiT9PeHSg\
    },
    \description\:{
      \ru\:"Авто в идеальном состоянии,
      вложений не требует",
      \uk\:"Авто в ідеальному стані,
      вкладень не потребує"
    },
    \doors\:2,
    \seats\:2,
    \country\:{
      \import\:{
        \id\:0
      }
    },
    \spareParts\:false
  } "

```

**Пример успешного ответа:**

```

{
  "damage":false,
  "custom":false,
  "year":2016,
  "price":{
    "value":80000,
    "currency":{
      "id":1
    }
  },
  "categories":{
    "main":{
      "id":1
    },
    "all":[
      {
        "id":1
      }
    ]
  },
  "brand":{
    "id":5
  },
  "model":{
    "id":963
  },
  "modification":"V12 AMG B-turbo",
  "body":{
    "id":6
  },
  "mileage":32,

```

(continues on next page)

```
"region":{
  "id":10
},
"city":{
  "id":10
},
"VIN":"JKBVNCB164A66XXXX",
"gearbox":{
  "id":1
},
"drive":{
  "id":2
},
"fuel":{
  "id":2,
  "consumption":{
    "city":10,
    "route":6,
    "combine":8
  }
},
"engine":{
  "volume":{
    "liters":3.5
  }
},
"power":{
  "hp":585,
  "kW":430
},
"color":{
  "id":10,
  "metallic":true
},
"post":{
  "auctions":true,
  "comments":{
    "allowed":true,
    "check":false
  },
  "exchanges":{
    "payment":{
      "id":2
    },
    "type":{
      "id":1
    }
  }
},
"video":{
  "key":"1LEiT9PeHSg"
},
"description":{
  "ru":"Авто в идеальном состоянии, вложений не требует",
  "uk":"Авто в ідеальному стані, вкладень не потребує"
},
```

(continues on next page)



(продолжение с предыдущей страницы)

```

"doors":2,
"seats":2,
"country":{
  "import":{
    "id":0
  }
},
"spareParts":false,
"user":{
  "id":7069830
},
"ip":"80.91.174.90",
"dates":{
  "created":"2017-09-18T12:07:43.840Z"
},
"status":{
  "id":13
},
"_id":20476120
}

```

### Список объявлений пользователя

Чтобы получить список объявлений, Вам необходимо выполнить GET запрос такого вида:

`https://developers.ria.com/auto/used/autos/ids?user_id=Ваш ID в системе RIA.com&api_key=YOUR_API_KEY`

или

```

curl -X GET "https://developers.ria.com/auto/used/autos/ids?user_id=Ваш ID&api_key=YOUR_API_KEY" -
↳H "accept: application/json"

```

, где `user_id` - Ваш ID в системе RIA.com, `api_key` - Ваш ключ.

### Пример запроса

```

curl -X GET "https://developers.ria.com/auto/used/autos/ids?user_id=7069830&api_key=YOUR_API_KEY" -
↳H "accept: application/json"

```

### Пример успешного ответа:

```

{
  "active": {
    "moderated": [

    ],
    "notModerated": [

    ]
  },
  "draft": [

  ],
  "awaitingPayment": [

  ]
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```
],  
  "archive": [  
    20438832,  
    20445742  
  ]  
}
```

Расшифровка параметров:

- *active* - Активные объявления
- *moderated* - Объявления находятся на модерации
- *notModerated* - Объявления не от модерировано
- *draft* - Объявления находятся в черновиках
- *awaitingPayment* - Объявления ожидают оплаты
- *archive* - Объявления в архиве

Полное описание сервиса «Список объявлений пользователя» описаний с помощью стандарта **DeFacto swagger 2.0** [здесь](#)

### Статистика объявления

Чтобы получить статистику об объявлении, Вам необходимо выполнить GET запрос такого вида:

```
https://developers.ria.com/auto/used/autos/advertisementId/statistic?user_id=Ваш  
ID&api_key=YOUR_API_KEY
```

или

```
curl -X GET "https://developers.ria.com/auto/used/autos/advertisementId/statistic?user_id=Ваш ID&  
api_key=YOUR_API_KEY" -H "accept: application/json"
```

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, *api\_key* - Ваш ключ.

### Пример запроса

```
curl -X GET "https://developers.ria.com/auto/used/autos/20268839/statistic?user_id=7069830&api_  
key=YOUR_API_KEY" -H "accept: application/json"``
```

### Пример успешного ответа:

```
{  
  "views": 329,  
  "clicks": 24  
}
```

Расшифровка параметров:

- *views* - Количество просмотров объявления
- *clicks* - Количество открытий номера телефона

Полное описание сервиса «Статистика объявления» описанный с помощью стандарта **DeFacto swagger 2.0** [здесь](#)

## Получение основной информации объявления

Чтобы получить основную информацию об объявлении, Вам необходимо выполнить GET запрос такого вида:

```
https://developers.ria.com/auto/used/autos/advertisementId?user_id=Ваш ID&api_key=YOUR_API_KEY
```

или

```
curl -X GET "https://developers.ria.com/auto/used/autos/advertisementId?user_id=Ваш ID&api_key=YOUR_API_KEY" -H "accept: application/json"
```

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, *api\_key* - Ваш ключ.

### Пример запроса

```
curl -X GET "https://developers.ria.com/auto/used/autos/20445742?user_id=7069830&api_key=YOUR_API_KEY" -H "accept: application/json"
```

### Пример успешного ответа:

```
{
  "_id": 20445742,
  "fuel": {
    "consumption": {
      "city": 10,
      "route": 6,
      "combine": 8
    },
    "id": 2
  },
  "brand": {
    "id": 5
  },
  "model": {
    "id": 963
  },
  "modification": "V12 AMG B-turbo",
  "body": {
    "id": 6
  },
  "doors": 2,
  "gearbox": {
    "id": 1
  },
  "color": {
    "id": 10,
    "metallic": true
  },
  "year": 2016,
  "mileage": 32,
  "price": {
    "value": 80000,
    "currency": {
      "id": 1
    }
  },
  "post": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"auctions": true,
"exchanges": {
  "type": {
    "id": 1
  },
  "payment": {
    "id": 2
  }
},
"comments": {
  "allowed": true,
  "check": false
}
},
"spareParts": false,
"user": {
  "id": 7069830,
  "type": "private"
},
"dates": {
  "created": "2017-09-13T06:26:35.000Z",
  "updated": "0000-00-00 00:00:00",
  "expired": "0000-00-00 00:00:00",
  "sold": "0000-00-00 00:00:00"
},
"status": {
  "id": 1
},
"partner": {
  "id": 1
},
"ip": 8091,
"description": {
  "ru": "Авто в идеальном состоянии, вложений не требует",
  "uk": "Авто в ідеальному стані, вкладень не потребує"
},
"region": {
  "id": 10
},
"city": {
  "id": 10
},
"moderated": true,
"damage": false,
"custom": false,
"photos": {
  "main": {
    "id": 196054982
  }
},
"credit": false,
"video": {
  "key": "lLEiT9PeHSg"
},
"categories": {
  "main": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "id": 1
  },
  "all": [
    {
      "id": 1
    }
  ]
},
"drive": {
  "id": 2
},
"VIN": "JKBVNCB164A66XXXX",
"confiscated": false,
"verified": {
  "serviceStation": false
},
"seats": 2,
"power": {
  "hp": 585,
  "kW": 430.56
},
"engine": {
  "volume": {
    "cc": 3500,
    "liters": 3.5
  }
}
}
}

```

Полное описание сервиса «Получение основной информации объявления» описаный с помощью стандарта **DeFacto swagger 2.0** [здесь](#)

### Получения списка фотографий объявления

Для получения списка фотографий объявления, Вам необходимо выполнить GET запрос такого вида:

„[https://developers.ria.com/auto/fotos/advertisementId?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/fotos/advertisementId?api_key=YOUR_API_KEY)“

, где *advertisementId* - ID нужного Вам объявления, *api\_key* - Ваш ключ.

### Пример запроса

```
https://developers.ria.com/auto/fotos/19860101?api_key=YOUR_API_KEY'
```

### Пример успешного ответа:

```

{
  "status": 1,
  "data": {
    "19860101": {
      "187203123": {
        "photo_id": 187203123,
        "auto_id": 19860101,
        "status": 0,
        "checked": 1,
        "standard": 0,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"date_add": "2017-06-04 13:27:33",
"description": null,
"url": "auto/photo/18720/1872031/187203123/187203123.jpg",
"formats": [
  "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203123/187203123f.jpg",
  "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203123/187203123fx.jpg"
]
},
"187203127": {
  "photo_id": 187203127,
  "auto_id": 19860101,
  "status": 0,
  "checked": 1,
  "standard": 0,
  "date_add": "2017-06-04 13:27:35",
  "description": null,
  "url": "auto/photo/18720/1872031/187203127/187203127.jpg",
  "formats": [
    "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203127/187203127f.jpg",
    "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203127/187203127fx.jpg"
  ]
},
"187203129": {
  "photo_id": 187203129,
  "auto_id": 19860101,
  "status": 0,
  "checked": 1,
  "standard": 0,
  "date_add": "2017-06-04 13:27:37",
  "description": null,
  "url": "auto/photo/18720/1872031/187203129/187203129.jpg",
  "formats": [
    "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203129/187203129f.jpg",
    "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203129/187203129fx.jpg"
  ]
},
"187203135": {
  "photo_id": 187203135,
  "auto_id": 19860101,
  "status": 0,
  "checked": 1,
  "standard": 0,
  "date_add": "2017-06-04 13:27:39",
  "description": null,
  "url": "auto/photo/18720/1872031/187203135/187203135.jpg",
  "formats": [
    "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203135/187203135f.jpg",
    "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203135/187203135fx.jpg"
  ]
},
"187203138": {
  "photo_id": 187203138,
  "auto_id": 19860101,
  "status": 0,
  "checked": 1,
  "standard": 0,
```

(continues on next page)



Основные параметры транспортного средства доступны к редактированию на протяжении часа, после добавления на сайт:

- “Регион”
- “Тип транспорта”, “Марка”, “Модель”, “Версия”, “Год выпуска”
- “VIN-номер”, “Коробка передач”, “Привод”, “Количество дверей”, “Цвет”, “Топливо”

Остальные параметры - “Цена”, “Пробег”, “Фото” автомобиля, подробное описание — можете изменять и корректировать в любое время.

Если же по истечении 60 мин. отправить запрос на изменение одного из выше перечисленных параметров, Вы получите ошибку.

### Пример ошибки

```
{
  "message": "StatusCodeError: 400 - {"message": "\"Validation failed\"\", \"errors\": [{"message\": \"
↪ \"You can edit categories.main.id for 60 minutes\", \"code\": 7}, {\"message\": \"You can edit brand.
↪ id for 60 minutes\", \"code\": 7}, {\"message\": \"You can edit model.id for 60 minutes\", \"code\": 7}
↪ ]}\"
}
```

### Удаление объявления

Для удаления объявления, Вам необходимо отправить **DELETE** запрос такого вида:

```
curl -X DELETE "https://developers.ria.com/auto/used/autos/advertisementId?user_id=Ваш ID&reason_
↪ id=id причины удаления&api_key=YOUR_API_KEY" -H "accept: application/json"
```

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, *reason\_id* - причина удаления, *api\_key* - Ваш ключ.

*reason\_id* состоит из 4 категорий:

- *reason\_id*=4 - Авто продано с помощью AUTO.RIA.com
- *reason\_id*=5 - Авто продано в другом месте
- *reason\_id*=6 - Я хочу разместить объявление заново
- *reason\_id*=7 - Я передумал продавать автомобиль

### Пример запроса

```
curl -X DELETE "https://developers.ria.com/auto/used/autos/20460133?user_id=7069830&reason_id=7&
↪ api_key=YOUR_API_KEY" -H "accept: application/json"
```

Полное описание сервиса «Удаление объявления» описанный с помощью стандарта **DeFacto swagger 2.0** [здесь](#)

### Получение информации об опциях объявления

Чтобы получить информации об опциях объявления, Вам необходимо выполнить GET запрос такого вида:

```
https://developers.ria.com/auto/used/autos/advertisementId/options?user_id=Ваш ID&api_key=YOUR_API_KEY
```



или `curl -X GET "https://developers.ria.com/auto/used/autos/advertisementId/options?user_id=Ваш ID&api_key=YOUR_API_KEY" -H "accept: application/json"`

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, *api\_key* - Ваш ключ.

### Пример запроса

```
curl -X GET "https://developers.ria.com/auto/used/autos/20438832/options?user_id=7069830&api_key=YOUR_API_KEY" -H "accept: application/json"
```

### Пример успешного ответа:

