
RIA.com Documentation

Выпуск 2.0

RIA.com Documentation

мар. 20, 2023

1	AUTO.RIA	1
1.1	AUTO.RIA Вживані авто	1
1.2	AUTO.RIA Нові авто	37
2	DOM.RIA	49
2.1	Параметри для роботи з API DOM.RIA	50
2.2	Список характеристик нерухомості	50
2.3	Інформація по id оголошення	68
2.4	Підрахунок середньої ціни	71
2.5	Пошук оголошень	78
2.6	Методи для роботи з областями, містами, районами	80
3	RIA.com	91
3.1	Перелік статусів замовлень	91
3.2	Перелік замовлень	92
3.3	Зміна статусу замовлення	93
4	Загальні Помилки Веб-Служби	95
5	Використання API Key	97
6	Ліміти Веб-Сервісів	99

Ліміти Веб-Сервісів

Обмеження поширюються на кількість запитів до веб-служб RIA.com, які Ви можете створити, використовуючи свій API ключ.

Годинний ліміт: 1,000 запитів на годину

Ці обмеження застосовуються до всіх веб-служб RIA.com. Перевищення цих лімітів призведе до тимчасового блокування Вашого API ключа. Залежно від перевищеної межі, блок буде автоматично скасовано протягом наступної години або календарного дня.

Годинний Ліміт

Погодинні лічильники для API ключа анулюються на постійній основі.

Наприклад: Якщо Ви зробите 500 запитів в 10:15 та 500 запитів в 10:25 - Ваш API ключ буде тимчасово заблоковано. Блок буде анульовано в 11:15, після чого Ви зможете зробити ще 500 запитів.

Обмеження Запитів

Якщо Ваш API ключ перевищує ліміти запитів, Ви отримаєте відповідь з HTTP статус-кодом 429 (занадто багато запитів).

Потрібно підвищити ліміти?

Якщо Ви створюєте додаток, який вимагає більш високих лімітів, ми будемо раді співпрацювати. Напишіть нам листа на info_developers@ria.com для отримання більш детальної інформації.

1.1 AUTO.RIA Вживані авто

1.1.1 Методи для роботи з Параметрами

Типи транспорту

Щоб отримати список типів транспорту, потрібно надіслати GET запит на адресу https://developers.ria.com/auto/categories/?api_key=YOUR_API_KEY. Результат буде приблизно наступним:

```
[
  {"name": "Легкові", "value": 1},
  {"name": "Мото", "value": 2},
  {"name": "Водний транспорт", "value": 3},
  {"name": "Спецтехніка", "value": 4},
  {"name": "Причепи", "value": 5},
  {"name": "Вантажівки", "value": 6},
  {"name": "Автобуси", "value": 7},
  {"name": "Автобудинки", "value": 8},
  {"name": "Повітряний транспорт", "value": 9},
  {"name": "Сільгосптехніка", "value": 10}
]
```

Типи кузова

Типи кузова залежать від типів транспорту. Тому для того, щоб отримати список типів кузова, необхідно надіслати GET запит на адресу https://developers.ria.com/auto/categories/:categoryId/bodystyles?api_key=YOUR_API_KEY, де *categoryId* - ідентифікатор типу транспорту, *api_key* - Ваш ключ.

Наприклад, для легкових автомобілів (https://developers.ria.com/auto/categories/1/bodystyles?api_key=YOUR_API_KEY) результат буде наступним:

```
[
  {"name": "Седан", "value": 3, "parentId": 0},
  {"name": "Позашляховик / Кросовер", "value": 5, "parentId": 0},
  {"name": "Мінівен", "value": 8, "parentId": 0},
  {"name": "Мікровен", "value": 449, "parentId": 0},
  {"name": "Хетчбек", "value": 4, "parentId": 0},
  {"name": "Універсал", "value": 2, "parentId": 0},
  {"name": "Купе", "value": 6, "parentId": 0},
  {"name": "Кабріолет", "value": 7, "parentId": 0},
  {"name": "Пікап", "value": 9, "parentId": 0},
  {"name": "Ліфтбек", "value": 307, "parentId": 0},
  {"name": "Фастбек", "value": 448, "parentId": 0},
  {"name": "Лімузин", "value": 252, "parentId": 0},
  {"name": "Родстер", "value": 315, "parentId": 0}
]
```

Також типи кузова можуть бути розділені на групи. Це актуально для спецтехніки. Тому існує спосіб отримати згруповані типи кузова надіславши GET запит за адресою https://developers.ria.com/auto/categories/:categoryId/bodystyles/_group?api_key=YOUR_API_KEY, де *categoryId* - ідентифікатор типу транспорту, *api_key* - Ваш ключ.

Наприклад, для мотоциклів (https://developers.ria.com/auto/categories/2/bodystyles/_group?api_key=YOUR_API_KEY) результат буде наступним:

```
[
  [
    {"name": "Інше", "value": 56, "parentId": 0},
  ]
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```

{"name": "Вантажні моторолери, мотоцикли, скутери, мопеди", "value": 429, "parentId": 56}
],
[
{"name": "Квадроцикли", "value": 35, "parentId": 0},
{"name": "Всюдихід-амфібія", "value": 43, "parentId": 35},
{"name": "Гольф-кар", "value": 44, "parentId": 35},
{"name": "Картинг", "value": 45, "parentId": 35},
{"name": "Квадроцикл дитячий", "value": 36, "parentId": 35},
{"name": "Квадроцикл спортивний", "value": 39, "parentId": 35},
{"name": "Квадроцикл утилітарний", "value": 41, "parentId": 35},
{"name": "Мотовсюдиход", "value": 42, "parentId": 35}
],
[
{"name": "Міні мотоцикли", "value": 31, "parentId": 0},
{"name": "Міні крос (Пітбайк)", "value": 33, "parentId": 31},
{"name": "Міні спорт", "value": 32, "parentId": 31}
],
[
{"name": "Мопеди", "value": 58, "parentId": 0},
{"name": "Максі-скутер", "value": 12, "parentId": 58},
{"name": "Мокик", "value": 427, "parentId": 58},
{"name": "Скутер / Мотороллер", "value": 11, "parentId": 58},
{"name": "Скутер для інвалідів", "value": 426, "parentId": 58},
{"name": "Скутер ретро", "value": 425, "parentId": 58},
{"name": "Скутери з кабіною", "value": 430, "parentId": 58}
],
[
{"name": "Мотоцикли", "value": 13, "parentId": 0},
{"name": "Боббер", "value": 421, "parentId": 13},
{"name": "Кафе рейсер", "value": 423, "parentId": 13},
{"name": "Мотоцикл Багатоцільовий (All-round)", "value": 25, "parentId": 13},
{"name": "Мотоцикл Без обтікачів (Naked bike)", "value": 15, "parentId": 13},
{"name": "Мотоцикл з коляскою", "value": 29, "parentId": 13},
{"name": "Мотоцикл Кастом", "value": 30, "parentId": 13},
{"name": "Мотоцикл Класік", "value": 14, "parentId": 13},
{"name": "Мотоцикл Кросс", "value": 19, "parentId": 13},
{"name": "Мотоцикл Круїзер", "value": 24, "parentId": 13},
{"name": "Мотоцикл Позашляховий (Enduro)", "value": 21, "parentId": 13},
{"name": "Мотоцикл Спорт-туризм", "value": 17, "parentId": 13},
{"name": "Мотоцикл Супермото (Motard)", "value": 22, "parentId": 13},
{"name": "Мотоцикл Тріал", "value": 20, "parentId": 13},
{"name": "Мотоцикл Туризм", "value": 16, "parentId": 13},
{"name": "Мотоцикл Чоппер", "value": 23, "parentId": 13},
{"name": "Скремблер", "value": 422, "parentId": 13},
{"name": "Спортбайк", "value": 18, "parentId": 13}
],
[
{"name": "Снігохід", "value": 46, "parentId": 0},
{"name": "Гірські снігоходи", "value": 434, "parentId": 46},
{"name": "Міні-розбірні снігоходи", "value": 432, "parentId": 46},
{"name": "Мотобуксир", "value": 424, "parentId": 46},
{"name": "Снігомопеди та снігоскутери", "value": 436, "parentId": 46},
{"name": "Снігоходи для шоловання та рибалки", "value": 433, "parentId": 46},
{"name": "Спортивні снігоходи", "value": 435, "parentId": 46},
{"name": "Утилітарні снігоходи", "value": 431, "parentId": 46}
],

```

(continues on next page)

(продолжение с предыдущей страницы)

```

{"name": "Траjk", "value": 57, "parentId": 0},
[
  {"name": "Трицикл", "value": 34, "parentId": 0},
  {"name": "Вантажні трицикли", "value": 428, "parentId": 34}
]

```

Формат даних при цьому відрізняється від звичайного - це колекція об'єктів у якій є інші колекції. Група типів кузовів завжди починається з її назви. Наприклад, у групу *Квадроцикли* входять типи кузовів *Квадроцикл дитячий*, *Квадроцикл спортивний*, *Квадроцикл утилітарний*, *Мотовсюдиxid* тощо.

Також, за необхідності, можна отримати просто весь список типів кузовів, надіславши GET запит за адресою https://developers.ria.com/auto/bodystyles?api_key=YOUR_API_KEY.

Марки

Марки залежать від типів транспорту. Тому для того, щоб отримати список марок необхідно надіслати GET запит за адресою https://developers.ria.com/auto/categories/:categoryId/marks?api_key=YOUR_API_KEY, де *categoryId* - ідентифікатор типу транспорту, *api_key* - Ваш ключ.

Наприклад, для легкових автомобілів (https://developers.ria.com/auto/categories/1/marks?api_key=YOUR_API_KEY) результат буде таким:

```

[
  { 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 }
  ...
]

```

Моделі

Моделі залежать від типів транспорту і марок. Тому список марок можна отримати за адресою http://api.auto.ria.com/categories/:categoryId/marks/:markId/models?api_key=YOUR_API_KEY, де *categoryId* - ідентифікатор типу транспорту, а *markId* - ідентифікатор марки, *api_key* - Ваш ключ.

Наприклад, для мотоциклів BMW (https://developers.ria.com/auto/categories/2/marks/9/models?api_key=YOUR_API_KEY) список моделей буде таким:

```

[
  { name: "Adventure", value: 25290 },
  { name: "C", value: 25291 },
  { name: "CS", value: 25292 },
  { name: "DKW", value: 28318 },

```

(continues on next page)

(продолжение с предыдущей страницы)

```

{ 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 }
]

```

Моделі, так само як і типи кузовів, можуть бути згруповані. Щоб отримати такий список, необхідно надіслати запит за адресою https://developers.ria.com/auto/categories/:categoryId/marks/:markId/models/_group?api_key=YOUR_API_KEY, де *categoryId* - ідентифікатор типу транспорту, а *markId* - ідентифікатор марки, *api_key* - Ваш ключ.

