

Таксовик: Документация API для партнёров

[Задача](#)

[Сценарии использования API](#)

[Расчёт стоимости поездки](#)

[Показ телефона таксопарка](#)

[Заказ такси в таксопарке](#)

[Для сервиса маршрутов общественного транспорта](#)

[Особенности API](#)

[Функции API](#)

[Создание маршрута по путевым точкам](#)

[URL](#)

[Метод](#)

[JSON-запрос](#)

[Пример JSON-запроса \(предпочтительный вариант\)](#)

[Пример JSON-запроса \(упрощённый вариант\)](#)

[JSON-ответ](#)

[Создание маршрута с рассчитанными параметрами](#)

[URL](#)

[Метод](#)

[JSON-запрос](#)

[Пример JSON-запроса](#)

[JSON-ответ](#)

[Пример JSON-ответа](#)

[Обновление даты маршрута по его модификатору](#)

[URL](#)

[Метод](#)

[JSON-запрос](#)

[Пример JSON-запроса](#)

[JSON-ответ](#)

[Просмотр телефона](#)

[URL](#)

[Метод](#)

[JSON-запрос](#)

[Пример JSON-запроса](#)

[JSON-ответ](#)

[Пример JSON-ответа](#)

[Создание заказа](#)

[URL](#)

[Метод](#)

[JSON-запрос](#)

[Пример JSON-запроса](#)

[JSON-ответ](#)

[Пример JSON-ответа](#)

[Авторизация](#)

[Получение истории заказов](#)

[Отмена заказа пользователем](#)

Задача

Позволить внешним службам с помощью типизированных запросов, содержащих информацию о маршруте (возможно, включая данные, рассчитанные с использованием картографических служб), получить список таксопарков с рассчитанными **стоимостями поездки по этому маршруту** и набором параметров для новых запросов (с целью получить **телефон конкретного таксопарка** или **создать заказ**).

Сценарии использования API

Расчёт стоимости поездки

1. Используя приложение партнёра, пользователь указывает предполагаемый маршрут
2. Партнёр формирует набор данных о маршруте одним из двух способов:
 - a. [для создания маршрута по путевым точкам](#)
 - b. [для создания маршрута с рассчитанными параметрами](#)
3. Партнёр отправляет их к нам на сервер для создания маршрута
4. API [возвращает](#) данные о тарифах и ценах при поездке по маршруту
5. Партнёр выводит эти данные пользователю

Показ телефона таксопарка

1. Прodelав шаги сценария «[Расчёт стоимости поездки](#)», пользователь может захотеть узнать телефон таксопарка с интересующими его тарифом и ценой
2. В этом случае партнёр формирует запрос [на показ телефона](#)
3. API [возвращает](#) номер телефона таксопарка и фиксирует в БД факт запроса
4. Партнёр показывает полученный телефон пользователю
5. В личном кабинете партнёра отобразится факт показа телефона

Заказ такси в таксопарке

1. Прodelав шаги сценария «[Расчёт стоимости поездки](#)», пользователь может захотеть заказать такси в таксопарке с выбранным тарифом
2. Возможно, пользователь захочет указать другую дату поездки (уточнить время подачи машины), в этом случае необходимо отправить запрос для [обновления времени поездки маршрута](#) и рассчитать новые цены (новую цену для отдельного тарифа отдельного таксопарка)
3. В этом случае, партнёр формирует запрос на [создание заказа](#)
4. API [возвращает](#) номер заказа и всю необходимую информацию о созданном заказе, а также фиксирует в БД факт заказа

5. Партнёр благодарит пользователя за заказ и для справки выводит ему информацию о таксопарке, маршруте и времени поездки
6. В личном кабинете партнёра отобразится факт совершения заказа

Для сервиса маршрутов общественного транспорта

На примере службы построения маршрутов общественного транспорта это может выглядеть следующим образом.

1. Пользователь строит маршрут из точки А в точку Б
2. Сервер партнёра помимо вычисления маршрута общественного транспорта, вычисляет необходимые параметры соответствующего автомобильного маршрута
3. Эти параметры отправляются на сервер Таксовика с помощью API. В ответ сервер отправляет отсортированный по возрастанию список цен с тарифами.
4. Сервер партнёра возвращает пользователю страничку, где помимо маршрута общественного транспорта, отображаются параметры поездки на такси (например, время в пути) и цены на эту поездку в различных таксопарках, отсортированные по возрастанию.
5. Напротив каждой цены отображено приглашение “Показать телефон” для заказа.
6. Если пользователь кликает на “Показать телефон”, сервер партнёра делает запрос на сервер Таксовика и отображает пользователю возвращаемый номер телефона.
7. Если пользователь хочет сделать заказ по указанному маршруту, партнёр отображает форму, чтобы пользователь мог указать всю информацию, необходимую для отправки запроса на создание заказа.
8. Если при заказе пользователь указал другое время поездки (отличное от указанного в запросе для расчёта стоимости поездок во всех таксопарках), необходимо предварительно (перед отправкой запроса для создания заказа), отправить запрос для обновления даты маршрута и получить новую стоимость, которую можно показать пользователю.

Особенности API

Все запросы принимают в формате **JSON**, все ответы также отправляются в этом формате. Все ключи являются обязательными, если не указано обратное. Для именования ключей передаваемых и принимаемых **JSON-объектов** используется соглашение «**lowerCamelCase**»: ключ всегда пишется в одно слово без дефисов и подчёркиваний, первая буква каждого следующего за первым слова ключа — заглавная.

Примеры:

```
price  
dateTime  
routeVariantId
```

Обращаем внимание, что обязательным параметром любого запроса является идентификатор партнёра (**partnerId**), который выдаётся каждому партнёру при подключении к сервису Таксовик.