```
[
  {
    "id": 125
  },
  {
    "id": 137
  },
  {
    "id": 188
  },
  {
    "id": 189
  },
  {
    "id": 190
  },
  {
    "id": 191
  },
  {
    "id": 211
  },
  {
    "id": 217
  },
  {
    "id": 225
  },
  {
    "id": 303
  },
  {
    "id": 354
  },
  {
    "id": 355
  },
  {
    "id": 437
  },
  {
    "id": 463
  },
  {
    "id": 481
  }
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
},  
{  
  "id": 525  
}  
]
```

Для расшифровки полученных параметров можно использовать сервис [Опции](#)

Полное описание сервиса «Получение информации об опциях объявления» описанный с помощью стандарта **DeFacto swagger 2.0** [здесь](#)

### Добавление опций

Для добавления опций в объявление, Вам необходимо выполнить PUT запрос такого вида:

```
curl -X POST "https://developers.ria.com/auto/used/autos/advertisementId/options?user_id=Ваш ID&  
↪api_key=YOUR_API_KEY" -H "accept: application/json" -H "content-type: application/json" -d "[ { \\  
↪id\": ID опции }]"
```

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, «id»: - ID нужной Вам опции, *api\_key* - Ваш ключ.

### Пример запроса

```
curl -X POST "https://developers.ria.com/auto/used/autos/20438832/options?user_id=7069830&api_  
↪key=YOUR_API_KEY" -H "accept: application/json" -H "content-type: application/json" -d "[ { \"id\  
↪\": 525 }]"`
```

### Пример успешного ответа:

```
{ "message": "Ok",  
  "errors": [],  
  "success": true }
```

Полное описание сервиса «Добавление опций» описанный с помощью стандарта **DeFacto swagger 2.0** [здесь](#)

### Удаление опций

Для удаления опций объявления, Вам необходимо отправить **DELETE** запрос такого вида:

```
curl -X DELETE "https://developers.ria.com/auto/used/autos/advertisementId/options/ids?user_id=Ваш_  
↪ID&api_key=YOUR_API_KEY" -H "accept: application/json"
```

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, «id»: - ID нужной Вам опции, *api\_key* - Ваш ключ.

### Пример запроса

```
curl -X DELETE "https://developers.ria.com/auto/used/autos/20438832/options/125,525,481?user_  
↪id=7069830&api_key=YOUR_API_KEY" -H "accept: application/json" `
```

## Получение фильтров на обмен по вашему объявлению

Для получения опций в объявление, Вам необходимо выполнить GET запрос такого вида:

```
https://developers.ria.com/auto/used/autos/advertisementId/exchangeFilters?user_id=Ваш
ID&api_key=YOUR_API_KEY
```

или

```
curl -X GET "https://developers.ria.com/auto/used/autos/advertisementId/exchangeFilters?user_
↳ id=Ваш ID&api_key=YOUR_API_KEY" -H "accept: application/json"
```

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, *api\_key* - Ваш ключ.

### Пример запроса

```
curl -X GET "https://developers.ria.com/auto/used/autos/20438832/exchangeFilters?user_id=7069830&
↳ api_key=YOUR_API_KEY" -H "accept: application/json"
```

### Пример успешного ответа:

```
[
  {
    "id": 2930510,    // Id добавленного фильтра
    "brand": {
      "id": 6
    },
    "model": {
      "id": 3460
    },
    "year": {
      "lte": 2013,
      "gte": 2010
    },
    "category": {
      "id": 1
    },
    "body": {
      "id": 3
    }
  }
]
```

Полное описание сервиса «Получение фильтров на обмен по вашему объявлению» описанный с помощью стандарта **DeFacto swagger 2.0** [здесь](#)

## Добавление нового фильтра на обмен

Для добавления нового фильтра на обмен в объявление, Вам необходимо выполнить POST запрос такого вида:

```
curl -X POST "https://developers.ria.com/auto/used/autos/advertisementId/exchangeFilter?user_
↳ id=Ваш ID&api_key=YOUR_API_KEY" -H "accept: application/json" -H "content-type: application/json"
↳ -d "{\ \}"
```

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, *-d "{ \ \ }"* - здесь Вы указываете массив параметров на обмен которые хотите добавить, *api\_key* - Ваш ключ.

### Пример запроса

```
curl -X POST "https://developers.ria.com/auto/used/autos/20438832/exchangeFilter?user_id=7069830&api_key=YOUR_API_KEY" -H "accept: application/json" -H "content-type: application/json" -d "{ \ \ \"brand\": { \"id\": 5 }, \"model\": { \"id\": 32 }, \"category\": { \"id\": 1 }, \"body\": { \ \ \"id\": 3 }, \"year\": { \"gte\": 2008, \"lte\": 2012 } }"
```

### Пример успешного ответа:

```
{"message": "Ok", "errors": [], "success": true}
```

Полное описание сервиса «Добавление нового фильтра на обмен» описанный с помощью стандарта DeFacto swagger 2.0 [здесь](#)

### Удаление существующего фильтра на обмен

Для удаления существующего фильтра на обмен, Вам необходимо выполнить DELETE запрос такого вида:

```
curl -X DELETE "https://developers.ria.com/auto/used/autos/advertisementId/exchangeFilter/filterId?user_id=Ваш ID&api_key=c" -H "accept: application/json"
```

, где *user\_id* - Ваш ID в системе RIA.com, *advertisementId* - ID нужного Вам объявления, *filterId* - ID выбраного Вами фильтра, *api\_key* - Ваш ключ.

### Пример запроса

```
curl -X DELETE "https://developers.ria.com/auto/used/autos/20438832/exchangeFilter/2930510?user_id=7069830&api_key=YOUR_API_KEY" -H "accept: application/json"
```

Полное описание сервиса «Удаление существующего фильтра на обмен» описанный с помощью стандарта DeFacto swagger 2.0 [здесь](#)

### Добавления фотографий к объявлению

Для добавления фотографий к объявлению, Вам необходимо выполнить POST запрос такого вида:

```
curl -X POST "https://developers.ria.com/auto/used/autos/advertisementId/photos/upload?user_id=Ваш ID&api_key=YOUR_API_KEY" -H "accept: application/json" -H "content-type: application/json" -d "{ \ \ \"main\": \"главная фотография\", \"links\": [ \"фотографии\" ] }"
```

Расшифровка параметров:

- *advertisementId* - ID нужного Вам объявления
- *user\_id* - Ваш ID в системе RIA.com
- *api\_key* - Ваш ключ
- *-d "{ \"main\": \"ссылка на главную фотографию\", \"links\": [ \"ссылки на фотографии\" ] }"* - здесь Вы указываете массив фотографий которые хотите добавить. В каждой ссылке на изображение должен присутствовать формат

Фотографии будут добавлены не по очередности, а в случайном порядке.

Полное описание сервиса «Добавления фотографий к объявлению» описаный с помощью стандарта DeFacto swagger 2.0 [здесь](#)

### Пример запроса

```
curl -X POST "https://developers.ria.com/auto/used/autos/21739303/photos/upload?user_id=7069830&
↪api_key=bVjbA3izvnu04o7Pliw41bXAlY3mIj03CfQpeMNC" -H "accept: application/json" -H "content-
↪type: application/json" -d '{"main": "https://cdn0.riastatic.com/photosnew/auto/photo/bmw_
↪320__213471725fx.jpg", "links": [ "https://cdn0.riastatic.com/photosnew/auto/photo/bmw_320__
↪213471725fx.jpg", "https://cdn0.riastatic.com/photosnew/auto/photo/bmw_320__213471726fx.jpg"
↪}]'
```

### Пример успешного ответа

```
{"message": "Ok", "errors": []}
```

## 1.2 AUTO.RIA новые авто

### 1.2.1 Методы для работы с параметрами новые авто

#### Типы транспорта

Что бы получить список типов транспорта нужно отправив GET запрос на адрес [https://developers.ria.com/auto/categories/?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/categories/?api_key=YOUR_API_KEY). Результат будет примерно следующим:

```
[
  { name: "Легковые", value: 1 },
  { name: "Мото", value: 2 },
  { name: "Водный транспорт", value: 3 },
  { name: "Спецтехника", value: 4 },
  { name: "Прицеп", value: 5 },
  { name: "Грузовик", value: 6 },
  { name: "Автобус", value: 7 },
  { name: "Автодом", value: 8 },
]
```

#### Марки

Марки зависят от типов транспорта. Поэтому для того, чтобы получить список марок необходимо отправить GET запрос по адресу [https://developers.ria.com/auto/new/marks?category\\_id=id&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/marks?category_id=id&api_key=YOUR_API_KEY), где *category\_id* - идентификатор типа транспорта, *api\_key* - Ваш ключ.

Например, для легковых автомобилей ([https://developers.ria.com/auto/new/marks?category\\_id=1&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/marks?category_id=1&api_key=YOUR_API_KEY)) результат будет следующим:

```
[
  {
    "name": "Acura",
    "marka_id": 98,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"eng": "acura",
"country_id": 392,
"category_id": 1
},
{
  "name": "Alfa Romeo",
  "marka_id": 3,
  "eng": "alfa-romeo",
  "country_id": 380,
  "category_id": 1
},
{
  "name": "Aston Martin",
  "marka_id": 5,
  "eng": "aston-martin",
  "country_id": 826,
  "category_id": 1
},
{
  "name": "Audi",
  "marka_id": 6,
  "eng": "audi",
  "country_id": 276,
  "category_id": 1
},
{
  "name": "Bentley",
  "marka_id": 8,
  "eng": "bentley",
  "country_id": 826,
  "category_id": 1
},
]
.....
```

## Модели

Модели зависят от типов транспорта и марок. Следовательно список марок можно получить по адресу [https://developers.ria.com/auto/new/models?marka\\_id=id&category\\_id=id&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/models?marka_id=id&category_id=id&api_key=YOUR_API_KEY), где *category\_id* - идентификатор типа транспорта а *marka\_id* - идентификатор марки, *api\_key*- Ваш ключ.

Например, для легкового автомобиля BMW ([https://developers.ria.com/auto/new/models?marka\\_id=9&category\\_id=1](https://developers.ria.com/auto/new/models?marka_id=9&category_id=1)) список моделей будет следующим:

```
[
  {
    "parent_id": 0,
    "name": "1 Series",
    "model_id": 2161,
    "marka_id": 9,
    "eng": "1-series",
    "category_id": "1"
  },
  {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"parent_id": 0,
"name": "2 Series",
"model_id": 43023,
"marka_id": 9,
"eng": "2-series",
"category_id": "1"
},
{
  "parent_id": 0,
  "name": "3 Series",
  "model_id": 3219,
  "marka_id": 9,
  "eng": "3-series",
  "category_id": "1"
},
{
  "parent_id": 3219,
  "name": "3 Series GT",
  "model_id": 43029,
  "marka_id": 9,
  "eng": "3-series-gt",
  "category_id": "1"
},
{
  "parent_id": 0,
  "name": "4 Series",
  "model_id": 42495,
  "marka_id": 9,
  "eng": "4-series",
  "category_id": "1"
},
]
.....

```

### Типы кузова в поколении новых авто

Тип кузова зависит от поколения авто. Поэтому для того, чтобы получить список типов кузова необходимо отправить GET запрос на адрес [https://developers.ria.com/auto/new/generation\\_bodystyles?generation\\_id=id&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/generation_bodystyles?generation_id=id&api_key=YOUR_API_KEY), где *generation\_id* - поколение авто, *api\_key* - Ваш ключ.

Например, для модели BMW X6 в поколении E71 (рестайлинг) ([https://developers.ria.com/auto/new/generation\\_bodystyles?generation\\_id=432&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/generation_bodystyles?generation_id=432&api_key=YOUR_API_KEY)), список типов кузова будет следующим:

```

[
  {
    "generation_bodystyle_id": 538,
    "bodystyle_id": 324
  }
]

```

### Поколение модели нового авто

Поколение зависит от модели нового авто. Следовательно список поколений можно получить отправив GET запрос на адрес [https://developers.ria.com/auto/new/generation?model\\_id=id&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/generation?model_id=id&api_key=YOUR_API_KEY), где *model\_id* - модель выбранного авто, *api\_key*- Ваш ключ.

Например, для модели BMW X6 ([https://developers.ria.com/auto/new/generation?model\\_id=2153&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/generation?model_id=2153&api_key=YOUR_API_KEY)) список поколений будет следующим:

```
[
  {
    "year_to": 2014,
    "year_from": 2012,
    "name": "E71 (рестайлинг)",
    "model_id": 2153,
    "marka_id": 9,
    "generation_id": 432,
    "eng": "e71-restajling"
  },
  {
    "year_to": 0,
    "year_from": 2014,
    "name": "F16",
    "model_id": 2153,
    "marka_id": 9,
    "generation_id": 433,
    "eng": "f16"
  },
  {
    "year_to": 2010,
    "year_from": 2008,
    "name": "E71",
    "model_id": 2153,
    "marka_id": 9,
    "generation_id": 3275,
    "eng": "e71"
  }
]
```

### Комплектации авто по модификации

Комплектации авто зависит от модификации (базы). Получить их список можно отправив GET запрос по адресу [https://developers.ria.com/auto/new/equip\\_base?base\\_id=id&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/equip_base?base_id=id&api_key=YOUR_API_KEY), где *base\_id* - id модификации (базы), *api\_key*- Ваш ключ.

Например, для модели BMW X6 E71 50i AT (407 л.с.) xDrive ([https://developers.ria.com/auto/new/equip\\_base?base\\_id=9036&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/equip_base?base_id=9036&api_key=YOUR_API_KEY)) список комплектации будет следующим:

```
[
  {
    "year": "2011",
    "name": "base",
    "equip_id": 9933,
    "base_id": 9036
  },
]
```

(continues on next page)



(продолжение с предыдущей страницы)

```

{
  "year": "2012",
  "name": "base",
  "equip_id": 18668,
  "base_id": 9036
},
{
  "year": "2013",
  "name": "base",
  "equip_id": 26251,
  "base_id": 9036
},
{
  "year": "2014",
  "name": "base",
  "equip_id": 36709,
  "base_id": 9036
},
{
  "year": "2015",
  "name": "base",
  "equip_id": 58678,
  "base_id": 9036
},
{
  "year": "2016",
  "name": "base",
  "equip_id": 59806,
  "base_id": 9036
},
{
  "year": "2017",
  "name": "base",
  "equip_id": 79274,
  "base_id": 9036
}
]

```

### Модификации (базы) по типу кузова в поколении

Модификация (базы) зависит от типа кузова в поколении. Получить их список можно отправив GET запрос по адресу [https://developers.ria.com/auto/new/generation\\_bodystyles\\_bases?generation\\_bodystyle\\_id=id&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/generation_bodystyles_bases?generation_bodystyle_id=id&api_key=YOUR_API_KEY), где *generation\_bodystyle\_id* - типов кузова, *api\_key*- Ваш ключ.

Например, для модели BMW X6 в поколении E71 (рестайлинг) ([https://developers.ria.com/auto/new/generation\\_bodystyles\\_bases?generation\\_bodystyle\\_id=538&api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/auto/new/generation_bodystyles_bases?generation_bodystyle_id=538&api_key=YOUR_API_KEY)) список модификаций будет следующим:

```

[
  {
    "name": "50i AT (407 л.с.) xDrive",
    "model_id": 2153,
    "marka_id": 9,
    "generation_bodystyle_id": 538,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"base_id": 9036
},
{
  "name": "40d AT (306 л.с.) xDrive",
  "model_id": 2153,
  "marka_id": 9,
  "generation_bodystyle_id": 538,
  "base_id": 15821
}
]
```

## 1.2.2 Список всех объявлений

Чтобы получить список объявлений, Вам необходимо выполнить GET запрос такого вида:

```
https://developers.ria.com/auto/new/autos?user_id=Ваш id&api_key=YOUR_API_KEY
```

, где *user\_id* - Ваш ID в системе RIA.com, *api\_key* - Ваш ключ.

**Пример запроса** `https://developers.ria.com/auto/new/autos?user_id=4784009&api_key=YOUR_API_KEY`

**Пример успешного ответа:**

```
{
  "archive": [
    1540934,
    1540935,
    1540936,
    1540937,
    1603783,
    1618647,
    1618648,
    1618787,
    1619070,
    1619076,
    1619077,
    1619078,
    1619079,
    1619080,
    1621260,
    1621420,
    1623522,
    1623562,
    1623565,
    1623630,
    1627132,
    1628973,
    1628974,
    1628975,
    1628979,
    1628980,
    1628981,
    1629094,
    1629326,
    1630728,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
1630729,  
1630730,  
1630731,  
1630732,  
1630733,  
1630734,  
1630735,  
1630736,  
1630737,  
1630738,  
1630739,  
1630740,  
1630741,  
1630742,  
1630819,  
1631564,  
1632314,  
1633318,  
1636224,  
1636225,  
  
],  
"active": [  
  1696177  
]  
}
```

Расшифровка параметров

- *active* - Активные объявления
- *archive* - Объявления в архиве

### 1.2.3 Создание нового объявления

Для создания нового объявления, вам необходимо выполнить **POST** запрос такого вида:

```
curl -X POST "https://developers.ria.com/auto/new/autos?user_id=4784009&marka_id=id&model_id=id&  
base_id=id&equip_id=id&currency_id=id&price=цена&in_stock=id&test_drive=id&api_key=YOUR_API_KEY"  
-H "accept: application/json"
```

Обязательными параметрами являются:

- *user\_id* - Ваш ID в системе RIA.com
- *marka\_id* - Марка автомобиля
- *model\_id* - Модель автомобиля
- *base\_id* - Модификация ( базы ) по типу кузова
- *equip\_id* - Комплектация автомобиля
- *currency\_id* - Валюта в которой продаете автомобиль (1 - доллары США, 2 - евро, 3 - гривна)

Дополнительные параметры:

- *price* - Цена
- *auto\_note* - Описание

- *version* - Версия (пример 2.0 TDI)
- *in\_stock* - В наличии (1 - в наличии, 0 - нет в наличии)
- *test\_drive* - Возможность пройти тест драйв (1 - есть возможность, 0 - нету возможности)

#### Пример запроса

```
curl -X POST "https://developers.ria.com/auto/new/autos?user_id=4784009&marka_id=9&model_id=2153&base_id=9036&equip_id=79274&currency_id=1&price=50000&in_stock=1&test_drive=1&api_key=YOUR_API_KEY" -H "accept: application/json"
```

#### Пример успешного ответа:

```
{
  "num_rows": 1,
  "last_insert_id": 1696388
}
```

### 1.2.4 Редактирование объявления

Для редактирования основной информации объявления, вам необходимо отправить **PUT** запрос такого вида:

```
curl -X PUT "https://developers.ria.com/auto/new/autos?user_id=Ваш id&auto_id=id&currency_id=id&price=Цена&auto_note=Test&in_stock=id&test_drive=id&api_key=YOUR_API_KEY" -H "accept: application/json" -H "Content-Length: 0"
```

Обязательными параметрами являются:

- *user\_id* - Ваш ID в системе RIA.com
- *auto\_id* - ID нужного Вам объявления
- *currency\_id* - Валюта в которой продаете автомобиль (1 - доллары США, 2 - евро, 3 - гривна)
- *Content-Length* - размер возвращаемого документа (Content-Length: 0)

Дополнительные параметры:

- *price* - Цена
- *auto\_note* - Описание
- *version* - Версия (пример 2.0 TDI)
- *in\_stock* - В наличии (1 - в наличии, 0 - нет в наличии)
- *test\_drive* - Возможность пройти тест драйв (1 - есть возможность, 0 - нету возможности)

#### Пример запроса

```
curl -X PUT "https://developers.ria.com/auto/new/autos?user_id=4784009&auto_id=1696388&currency_id=2&price=55000&auto_note=Test&in_stock=0&test_drive=0&api_key=YOUR_API_KEY" -H "accept: application/json" -H "Content-Length: 0"
```

#### Пример успешного ответа:

```
{"num_rows":1,
"last_insert_id":0}
```

### 1.2.5 Удаление объявления

Для удаления объявления, Вам необходимо отправить **DELETE** запрос такого вида:

```
curl -X DELETE "https://developers.ria.com/auto/new/autos?user_id=id&auto_id=id&api_key=YOUR_API_KEY"
-H "accept: application/json"
```

, где *auto\_id* - ваш ID в системе RIA.com, *auto\_id* - id нужного Вам объявления, *api\_key* - ваш ключ.

#### Пример запроса

```
curl -X DELETE "https://developers.ria.com/auto/new/autos?user_id=4784009&auto_id=1696388&
api_key=YOUR_API_KEY" -H "accept: application/json"
```

#### Пример успешного ответа:

```
{"num_rows":1,
"last_insert_id":0}
```

### 1.2.6 Статистика объявления

Чтобы получить статистику об объявлении, Вам необходимо выполнить GET запрос такого вида:

```
https://developers.ria.com/auto/new/stat?user_id=Ваш ID&auto_id=ID объявления&date_from= дата
c&date_to=дата по&api_key=YOUR_API_KEY
```

или

```
curl -X GET "https://developers.ria.com/auto/new/stat?user_id=Ваш ID&auto_id=ID объявления&date_
↳from= фильтр по диапазону дат&date_to=фильтр по диапазону дат&api_key=YOUR_API_KEY" -H "accept:
↳application/json"
```

, где *user\_id* - Ваш ID в системе RIA.com, *auto\_id* - ID нужного Вам объявления, *date\_from*, *date\_to* - фильтр по диапазону дат (год - месяц - день), *api\_key* - Ваш ключ.

#### Пример запроса

```
curl -X GET "https://developers.ria.com/auto/new/stat?user_id=4784009&auto_id=1696177&date_
↳from=2018-03-06&date_to=2018-07-10&api_key=YOUR_API_KEY" -H "accept: application/json"
```

#### Пример успешного ответа:

```
{
  "viewsStatistics": [
    {
      "salon_id": 2407,
      "model_id": 1038,
      "marka_id": 13,
      "date": "2018-05-15",
      "count": 1,
      "auto_id": 1696177,
      "type": "views"
    },
    {
      "salon_id": 2407,
      "model_id": 1038,
      "marka_id": 13,
      "date": "2018-05-20",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"count": 105,
"auto_id": 1696177,
"type": "views"
},
{
"salon_id": 2407,
"model_id": 1038,
"marka_id": 13,
"date": "2018-03-09",
"count": 23,
"auto_id": 1696177,
"type": "views"
},
{
"salon_id": 2407,
"model_id": 1038,
"marka_id": 13,
"date": "2018-04-27",
"count": 1,
"auto_id": 1696177,
"type": "views"
},
{
"salon_id": 2407,
"model_id": 1038,
"marka_id": 13,
"date": "2018-03-08",
"count": 30,
"auto_id": 1696177,
"type": "views"
}
],
"phonesStatistics": [
{
"salon_id": 2407,
"model_id": 1038,
"marka_id": 13,
"date": "2018-04-15",
"count": 1,
"auto_id": 1696177,
"type": "auto_phones_final"
},
{
"salon_id": 2407,
"model_id": 1038,
"marka_id": 13,
"date": "2018-04-14",
"count": 1,
"auto_id": 1696177,
"type": "auto_phones_final"
},
{
"salon_id": 2407,
"model_id": 1038,
"marka_id": 13,
"date": "2018-06-03",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "count": 1,  
    "auto_id": 1696177,  
    "type": "auto_phones_final"  
  },  
  {  
    "salon_id": 2407,  
    "model_id": 1038,  
    "marka_id": 13,  
    "date": "2018-05-21",  
    "count": 21,  
    "auto_id": 1696177,  
    "type": "auto_phones_final"  
  }  
]
```

Расшифровка параметров:

- *viewsStatistics* - Количество просмотров объявления
- *phonesStatistics* - Количество открытий номера телефона
- *salon\_id* - id салона
- *model\_id* - id модели автомобиля
- *marka\_id* - id марки автомобиля
- *date* - дата
- *count* - количество
- *auto\_id* - id объявления
- *type* - «views» - Просмотры объявления
- *type* - «auto\_phones\_final» - Открытие номеров телефона

Полное описание сервиса «Статистика объявления» описанный с помощью стандарта **DeFacto swagger 2.0** [здесь](#)





### **Лимиты Веб-Сервисов**

Ограничения распространяются на количество запросов к веб-службам RIA.com, которые Вы можете создать, используя свой API ключ.

Часовой лимит: 1,000 запросов в час Эти ограничения применяются ко всем веб-службам RIA.com. Превышение этих лимитов приведет к временному блокированию Вашего API ключа. В зависимости от превышенного предела, блок будет автоматически отменен в течении следующего часа или календарного дня.

#### **Часовой Лимит**

Почасовые счетчики для API ключа аннулируются на постоянной основе.

Пример: Пример: Если Вы сделаете 500 запросов в 10:15 и 500 запросов в 10:25, Ваш API ключ будет временно заблокирован. Блок будет аннулирован в 11:15, после чего Вы сможете сделать еще 500 запросов.

#### **Ограничение Запросов**

Если Ваш API ключ превышает лимиты запросов, Вы получите ответ с HTTP статус-кодом 429 (слишком много запросов).

#### **Нужно повысить лимиты?**

Если Вы создаете приложение, которое требует более высоких лимитов, мы будем рады сотрудничать. Свяжитесь с нами для более подробной информации.

## 2.1 Параметры для работы с DOM.RIA API

### 2.1.1 Продажа квартир/комнат

### 2.1.2 Аренда квартир/комнат

### 2.1.3 Продажа домов

### 2.1.4 Аренда домов

### 2.1.5 Продажа офисов

### 2.1.6 Аренда офисов

### 2.1.7 Продажа/аренда коммерческой недвижимости

### 2.1.8 Продажа/аренда земельных участков

### 2.1.9 Продажа/аренда гаражей

## 2.2 Список характеристик недвижимости

Получить список областей можно отправив GET запрос по адресу `https://developers.ria.com/dom/options?category=id&realty_type=id&operation_type=id&api_key=YOUR_API_KEY`, где

- *category* - тип объекта
- *realty\_type* - тип недвижимости
- *operation\_type* - тип операции

или