Наприклад, для легкових автомобілів BMW (https://developers.ria.com/auto/categories/1/marks/9/models/_group?api_key=YOUR_API_KEY) список моделей буде таким:

```

[
  [
    { 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 },
]

```

(continues on next page)

(продолжение с предыдущей страницы)

```
[
  { name: "Z1", value: 97 },
  { name: "Z3", value: 98 },
  { name: "Z4", value: 99 },
  { name: "Z8", value: 100 }
]
```

Формат даних такий самий як і у випадку з типом кузова - колекція об'єктів у якій можуть бути інші об'єкти. Група моделей завжди починається з її назви.

Також, за необхідності, можна просто отримати список усіх моделей, надіславши GET запит за адресою https://developers.ria.com/auto/models?api_key=YOUR_API_KEY.

Області

Отримати список областей можна надіславши GET запит за адресою https://developers.ria.com/auto/states?api_key=YOUR_API_KEY.

Результат буде таким:

```
[
  {"name": "Київська", "value": 10},
  {"name": "Вінницька", "value": 1},
  {"name": "Волинська", "value": 18},
  {"name": "Дніпропетровська", "value": 11},
  {"name": "Донецька", "value": 13},
  {"name": "Житомирська", "value": 2},
  {"name": "Закарпатська", "value": 22},
  {"name": "Запорізька", "value": 14},
  {"name": "Івано-Франківська", "value": 15},
  {"name": "Кіровоградська", "value": 16},
  {"name": "Луганська", "value": 17},
  {"name": "Львівська", "value": 5},
  {"name": "Миколаївська", "value": 19},
  {"name": "Одеська", "value": 12},
  {"name": "Полтавська", "value": 20},
  {"name": "Рівненська", "value": 9},
  {"name": "Сумська", "value": 8},
  {"name": "Тернопільська", "value": 3},
  {"name": "Харківська", "value": 7},
  {"name": "Херсонська", "value": 23},
  {"name": "Хмельницька", "value": 4},
  {"name": "Черкаська", "value": 24},
  {"name": "Чернівецька", "value": 25},
  {"name": "Чернігівська", "value": 6}
]
```

Міста

Міста залежать від областей - тому, щоб отримати їх список, необхідно надіслати GET запит за адресою https://developers.ria.com/auto/states/:stateId/cities?api_key=YOUR_API_KEY, де *stateId* - ідентифікатор області, *api_key* - Ваш ключ.

Наприклад, для Вінницької області (https://developers.ria.com/auto/states/1/cities?api_key=YOUR_API_KEY) список міст буде таким:

```
[
  {"name": "Вінниця", "value": 1},
  {"name": "Жмеринка", "value": 27},
  {"name": "Козятин", "value": 30},
  {"name": "Крижопіль", "value": 31},
  {"name": "Липовець", "value": 32},
  {"name": "Літин", "value": 33},
  {"name": "Могилів-Подільський", "value": 34},
  {"name": "Муровані Курилівці", "value": 35},
  {"name": "Немирів", "value": 36},
  {"name": "Оратів", "value": 37},
  {"name": "Піщанка", "value": 38},
  {"name": "Погребище", "value": 39},
  {"name": "Теплик", "value": 40},
  {"name": "Тиврів", "value": 41},
  {"name": "Томашпіль", "value": 42},
  {"name": "Тростянець", "value": 43},
  {"name": "Тульчин", "value": 44},
  {"name": "Хмільник", "value": 45},
  {"name": "Чернівці", "value": 46},
  {"name": "Чечельник", "value": 47},
  {"name": "Шаргород", "value": 48},
  {"name": "Ямпіль", "value": 49},
  {"name": "Бар", "value": 597},
  {"name": "Бершадь", "value": 599},
  {"name": "Гайсин", "value": 602},
  {"name": "Іллінці", "value": 603},
  {"name": "Калинівка", "value": 604},
  {"name": "Гнівань", "value": 609},
  {"name": "Ладизин", "value": 644},
  {"name": "Якушинці", "value": 1595}
]
```

Типи приводу

Типи приводу також залежать від типу транспорту - тому, щоб отримати їхній список, необхідно надіслати GET запит за адресою https://developers.ria.com/auto/categories/:categoryId/driverTypes?api_key=YOUR_API_KEY, де *categoryId* - ідентифікатор типу транспорту, *api_key* - Ваш ключ.

Наприклад, список типів приводу для мотоциклів (https://developers.ria.com/auto/categories/2/driverTypes?api_key=YOUR_API_KEY) таким чином:

```
[
  {"name": "Кардан", "value": 4},
  {"name": "Ремінь", "value": 5},
  {"name": "Ланцюг", "value": 6}
]
```

Типи палива

Типи палива можна отримати надіславши GET запит за адресою https://developers.ria.com/auto/type?api_key=YOUR_API_KEY. Відповідь буде виглядати наступним чином:

```
[
  {"name": "Бензин", "value": 1},
  {"name": "Дизель", "value": 2},
  {"name": "Газ", "value": 3},
  {"name": "Газ / Бензин", "value": 4},
  {"name": "Гібрид", "value": 5},
  {"name": "Електро", "value": 6},
  {"name": "Інше", "value": 7},
  {"name": "Газ метан", "value": 8},
  {"name": "Газ пропан-бутан", "value": 9}
]
```

Коробки передач

Коробки передач залежать від типу транспорту - тому, щоб отримати їх список необхідно надіслати GET запит за адресою https://developers.ria.com/auto/categories/:categoryId/gearboxes?api_key=YOUR_API_KEY, де *categoryId* - ідентифікатор типу транспорту, *api_key*- Ваш ключ .

Наприклад, список коробок передач для мотоциклів (https://developers.ria.com/auto/categories/2/gearboxes?api_key=YOUR_API_KEY) буде виглядати наступним чином:

```
[
  {"name": "Ручна / Механіка", "value": 1},
  {"name": "Автомат", "value": 2},
  {"name": "Типтронік", "value": 3},
  {"name": "Робот", "value": 4},
  {"name": "Варіатор", "value": 5}
]
```

Опції

Опції залежать від типу транспорту. Отримати їхній список можна надіславши GET запит за адресою https://developers.ria.com/auto/categories/:categoryId/options?api_key=YOUR_API_KEY, де *categoryId* - ідентифікатор типу транспорту, *api_key*- Ваш ключ.

Наприклад, список опцій для легкових автомобілів (https://developers.ria.com/auto/categories/1/options?api_key=YOUR_API_KEY) матиме приблизно такий вигляд:

```
[
  {"name": "Антиблокувальна система (ABS)", "value": 217},
  {"name": "Антипробуксовочна система (ASR)", "value": 606},
  {"name": "Блокування замків задніх дверей", "value": 622},
  {"name": "Броньований кузов", "value": 515},
  {"name": "Вибір режиму руху", "value": 610},
  {"name": "Датчик втоми водія", "value": 617},
  {"name": "Датчик проникнення в салон (датчик об`єму)", "value": 623},
  {"name": "Датчик тиску в шинах", "value": 620},
  {"name": "Допомога при спуску", "value": 611},
  {"name": "Допомога при старті в гору", "value": 609},
  {"name": "Запобігання зіткнення", "value": 612},
  {"name": "Імобілайзер", "value": 604}
  ...
]
```

Кольори

Отримати список усіх кольорів можна якщо надіслати GET запит за адресою https://developers.ria.com/auto/colors?api_key=YOUR_API_KEY. Результат буде наступним:

```
[
  {"name": "Бежевий", "value": 1},
  {"name": "Чорний", "value": 2},
  {"name": "Синій", "value": 3},
  {"name": "Коричневий", "value": 5},
  {"name": "Зелений", "value": 7},
  {"name": "Сірий", "value": 8},
  {"name": "Помаранчевий", "value": 9},
  {"name": "Фіолетовий", "value": 12},
  {"name": "Червоний", "value": 13},
  {"name": "Білий", "value": 15},
  {"name": "Жовтий", "value": 16}
]
```

Країна виробник

Отримати список країн виробників можна якщо надіслати GET запит за адресою https://developers.ria.com/auto/countries?api_key=YOUR_API_KEY. Результат буде наступним:

```
[
  {"name": "Австрія", "value": 40},
  {"name": "Англія", "value": 826},
  {"name": "Аргентина", "value": 32},
  {"name": "Бельгія", "value": 56},
  {"name": "Болгарія", "value": 100},
  {"name": "Данія", "value": 208},
  {"name": "Естонія", "value": 233},
  {"name": "Індія", "value": 356},
  {"name": "Китай", "value": 158},
  {"name": "Корея", "value": 408},
  {"name": "Німеччина", "value": 276},
  .....
]
```

Інші параметри

Розмитнення

Параметр розмитнення (`custom`) може набувати тільки двох значень:

- 1 - нерозмитнений
- 0 - розмитнений

Після ДТП

Цей параметр (`damage`) набуває таких значень:

1 - оголошення після ДТП

0 - інші оголошення

Взято в кредит

Цей параметр (`under_credit`) може набувати таких значень:

1 - оголошення взяті в кредит

0 - інші оголошення

Конфіскація

Цей параметр (`confiscated_car`) може набувати таких значень:

1 - тільки конфісковані оголошення

0 - тільки неконфісковані оголошення

Ціна

Цей параметр може набувати значень, які Ви задасте:

`price_ot=` - ціна від

`price_do=` - ціна до

Валюта

Цей параметр може набувати таких значень:

1 - долари США

2 - євро

3 - гривня

Авто не в Україні

Даний параметр (`abroad`) може набувати таких значень:

0 - за замовчуванням показати все

1 - включити в запит пошуку

2 - не включати до запиту пошуку

Не на ходу

Даний параметр (`auto_repairs`) може набувати таких значень:

1 - Лише не на ходу

2 - Приховати не на ходу

Продавець

Цей параметр може набувати таких значень:

- 0 - за замовчуванням усі
- 1 - приватна особа
- 2 - компанія

Пригнаний з

Для цього параметра можна використовувати елементи, отримані за допомогою сервісу **Країна виробник**

`matched_country=` - Пригнаний з

Об'єм двигуна

Цей параметр може набувати значень, які Ви задасте:

`engineVolumeFrom=` - об'єм від
`engineVolumeTo=` - об'єм до

Витрата палива

Цей параметр може набувати значень, які Ви задасте:

`fuelRateFrom=` - витрата від
`fuelRateTo=` - витрата до

Цикл

`fuelRatesType=` може дорівнювати трьом значенням

`city` - міський цикл
`combine` - шосе
`route` - змішаний

Потужність двигуна

Цей параметр може набувати значень, які Ви задасте:

`powerFrom=` - потужність від
`powerTo=` - потужність до

Одиниця виміру потужності (`power_name`)

- 1 - кінські сили
- 2 - кВт

Сортування

Цей параметр може набувати таких значень:

- 0 - за замовчуванням звичайне сортування
- 1 - від дешевих до дорогих
- 2 - від дорогих до дешевих

Період подачі

Цей параметр (top) може набувати таких значень:

- 0 - за замовчуванням за весь час
- 1 - за годину
- 2 - за сьогодні
- 3 - за три дні
- 4 - за тиждень
- 5 - за місяць
- 6 - за 3 місяці
- 8 - за 3 години
- 9 - за 6 годин
- 10 - за 2 дні
- 11 - за добу
- 14 - за 12 годин

Актуальність

Цей параметр (saledParam) може набувати таких значень:

- 0 - всі автомобілі
- 1 - показувати тільки продані авто
- 2 - не відображати продані автомобілі

VIN код

Цей параметр може набувати таких значень:

- 0 - не включати до запити пошуку
- 1 - включити до запити пошуку

Колісна формула

Цей параметр (`wheelFormulaId`) може набувати таких значень:

0 - за замовчуванням

1 - 4x2

2 - 4x4

3 - 6x2

4 - 6x4

5 - 6x6

7 - 8x2

8 - 8x4

9 - 8x6

10 - 8x8

11 - 10x4

12 - 10x8

Кількість осей

Цей параметр (`axleId`) може набувати таких значень:

0 - за замовчуванням

1 - одна вісь

2 - дві осі

3 - три осі

4 - чотири осі

5 - більше чотирьох осей

Тільки з Фото

Цей параметр може набувати таких значень:

0 - не включати в запит пошуку

1 - включити в запит пошуку

Тільки з Відео

Даний параметр може набувати таких значень:

0 - не включати в запит пошуку

1 - включити в запит пошуку

Кількість відображуваних id

Цей параметр може набувати таких значень від 1 до 100

1.1.2 API Пошуку

Щоб звернутися до RIA API search, Вам необхідно виконати GET запит такого вигляду:

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

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

Він складається з:

- **SEARCH** — назва методу API, до якого Ви хочете звернутися.
- **API_KEY** — ключ доступу. Для отримання ключа доступу потрібно зареєструватися на платформі developers.ria.com
- **PARAMETERS** — вхідні параметри, послідовність пар name=value, розділених амперсандом. Список параметрів вказано *тут*

У відповідь на такий запит Ви отримаєте відповідь у форматі JSON:

```
[
  {
    "additional_params": {
      "lang_id": 2,           // Російська мова
      "page": 0,            // Порядковий номер сторінки
      "view_type_id": 0,
      "target": "search",
      "section": "auto",    // Пошук по авто
      "catalog_name": "",
      "elastica": true,
      "nodejs": true
    },
    "result": {             // Результат пошуку
      "search_result": {
        "ids": [           // id оголошень
          ....
          ....
          ....
        ],
        "count": 11,       // Кількість id оголошень доступних за ↵
        ↵ заданими параметрами
        "last_id": 0,
        "qs": {
          "fields": [
            "_id"
          ],
          "size": 50,      // Кількість id оголошень, що ↵
        ↵ відображаються
        "from": 0,
      }
    }
  ]
```

Приклад

Припустимо, Ви шукаєте:

- Легкові машини (`category_id=1`)
- Кузов *Седан, Універсал* (`bodystyle[0]=3&bodystyle[4]=2`)
- Японія (`brandOrigin[0]=276`)
 - Toyota (`marka_id[0]=79`)
 - Всі моделі (`model_id[0]=0`)
 - Рік випуску від 2010 до 2017р. (`s_yers[0]=2010&po_yers[0]=2017`)
- Німеччина (`brandOrigin[1]=392`)
 - Volkswagen (`marka_id[1]=84`)
 - Всі моделі (`model_id[1]=0`)
 - Рік випуску від 2012 до 2016р. (`s_yers[1]=2012&po_yers[1]=2016`)
- Ціна від 1000 до 60000 (`price_ot=1000&price_do=60000`)
 - Ціна вказана в доларах США (`currency=1`)
- Можливий торг (`auctionPossible=1`)
- Область
 - Вінницька - пошук по всіх містах цієї області (`state[0]=1&city[0]=0`)
 - Житомирська - пошук по всіх містах цієї області (`state[1]=2&city[1]=0`)
 - Київська - пошук по всіх містах цієї області (`state[2]=10&city[2]=0`)
- Не відображати авто, які знаходяться не в Україні (`abroad=2`)
- Не відображати нерозмитнені авто (`custom=1`)
- Гаражне зберігання (`auto_options[477]=477`)
- Паливо
 - Бензин (`type[0]=1`)
 - Дизель (`type[1]=2`)
 - Газ/бензин (`type[3]=4`)
 - Електро (`type[5]:6`)
- КПП
 - Ручна / Механіка (`gearbox[0]=1`)
 - Автомат (`gearbox[1]=2`)
 - Типтронік (`gearbox[2]=3`)
- Об'єм 1.4 - 3.2 л. (`engineVolumeFrom=1.4&engineVolumeTo=3.2`)
- Потужність 90 - 250 (`powerFrom=90&powerTo=250`)
 - Одиниця виміру потужності - к.с. (`power_name=1`)
- Тільки з фото (`with_photo=1`)

У підсумку ми отримуємо запит такого [типу]:

[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&...)

У разі успішного виконання запиту за вказаними параметрами результат буде зі статусом **200 ОК**.

Приклад успішної відповіді:

```
[
  {
    "additional_params": {
      "lang_id": 2,           // Російська мова
      "page": 0,             // Порядковий номер сторінки
      "view_type_id": 0,
      "target": "search",
      "section": "auto",     // Пошук по авто
      "catalog_name": "",
      "elastica": true,
      "nodejs": true
    },
    "result": {              // Результат пошуку
      "search_result": {
        "ids": [
          "19519211",
          "19684763",
          "19493489",
          "19714833",
          "19393702",
          "19692238",       // ід оголошень
          "19574398",
          "18154136",
          "19391327",
          "18693600",
          "19431563",
          "18047892"
        ],
        "count": 1,         // Кількість ід оголошень доступних за заданими ↵
        ↪ параметрами
        "last_id": 0,
        "qs": {
          "fields": [
            "_id"
          ],
          "size": 50,       // Кількість відображуваних ід оголошень
          "from": 0,
        }
      }
    }
  }
]
```

Важливо

Максимальна кількість ід оголошень, що відображаються, за один запит дорівнює 100(*countpage*). У разі якщо результат пошуку перевищує це значення можна додати параметр *page* (порядковий номер сторінки) за допомогою якого можна переглянути всі результати пошуку.

```
"additional_params": {
  "lang_id": 2,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"page": "1", // Порядковий номер сторінки
"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",
      . . . . .
      "19860637",
      "19747721",
      "18050784",
      "19810589",
      "19746765",
      "18412097",
      "18545537"
    ],
    "count": 1578, // Кількість id за результатами пошуку
    "last_id": 0,
    "qs": {
      "fields": [
        "_id"
      ],
      "size": 100,
      "from": 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

- 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
- 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 - выберет для подсчета средней цены все объявления с пробегом от 10 до 100 тыс. км.

Если передать массив значений в параметре *опции* 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 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,
  ]
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    45800,
    50000,
    1800,
    4350,
    4400,
    2850
  ],
  classifieds: [
    14663610,
    14226353,
    14138132,
    13969588,
    14697569,
    13386778,
    13279188,
    14555863,
    14754932,
    14816842,
    14664706,
    13873344,
    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&ra

Средняя цена для Honda Accord в Киеве - https://developers.ria.com/auto/average_price?api_key=YOUR_API_KEY&m

1.1.4 AUTO info Інформація по id оголошення

Щоб звернутися до RIA API info, Вам необхідно виконати GET запит такого виду:

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

або

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

Він складається з:

- **info** — назва методу API, до якого Ви хочете звернутися.
- **API_KEY** — ключ доступу. Для отримання ключа доступу потрібно зареєструватися на платформі developers.ria.com
- **auto_id=** — номер оголошення за яким потрібна інформація

Приклад

Припустимо, Вам потрібна інформація по *19050985* id.

Цей запит матиме такий вигляд:

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

або

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

В разі успішного виконання запиту результат буде зі статусом **200 OK**.

Приклад успішної відповіді:

```
[
  {
    "userId": 489269, // id_
    ↪ користувача
    "chipsCount": 0,
    "locationCityName": "Дніпро", // Місто
    "auctionPossible": false, // Торг
    "exchangePossible": false, // Обмін_
    ↪
    "realtyExchange": false, // Обмін_
    ↪ на нерухомість
    "exchangeType": "Будь-який", // Тип_
    ↪ обміна
    "exchangeTypeId": 0, // _
    ↪ Ідентифікатор типу обміну
    "addDate": "2017-04-24 23:02:06", // Дата_
    ↪ і час додавання
    "updateDate": "2017-04-24 23:02:06", // Дата_
    ↪ і час оновлення
    "expireDate": "2017-07-24 23:02:06", // Дата_
    ↪ і час закінчення
    "userHideADSStatus": false, // _
    ↪ Користувач приховав статус
  }
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"userPhoneData": {
  "phoneId": "xxxxxxxxxx", // id
  ↪ телефону користувача (дані не виводяться)
  "phone": "(xxx) xxx xx xx" //
  ↪ Телефон користувача (дані не виводяться)
},
"USD": 33500, //
↪ Вартість у доларах США
"UAH": 880380, //
↪ Вартість у гривнях
"EUR": 30815, //
↪ Вартість у євро
"isAutoAddedByPartner": false,
"partnerId": 0,
"levelData": {
  "level": 10, //
  ↪ Рівень топа
  "label": 3, // Мітка
  "hotType": "терміново", // Мітка
  ↪ "ТОР" або "Терміново"
  "expireDate": "2017-05-04 23:02:07" // Дата
  ↪ і час закінчення мітки "ТОР" "Терміново"
},
"autoData": { //
  ↪ Характеристики автомобіля
  "description": "Отже. Настав час терміново продати свій улюблений автомобіль. " +
    "\r\nКупував його в серпні 2013 року. \Повністю обслужений на о", // Опис