```
curl -X GET "https://developers.ria.com/dom/options?category=id&realty_type=id&operation_type=id&api_key=YOUR_API_KEY" -H "accept: application/json"
```

Для получение информации на украинском языке нужно добавить параметр *lang\_id=4*

Полное описание сервиса «Список характеристик недвижимости» описанный с помощью стандарта DeFacto swagger 2.0 [здесь](#)

### Пример

Допусти Вам нужно получить список характеристик объекта „Дома“, а тип недвижимости - „Дачи“, тип операции - „Продажа“

Данный запрос будет выглядеть так:

```
https://developers.ria.com/dom/options?category=4&realty_type=7&operation_type=1&api_key=YOUR_API_KEY
```

или

```
curl -X GET "https://developers.ria.com/dom/options?category=4&realty_type=7&operation_type=1&api_key=YOUR_API_KEY" -H "accept: application/json"
```

Результат будет следующим:

```
[
  {
    "group_name": "комнаты",
    "group_prio": 150,
    "items": [
      {
        "characteristic_id": 209,
        "label": "комнат",
        "label_uk": "кімнат",
        "type_on_add": "text",
        "type_on_search": "text",
        "data_type": "int",
        "required": 1,
        "characteristic_category": "main",
        "prio": 150,
        "name": "Комнат",
        "group_name": "комнаты",
        "group_name_uk": "кімнати",
        "field_name": "rooms_count",
        "display_label_search": 0,
        "display_label_add": 0,
        "group_orientation_search": "horizontal",
        "group_orientation_add": "horizontal",
        "empty_value": 0,
        "type": "characteristic",
        "group_prio": 0
      }
    ]
  },
  {
    "group_name": "Тип стен",
    "group_prio": 100,
    "items": [
      {
        "characteristic_id": 149,
        "label": "тип стен",
        "label_uk": "тип стін",
        "type_on_add": "select",
        "type_on_search": "checkbox",
        "data_type": "int",
        "required": 1,
        "characteristic_category": "main",
        "prio": 100,
        "name": "тип стен для дома",
        "group_name": "Тип стен",
        "group_name_uk": "Тип стін",
        "children": {
          "129": {
            "characteristic_id": "129",
            "name": "кирпич",
            "name_uk": "цегла",
            "prio": "200"
          }
        }
      }
    ]
  }
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
},
"130": {
  "characteristic_id": "130",
  "name": "кирпич силикатный",
  "name_uk": "цегла силікатна",
  "prio": "190"
},
"131": {
  "characteristic_id": "131",
  "name": "кирпич саманный",
  "name_uk": "цегла саманна",
  "prio": "180"
},
"132": {
  "characteristic_id": "132",
  "name": "дерево и кирпич",
  "name_uk": "дерево та цегла",
  "prio": "170"
},
"133": {
  "characteristic_id": "133",
  "name": "панель",
  "name_uk": "панель",
  "prio": "160"
},
"134": {
  "characteristic_id": "134",
  "name": "пеноблок",
  "name_uk": "пеноблок",
  "prio": "150"
},
"135": {
  "characteristic_id": "135",
  "name": "керамзитобетон",
  "name_uk": "керамзитобетон",
  "prio": "140"
},
"136": {
  "characteristic_id": "136",
  "name": "монолит",
  "name_uk": "моноліт",
  "prio": "130"
},
"138": {
  "characteristic_id": "138",
  "name": "сруб",
  "name_uk": "зруб",
  "prio": "110"
},
"139": {
  "characteristic_id": "139",
  "name": "брус",
  "name_uk": "брус",
  "prio": "100"
},
"140": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"characteristic_id": "140",
"name": "каркасно-щитовой",
"name_uk": "каркасно-щитовий",
"prio": "90"
},
"141": {
  "characteristic_id": "141",
  "name": "глинобитный",
  "name_uk": "глинобитний",
  "prio": "80"
},
"142": {
  "characteristic_id": "142",
  "name": "пенобетон",
  "name_uk": "пінобетон",
  "prio": "70"
},
"143": {
  "characteristic_id": "143",
  "name": "газобетон",
  "name_uk": "газобетон",
  "prio": "60"
},
"144": {
  "characteristic_id": "144",
  "name": "сэндвич-панели",
  "name_uk": "сєндвіч-панєлі",
  "prio": "50"
},
"145": {
  "characteristic_id": "145",
  "name": "метал",
  "name_uk": "метал",
  "prio": "40"
},
"146": {
  "characteristic_id": "146",
  "name": "поротерм",
  "name_uk": "поротерм",
  "prio": "30"
},
"147": {
  "characteristic_id": "147",
  "name": "ракушечник (ракушняк)",
  "name_uk": "ракушняк",
  "prio": "20"
},
"148": {
  "characteristic_id": "148",
  "name": "инкерманский камень",
  "name_uk": "інкерманський камінь",
  "prio": "10"
},
"1433": {
  "characteristic_id": "1433",
  "name": "шлакобетон",
```

(continues on next page)

```
    "name_uk": "шлакобетон",
    "prio": "65"
  },
  "1441": {
    "characteristic_id": "1441",
    "name": "шлакоблок",
    "name_uk": "шлакоблок",
    "prio": "55"
  },
  "1442": {
    "characteristic_id": "1442",
    "name": "наливной",
    "name_uk": "наливний",
    "prio": "5"
  },
  "1444": {
    "characteristic_id": "1444",
    "name": "бутовый камень",
    "name_uk": "бутовий камінь",
    "prio": "4"
  },
  "1445": {
    "characteristic_id": "1445",
    "name": "мергель",
    "name_uk": "мергель",
    "prio": "3"
  },
  "1447": {
    "characteristic_id": "1447",
    "name": "крупноблочный известняк",
    "name_uk": "великоблочний вапняк",
    "prio": "2"
  },
  "1463": {
    "characteristic_id": "1463",
    "name": "термоблок",
    "name_uk": "термоблок",
    "prio": "1"
  },
  "1597": {
    "characteristic_id": "1597",
    "name": "сип панель ",
    "name_uk": "сіп панель",
    "prio": "56"
  },
  "1598": {
    "characteristic_id": "1598",
    "name": "контейнер ",
    "name_uk": "контейнер",
    "prio": "7"
  }
},
"field_name": "wall_type",
"display_label_search": 0,
"display_label_add": 0,
"group_orientation_search": "horisontal",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "group_orientation_add": "horisontal",
    "empty_value": 0,
    "type": "characteristic",
    "group_prio": 0
  }
]
},
{
  "group_name": "площадь помещений",
  "group_prio": 90,
  "items": [
    {
      "characteristic_id": 215,
      "label": "общая пл.",
      "label_uk": "загальна пл.",
      "type_on_add": "text",
      "type_on_search": "text",
      "data_type": "float",
      "required": 1,
      "sufix": "м.кв.",
      "characteristic_category": "main",
      "prio": 4,
      "name": "площадь дома",
      "group_name": "площадь помещений",
      "group_name_uk": "площа приміщень",
      "group_prio": 90,
      "field_name": "total_square_meters",
      "display_label_search": 0,
      "display_label_add": 0,
      "group_orientation_search": "horisontal",
      "group_orientation_add": "horisontal",
      "empty_value": 0,
      "type": "characteristic"
    },
    {
      "characteristic_id": 216,
      "label": "жилая",
      "label_uk": "житлова",
      "type_on_add": "text",
      "type_on_search": "text",
      "data_type": "float",
      "required": 0,
      "sufix": "кв. м",
      "characteristic_category": "main",
      "prio": 3,
      "name": "жилая площадь",
      "group_name": "площадь помещений",
      "group_name_uk": "площа приміщень",
      "group_prio": 90,
      "field_name": "living_square_meters",
      "display_label_search": 0,
      "display_label_add": 0,
      "group_orientation_search": "horisontal",
      "group_orientation_add": "horisontal",
      "empty_value": 0,
      "type": "characteristic"
    }
  ]
}

```

(continues on next page)

```
    },
    {
      "characteristic_id": 218,
      "label": "кухня",
      "label_uk": "кухня",
      "type_on_add": "text",
      "type_on_search": "text",
      "data_type": "float",
      "required": 0,
      "sufix": "кв. м",
      "characteristic_category": "main",
      "prio": 2,
      "name": "кухня",
      "group_name": "площадь помещений",
      "group_name_uk": "площа приміщень",
      "group_prio": 90,
      "field_name": "kitchen_square_meters",
      "display_label_search": 0,
      "display_label_add": 0,
      "group_orientation_search": "horisontal",
      "group_orientation_add": "horisontal",
      "empty_value": 0,
      "type": "characteristic"
    }
  ]
},
{
  "group_name": "площадь участка",
  "group_prio": 83,
  "items": [
    {
      "characteristic_id": 219,
      "label": "участок",
      "label_uk": "ділянка",
      "type_on_add": "text",
      "type_on_search": "text",
      "data_type": "float",
      "required": 0,
      "characteristic_category": "main",
      "prio": 10,
      "name": "площадь участка",
      "group_name": "площадь участка",
      "group_name_uk": "площа ділянки",
      "group_prio": 85,
      "field_name": "ares_count",
      "display_label_search": 0,
      "display_label_add": 0,
      "group_orientation_search": "horisontal",
      "group_orientation_add": "horisontal",
      "empty_value": 0,
      "type": "characteristic"
    },
    {
      "characteristic_id": 226,
      "label": "единица измерения",
      "label_uk": "одиниця виміру",
```

(continues on next page)



(продолжение с предыдущей страницы)

```

"type_on_add": "select",
"type_on_search": "select",
"data_type": "int",
"required": 0,
"characteristic_category": "main",
"prio": 2,
"name": "единица измерения площади участка",
"group_name": "площадь участка",
"group_name_uk": "площа ділянки",
"group_prio": 83,
"children": {
  "223": {
    "characteristic_id": "223",
    "name": "сотка",
    "name_uk": "сотка",
    "prio": "3"
  },
  "224": {
    "characteristic_id": "224",
    "name": "Га (гектар)",
    "name_uk": "Га (гектар)",
    "prio": "2"
  },
  "225": {
    "characteristic_id": "225",
    "name": "кв. м",
    "name_uk": "кв. м",
    "prio": "1"
  }
},
"field_name": "lot_unit",
"display_label_search": 0,
"display_label_add": 0,
"group_orientation_search": "horizontal",
"group_orientation_add": "horizontal",
"empty_value": 0,
"type": "characteristic"
}
]
},
{
  "group_name": "этаж",
  "group_prio": 75,
  "items": [
    {
      "characteristic_id": 1494,
      "label": "у озера",
      "label_uk": "біля озера",
      "type_on_add": "checkbox",
      "type_on_search": "checkbox",
      "data_type": "int",
      "required": 0,
      "characteristic_category": "main",
      "prio": 4,
      "name": "у озера",
      "group_name": "этаж",

```

(continues on next page)

```
"group_name_uk": "поверх",
"group_prio": 75,
"display_label_search": 1,
"display_label_add": 1,
"group_orientation_search": "horisontal",
"group_orientation_add": "horisontal",
"empty_value": 0,
"type": "characteristic"
},
{
  "characteristic_id": 1493,
  "label": "у реки",
  "label_uk": "біля річки",
  "type_on_add": "checkbox",
  "type_on_search": "checkbox",
  "data_type": "int",
  "required": 0,
  "characteristic_category": "main",
  "prio": 3,
  "name": "у реки",
  "group_name": "этаж",
  "group_name_uk": "поверх",
  "group_prio": 75,
  "display_label_search": 1,
  "display_label_add": 1,
  "group_orientation_search": "horisontal",
  "group_orientation_add": "horisontal",
  "empty_value": 0,
  "type": "characteristic"
},
{
  "characteristic_id": 230,
  "label": "мансардный этаж",
  "label_uk": "мансардний поверх",
  "type_on_add": "checkbox",
  "type_on_search": "checkbox",
  "data_type": "int",
  "required": 0,
  "characteristic_category": "main",
  "prio": 2,
  "name": "мансардный этаж",
  "group_name": "этаж",
  "group_name_uk": "поверх",
  "group_prio": 75,
  "display_label_search": 0,
  "display_label_add": 0,
  "group_orientation_search": "horisontal",
  "group_orientation_add": "horisontal",
  "empty_value": 0,
  "type": "characteristic"
},
{
  "characteristic_id": 231,
  "label": "подвальный / цокольный этаж",
  "label_uk": "підвальний / цокольний поверх",
  "type_on_add": "checkbox",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "type_on_search": "checkbox",
    "data_type": "int",
    "required": 0,
    "characteristic_category": "main",
    "prio": 1,
    "name": "подвальный / цокольный этаж",
    "group_name": "этаж",
    "group_name_uk": "поверх",
    "group_prio": 75,
    "display_label_search": 0,
    "display_label_add": 0,
    "group_orientation_search": "horizontal",
    "group_orientation_add": "horizontal",
    "empty_value": 0,
    "type": "characteristic"
  }
]
},
{
  "group_name": "характеристика здания",
  "group_prio": 70,
  "items": [
    {
      "characteristic_id": 443,
      "label": "год постройки",
      "label_uk": "рік побудови",
      "type_on_add": "select",
      "type_on_search": "checkbox",
      "data_type": "int",
      "required": 0,
      "default_value": "не указано",
      "characteristic_category": "main",
      "prio": 160,
      "name": "год постройки",
      "group_name": "характеристика здания",
      "group_name_uk": "характеристика будівлі",
      "group_prio": 70,
      "children": {
        "421": {
          "characteristic_id": "421",
          "name": "2015, IV кв.",
          "name_uk": "2015, IV кв.",
          "prio": "25"
        },
        "422": {
          "characteristic_id": "422",
          "name": "2013",
          "name_uk": "2013",
          "prio": "20"
        },
        "423": {
          "characteristic_id": "423",
          "name": "2012",
          "name_uk": "2012",
          "prio": "19"
        }
      }
    },
  ]
},