  "version": "",
  "onModeration": false,
  "year": 2013, // Рік
  ↪ випуску
  "autoId": 19050985, // id
  ↪ оголошення
  "statusId": 0, // -
  "withVideo": false, //
  ↪ Наявність відео
  "race": "60 тыс. км", // Пробіг
  "raceInt": 60, // -
  "fuelName": "Дизель", // Тип
  ↪ палива
  "gearboxName": "Автомат", // Тип
  ↪ приводу
  "isSold": false, // Цей
  ↪ параметр показує чи машину продано чи ні
  "mainCurrency": "USD", //
  ↪ Основна валюта
  "fromArchive": false,
  "categoryId": 1, // Тип
  ↪ транспорту
  "custom": 0
},
"markName": "Volvo", // Марка
  ↪ автомобіля
  "markId": 85, // id
  ↪ марки автомобіля

```

(continues on next page)

(продолжение с предыдущей страницы)

```

"modelName": "XC90", // u
↪ Модель автомобіля
"modelId": 824, // id u
↪ Моделі автомобіля
"photoData": { // Фото
  "count": 15, // u
  ↪ Кількість фотографій
  "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", // u
↪ Посилання на оголошення
"title": "Volvo XC90", // Назва
"stateData": {
  "name": "Дніпро", // Місто
  "regionName": "Дніпропетровська", // u
  ↪ Область
  "linkToCatalog": "/city/dnepropetrovsk/",
  "title": "Пошук оголошень по місту Дніпро",
  "stateId": 11 // id u
  ↪ Міста
},
"oldTop": {
  "isActive": true,
  "expireDate": ""
},
"canSetSpecificPhoneToAdvert": false,
"dontComment": 0,
"sendComments": 0,
"badges": [
],
"checkedVin": { // u
  ↪ Перевірка VIN-коду
  "isShow": false, // u
  ↪ Відображати VIN-код
  "linkToReport": "/vin-check/auto/19050985/",
  "hasRestrictions": false, // u
  ↪ Обмеження
  "checkDate": "04.05.2017",
  "isChecked": false // VIN-
  ↪ код перевірено
},
"isLeasing": 0, // Лізинг
"dealer": {
  "link": "",
  "logo": "",
  "type": "",
  "id": 0,
  "name": "",
  "packageId": 0,
  "typeId": 0
},
"withInfoBar": false, // u
↪ Інформаційна панель

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"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

Додавання оголошення

Щоб додати оголошення, Вам необхідно виконати POST запит такого типу:

```
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 "{ \ \ }"
```

де *user_id* - Ваш ID в системі RIA.com, *-d "{ \ \ }"* - тут Ви вказуєте масив параметрів які хочете додати, *api_key* - Ваш ключ.

Оголошення за замовчуванням додаються в чернетки, щоб опублікувати їх, Вам потрібно зайти в свій особистий кабінет на AUTO.RIA

Також варто зауважити, що існують параметри обов'язкові для заповнення. Детально з параметрами можна ознайомитися [тут](#).

Приклад запити

```
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\": \"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}"
```

Розгорнутый приклад

```
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,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    \ "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,
    вкладень не потребує\ "
  },
  \ "doors\ ":2,
  \ "seats\ ":2,
  \ "country\ ":{
    \ "import\ ":{
      \ "id\ ":0
    }
  },
  \ "spareParts\ ":false
} "

```

Приклад успішної відповіді:

```

{
  "damage":false,
  "custom":false,
  "year":2016,
  "price":{
    "value":80000,
    "currency":{
      "id":1
    }
  },
  "categories":{
    "main":{
      "id":1
    }
  },

```

(continues on next page)

```
"all":[
  {
    "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":"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
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    },
    "exchanges":{
      "payment":{
        "id":2
      },
      "type":{
        "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,
  "user":{
    "id":7069830
  },
  "ip":"80.91.174.90",
  "dates":{
    "created":"2017-09-18T12:07:43.840Z"
  },
  "status":{
    "id":13
  },
  "_id":20476120
}

```

Список оголошень користувача

Щоб отримати список оголошень, Вам необхідно виконати GET запит такого формату:

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

або

```

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

```

де `user_id` - Ваш ID в системі RIA.com, `api_key` - Ваш ключ.

Приклад запиту

```

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

```

Приклад успішної відповіді:

```
{
  "active": {
    "moderated": [

    ],
    "notModerated": [

    ]
  },
  "draft": [

  ],
  "awaitingPayment": [

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

Розшифрування параметрів:

- *active* - Активні оголошення
- *moderated* - Оголошення перебувають на модерації
- *notModerated* - Оголошення не відмодеровано
- *draft* - Оголошення знаходяться в чернетках
- *awaitingPayment* - Оголошення очікують оплати
- *archive* - Оголошення в архіві

Повний опис сервісу «Список оголошень користувача» описано за допомогою стандарту **DeFacto swagger 2.0** [тут](#)

Статистика оголошення

Щоб отримати статистику про оголошення, Вам необхідно виконати GET запит такого типу:

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

або

```
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"
```

де *user_id* - Ваш ID в системі RIA.com, *advertisementId* - ID потрібного Вам оголошення, *api_key* - Ваш ключ.

Приклад запиту

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

Приклад успішної відповіді:

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

Розшифрування параметрів:

- *views* - Кількість переглядів оголошення
- *clicks* - Кількість відкриттів номера телефону

Повний опис сервісу «Статистика оголошення» описаний за допомогою стандарту **DeFacto swagger 2.0** [тут](#)

Отримання основної інформації про оголошення

Щоб отримати основну інформацію про оголошення, Вам необхідно виконати GET запит такого вигляду:

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

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

де *user_id* - Ваш ID в системі RIA.com, *advertisementId* - ID потрібного Вам оголошення, *api_key* - Ваш ключ.

Приклад запиту

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

Приклад успішної відповіді:

```
{
  "_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": {
```

(continues on next page)

```
"id": 1
},
"color": {
  "id": 10,
  "metallic": true
},
"year": 2016,
"mileage": 32,
"price": {
  "value": 80000,
  "currency": {
    "id": 1
  }
},
"post": {
  "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
```

(continues on next page)

(продолжение с предыдущей страницы)

```
},
"moderated": true,
"damage": false,
"custom": false,
"photos": {
  "main": {
    "id": 196054982
  }
},
"credit": false,
"video": {
  "key": "LLEiT9PeHSg"
},
"categories": {
  "main": {
    "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
  }
}
}
```

Повний опис сервісу «Отримання основної інформації про оголошення» описаний за допомогою стандарту **DeFacto swagger 2.0** [тут](#)

Отримання списку фотографій оголошення

Для отримання списку фотографій оголошення, Вам необхідно виконати GET запит такого типу:

„https://developers.ria.com/auto/fotos/advertisementId?api_key=YOUR_API_KEY“

де *advertisementId* - ID потрібного Вам оголошення, *api_key* - Ваш ключ.

Приклад запиту

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

Приклад успішної відповіді:

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

(continues on next page)

(продолжение с предыдущей страницы)

```

    "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,
    "date_add": "2017-06-04 13:27:40",
    "description": null,
    "url": "auto/photo/18720/1872031/187203138/187203138.jpg",
    "formats": [
      "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203138/187203138f.jpg",
      "https://cdn.riastatic.com/photos/auto/photo/18720/1872031/187203138/187203138fx.jpg"
    ]
  }
}
}
}
}
}

```

Зміна основної інформації оголошення

Для зміни основної інформації оголошення, Вам необхідно відправити **PUT** запит такого типу:

```

curl -X PUT "https://developers.ria.com/auto/used/autos/advertisementId?user_id=Ваш_ID&api_
↵key=YOUR_API_KEY" -H "accept: application/json" -H "content-type: application/json" -d "{ \ \ }
↵}"

```

де *user_id* - Ваш ID в системі RIA.com, *advertisementId* - ID потрібного Вам оголошення, *-d "{ \ \ }"* - тут Ви вказуєте масив параметрів які хочете змінити, *api_key* - Ваш ключ.

Детально з параметрами і структурою можна ознайомитися *тут*

Приклад запиту

Припустимо, Вам потрібно змінити ціну і валюту, в якій ви продаєте ваш автомобіль. Цей запит матиме такий вигляд:

```

curl -X PUT "https://developers.ria.com/auto/used/autos/20438832?user_id=7069830&api_key=YOUR_API_
↵KEY" -H "accept: application/json" -H "content-type: application/json" -d "{ \"price\": { \
↵\"value\":10000, \"currency\": { \"id\":1 } } }"

```

или изменить главное фото объявления, данный запрос будет иметь следующий вид:

```

curl -X PUT "https://developers.ria.com/auto/used/autos/20438832?user_id=7069830&api_key=YOUR_API_
↵KEY" -H "accept: application/json" -H "content-type: application/json" -d "{ \"photos\": { \
↵\"main\": { \"id\": 16652206 } } }"

```

Об'єкт *price* містить у собі параметри *value* - значення, *currency* - ID валюти, в якій ви продаєте ваш автомобіль, у нашому випадку це долар.

Приклад успішної відповіді:

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

```

Повний опис сервісу «Зміна основної інформації оголошення» описаний за допомогою стандарту DeFacto swagger 2.0 [тут](#)

Важливо

Основні параметри транспортного засобу доступні до редагування протягом години, після додавання на сайт:

- «Region»
- «Тип транспорту», «Марка», «Модель», «Версія», «Рік випуску»
- «VIN-номер», «Коробка передач», «Привід», «Кількість дверей», «Колір», «Паливо»

Решту параметрів - «Ціна», «Пробіг», «Фото» автомобіля, докладний опис - можете змінювати і коригувати в будь-який час.

Якщо ж після закінчення 60 хв. відправити запит на зміну одного з вище перерахованих параметрів, Ви отримаєте помилку.

Приклад помилки

```
{
  "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}
↪ ]}"
}
```

Видалення оголошення

Для видалення оголошення, Вам необхідно відправити **DELETE** запит такого формату:

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

де *user_id* - Ваш ID в системі RIA.com, *advertisementId* - ID потрібного Вам оголошення, *reason_id* - причина видалення, *api_key* - Ваш ключ.

reason_id складається з 4 категорій:

- *reason_id*=4 - Авто продано за допомогою AUTO.RIA.com
- *reason_id*=5 - Авто продано в іншому місці
- *reason_id*=6 - Я хочу розмістити оголошення знову
- *reason_id*=7 - Я передумав продавати автомобіль

Приклад запиту

```
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"
```

Повний опис сервісу «Видалення оголошення» описаний за допомогою стандарту **DeFacto swagger 2.0** [тут](#)

Отримання інформації про опції оголошення

Щоб отримати інформацію про опції оголошення, Вам необхідно виконати GET запит такого формату:

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

або `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"`

де *user_id* - Ваш ID в системі RIA.com, *advertisementId* - ID потрібного Вам оголошення, *api_key* - Ваш ключ.

Приклад запиту

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

Приклад успішної відповіді:

```
[
  {
    "id": 125
  },
  {
    "id": 137
  },
  {
    "id": 188
  },
  {
    "id": 189
  },
  {
    "id": 190
  },
  {
    "id": 191
  },
  {
    "id": 211
  },
  {
    "id": 217
  },
  {
    "id": 225
  },
  {
    "id": 303
  },
  {
    "id": 354
  },
  {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[
  "id": 355
},
{
  "id": 437
},
{
  "id": 463
},
{
  "id": 481
},
{
  "id": 525
}
]
```

Для розшифровки отриманих параметрів можна використовувати сервіс Опції

Повний опис сервісу «Отримання інформації про опції оголошення» описаний за допомогою стандарту **DeFacto swagger 2.0** [тут](#)

Додавання опцій

Для додавання опцій в оголошення, Вам необхідно виконати PUT запит такого виду:

```
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 опції }]"
```

де *user_id* - Ваш ID в системі RIA.com, *advertisementId* - ID потрібного Вам оголошення, «id»: - ID потрібної Вам опції, *api_key* - Ваш ключ.

Приклад запиту

```
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 }]"
```

Приклад успішної відповіді:

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

Повний опис сервісу «Додавання опцій» описаний за допомогою стандарту **DeFacto swagger 2.0** [тут](#)

Видалення опцій

Для видалення опцій оголошення, Вам необхідно відправити **DELETE** запит такого формату:

```
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"
```

де *user_id* - Ваш ID в системі RIA.com, *advertisementId* - ID потрібного Вам оголошення, «id»: - ID потрібної Вам опції, *api_key* - Ваш ключ.

Приклад запиту

```
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"
```

Додавання фото до оголошення

Для додавання фотографій до оголошення, Вам необхідно виконати POST запит такого типу:

```
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": "головне фото", "links": [ "фотографії" ]}'
```

Розшифрування параметрів:

- *advertisementId* - ID потрібного Вам оголошення
- *user_id* - Ваш ID у системі RIA.com
- *api_key* - Ваш ключ
- *-d '{"main": "посилання на головну фотографію", "links": ["посилання на фотографії"]}'* - тут Ви вказуєте масив фотографій, які хочете додати. У кожному посиланні на зображення має бути присутній формат

Фотографії будуть додані не за черговістю, а у випадковому порядку.

Повний опис сервісу «Додавання фотографій до оголошення» описаний за допомогою стандарту DeFacto swagger 2.0 [тут](#)

Приклад запиту

```
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" ]}'
```

Приклад успішної відповіді

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

1.2 AUTO.RIA Нові авто

1.2.1 Методи для роботи із загальнодоступними даними розділу «Нові авто»

Типи транспорту

Щоб отримати список типів транспорту, потрібно надіслати GET запит на адресу https://developers.ria.com/auto/categories/?api_key=YOUR_API_KEY. Результат буде наступним:

```
[
  {"name": "Легкові", "value": 1},
  {"name": "Мото", "value": 2},
  {"name": "Водний транспорт", "value": 3},
  {"name": "Спецтехніка", "value": 4},
  {"name": "Причепи", "value": 5},
  {"name": "Вантажівки", "value": 6},
  {"name": "Автобуси", "value": 7},
  {"name": "Автобудинки", "value": 8},
  {"name": "Повітряний транспорт", "value": 9},
  {"name": "Сільгосптехніка", "value": 10}
]
```

Марки

Марки залежать від типів транспорту. Тому для того, щоб отримати список марок, необхідно надіслати GET запит за адресою https://developers.ria.com/auto/new/marks?category_id=id&api_key=YOUR_API_KEY, де *category_id* - ідентифікатор типу транспорту, *api_key* - Ваш ключ.

Наприклад, для легкових автомобілів (https://developers.ria.com/auto/new/marks?category_id=1&api_key=YOUR_API_KEY) результат буде наступним:

```
[
  {
    "category_id": 1,
    "cnt": 1,
    "country_id": 0,
    "eng": "abarth",
    "marka_id": 5166,
    "name": "Abarth",
    "slang": "Абарт",
    "value": 5166
  },
  {
    "category_id": 1,
    "cnt": 369,
    "country_id": 392,
    "eng": "acura",
    "marka_id": 98,
    "name": "Acura",
    "slang": "Акура",
    "value": 98
  },
  .....
]
```

Моделі

Моделі залежать від типів транспорту і марок. Відповідно список марок можна отримати за адресою https://developers.ria.com/auto/new/models?marka_id=id&category_id=id&api_key=YOUR_API_KEY, де *category_id* - ідентифікатор типу транспорту, а *marka_id* - ідентифікатор марки, *api_key* - Ваш ключ.

Наприклад, для легкового автомобіля BMW (https://developers.ria.com/auto/new/models?marka_id=9&category_id=1& список моделей буде наступним:

```
[
  {
    "category_id": "1",
    "count": 0,
    "eng": "ix1",
    "marka_id": 9,
    "model_id": 63528,
    "name": "iX1",
    "parent_id": 0,
    "slang": null,
    "value": 63528
  },
  {
    "category_id": "1",
    "count": 0,
    "eng": "1500",
    "marka_id": 9,
    "model_id": 63521,
    "name": "1500",
    "parent_id": 0,
    "slang": null,
    "value": 63521
  },
  .....
]
```

Покоління моделі нового авто

Покоління залежить від моделі нового авто. Отже список поколінь можна отримати надіславши GET запит на адресу https://developers.ria.com/auto/new/generations?model_id=id&api_key=YOUR_API_KEY, де *model_id* - модель обраного авто, *api_key* - Ваш ключ.

Наприклад, для моделі BMW X6 (https://developers.ria.com/auto/new/generations?model_id=2153&api_key=YOUR_API_KEY) список поколінь буде таким:

```
[
  {
    "eng": "e71-f1",
    "generation_id": 432,
    "marka_id": 9,
    "model_id": 2153,
    "name": "E71 (FL)",
    "year_from": 2012,
    "year_to": 2014
  },
  {
    "eng": "f16",
    "generation_id": 433,
    "marka_id": 9,
    "model_id": 2153,
    "name": "F16",
    "year_from": 2014,
    "year_to": 0
  }
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
},
{
  "eng": "e71",
  "generation_id": 3275,
  "marka_id": 9,
  "model_id": 2153,
  "name": "E71",
  "year_from": 2008,
  "year_to": 2011
},
{
  "eng": "g06",
  "generation_id": 7541,
  "marka_id": 9,
  "model_id": 2153,
  "name": "G06",
  "year_from": 2019,
  "year_to": 0
}
]
```

Типи кузова в поколінні нових авто

Тип авто кузова залежить від покоління авто. Тому для того, щоб отримати список типів кузова, необхідно надіслати GET запит на адресу https://developers.ria.com/auto/new/generation_bodystyles?generation_id=id&api_key=YOUR_API_KEY, де *generation_id* - покоління авто, *api_key* - Ваш ключ.

Наприклад, для моделі BMW X6 у поколінні E71 (рестайлінг) (https://developers.ria.com/auto/new/generation_bodystyles?generation_id=432&api_key=YOUR_API_KEY), список типів кузова буде таким:

```
[
  {
    "bodystyle_id": 324,
    "eng": "krossover",
    "generation_bodystyle_id": 538,
    "name": "Кросовер"
  }
]
```

Модифікації (бази) за типом кузова в поколінні

Модифікація (бази) залежить від типу кузова в поколінні. Отримати цей список можна надіславши GET запит за адресою https://developers.ria.com/auto/new/generation_bodystyles_bases?generation_bodystyle_id=id&api_key=YOUR_API_KEY, де *generation_bodystyle_id* - id типів кузова, *api_key* - Ваш ключ.

Наприклад, для моделі BMW X6 у поколінні E71 (рестайлінг) (https://developers.ria.com/auto/new/generation_bodystyles_bases?generation_bodystyle_id=538&api_key=YOUR_API_KEY), список модифікацій буде таким:

```
[
  {
    "base_id":9036,
    "generation_bodystyle_id":538,
    "marka_id":9,
    "model_id":2153,
    "name":"50i Steptronic (407 к.с.) xDrive"
  },
  {
    "base_id":15821,
    "generation_bodystyle_id":538,
    "marka_id":9,
    "model_id":2153,
    "name":"40d Steptronic (306 к.с.) xDrive"
  },
  {
    "base_id":166827,
    "generation_bodystyle_id":538,
    "marka_id":9,
    "model_id":2153,
    "name":"30d Steptronic (245 к.с.) xDrive"
  }
]
```

Комплектації авто за модифікацією

Комплектації авто залежать від модифікації (бази). Отримати їх список можна надіславши GET запит за адресою https://developers.ria.com/auto/new/equips?base_id=id&api_key=YOUR_API_KEY, де *base_id* - id модифікації (бази), *api_key*- Ваш ключ.