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"424": {
  "characteristic_id": "424",
  "name": "2011 ",
  "name_uk": "2011",
  "prio": "18"
},
"425": {
  "characteristic_id": "425",
  "name": "2010",
  "name_uk": "2010",
  "prio": "17"
},
"426": {
  "characteristic_id": "426",
  "name": "2009",
  "name_uk": "2009",
  "prio": "16"
},
"427": {
  "characteristic_id": "427",
  "name": "2008",
  "name_uk": "2008",
  "prio": "15"
},
"428": {
  "characteristic_id": "428",
  "name": "2007",
  "name_uk": "2007",
  "prio": "14"
},
"429": {
  "characteristic_id": "429",
  "name": "2006",
  "name_uk": "2006",
  "prio": "13"
},
"434": {
  "characteristic_id": "434",
  "name": "2001-2005",
  "name_uk": "2001-2005",
  "prio": "8"
},
"435": {
  "characteristic_id": "435",
  "name": "1990-2000",
  "name_uk": "1990-2000",
  "prio": "7"
},
"436": {
  "characteristic_id": "436",
  "name": "1980-1989",
  "name_uk": "1980-1989",
  "prio": "6"
},
"437": {
  "characteristic_id": "437",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"name": "1970-1979",
"name_uk": "1970-1979",
"prio": "5"
},
"438": {
  "characteristic_id": "438",
  "name": "1960-1969",
  "name_uk": "1960-1969",
  "prio": "4"
},
"439": {
  "characteristic_id": "439",
  "name": "1944-1959",
  "name_uk": "1944-1959",
  "prio": "3"
},
"440": {
  "characteristic_id": "440",
  "name": "1917-1942",
  "name_uk": "1917-1942",
  "prio": "2"
},
"441": {
  "characteristic_id": "441",
  "name": "раньше 1917",
  "name_uk": "до 1917",
  "prio": "1"
},
"442": {
  "characteristic_id": "442",
  "name": "не указано",
  "name_uk": "не вказано",
  "prio": "36"
},
"1448": {
  "characteristic_id": "1448",
  "name": "2014",
  "name_uk": "2014",
  "prio": "21"
},
"1449": {
  "characteristic_id": "1449",
  "name": "Сдача в 2015",
  "name_uk": "Здача в 2015",
  "prio": "30"
},
"1450": {
  "characteristic_id": "1450",
  "name": "Сдача в 2016",
  "name_uk": "Здача в 2016",
  "prio": "31"
},
"1454": {
  "characteristic_id": "1454",
  "name": "2015, III кв.",
  "name_uk": "2015, III кв.",
```

(continues on next page)

```
"prio": "24"
},
"1455": {
  "characteristic_id": "1455",
  "name": "2015, II кв.",
  "name_uk": "2015, II кв.",
  "prio": "23"
},
"1456": {
  "characteristic_id": "1456",
  "name": "2015, I кв.",
  "name_uk": "2015, I кв.",
  "prio": "22"
},
"1457": {
  "characteristic_id": "1457",
  "name": "2016, I кв.",
  "name_uk": "2016, I кв.",
  "prio": "26"
},
"1458": {
  "characteristic_id": "1458",
  "name": "2016, II кв.",
  "name_uk": "2016, II кв.",
  "prio": "27"
},
"1459": {
  "characteristic_id": "1459",
  "name": "2016, III кв.",
  "name_uk": "2016, III кв.",
  "prio": "28"
},
"1460": {
  "characteristic_id": "1460",
  "name": "2016, IV кв.",
  "name_uk": "2016, IV кв.",
  "prio": "29"
},
"1468": {
  "characteristic_id": "1468",
  "name": "Сдача в 2017",
  "name_uk": "Здача в 2017",
  "prio": "32"
},
"1469": {
  "characteristic_id": "1469",
  "name": "Сдача в 2018",
  "name_uk": "Здача в 2018",
  "prio": "33"
},
"1470": {
  "characteristic_id": "1470",
  "name": "Сдача в 2019",
  "name_uk": "Здача в 2019",
  "prio": "34"
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "1471": {
      "characteristic_id": "1471",
      "name": "Сдача в 2020",
      "name_uk": "Здача в 2020",
      "prio": "35"
    }
  },
  "display_label_search": 0,
  "display_label_add": 0,
  "group_orientation_search": "horisontal",
  "group_orientation_add": "horisontal",
  "empty_value": 0,
  "type": "characteristic"
}
]
},
{
  "group_name": "цена",
  "group_prio": 45,
  "items": [
    {
      "characteristic_id": 234,
      "label": "цена",
      "label_uk": "ціна",
      "type_on_add": "text",
      "type_on_search": "text",
      "data_type": "int",
      "required": 1,
      "characteristic_category": "main",
      "prio": 96,
      "name": "цена",
      "group_name": "цена",
      "group_name_uk": "ціна",
      "group_prio": 64,
      "field_name": "price",
      "display_label_search": 0,
      "display_label_add": 0,
      "group_orientation_search": "horisontal",
      "group_orientation_add": "horisontal",
      "empty_value": 0,
      "type": "characteristic"
    },
    {
      "characteristic_id": 1011,
      "label": "цена договорная",
      "label_uk": "ціна договірна",
      "type_on_add": "checkbox",
      "type_on_search": "checkbox",
      "data_type": "int",
      "required": 0,
      "characteristic_category": "main",
      "prio": 95,
      "name": "цена договорная",
      "group_name": "цена",
      "group_name_uk": "ціна",
      "group_prio": 64,

```

(continues on next page)

```
"field_name": "is_negotiable",
"display_label_search": 0,
"display_label_add": 0,
"group_orientation_search": "horisontal",
"group_orientation_add": "horisontal",
"empty_value": 0,
"type": "characteristic"
},
{
  "characteristic_id": 242,
  "label": "тип валюты",
  "label_uk": "тип валюты",
  "type_on_add": "select",
  "type_on_search": "select",
  "data_type": "int",
  "required": 1,
  "default_value": "$",
  "characteristic_category": "main",
  "prio": 85,
  "name": "тип валюты (USD)",
  "group_name": "цена",
  "group_name_uk": "ціна",
  "group_prio": 64,
  "children": {
    "239": {
      "characteristic_id": "239",
      "name": "$",
      "name_uk": "$",
      "prio": "3"
    },
    "240": {
      "characteristic_id": "240",
      "name": "грн",
      "name_uk": "грн",
      "prio": "2"
    },
    "241": {
      "characteristic_id": "241",
      "name": "€",
      "name_uk": "€",
      "prio": "1"
    }
  }
},
"field_name": "currency_type",
"display_label_search": 0,
"display_label_add": 0,
"group_orientation_search": "horisontal",
"group_orientation_add": "horisontal",
"empty_value": 0,
"type": "characteristic"
},
{
  "characteristic_id": 1464,
  "label": "стартовая цена",
  "label_uk": "стартова ціна",
  "type_on_add": "checkbox",
```

(continues on next page)



(продолжение с предыдущей страницы)

```

    "type_on_search": "checkbox",
    "data_type": "int",
    "required": 0,
    "characteristic_category": "main",
    "prio": 45,
    "name": "стартовая цена",
    "group_name": "цена",
    "group_name_uk": "ціна",
    "field_name": "start_price",
    "display_label_search": 0,
    "display_label_add": 0,
    "group_orientation_search": "horizontal",
    "group_orientation_add": "horizontal",
    "empty_value": 0,
    "type": "characteristic",
    "group_prio": 0
  },
  {
    "characteristic_id": 273,
    "label": "возможен торг",
    "label_uk": "можливий торг",
    "type_on_add": "checkbox",
    "type_on_search": "checkbox",
    "data_type": "int",
    "required": 0,
    "characteristic_category": "main",
    "prio": 42,
    "name": "возможен торг",
    "group_name": "цена",
    "group_name_uk": "ціна",
    "field_name": "is_bargain",
    "display_label_search": 0,
    "display_label_add": 0,
    "group_orientation_search": "horizontal",
    "group_orientation_add": "horizontal",
    "empty_value": 0,
    "type": "characteristic",
    "group_prio": 0
  },
  {
    "characteristic_id": 274,
    "label": "возможна рассрочка / кредит",
    "label_uk": "можлива розстрочка / кредит",
    "type_on_add": "checkbox",
    "type_on_search": "checkbox",
    "data_type": "int",
    "required": 0,
    "characteristic_category": "main",
    "prio": 40,
    "name": "возможна рассрочка / кредит",
    "group_name": "цена",
    "group_name_uk": "ціна",
    "display_label_search": 0,
    "display_label_add": 0,
    "group_orientation_search": "horizontal",
    "group_orientation_add": "horizontal",

```

(continues on next page)

```
"empty_value": 0,
"type": "characteristic",
"group_prio": 0
},
{
  "characteristic_id": 265,
  "label": "возможен обмен",
  "label_uk": "можливий обмін",
  "type_on_add": "select",
  "type_on_search": "select",
  "data_type": "int",
  "required": 0,
  "default_value": "нет",
  "characteristic_category": "main",
  "prio": 35,
  "name": "возможен обмен",
  "group_name": "цена",
  "group_name_uk": "ціна",
  "children": {
    "257": {
      "characteristic_id": "257",
      "name": "нет",
      "name_uk": "ні",
      "prio": "8"
    },
    "258": {
      "characteristic_id": "258",
      "name": "на авто + моя доплата",
      "name_uk": "на авто + моя доплата",
      "prio": "7"
    },
    "259": {
      "characteristic_id": "259",
      "name": "на авто без доплаты",
      "name_uk": "на авто без доплати",
      "prio": "6"
    },
    "260": {
      "characteristic_id": "260",
      "name": "на авто + ваша доплата",
      "name_uk": "на авто + ваша доплата",
      "prio": "5"
    },
    "261": {
      "characteristic_id": "261",
      "name": "на недвижимость + моя доплата",
      "name_uk": "на нерухомість + моя доплата",
      "prio": "4"
    },
    "262": {
      "characteristic_id": "262",
      "name": "на недвижимость без доплаты",
      "name_uk": "на нерухомість без доплати",
      "prio": "3"
    },
    "263": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "characteristic_id": "263",
    "name": "на недвижимость + ваша доплата",
    "name_uk": "на нерухомість + ваша доплата",
    "prio": "2"
  },
  "264": {
    "characteristic_id": "264",
    "name": "рассмотрю любой вариант",
    "name_uk": "розгляну будь-який варіант",
    "prio": "1"
  }
},
"field_name": "is_exchange",
"display_label_search": 0,
"display_label_add": 0,
"group_orientation_search": "horisontal",
"group_orientation_add": "horisontal",
"empty_value": 0,
"type": "characteristic",
"group_prio": 0
},
{
  "characteristic_id": 1437,
  "label": "тип предложения",
  "label_uk": "тип пропозиції",
  "type_on_add": "select",
  "type_on_search": "checkbox",
  "data_type": "int",
  "required": 0,
  "default_value": "от посредника",
  "characteristic_category": "main",
  "prio": 30,
  "name": "тип предложения",
  "group_name": "цена",
  "group_name_uk": "ціна",
  "group_prio": 64,
  "children": {
    "1434": {
      "characteristic_id": "1434",
      "name": "от посредника",
      "name_uk": "від посередника",
      "prio": "5"
    },
    "1435": {
      "characteristic_id": "1435",
      "name": "от представителя хозяина (без комиссионных)",
      "name_uk": "від представника власника (без комісійних)",
      "prio": "4"
    },
    "1436": {
      "characteristic_id": "1436",
      "name": "от собственника",
      "name_uk": "від власника",
      "prio": "2"
    },
    "1473": {