Наприклад, для моделі BMW X6 E71 50i AT (407 к.с.) xDrive (https://developers.ria.com/auto/new/equips?base_id=9036&api_key=YOUR_API_KEY) список комплектації буде таким:

```
[
  {
    "base_id":9036,
    "cnt":0,
    "equip_id":79274,
    "min_price":null,
    "name": "Base",
    "year":"2022"
  }
]
```

1.2.2 Методи для роботи з Кабінетом автосалону

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

(continues on next page)

(продолжение с предыдущей страницы)

```
"active": [  
  1696177  
]  
}
```

Расшифровка параметров

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

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  
}
```

Редагування оголошення

Для редагування основної інформації оголошення, вам необхідно надіслати **PUT** запит такого типу:

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

Обов'язковими параметрами є:

- *user_id* - Ваш ID у системі RIA.com
- *auto_id* - ID потрібного Вам оголошення
- *currency_id* - Валюта в якій продаєте автомобіль (1 - долари США, 2 - євро, 3 - гривня)
- *Content-Length* - розмір документа, що повертається (Content-Length: 0)

Додаткові параметри:

- *price* - Ціна
- *auto_note* - Опис
- *version* - Версія (наприклад 2.0 TDI)
- *in_stock* - У наявності (1 - у наявності, 0 - немає в наявності)
- *test_drive* - Можливість пройти тест драйв (1 - є можливість, 0 - немає можливості)
-

Приклад запиту

```
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"
```

Приклад успішної відповіді:

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

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}
```

Отримання даних про дзвінок

GET API для отримання даних про дзвінок

Щоб отримати дані про дзвінки, вам необхідно виконати GET-запит такого формату:

`https://developers.ria.com/auto/new/calls_stat?user_id=Ваш_ID&from=ДАТА_ПОЧАТКУ_ВИБІРКИ&to=ДАТА_КІНЦЯ_ВИБІРКИ&api_key=YOUR_API_KEY`
або

```
curl -X GET "https://developers.ria.com/auto/new/calls_stat?user_id=Ваш_ID&from=ДАТА_ПОЧАТКУ_
↳ВИБІРКИ&to=ДАТА_КІНЦЯ_ВИБІРКИ&api_key=YOUR_API_KEY" -H "accept: application/json"
```

де *user_id* - Ваш ID в системі RIA.com; *from*, *to* - фільтр за діапазоном дат (рік - місяць - день); *api_key* - Ваш ключ.

Приклад запиту

```
curl -X GET "https://developers.ria.com/auto/new/calls_stat?from=2021-06-01&to=2021-07-31&user_
↳id=1505029&api_key=YOUR_API_KEY" -H "accept: application/json"
```

Приклад успішної відповіді: [1]:

```
[
  {
    "call_id": 22346351,
    "caller_phone": "0951174074",
    "date": "2021-07-31T15:57:01",
    "destination_phone": "0731375686",
    "duration": 0,
    "link": "",
    "marka": "Acura",
    "marka_id": 98,
    "model": "MDX",
    "model_id": 955,
    "phone": "0730946407",
    "price": 0,
    "status": "CANCEL",
    "waighting": 11
  },
  {
    "call_id": 22345654,
    "caller_phone": "0968404829",
    "date": "2021-07-31T14:42:24",
    "destination_phone": "0731375686",
    "duration": 0,
    "link": "",
    "marka": null,
    "marka_id": 0,
    "model": null,
    "model_id": 0,
    "phone": "0504103684",
    "price": 0,
    "status": "BUSY",
    "waighting": 48
  },
  {
    "call_id": 22326708,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"caller_phone": "0674294559",
"date": "2021-07-30T17:00:12",
"destination_phone": "0731375686",
"duration": 57,
"link": "https://track.ria.com/call_link/56w456gw45v5765ee5ser_w45/2021/07/30/
↪380731375686=380674294559=17-00-12.wav",
"marka": "Acura",
"marka_id": 98,
"model": "MDX",
"model_id": 955,
"phone": "0935063003",
"price": 0,
"status": "ANSWER",
"waighting": 22
},
{
"call_id": 22325473,
"caller_phone": "09314226064",
"date": "2021-07-30T15:30:21",
"destination_phone": "0731375686",
"duration": 0,
"link": "",
"marka": "Acura",
"marka_id": 98,
"model": "MDX",
"model_id": 955,
"phone": "0936073263",
"price": 0,
"status": "CANCEL",
"waighting": 44
}
]
```

Розшифрування параметрів:

- *call_id* - Унікальний ID дзвінка
- *caller_phone* - Номер телефону, з якого був дзвінок
- *date* - Дата і час дзвінка
- *destination_phone* - Реальний номер на який переадресовувався дзвінок
- *duration* - Тривалість дзвінка в секундах
- *link* - Посилання на запис розмови
- *marka* - Марка, якою цікавилися під час розмови
- *marka_id* - ID марки
- *model* - Модель, якою цікавилися під час розмови
- *model_id* - ID моделі
- *phone* - Підмінний номер телефону, на який дзвонили
- *status* - Статус дзвінка (ANSWER, NOANSWER, CANCEL, BYSY, FAILED, CONGESTION)
- *waighting* - Час у секундах, перш ніж відбулося з'єднання

POST API для отримання даних про дзвінок

Другий спосіб передачі даних про дзвінок:

1. У вас є back-end із веб-сервером.
2. Ви передаєте свій url API (наприклад: https://your_site.com/api/call/call_from_auto_ria).
3. Щойно клієнт ставить слухавку, на цю адресу ми відправляємо всі дані (список параметрів в [п.1][1]).
4. Ви обробляєте отримані дані на своєму боці.

Ліміти Веб-Сервісів

Обмеження поширюються на кількість запитів до веб-служб RIA.com, які Ви можете створити, використовуючи свій API ключ.

Годинний ліміт: 1,000 запитів на годину.

Ці обмеження застосовуються до всіх веб-служб RIA.com. Перевищення цих лімітів призведе до тимчасового блокування Вашого API ключа. Залежно від перевищеної межі, блок буде автоматично скасовано протягом наступної години або календарного дня.

Погодинні лічильники для API ключа анулюються на постійній основі.

Наприклад: Якщо Ви зробите 500 запитів о 10:15 і 500 запитів о 10:25, Ваш API ключ буде тимчасово заблоковано. Блок буде анульовано об 11:15, після чого Ви зможете зробити ще 500 запитів.

Обмеження Запитів

Якщо Ваш API ключ перевищує ліміти запитів, Ви отримаєте відповідь з HTTP статус-кодом 429 (занадто багато запитів).

Потрібно підвищити ліміти?

Якщо Ви створюєте застосунок, який вимагає більш високих лімітів, ми будемо раді співпрацювати. Зв'яжіться з нами для більш детальної інформації.

2.1 Параметри для роботи з API DOM.RIA

2.1.1 Продаж квартир/кімнат

2.1.2 Оренда квартир/кімнат

2.1.3 Продаж будинків

2.1.4 Оренда будинків

2.1.5 Продаж офісів

2.1.6 Оренда офісів

2.1.7 Продаж/оренда комерційної нерухомості

2.1.8 Продаж/оренда земельних ділянок

2.1.9 Продаж/оренда гаражів

2.2 Список характеристик нерухомості

Отримати список областей можна зробивши GET запит https://developers.ria.com/dom/options?category=id&realty_type, де

- *category* - тип об'єкта
- *realty_type* - тип нерухомості
- *operation_type* - тип операції

або

```
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"
```

Повний опис сервісу «Список характеристик нерухомості» описаний за допомогою стандарту DeFacto swagger 2.0 [тут](#)

Приклад

Припустимо Вам потрібно отримати список характеристик об'єкта „Будинки“, а тип нерухомості - „Дачі“, тип операції - „Продаж“

Цей запит матиме такий вигляд:

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

або

```
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 буде наступним:

```
[
{
  "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": "horisontal",
      "group_orientation_add": "horisontal",
      "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"
        },
        "130": {
          "characteristic_id": "130",
          "name": "кирпич силикатный",
          "name_uk": "цегла силікатна",

```

(continues on next page)

```
"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": {
  "characteristic_id": "140",
  "name": "каркасно-щитовой",
  "name_uk": "каркасно-щитовий",
  "prio": "90"
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"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": "сэндвіч-панелі",
  "prio": "50"
},
"145": {
  "characteristic_id": "145",
  "name": "метал",
  "name_uk": "метал",
  "prio": "40"
},
"146": {
  "characteristic_id": "146",
  "name": "порогтерм",
  "name_uk": "порогтерм",
  "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",
  "name_uk": "шлакобетон",
  "prio": "65"
},
"1441": {
  "characteristic_id": "1441",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "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",
"group_orientation_add": "horisontal",
"empty_value": 0,
"type": "characteristic",
"group_prio": 0
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
]
},
{
  "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"
    },
    {
      "characteristic_id": 218,
      "label": "кухня",
      "label_uk": "кухня",
```

(continues on next page)

```
"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",
      "type_on_add": "select",
      "type_on_search": "select",
      "data_type": "int",
      "required": 0,
      "characteristic_category": "main",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "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": "horisontal",
    "group_orientation_add": "horisontal",
    "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",
      "group_name_uk": "поверх",
      "group_prio": 75,
      "display_label_search": 1,
      "display_label_add": 1,
      "group_orientation_search": "horisontal",

```

(continues on next page)

```
"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",
  "type_on_search": "checkbox",
  "data_type": "int",
  "required": 0,
  "characteristic_category": "main",
  "prio": 1,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "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"
        },
        "424": {
          "characteristic_id": "424",
          "name": "2011",
          "name_uk": "2011",
          "prio": "18"
        }
      }
    }
  ]
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```
},
"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",
  "name": "1970-1979",
  "name_uk": "1970-1979",
  "prio": "5"
},
"438": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"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.",
"prio": "24"
},
"1455": {
"characteristic_id": "1455",
"name": "2015, II кв.",
```

(continues on next page)

```
"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"
},
"1471": {
  "characteristic_id": "1471",
  "name": "Сдача в 2020",
  "name_uk": "Здача в 2020",
  "prio": "35"
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    }
  },
  "display_label_search": 0,
  "display_label_add": 0,
  "group_orientation_search": "horizontal",
  "group_orientation_add": "horizontal",
  "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": "horizontal",
      "group_orientation_add": "horizontal",
      "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,
      "field_name": "is_negotiable",
      "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"
},
{
  "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",
  "type_on_search": "checkbox",
  "data_type": "int",
  "required": 0,
  "characteristic_category": "main",
  "prio": 45,
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "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": "horisontal",
    "group_orientation_add": "horisontal",
    "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": "horisontal",
    "group_orientation_add": "horisontal",
    "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": "horisontal",
    "group_orientation_add": "horisontal",
    "empty_value": 0,
    "type": "characteristic",
    "group_prio": 0
  },
  {

```

(continues on next page)

```
"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": {
    "characteristic_id": "263",
    "name": "на недвижимость + ваша доплата",
    "name_uk": "на нерухомість + ваша доплата",
    "prio": "2"
  },
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "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": "horizontal",
  "group_orientation_add": "horizontal",
  "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": {
      "characteristic_id": "1473",
      "name": "от представителя застройщика",
      "name_uk": "від представника забудовника",
      "prio": "3"
    }
  }
},

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "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 Інформація по id оголошення

Щоб звернутися до API info, Вам необхідно виконати GET запит такого виду:

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

або

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

Повний опис сервісу «Інформація по id оголошення» описаний за допомогою стандарту DeFacto swagger 2.0 [тут](#)

Припустимо Вам потрібна інформація щодо id *13825265* .

Цей запит матиме такий вигляд:

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

або

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

У разі успішного виконання запиту результат буде зі статусом **200 OK**.

Приклад успішної відповіді:

```

{
  "street_name": "Онуфрія Трутенка вулиця",
  "metro_station_id": "15",
  "rooms_count": 2,
  "advert_title": 1,
  "type": "realty",
  "is_commercial": 0,
  "state_name": "Київська",
  "street_id": 5581,
  "state_id": 10,
  "beautiful_url": "realty-perevireno-prodaja-kvartira-kiev-goloseevskiy-onufriya-trutenko-ulitsa-
  ↪13825265.html",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

"levels": 64,
"exchange_required_type": 13,
"district_type_name": "Район",
"description": "Светлая квартира, расположена на 23 этаже / 23 эт. дома в ЖК "4 Сезона";, погoлок 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. "Васильковская" ( 5 мин.пешком), м. "Ипподром" (10 мин.пешком), м. "Выставочный Центр"(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. будинку в ЖК "4Сезони";, стеця 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. "Васильківська" ( 5 хв. пішки), м. "Іпподром" (10 хв. пішки), м. "Вистовковий Центр"(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,
  "214": 93,
  "216": 50,
  "218": 20,
  "227": 23,
  "228": 23,
  "234": 60000,

```

(continues on next page)

(продолжение с предыдущей страницы)

```
"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,
    "file": "dom/photo/8070/807045/80704542/80704542.jpg",
    "o": "h"
  },
  "80704543": {
    "id": 80704543,
    "file": "dom/photo/8070/807045/80704543/80704543.jpg",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    "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_pанорамас": 1
}
```

2.4 Підрахунок середньої ціни

“Підрахунок середньої ціни” — сервіс, заснований на актуальних статистичних даних DOM.RIA

Тепер ви можете:

- дізнаватися актуальні середні ціни на нерухомість;
- стежити за змінами цін у коротко- і довгостроковому періодах;
- аналізувати та прогнозувати зміни цін і попиту на нерухомість;
- розміщувати отриману інформацію на вашому сайті.

Ознайомтеся з технічною документацією, щоб отримати доступ і експортувати необхідну інформацію в програму вашої компанії.

З повним описом сервісу «Підрахунок середньої ціни» можна ознайомитися за допомогою стандарту DeFacto **swagger 2.0** [тут](#)

У разі успішного підрахунку середньої ціни за вказаними параметрами результат буде зі статусом **200 OK**.

2.4.1 Формат даних у запиті

Параметри **category**, **sub_category**, **operation** є обов'язковим для заповнення.

Детальну інформацію про застосування цих параметрів можна отримати за допомогою стандарту DeFacto **swagger 2.0** [тут](#)

Приклад успішного запиту:

```
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"
```

Приклад успішної відповіді:

```
{
  "total": 12407,
  "arithmeticMean": 785340.3,
  "prices": [
    {
      "count": 229,
      "value": 60000
    },
    {
      "count": 234,
      "value": 160000
    },
    {
      "count": 239,
      "value": 260000
    },
    {
      "count": 243,
      "value": 360000
    }
  ]
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "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
},
{
  "count": 234,
  "value": 1760000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "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
},
{
  "count": 134,
  "value": 3160000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "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
},
{
  "count": 46,
  "value": 4560000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "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
},
{
  "count": 9,
  "value": 5960000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "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
},
{
  "count": 1,
  "value": 7360000
},
},
```

(continues on next page)

(продолжение с предыдущей страницы)

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

Розшифрування параметрів:

- *total* - загальна кількість оголошень, що беруть участь у підрахунку.
- *arithmeticMean* - середнє арифметичне.
- *count* - кількість оголошень
- *value* - ціна в доларах США

2.5 Пошук оголошень

Щоб скористатись DOM.RIA API search, Вам необхідно виконати GET запит такого виду:

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

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

Він складається з:

- **SEARCH** — назва методу API, до якого Ви хочете звернутися.
- **API_KEY** — ключ доступу. Для отримання ключа доступу потрібно зареєструватися на платформі developers.ria.com
- **PARAMETERS** — вхідні параметри, послідовність параметрів name=value, розділених амперсандом. Список параметрів вказано вище.

На такий запит Ви отримаєте відповідь у форматі JSON:

```
{
  "items": [
    //id оголошень
  ],
  "count":
    // кількість
}
```

Повний опис сервісу «Пошук оголошень» описаний за допомогою стандарту DeFacto swagger 2.0

- Квартир
- Будинків
- Комерційної нерухомості
- Офісів
- Земельних ділянок
- Гаражів

Приклад

Припустимо, Ви шукаєте:

- Об'єкт - Квартири
- Тип нерухомості - квартира
- Операція - продаж
- Область - Київська
- Місто - Київ
- Район
 - Оболонський
 - Печерський
 - Подільський
- Кількість кімнат від 1 до 3
- Загальна площа від 60 до 90
- Житлова площа від 30 до 50
- Кухня від 4 до 9
- Кухня від 3 до 7
- Рік побудови - не вказано
- Ціна від 20000 до 90000
- Ціна за об'єкт
- Тип валюти долари США
- Можливий торг
- Тип пропозиції - від посередника

В результаті ми отримуємо запит такого [типу]:

https://developers.ria.com/auto/search?api_key=YOUR_API_KEY&category=1&realty_type=2&operation_type=1...

або

```
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&
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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"
```

У разі успішного виконання запити за вказаними параметрами результат буде зі статусом **200 ОК**.

Приклад успішної відповіді:

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

2.6 Методи для роботи з областями, містами, районами

2.6.1 Области

Отримати список областей можна надіславши GET запит за адресою https://developers.ria.com/dom/states?api_key=YOUR_API_KEY.

Повний опис сервісу «Методи для роботи з областями, містами, районами» описаний за допомогою стандарту DeFacto swagger 2.0 [тут](#)

Результат буде наступним:

```
[
  {
    "lang_id": 4,
    "stateID": 1,
    "name": "Вінницька",
    "eng_name": "vinnicya",
    "declension": "Вінницької області",
    "center_declension": "Вінниці",
    "region_name": "Вінниця",
    "translit": "vinnitskaya"
  },
  {
    "lang_id": 4,
    "stateID": 18,
    "name": "Волинська",
    "eng_name": "luck",
    "declension": "Волинської області",
    "center_declension": "Луцька",
    "region_name": "Луцьк",
    "translit": "volynskaya"
  },
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "lang_id": 4,
  "stateID": 11,
  "name": "Дніпропетровська",
  "eng_name": "dneprpetrovsk",
  "declension": "Дніпропетровської області",
  "center_declension": "Дніпра",
  "region_name": "Дніпро",
  "translit": "dneprpetrovskaya"
},
{
  "lang_id": 4,
  "stateID": 13,
  "name": "Донецька",
  "eng_name": "doneck",
  "declension": "Донецької області",
  "center_declension": "Донецька",
  "region_name": "Донецьк",
  "translit": "donetskaya"
},
{
  "lang_id": 4,
  "stateID": 2,
  "name": "Житомирська",
  "eng_name": "jytomir",
  "declension": "Житомирської області",
  "center_declension": "Житомира",
  "region_name": "Житомир",
  "translit": "zhitomirskaya"
},
{
  "lang_id": 4,
  "stateID": 22,
  "name": "Закарпатська",
  "eng_name": "ujgorod",
  "declension": "Закарпатської області",
  "center_declension": "Ужгорода",
  "region_name": "Ужгород",
  "translit": "zakarpatskaya"
},
{
  "lang_id": 4,
  "stateID": 14,
  "name": "Запорізька",
  "eng_name": "zaporijjya",
  "declension": "Запорізької області",
  "center_declension": "Запоріжжя",
  "region_name": "Запоріжжя",
  "translit": "zaporozhskaya"
},
{
  "lang_id": 4,
  "stateID": 15,
  "name": "Івано-Франківська",
  "eng_name": "ivano-frankivsk",
  "declension": "Івано-Франківської області",
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"center_declension": "Івано-Франківська",
"region_name": "Івано-Франківськ",
"translit": "ivano-frankovskaya"
},
{
  "lang_id": 4,
  "stateID": 10,
  "name": "Київська",
  "eng_name": "kiyv",
  "declension": "Київської області",
  "center_declension": "Києва",
  "region_name": "Київ",
  "translit": "kievskaya"
},
{
  "lang_id": 4,
  "stateID": 16,
  "name": "Кіровоградська",
  "eng_name": "kirovograd",
  "declension": "Кіровоградської області",
  "center_declension": "Кропивницького",
  "region_name": "Кропивницький",
  "translit": "kirovogradskaya"
},
{
  "lang_id": 4,
  "stateID": 17,
  "name": "Луганська",
  "eng_name": "lugansk",
  "declension": "Луганської області",
  "center_declension": "Луганська",
  "region_name": "Луганськ",
  "translit": "luganskaya"
},
{
  "lang_id": 4,
  "stateID": 5,
  "name": "Львівська",
  "eng_name": "lviv",
  "declension": "Львівської області",
  "center_declension": "Львова",
  "region_name": "Львів",
  "translit": "lvovskaya"
},
{
  "lang_id": 4,
  "stateID": 19,
  "name": "Миколаївська",
  "eng_name": "mikolayv",
  "declension": "Миколаївської області",
  "center_declension": "Миколаєва",
  "region_name": "Миколаїв",
  "translit": "nikolaevskaya"
},
{
  "lang_id": 4,
```

(continues on next page)