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "characteristic_id": "1473",
    "name": "от представителя застройщика",
    "name_uk": "від представника забудовника",
    "prio": "3"
  },
  "1506": {
    "characteristic_id": "1506",
    "name": "от застройщика",
    "name_uk": "від забудовника",
    "prio": "1"
  }
},
"display_label_search": 1,
"display_label_add": 1,
"group_orientation_search": "horizontal",
"group_orientation_add": "horizontal",
"empty_value": 0,
"type": "characteristic"
}
]
}
```

## 2.3 Информация по id объявления

Чтобы обратиться к RIA API info, Вам необходимо выполнить GET запрос такого вида:

```
https://developers.ria.com/dom/info/id_объявления?api_key=YOUR_API_KEY
```

или

```
curl -i -X GET "https://developers.ria.com/dom/info/id_объявления?api_key=YOUR_API_KEY"
```

Полное описание сервиса «Информация по id объявления» описанный с помощью стандарта DeFacto swagger 2.0 [здесь](#)

Допустим Вам нужна информация по *13825265* id.

Данный запрос будет выглядеть так:

```
https://developers.ria.com/dom/info/13825265?api_key=YOUR_API_KEY
```

или

```
curl -i -X GET "https://developers.ria.com/dom/info/13825265?api_key=YOUR_API_KEY"
```

В случае успешного выполнения запроса результат будет со статусом **200 ОК**.

Пример успешного ответа:

```
{
  "street_name": "Онуфрия Трутенко улица",
  "metro_station_id": "15",
  "rooms_count": 2,
  "advert_title": 1,
  "type": "realty",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"is_commercial": 0,
"state_name": "Киевская",
"street_id": 5581,
"state_id": 10,
"beautiful_url": "realty-perevireno-prodaja-kvartira-kiev-goloseevskiy-onufriya-trutenko-ulitsa-
↪13825265.html",
"levels": 64,
"exchange_required_type": 13,
"district_type_name": "Район",
"description": "Светлая квартира, расположена на 23 этаже / 23 эт. дома в ЖК &quot; 4 Сезона&
↪quot;; потолок Н-3,5 м., панорамные окна с видом на город. Состояние от застройщика, у Вас есть
↪возможность воплотить дизайнерские фантазии в доме. Дом имеет внешнее утепление, парковку,
↪подземный паркинг, во дворе оборудованная детская площадка. Шикарный район, в шаговой дистанции
↪м. &quot; Васильковская&quot;; ( 5 мин.пешком), м. &quot;Ипподром&quot;; (10 мин.пешком), м. &quot;
↪Выставочный Центр&quot;;(18 мин. пешком)  Голосеевский парк,озёра и пруды, детский сад, школа,
↪магазины, салоны, кафе и рестораны, фитнес центр. ",
"advert_publish_type": 1,
"currency_type": "$",
"district_type_id": 1,
"longitude": "30.480281005166944",
"inspected_at": "2017-10-02 14:51:02",
"levels_expired": "2018-01-01 12:16:32",
"is_exchange": "рассмотрю любой вариант",
"district_id": "15184",
"metro_station_name": "Васильковская",
"wall_type": "кирпич",
"publishing_date": "2017-10-02 14:51:07",
"description_uk": "Світла квартира, розташована на 23 поверсі / 23 пов. будинки в ЖК &quot;4
↪Сезони&quot;; стеля Н-3,5 м., панорамні вікна з видом на місто. Стан від забудовника, у Вас є
↪можливість втілити дизайнерські фантазії в будинку. Будинок має зовнішнє утеплення, місце для
↪паркування, підземний паркінг, у дворі обладнаний дитячий майданчик. Шикарний район, в кроковій
↪дистанції м. &quot;Васильківська&quot;; ( 5 хв. пішки), м. &quot;Іпподром&quot;; (10 хв. пішки), м.
↪&quot;Вистовковий Центр&quot;;(18 хв. пішки), Голосіївський парк, озера і ставки, дитячий садок,
↪школа, магазини, салони, кафе і ресторани, фітнес-центр. ",
"youtube_link": "",
"price_type": "за объект",
"inspected": 1,
"is_bargain": 273,
"price": 60000,
"floor": 23,
"latitude": "50.39161687881484",
"price_total": 60000,
"main_photo": "dom/photo/8070/807045/80704539/80704539.jpg",
"price_item": 645,
"realty_type_name": "Квартира",
"building_number_str": "3 Г",
"city_name": "Киев",
"metro_station_brunch": 2,
"realty_type_parent_id": 1,
"living_square_meters": 50,
"realty_type_id": 2,
"user_ip": 0,
"city_id": 10,
"characteristics_values": {
  "118": 108,
  "209": 2,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"214": 93,
"216": 50,
"218": 20,
"227": 23,
"228": 23,
"234": 60000,
"242": 239,
"247": 252,
"265": 264,
"273": 273,
"443": 1449,
"475": 473,
"480": 476,
"516": 512,
"791": 791,
"806": 802,
"808": 808,
"892": 890,
"1016": 1016,
"1020": 1023,
"1032": 1028,
"1157": 1149,
"1403": 1400,
"1437": 1436,
"1504": 1504,
"1571": 1571,
"1575": 1575,
"1581": 1581
},
"_id": "realty-13825265",
"floors_count": 23,
"created_at": "2017-09-20 21:12:42",
"user_id": 7160632,
"kitchen_square_meters": 20,
"user_package_id": 0,
"total_square_meters": 93,
"realty_id": 13825265,
"realty_sale_type": 2,
"photos": {
  "80704539": {
    "id": 80704539,
    "file": "dom/photo/8070/807045/80704539/80704539.jpg",
    "o": "h"
  },
  "80704540": {
    "id": 80704540,
    "file": "dom/photo/8070/807045/80704540/80704540.jpg",
    "o": "h"
  },
  "80704541": {
    "id": 80704541,
    "file": "dom/photo/8070/807045/80704541/80704541.jpg",
    "o": "h"
  },
  "80704542": {
    "id": 80704542,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "file": "dom/photo/8070/807045/80704542/80704542.jpg",
    "o": "h"
  },
  "80704543": {
    "id": 80704543,
    "file": "dom/photo/8070/807045/80704543/80704543.jpg",
    "o": "v"
  },
  "80704545": {
    "id": 80704545,
    "file": "dom/photo/8070/807045/80704545/80704545.jpg",
    "o": "v"
  },
  "80704546": {
    "id": 80704546,
    "file": "dom/photo/8070/807045/80704546/80704546.jpg",
    "o": "v"
  },
  "80704547": {
    "id": 80704547,
    "file": "dom/photo/8070/807045/80704547/80704547.jpg",
    "o": "v"
  },
  "80704548": {
    "id": 80704548,
    "file": "dom/photo/8070/807045/80704548/80704548.jpg",
    "o": "v"
  },
  "80704550": {
    "id": 80704550,
    "file": "dom/photo/8070/807045/80704550/80704550.jpg",
    "o": "h"
  },
  "80704552": {
    "id": 80704552,
    "file": "dom/photo/8070/807045/80704552/80704552.jpg",
    "o": "h"
  }
},
"date_end": "2018-03-02 14:51:07",
"district_name": "Голосеевский",
"advert_type_id": 1,
"advert_type_name": "продажа",
"is_show_building_no": 1,
"realty_type_parent_name": "Квартиры",
"user": {
  "name": "Владимир",
  "image": "",
  "good_partner_top": ""
},
"priceArr": {
  "1": "60 000",
  "2": "50 588",
  "3": "1 680 672"
},
"with_p panoramas": 1
}
```

## 2.4 Подсчёт средней цены

“Подсчёт средней цены” — сервис, основанный на актуальных статистических данных DOM.RIA

Теперь вы можете:

- узнавать актуальные средние цены на недвижимость;
- следить за изменениями цен в кратко- и долгосрочном периодах;
- анализировать и прогнозировать изменения цен и спроса на недвижимость;
- размещать полученную информацию на вашем сайте.

Ознакомьтесь с технической документацией, чтобы получить доступ и экспортировать необходимую информацию в программу вашей компании.

С полным описанием сервиса «Подсчёт средней цены» можно ознакомиться с помощью стандарта DeFacto **swagger 2.0** [здесь](#)

В случае успешного подсчета средней цены по указанным параметрам результат будет со статусом **200 ОК**.

### 2.4.1 Формат данных в запросе

Параметры **category**, **sub\_category**, **operation** являются обязательным для заполнения.

Детальную информацию про применение этих параметров можно получить с помощью стандарта DeFacto **swagger 2.0** [здесь](#)

Пример успешного запроса:

```
https://developers.ria.com/dom/average_price?category=1&sub_category=2&operation=1&state_id=1&
city_id=1&date_from=2017-06&date_to=2017-07&api_key=YOUR_API_KEY
```

или

```
curl -X GET "https://developers.ria.com/dom/average_price?category=1&sub_category=2&
operation=1&state_id=1&city_id=1&date_from=2017-06&date_to=2017-07&api_key=YOUR_API_KEY" -H
"accept: application/json"
```

Пример успешного ответа:

```
{
  "total": 12407,
  "arithmeticMean": 785340.3,
  "prices": [
    {
      "count": 229,
      "value": 60000
    },
    {
      "count": 234,
      "value": 160000
    },
    {
      "count": 239,
      "value": 260000
    },
  ],
}
```

(continues on next page)



(продолжение с предыдущей страницы)

```
{
  "count": 243,
  "value": 360000
},
{
  "count": 247,
  "value": 460000
},
{
  "count": 250,
  "value": 560000
},
{
  "count": 252,
  "value": 660000
},
{
  "count": 254,
  "value": 760000
},
{
  "count": 255,
  "value": 860000
},
{
  "count": 255,
  "value": 960000
},
{
  "count": 255,
  "value": 1060000
},
{
  "count": 254,
  "value": 1160000
},
{
  "count": 252,
  "value": 1260000
},
{
  "count": 250,
  "value": 1360000
},
{
  "count": 247,
  "value": 1460000
},
{
  "count": 243,
  "value": 1560000
},
{
  "count": 239,
  "value": 1660000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "count": 234,
  "value": 1760000
},
{
  "count": 229,
  "value": 1860000
},
{
  "count": 224,
  "value": 1960000
},
{
  "count": 217,
  "value": 2060000
},
{
  "count": 211,
  "value": 2160000
},
{
  "count": 204,
  "value": 2260000
},
{
  "count": 197,
  "value": 2360000
},
{
  "count": 189,
  "value": 2460000
},
{
  "count": 182,
  "value": 2560000
},
{
  "count": 174,
  "value": 2660000
},
{
  "count": 166,
  "value": 2760000
},
{
  "count": 158,
  "value": 2860000
},
{
  "count": 150,
  "value": 2960000
},
{
  "count": 142,
  "value": 3060000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "count": 134,
  "value": 3160000
},
{
  "count": 126,
  "value": 3260000
},
{
  "count": 119,
  "value": 3360000
},
{
  "count": 111,
  "value": 3460000
},
{
  "count": 104,
  "value": 3560000
},
{
  "count": 97,
  "value": 3660000
},
{
  "count": 90,
  "value": 3760000
},
{
  "count": 83,
  "value": 3860000
},
{
  "count": 77,
  "value": 3960000
},
{
  "count": 71,
  "value": 4060000
},
{
  "count": 65,
  "value": 4160000
},
{
  "count": 60,
  "value": 4260000
},
{
  "count": 55,
  "value": 4360000
},
{
  "count": 50,
  "value": 4460000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "count": 46,
  "value": 4560000
},
{
  "count": 41,
  "value": 4660000
},
{
  "count": 37,
  "value": 4760000
},
{
  "count": 34,
  "value": 4860000
},
{
  "count": 30,
  "value": 4960000
},
{
  "count": 27,
  "value": 5060000
},
{
  "count": 24,
  "value": 5160000
},
{
  "count": 22,
  "value": 5260000
},
{
  "count": 19,
  "value": 5360000
},
{
  "count": 17,
  "value": 5460000
},
{
  "count": 15,
  "value": 5560000
},
{
  "count": 14,
  "value": 5660000
},
{
  "count": 12,
  "value": 5760000
},
{
  "count": 10,
  "value": 5860000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "count": 9,
  "value": 5960000
},
{
  "count": 8,
  "value": 6060000
},
{
  "count": 7,
  "value": 6160000
},
{
  "count": 6,
  "value": 6260000
},
{
  "count": 5,
  "value": 6360000
},
{
  "count": 5,
  "value": 6460000
},
{
  "count": 4,
  "value": 6560000
},
{
  "count": 3,
  "value": 6660000
},
{
  "count": 3,
  "value": 6760000
},
{
  "count": 2,
  "value": 6860000
},
{
  "count": 2,
  "value": 6960000
},
{
  "count": 2,
  "value": 7060000
},
{
  "count": 2,
  "value": 7160000
},
{
  "count": 1,
  "value": 7260000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```