(продолжение с предыдущей страницы)

```
"stateID": 12,
"name": "Одеська",
"eng_name": "odesa",
"declension": "Одеської області",
"center_declension": "Одеси",
"region_name": "Одеса",
"translit": "odesskaya"
},
{
  "lang_id": 4,
  "stateID": 20,
  "name": "Полтавська",
  "eng_name": "poltava",
  "declension": "Полтавської області",
  "center_declension": "Полтави",
  "region_name": "Полтава",
  "translit": "poltavskaya"
},
{
  "lang_id": 4,
  "stateID": 9,
  "name": "Рівненська",
  "eng_name": "rivne",
  "declension": "Рівенської області",
  "center_declension": "Рівного",
  "region_name": "Рівне",
  "translit": "rovenskaya"
},
{
  "lang_id": 4,
  "stateID": 8,
  "name": "Сумська",
  "eng_name": "sumi",
  "declension": "Сумської області",
  "center_declension": "Сум",
  "region_name": "Суми",
  "translit": "sumskaya"
},
{
  "lang_id": 4,
  "stateID": 3,
  "name": "Тернопільська",
  "eng_name": "ternopil",
  "declension": "Тернопільської області",
  "center_declension": "Тернополя",
  "region_name": "Тернопіль",
  "translit": "ternopolskaya"
},
{
  "lang_id": 4,
  "stateID": 7,
  "name": "Харківська",
  "eng_name": "harkiv",
  "declension": "Харківської області",
  "center_declension": "Харкова",
  "region_name": "Харків",
```

(continues on next page)

```
    "translit": "kharkovskaya"
  },
  {
    "lang_id": 4,
    "stateID": 23,
    "name": "Херсонська",
    "eng_name": "herson",
    "declension": "Херсонської області",
    "center_declension": "Херсону",
    "region_name": "Херсон",
    "translit": "khersonskaya"
  },
  {
    "lang_id": 4,
    "stateID": 4,
    "name": "Хмельницька",
    "eng_name": "hmelnickiy",
    "declension": "Хмельницької області",
    "center_declension": "Хмельницького",
    "region_name": "Хмельницький",
    "translit": "khmelnytskaya"
  },
  {
    "lang_id": 4,
    "stateID": 24,
    "name": "Черкаська",
    "eng_name": "cherkasi",
    "declension": "Черкаської області",
    "center_declension": "Черкас",
    "region_name": "Черкаси",
    "translit": "cherkasskaya"
  },
  {
    "lang_id": 4,
    "stateID": 25,
    "name": "Чернівецька",
    "eng_name": "chernivci",
    "declension": "Чернівецької області",
    "center_declension": "Чернівців",
    "region_name": "Чернівці",
    "translit": "chernovytiskaya"
  },
  {
    "lang_id": 4,
    "stateID": 6,
    "name": "Чернігівська",
    "eng_name": "chernigiv",
    "declension": "Чернігівської області",
    "center_declension": "Чернігова",
    "region_name": "Чернігів",
    "translit": "chernigovskaya"
  }
]
```

2.6.2 Міста

Міста залежать від областей, тому, щоб отримати їх список, необхідно надіслати GET запит за адресою https://developers.ria.com/dom/cities/:stateId?api_key=YOUR_API_KEY, де *stateId* - ідентифікатор області. Для отримання інформації українською мовою потрібно додати параметр *lang_id=4*

Наприклад, для Львівської області (https://developers.ria.com/dom/cities/5?api_key=YOUR_API_KEY&lang_id=4) список міст буде таким:

```
[
  {
    "lang_id": 4,
    "cityID": 287,
    "stateID": 5,
    "name": "Борислав",
    "eng": "borislav",
    "declension": ""
  },
  {
    "lang_id": 4,
    "cityID": 288,
    "stateID": 5,
    "name": "Броди",
    "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": "Жидачів",
    "eng": "jidachiv",
    "declension": ""
  },
],
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "lang_id": 4,
  "cityID": 293,
  "stateID": 5,
  "name": "Жовква",
  "eng": "jovkva",
  "declension": ""
},
{
  "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": ""
},
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{
  "lang_id": 4,
  "cityID": 632,
  "stateID": 5,
  "name": "Новий Розділ",
  "eng": "noviy-rozdil",
  "declension": "Новороздільський"
},
{
  "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": ""
},
}
```

(continues on next page)

```
{
  "lang_id": 4,
  "cityID": 614,
  "stateID": 5,
  "name": "Соснівка",
  "eng": "sosnovka",
  "declension": null
},
{
  "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 Райони

Райони залежать від міст, тому, щоб отримати їхній список, необхідно надіслати GET запит за адресою https://developers.ria.com/dom/cities_districts/:city_id?api_key=YOUR_API_KEY, де *city_id* - ідентифікатор міста.

Наприклад, для міста Львова (https://developers.ria.com/dom/cities_districts/5?api_key=YOUR_API_KEY) список районів буде таким:

```
[
[
{
"name": "Район",
"value": ""
},
{
"city_id": 5,
"area_id": 20169,
"name": "Батальна",
"type": 1,
"value": 20169
},
{
"city_id": 5,
"area_id": 17764,
"name": "Білогорща",
"type": 1,
"value": 17764
},
{
"city_id": 5,
"area_id": 17765,
"name": "Богданівка (Залізничний)",
"type": 1,
"value": 17765
},
{
"city_id": 5,
"area_id": 17781,
"name": "Богданівка (Франківський)",
"type": 1,
"value": 17781
},
{
"city_id": 5,
"area_id": 15771,
"name": "Боднарівка",
"type": 1,
"value": 15771
},
{
"city_id": 5,
"area_id": 17769,
"name": "Великі Кривчиці",
"type": 1,
"value": 17769
}
...
]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
]
]
```

Ми почали створювати нові персоналізовані API, які допоможуть Вам керувати всіма Вашими оголошеннями на RIA.com

При цьому, немає необхідності відкривати багато вкладок, тепер всі API доступні в одній адмін-панелі. Зареєструйтеся або авторизуйтеся на developers.ria.com, щоб використовувати персоналізовані API. За замовчуванням усі права на доступ до сервісів не активні, щоб активувати потрібний Вам сервіс перейдіть у вкладку «Мої дозволи»

Усі методи та дані описані за допомогою стандарту DeFacto swagger 2.0 [тут](#)

3.1 Перелік статусів замовлень

Щоб отримати список типів статусів, потрібно надіслати GET запит на адресу https://developers.ria.com/ria/basket/status?api_key=YOUR_API_KEY. Результат буде приблизно таким:

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

```
{
  "status": true,
  "data": [
    {
      "id": 0,
      "name": "Нові та Невиконані"
    },
    {
      "id": 1,
      "name": "Виконані"
    },
    {
      "id": 2,
      "name": "Невиконані"
    }
  ]
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

},
{
  "id": 6,
  "name": "Видалені"
},
{
  "id": 4,
  "name": "Всі"
},
{
  "id": 7,
  "name": "Зворотній дзвінок"
}
]
}

```

3.2 Перелік замовлень

Щоб отримати список замовлень, потрібно надіслати GET запит на адресу https://developers.ria.com/ria/basket/orders/?user_id=*amp;api_key=YOUR_API_KEY. Де **user_id** - потрібно вказати Ваш id.

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

Результат буде орієнтовно наступним:

```

{
  "status":true,
  "total":2,
  "orders":[
    {
      "order_id":212121,           // id замовлення
      "status":6,                // Статус "Видалені"
      "total_order_sum":100,     // Загальна сума замовлення
      "created_at":"2023-01-11 03:39:30", // Час створення
      "recipient":{"
        "user_id":000000,       // id користувача
        "name":"Андрій Сергійович",
        "email":"mail@gmail.com",
        "phone":"....."      // Телефон
      }},
      "delivery":{"             // Доставка
        "state_id":1,          // id Області
        "city_id":1,          // id Міста
        "address":"Вінницька область, Вінниця",
        "service":"DHL"       // Служба доставки
        "department":"Відділення №1: вулиця Соборна, 15; оф. 6" // Інформація про відділення
      }},
      "items":[
        {
          "adv_id":122724239,    // id оголошення
          "name":"Диффузор 1j0121207m",
          "price":100,          // Ціна
          "count":1,           // Кількість

```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "note": "" // Нотатки
        "vendor_code": "", // Код постачальника
        "catalog_numbers": "" // Каталогний номер
    }
  ],
  ...
]
```

3.3 Зміна статусу замовлення

Щоб змінити статус замовлення, потрібно надіслати PUT запит на адресу

```
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}"
```

де *user_id* - id користувача, *order_id* - id замовлення, *status_id* - id статусу замовлення,

Відповідь буде наступною:

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

Загальні Помилки Веб-Служби

Певні поширені помилки будуть повернуті стандартним способом з усіх веб-сервісів API RIA.com. Крім того, деякі повідомлення про помилки можуть бути специфічними для деяких сервісів (дивитись документацію для конкретного сервісу). Нижче описано поширені помилки:

Використання API Key

Після реєстрації, Вам на пошту буде надано власний унікальний API ключ у вигляді рядка з 40 символів. Ваш API ключ:

- Унікальний, ідентифікує вас.
- Надає Вам доступ до веб-сервісів RIA.com.
- Повністю приватний, не може передаватися іншим особам.

Щоб використовувати свій ключ, просто вставте його як параметр URL-адреси під час створення запитів веб-сервісу. Наприклад:

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

Незалежно від використовуваного HTTP методу, API ключ завжди потрібно передавати як параметр GET у запиті URL. Таким чином, навіть якщо Ви будете використовувати метод POST або PUT для конкретного сервісу, параметр запиту *api_key* повинен завжди вказуватися в URL запиті.

Альтернативний Метод

Залежно від того, як Ви будете використовувати API, іноді може бути легше передати API ключ як базову аутентифікацію HTTP. Якщо Ви хочете використовувати цей метод, надішліть Ваш API ключ як ім'я користувача, залишивши пароль порожнім. Наприклад:

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

Ліміти Веб-Сервісів

Обмеження поширюються на кількість запитів до веб-служб RIA.com, які Ви можете створити використовуючи свій API ключ. .

- **Hourly Limit:** Годинний ліміт: 1,000 запитів на годину

Ці обмеження застосовуються до всіх веб-служб RIA.com. Перевищення цих лімітів призведе до тимчасового блокування Вашого API ключа. Залежно від перевищеної межі, блок буде автоматично скасовано протягом наступної години або календарного дня.

Розуміння Граничних Значень Лімітів

Годинний Ліміт

Погодинні лічильники для API ключа анулюються на постійній основі

Приклад: Якщо Ви зробите 500 запитів о 10:15 і 500 запитів о 10:25, Ваш API ключ буде тимчасово заблоковано. Блок буде анульовано об 11:15, після чого Ви зможете зробити ще 500 запитів.

Обмеження Запитів

Якщо Ваш API ключ перевищує ліміти запитів, Ви отримаєте відповідь з HTTP статус-кодом 429 (занадто багато запитів).

Потрібно підвищити ліміти?

Якщо Ви створюєте застосунок, який потребує вищих лімітів, ми будемо раді співпрацювати. Зв'яжіться з нами для більш детальної інформації.