{
  "count": 1,
  "value": 7360000
},
{
  "count": 1,
  "value": 7460000
},
{
  "count": 1,
  "value": 7560000
},
{
  "count": 1,
  "value": 7660000
},
{
  "count": 1,
  "value": 7760000
}
]
}

```

Расшифровка параметров:

- *total* - общее количество объявлений, участвующих в подсчете.
- *arithmeticMean* - среднее арифметическое.
- *count* - количество объявлений
- *value* - цена в долларах США

## 2.5 Поиск объявлений

Чтобы обратиться к DOM.RIA API search, Вам необходимо выполнить GET запрос такого вида:

`https://developers.ria.com/dom/search?api_key=YOUR_API_KEY&PARAMETERS`

или `curl -X GET "https://developers.ria.com/dom/search?api_key=YOUR_API_KEY&PARAMETERS" -H "accept: application/json"`

Он состоит из:

- **SEARCH** — название метода API, к которому Вы хотите обратиться.
- **API\_KEY** — ключ доступа. Для получения ключа доступа нужно зарегистрироваться на портале `developers.ria.com`
- **PARAMETERS** — входные параметры, последовательность пар `name=value`, разделенных амперсандом. Список параметров указан выше.

В ответ на такой запрос Вы получите ответ в формате JSON:

```

{
  "items": [
    //id объявлений
  ],

```

(continues on next page)

(продолжение с предыдущей страницы)

```

"count":           // количество
}

```

Полное описание сервиса «Поиск объявлений» описанный с помощью стандарта DeFacto swagger 2.0

- Квартир
- Домов
- Коммерческой недвижимости
- Офисов
- Земельных участков
- Гаражей

### Пример

Допустим Вы ищете:

- Объект - Квартиры
- Тип недвижимости - квартира
- Операция - продажа
- Область - Киевская
- Город - Киев
- Район
  - Оболонский
  - Печерский
  - Подольский
- Количество комнат от 1 до 3
- Общая площадь от 60 до 90
- Жилая площадь от 30 до 50
- Кухня от 4 до 9
- Кухня от 3 до 7
- Год постройки - не указан
- Цена от 20000 до 90000
- Цена за объект
- Тип валюты доллары США
- Возможен торг
- Тип предложения - от посредника

В итоге мы получаем запрос такого [вида]:

[https://developers.ria.com/auto/search?api\\_key=YOUR\\_API\\_KEY&category=1&realty\\_type=2&operation\\_type=1...](https://developers.ria.com/auto/search?api_key=YOUR_API_KEY&category=1&realty_type=2&operation_type=1...)

или

```
curl -X GET "https://developers.ria.com/dom/search?category=1&realty_type=2&operation_type=1&state_
↳id=10&city_id=10&district_id=15187&district_id=15189&district_id=15188&
↳characteristic[209][from]=1&characteristic[209][to]=3&
characteristic[214][from]=60&characteristic[214][to]=90&characteristic[216][from]=30&
↳characteristic[216][to]=50&
characteristic[218][from]=4&characteristic[218][to]=9&characteristic[227][from]=3&
↳characteristic[227][to]=7&
characteristic[443]=442&characteristic[234][from]=20000&characteristic[234][to]=90000&
characteristic[242]=239&characteristic[273]=273&characteristic[1437]=1434&
api_key=YOUR_API_KEY" -H "accept: application/json"
```

В случае успешного выполнения запроса по указанным параметрам результат будет со статусом **200 ОК**.

Пример успешного ответа:

```
{
  "items": [
    13336044,
    14076392,
    13814523,
    14064249,
    13078930,
    13927706,
    13962022
  ],
  "count": 7
}
```

## 2.6 Методы для работы с областями, городами, районами

### 2.6.1 Области

Получить список областей можно отправив GET запрос по адресу [https://developers.ria.com/dom/states?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/dom/states?api_key=YOUR_API_KEY). Для получение информации на украинском языке нужно добавить параметр `lang_id=4`

Полное описание сервиса «Методы для работы с областями, городами, районами» описанный с помощью стандарта DeFacto swagger 2.0 [здесь](#)

Результат будет следующим:

```
[
  {
    "lang_id": 2,
    "stateID": 1,
    "name": "Винницкая",
    "eng_name": "vinnica",
    "declension": "Винницкой области",
    "center_declension": "Винницы",
    "region_name": "Винница"
  },
  {
    "lang_id": 2,
    "stateID": 18,
```

(continues on next page)



(продолжение с предыдущей страницы)

```
"name": "Волынская",
"eng_name": "luck",
"declension": "Волынской области",
"center_declension": "Луцка",
"region_name": "Луцк"
},
{
  "lang_id": 2,
  "stateID": 11,
  "name": "Днепропетровская",
  "eng_name": "dneprpetrovsk",
  "declension": "Днепропетровской области",
  "center_declension": "Днепропетровска",
  "region_name": "Днепропетровск"
},
{
  "lang_id": 2,
  "stateID": 13,
  "name": "Донецкая",
  "eng_name": "doneck",
  "declension": "Донецкой области",
  "center_declension": "Донецка",
  "region_name": "Донецк"
},
{
  "lang_id": 2,
  "stateID": 2,
  "name": "Житомирская",
  "eng_name": "jitomir",
  "declension": "Житомирской области",
  "center_declension": "Житомира",
  "region_name": "Житомир"
},
{
  "lang_id": 2,
  "stateID": 22,
  "name": "Закарпатская",
  "eng_name": "ujgorod",
  "declension": "Закарпатской области",
  "center_declension": "Ужгорода",
  "region_name": "Ужгород"
},
{
  "lang_id": 2,
  "stateID": 14,
  "name": "Запорожская",
  "eng_name": "zaporoje",
  "declension": "Запорожской области",
  "center_declension": "Запорожья",
  "region_name": "Запорожье"
},
{
  "lang_id": 2,
  "stateID": 15,
  "name": "Ивано-Франковская",
  "eng_name": "ivano-frankovsk",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"declension": "Ивано-Франковской области",
"center_declension": "Ивано-Франковска",
"region_name": "Ивано-Франковск"
},
{
  "lang_id": 2,
  "stateID": 10,
  "name": "Киевская",
  "eng_name": "kiev",
  "declension": "Киевской области",
  "center_declension": "Киева",
  "region_name": "Киев"
},
{
  "lang_id": 2,
  "stateID": 16,
  "name": "Кировоградская",
  "eng_name": "kirovograd",
  "declension": "Кировоградской области",
  "center_declension": "Кировограда",
  "region_name": "Кировоград"
},
{
  "lang_id": 2,
  "stateID": 17,
  "name": "Луганская",
  "eng_name": "lugansk",
  "declension": "Луганской области",
  "center_declension": "Луганска",
  "region_name": "Луганск"
},
{
  "lang_id": 2,
  "stateID": 5,
  "name": "Львовская",
  "eng_name": "lvov",
  "declension": "Львовской области",
  "center_declension": "Львова",
  "region_name": "Львов"
},
{
  "lang_id": 2,
  "stateID": 19,
  "name": "Николаевская",
  "eng_name": "nikolaev",
  "declension": "Николаевской области",
  "center_declension": "Николаева",
  "region_name": "Николаев"
},
{
  "lang_id": 2,
  "stateID": 12,
  "name": "Одесская",
  "eng_name": "odessa",
  "declension": "Одесской области",
  "center_declension": "Одессы",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"region_name": "Одесса"
},
{
  "lang_id": 2,
  "stateID": 20,
  "name": "Полтавская",
  "eng_name": "poltava",
  "declension": "Полтавской области",
  "center_declension": "Полтавы",
  "region_name": "Полтава"
},
{
  "lang_id": 2,
  "stateID": 21,
  "name": "Республика Крым",
  "eng_name": "simferopol",
  "declension": "Республики Крым",
  "center_declension": "Симферополя",
  "region_name": "Республика Крым"
},
{
  "lang_id": 2,
  "stateID": 9,
  "name": "Ровенская",
  "eng_name": "rovno",
  "declension": "Ровенской области",
  "center_declension": "Ровно",
  "region_name": "Ровно"
},
{
  "lang_id": 2,
  "stateID": 8,
  "name": "Сумская",
  "eng_name": "sumiy",
  "declension": "Сумской области",
  "center_declension": "Сум",
  "region_name": "Сумы"
},
{
  "lang_id": 2,
  "stateID": 3,
  "name": "Тернопольская",
  "eng_name": "ternopol",
  "declension": "Тернопольской области",
  "center_declension": "Тернополя",
  "region_name": "Тернополь"
},
{
  "lang_id": 2,
  "stateID": 7,
  "name": "Харьковская",
  "eng_name": "harkov",
  "declension": "Харьковской области",
  "center_declension": "Харькова",
  "region_name": "Харьков"
},
},
```

(continues on next page)

```
{
  "lang_id": 2,
  "stateID": 23,
  "name": "Херсонская",
  "eng_name": "herson",
  "declension": "Херсонской области",
  "center_declension": "Херсона",
  "region_name": "Херсон"
},
{
  "lang_id": 2,
  "stateID": 4,
  "name": "Хмельницкая",
  "eng_name": "hmelnickiy",
  "declension": "Хмельницкой области",
  "center_declension": "Хмельницкого",
  "region_name": "Хмельницкий"
},
{
  "lang_id": 2,
  "stateID": 24,
  "name": "Черкасская",
  "eng_name": "cherkassiy",
  "declension": "Черкасской области",
  "center_declension": "Черкасс",
  "region_name": "Черкассы"
},
{
  "lang_id": 2,
  "stateID": 6,
  "name": "Черниговская",
  "eng_name": "chernigov",
  "declension": "Черниговской области",
  "center_declension": "Чернигова",
  "region_name": "Чернигов"
},
{
  "lang_id": 2,
  "stateID": 25,
  "name": "Черновицкая",
  "eng_name": "chernovciy",
  "declension": "Черновицкой области",
  "center_declension": "Черновцов",
  "region_name": "Черновцы"
}
}
```

## 2.6.2 Города

Города зависят от областей, поэтому, чтобы получить их список, необходимо послать GET запрос по адресу [https://developers.ria.com/dom/cities/:stateId?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/dom/cities/:stateId?api_key=YOUR_API_KEY), где *stateId* - идентификатор области. Для получение информации на украинском языке нужно добавить параметр *lang\_id=4*

Например, для Львовской области ([https://developers.ria.com/dom/cities/5?api\\_key=YOUR\\_API\\_KEY&lang\\_id=4](https://developers.ria.com/dom/cities/5?api_key=YOUR_API_KEY&lang_id=4)) список городов будет следующим:

```
[
  {
    "lang_id": 4,
    "cityID": 287,
    "stateID": 5,
    "name": "Борислав",
    "eng": "borislav",
    "declension": ""
  },
  {
    "lang_id": 4,
    "cityID": 288,
    "stateID": 5,
    "name": "Бродя",
    "eng": "brodi",
    "declension": ""
  },
  {
    "lang_id": 4,
    "cityID": 289,
    "stateID": 5,
    "name": "Буськ",
    "eng": "busk",
    "declension": ""
  },
  {
    "lang_id": 4,
    "cityID": 290,
    "stateID": 5,
    "name": "Городок",
    "eng": "gorodok",
    "declension": ""
  },
  {
    "lang_id": 4,
    "cityID": 291,
    "stateID": 5,
    "name": "Дрогобич",
    "eng": "drogobich",
    "declension": ""
  },
  {
    "lang_id": 4,
    "cityID": 292,
    "stateID": 5,
    "name": "Жидачив",
    "eng": "jidachiv",
    "declension": ""
  },
  {
    "lang_id": 4,
    "cityID": 293,
    "stateID": 5,
    "name": "Жовква",
    "eng": "jovkva",
    "declension": ""
  },
],
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "lang_id": 4,
  "cityID": 294,
  "stateID": 5,
  "name": "Золочев",
  "eng": "zolochev",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 295,
  "stateID": 5,
  "name": "Кам'янка-Бузька",
  "eng": "kam_yanka-buzka",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 5,
  "stateID": 5,
  "name": "Львів",
  "eng": "lviv",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 297,
  "stateID": 5,
  "name": "Миколаїв",
  "eng": "mikolayv",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 646,
  "stateID": 5,
  "name": "Моршин",
  "eng": "Morshun",
  "declension": null
},
{
  "lang_id": 4,
  "cityID": 298,
  "stateID": 5,
  "name": "Мостиська",
  "eng": "mostiska",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 632,
  "stateID": 5,
  "name": "Новий Розділ",
  "eng": "noviy-rozdil",
  "declension": "Новороздільский"
},
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "lang_id": 4,
  "cityID": 299,
  "stateID": 5,
  "name": "Перемишляни",
  "eng": "peremishlyani",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 300,
  "stateID": 5,
  "name": "Пустомити",
  "eng": "pustomiti",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 301,
  "stateID": 5,
  "name": "Радехів",
  "eng": "radehiv",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 302,
  "stateID": 5,
  "name": "Самбір",
  "eng": "sambir",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 303,
  "stateID": 5,
  "name": "Сколе",
  "eng": "skole",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 304,
  "stateID": 5,
  "name": "Сокаль",
  "eng": "sokal",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 614,
  "stateID": 5,
  "name": "Соснівка",
  "eng": "sosnovka",
  "declension": null
},
}
```

(continues on next page)

```
{
  "lang_id": 4,
  "cityID": 305,
  "stateID": 5,
  "name": "Старий Самбір",
  "eng": "stariyi_sambir",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 306,
  "stateID": 5,
  "name": "Стрий",
  "eng": "striyi",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 307,
  "stateID": 5,
  "name": "Трускавець",
  "eng": "truskavec",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 308,
  "stateID": 5,
  "name": "Турка",
  "eng": "turka",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 309,
  "stateID": 5,
  "name": "Червоноград",
  "eng": "chervonograd",
  "declension": ""
},
{
  "lang_id": 4,
  "cityID": 310,
  "stateID": 5,
  "name": "Яворів",
  "eng": "yavoriv",
  "declension": ""
}
}
```

### 2.6.3 Районы

Районы зависят от городов, поэтому, чтобы получить их список, необходимо послать GET запрос по адресу [https://developers.ria.com/dom/cities\\_districts/:city\\_id?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/dom/cities_districts/:city_id?api_key=YOUR_API_KEY), где *city\_id* - идентификатор города. Для получение информации на украинском языке нужно добавить



параметр `lang_id=4`

Например, для города Львова ([https://developers.ria.com/dom/cities\\_districts/5?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/dom/cities_districts/5?api_key=YOUR_API_KEY)) список районов будет следующим:

```
[
  [
    {
      "name": "Район",
      "value": ""
    },
    {
      "city_id": 5,
      "area_id": 15754,
      "name": "Аеропорт",
      "type": 1,
      "value": 15754
    },
    {
      "city_id": 5,
      "area_id": 15765,
      "name": "Арсен",
      "type": 1,
      "value": 15765
    },
    {
      "city_id": 5,
      "area_id": 15750,
      "name": "Білогорща",
      "type": 1,
      "value": 15750
    },
    {
      "city_id": 5,
      "area_id": 15771,
      "name": "Бондарівка",
      "type": 1,
      "value": 15771
    },
    {
      "city_id": 5,
      "area_id": 15751,
      "name": "Виговського",
      "type": 1,
      "value": 15751
    },
    {
      "city_id": 5,
      "area_id": 15763,
      "name": "Винники",
      "type": 1,
      "value": 15763
    },
    {
      "city_id": 5,
      "area_id": 15087,
      "name": "Галицький",
      "type": 1,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"value": 15087
},
{
  "city_id": 5,
  "area_id": 15777,
  "name": "Голоско",
  "type": 1,
  "value": 15777
},
{
  "city_id": 5,
  "area_id": 15088,
  "name": "Залізничний",
  "type": 1,
  "value": 15088
},
{
  "city_id": 5,
  "area_id": 15778,
  "name": "Замарстинів",
  "type": 1,
  "value": 15778
},
{
  "city_id": 5,
  "area_id": 15779,
  "name": "Збоїща",
  "type": 1,
  "value": 15779
},
{
  "city_id": 5,
  "area_id": 15760,
  "name": "Знесення",
  "type": 1,
  "value": 15760
},
{
  "city_id": 5,
  "area_id": 15767,
  "name": "Новий Львів",
  "type": 1,
  "value": 15767
},
{
  "city_id": 5,
  "area_id": 15773,
  "name": "Новий Світ",
  "type": 1,
  "value": 15773
},
{
  "city_id": 5,
  "area_id": 15768,
  "name": "Пасіки",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"type": 1,
"value": 15768
},
{
  "city_id": 5,
  "area_id": 15772,
  "name": "Персеньківка",
  "type": 1,
  "value": 15772
},
{
  "city_id": 5,
  "area_id": 15769,
  "name": "Пироговка",
  "type": 1,
  "value": 15769
},
.....

{
  "city_id": 5,
  "area_id": 15775,
  "name": "Ринок Південний",
  "type": 1,
  "value": 15775
},
{
  "city_id": 5,
  "area_id": 15748,
  "name": "Рясне",
  "type": 1,
  "value": 15748
},
{
  "city_id": 5,
  "area_id": 15766,
  "name": "Санта Барбара",
  "type": 1,
  "value": 15766
},
{
  "city_id": 5,
  "area_id": 15753,
  "name": "Сигнівка",
  "type": 1,
  "value": 15753
},
{
  "city_id": 5,
  "area_id": 15090,
  "name": "Сихівський",
  "type": 1,
  "value": 15090
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "city_id": 5,
  "area_id": 15752,
  "name": "Скнилів",
  "type": 1,
  "value": 15752
},
{
  "city_id": 5,
  "area_id": 15747,
  "name": "Снопківська",
  "type": 1,
  "value": 15747
},
{
  "city_id": 5,
  "area_id": 15764,
  "name": "ТЦ Іскра",
  "type": 1,
  "value": 15764
},
{
  "city_id": 5,
  "area_id": 15091,
  "name": "Франківський",
  "type": 1,
  "value": 15091
},
{
  "city_id": 5,
  "area_id": 15745,
  "name": "Цитадель",
  "type": 1,
  "value": 15745
},
{
  "city_id": 5,
  "area_id": 15092,
  "name": "Шевченківський",
  "type": 1,
  "value": 15092
}
],
[
  {
    "name": "Пригород",
    "value": ""
  },
  {
    "city_id": 5,
    "area_id": 7494,
    "name": "Брюховичі",
    "type": 2,
    "value": 7494
  },
  {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"city_id": 5,
"area_id": 15742,
"name": "Зимна Вода",
"type": 2,
"value": 15742
},
{
"city_id": 5,
"area_id": 15743,
"name": "Малечковичі",
"type": 2,
"value": 15743
},
{
"city_id": 5,
"area_id": 15741,
"name": "Сокільники",
"type": 2,
"value": 15741
}
],
[
{
"name": "Село",
"value": ""
},
{
"city_id": 5,
"area_id": 7495,
"name": "Винники",
"type": 3,
"value": 7495
},
{
"city_id": 5,
"area_id": 17513,
"name": "Липники",
"type": 3,
"value": 17513
},
{
"city_id": 5,
"area_id": 7497,
"name": "Рудне",
"type": 3,
"value": 7497
}
]
]
```



Мы начали создавать новые персонализированные API, которые помогут Вам управлять всеми Вашими объявлениями на RIA.com

При этом, нет необходимости открывать много вкладок, теперь все API доступны в одной админ-панели. Зарегистрируйтесь или авторизируйтесь на Developers.RIA.com <<https://developers.ria.com/>>‘\_’, чтобы использовать персонализированные API. По умолчанию все права на доступ к сервисам не активны, что бы активировать нужный Вам сервис перейдите во вкладку \_»Мои разрешения» <<https://developers.ria.com/account/credentials>>‘\_

Все методы и данные описаны с помощью стандарта DeFacto swagger 2.0 \_‘здесь <<http://swagger.ria.com/ui/?api=ria/basket>>‘\_

### 3.1 Перечень статусов заказов

Что бы получить список типов статусов нужно отправив GET запрос на адрес [https://developers.ria.com/ria/basket/status?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/ria/basket/status?api_key=YOUR_API_KEY). Результат будет примерно следующим:

```
curl -X GET https://developers.ria.com/ria/basket/status?api_key=YOUR_API_KEY
```

```
{
  "status": true,
  "data": [
    {
      "id": 0,
      "name": "Новые и Невыполненные"
    },
    {
      "id": 1,
      "name": "Выполненные"
    },
    {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "id": 2,
    "name": "Невыполненные"
  },
  {
    "id": 6,
    "name": "Удаленные"
  },
  {
    "id": 4,
    "name": "Все"
  },
  {
    "id": 7,
    "name": "Обратный звонок"
  }
]
}

```

## 3.2 Перечень заказов

Что бы получить список заказов нужно отправив GET запрос на адрес [https://developers.ria.com/ria/basket/orders/?api\\_key=YOUR\\_API\\_KEY](https://developers.ria.com/ria/basket/orders/?api_key=YOUR_API_KEY).

```
curl -X GET https://developers.ria.com/ria/basket/orders?api_key=YOUR_API_KEY
```

Результат будет примерно следующим:

```

{
  "status": true,
  "total": 2,
  "orders": [
    {
      "order_id": 21, // id заказа
      "status": 6, // Статус "Удаленные"
      "total_order_sum": 100, // Общая сумма заказа
      "created_at": "2017-04-06 16:35:31", // Время создания
      "recipient": {
        "user_id": 000000, // id пользователя
        "name": "Андрій Сергійович",
        "email": "mail@gmail.com",
        "phone": "....." // Телефон
      },
      "delivery": { // Доставка
        "state_id": 1, // id Области
        "city_id": 1, // id Города
        "address": "Винницькая область, Винниця",
        "service": "DHL" // Служба доставки
      },
      "department": "Отделение №1: ул. Интернаціональна, 20а" // Информация об отделении
    },
    "items": [
      {
        "adv_id": 3612261, // id объявления
        "name": "Детали двигателя Головка блока Легковой Audi A3 Cabrio v2 v3",
        "price": 100, // Цена
      }
    ]
  ]
}

```

(continues on next page)



(продолжение с предыдущей страницы)

```
        "count":1,                // Количество
        "note":""                // Заметки
        "vendor_code": ""        // Код поставщика
    }
  ],
  ...
]
```

### 3.3 Изменение статуса заказа

Что бы изменить статус заказа нужно отправить PUT запрос на адрес

```
curl -X PUT "https://developers.ria.com/ria/basket/orders?user_id=*id*&order_id=*id*&api_key=YOUR_
↳API_KEY" -H "accept: application/json" -H "content-type: application/json"
-d "{ \"status\": 0}"
```

, где *user\_id* - id пользователя, *order\_id* - id заказа, *status\_id* - id статуса заказа,

Ответ будет следующим:

```
{
  "status": true
}
```



---

## Общие Ошибки Веб-Службы

---

Определенные распространенные ошибки будут возвращены стандартным способом из всех веб-сервисов API RIA.com. Кроме того, некоторые сообщения об ошибках могут быть специфичными для некоторых сервисов (см. документацию для конкретного сервиса). Ниже описаны распространенные ошибки:



---

## Использование API Key

---

После [регистрации](#), Вам на почту будет предоставлен собственный, уникальный API ключ в виде строки из 40 символов. Ваш API ключ:

- Уникальный, идентифицирует вас.
- Предоставляет Вам доступ к веб-сервисам RIA.
- Полностью приватен, не может передаваться другим лицам.

Чтобы использовать свой ключ, просто вставьте его в качестве параметра URL-адреса при создании запросов веб-сервиса. Например:

```
GET http://developers.ria.com/states?api_key=YOUR_KEY_HERE
```

Независимо от используемого HTTP метода, API ключ всегда должен передаваться в качестве параметра GET в запросе URL. Таким образом, даже если Вы будете использовать метод POST или PUT для конкретного сервиса, параметр запроса *api\_key* должен всегда указываться в URL запросе.

### **Альтернативный Метод**

В зависимости от того, как Вы будете использовать API, иногда может быть легче передать API ключ в качестве базовой аутентификации HTTP. Если Вы хотите использовать этот метод, отправьте Ваш API ключ в качестве имени пользователя, оставив пароль пустым. Например:

```
GET http://YOUR_KEY_HERE@developers.ria.com/states
```



---

## Лимиты Веб-Сервисов

---

Ограничения распространяются на количество запросов к веб-службам RIA.com, которые Вы можете создать, используя свой API ключ. .

- **Hourly Limit:** Часовой лимит: 1,000 запросов в час

Эти ограничения применяются ко всем веб-службам RIA.com. Превышение этих лимитов приведет к временному блокированию Вашего API ключа. В зависимости от превышенного предела, блок будет автоматически отменен в течении следующего часа или календарного дня.

### Понимание Предельных Значений Лимитов

#### Часовой Лимит

Почасовые счетчики для API ключа аннулируются на постоянной основе.

*Пример:* Пример: Если Вы сделаете 500 запросов в 10:15 и 500 запросов в 10:25, Ваш API ключ будет временно заблокирован. Блок будет аннулирован в 11:15, после чего Вы сможете сделать еще 500 запросов.

#### Ограничение Запросов

Если Ваш API ключ превышает лимиты запросов, Вы получите ответ с HTTP статус-кодом 429 (слишком много запросов).

#### Нужно повысить лимиты?

Если Вы создаете приложение, которое требует более высоких лимитов, мы будем рады сотрудничать. [Свяжитесь с нами](#) для более подробной информации.