Функции API

Создание маршрута по путевым точкам

Информацию о путевых точках можно передать двумя способами в виде координат (более предпочтительный вариант) и в виде текстовых адресов (менее предпочтительный вариант).

Обращаем внимание, что второй вариант может привести к существенным расхождениям предполагаемого маршрута с тем, который будет реально построен и рассчитан с помощью API. В этом случае рассчитанная стоимость поездки может существенно отличаться от стоимости поездки, рассчитанной с использованием точных координат путевых точек, поэтому, если есть возможность рассчитать координаты до отправки путевых точек нашему API, настоятельно рекомендуем это сделать.

URL <https://taxovik.ru/api/routes/>

Метод **POST**

JSON-запрос

<code>partnerId</code>	Уникальный идентификатор партнёра
<code>route</code>	Содержит информацию о предполагаемом маршруте.
<code>waypoints</code> <code>route.waypoints</code>	Массив, содержащий путевые точки маршрута.
<code>coordinates</code> <code>route.waypoints[].coordinates</code>	Массив координат путевой точки маршрута. Первое число — градусы северной широты, второе число — градусы восточной долготы. (более предпочтительный вариант). Пример: <code>[55.542805, 37.54064]</code>
<code>address</code> <code>route.waypoints[].address</code>	Адрес путевой точки в любой форме. Примеры: <code>"м. Фрузенская"</code> <code>"пр-кт. Сахарова, 4"</code>
<code>dateTime</code> <code>route.dateTime</code>	Объект, содержащий информацию о дате и времени предполагаемой поездки. Стоимость поездки существенно зависит от времени поездки. Пример: <code>{"date": "13.02.2015", "time": "21:33"}</code>

Пример JSON-запроса (предпочтительный вариант)

```
{
  "partnerId": 31,
  "route": {
    "waypoints": [{
      "coordinates": [55.542997, 37.540488]
    }, {
      "coordinates": [55.727232, 37.580328]
    }],
    "dateTime": {
      "date": "28.03.2015",
      "time": "13:00"
    }
  }
}
```

Пример JSON-запроса с помощью CURL

```
curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d '{
  "partnerId": "31", "route": { "waypoints": [{ "coordinates": [55.542997, 37.540488] }, {
  "coordinates": [55.727232, 37.580328] }], "dateTime": { "date": "28.03.2015", "time": "13:00" } }
}' https://taxovik.ru/api/routes/
```

Пример JSON-запроса (упрощённый вариант)

```
{
  "partnerId": 31,
  "route": {
    "waypoints": [{
      "address": "Южнобутовская улица, 17"
    }, {
      "address": "Шереметьево терминал С"
    }],
    "dateTime": {
      "date": "28.03.2015",
      "time": "13:00"
    }
  }
}
```

JSON-ответ

Аналогичен [JSON-ответу](#) для создания маршрута с рассчитанными параметрами.

Создание маршрута с рассчитанными параметрами

URL https://taxovik.ru/api/route_variants/

Метод **POST**

JSON-запрос

partnerId	Уникальный идентификатор партнёра
-----------	-----------------------------------

routeVariant	Общий ключ для хранения всей информации о маршруте предстоящей поездки
waypoints routeVariant.waypoints	Массив путевых точек поездки (0-й элемент — начало маршрута, 1-й элемент — конец маршрута)
coordinates routeVariant.waypoints[].coordinates	Массив координат путевой точки маршрута. Первое число — градусы северной широты, второе число — градусы восточной долготы. Пример: [55.542805, 37.54064]
geocode routeVariant.waypoints[].geocode	Географический идентификатор путевой точки вида «Страна, (Субъект), Город, Улица, Дом». Используется для записи уникального понятного человеку названия путевой точки. Примеры: "Россия, Москва, Сокольническая линия, метро Фрунзенская" "Россия, Московская область, Люберецкий район, посёлок городского типа Малаховка, Пионерская улица, 10"
waypointType routeVariant.waypoints[].waypointType	Тип путевой точки. Принимает одно из следующих значений: 'town' // находится в пределах МКАД 'suburb' // находится за пределами города 'airport' // является аэропортом 'railroad' // является вокзалом
crossingPoint routeVariant.crossingPoint необязательный параметр	Координаты точки пересечения маршрутом МКАДа. Обязательный параметр в маршрутах из аэропорта или в аэропорт (используется для определения округа, в котором машина въезжает/выезжает за пределы города). Пример: [55.581817, 37.569914]
routeParams routeVariant.routeParams	Числовые параметры маршрута (расстоянии и оценочное время) при движении по МКАД/за МКАД/подаче за МКАД/всего по маршруту.
total routeVariant.routeParams.total	Суммарная длина (в километрах) и оценочное время (в минутах), которое машина проведёт в пути по маршруту. Ассоциативный массив. Пример:

	<pre>{km: 60, min: 42}</pre> означает, что длина маршрута составляет 60 километров, оценочное время преодоления маршрута — 42 минуты.
ringRoad <pre>routeVariant.routeParams.ringRoad</pre>	Суммарное расстояние (в километрах) и оценочное время (в минутах), которое машина проедет по МКАД, двигаясь по маршруту. Ассоциативный массив. Пример: <pre>{km: 10, min: 15}</pre> означает, что маршрут суммарно содержит в себе 10 километров передвижения по МКАД, оценочное время преодоления этого участка (участков) маршрута составит 15 минут.
suburb <pre>routeVariant.routeParams.suburb</pre>	Суммарное расстояние (в километрах) и оценочное время (в минутах), которое машина проедет за пределами МКАД, двигаясь по маршруту. Ассоциативный массив. Пример: <pre>{km: 15, min: 20}</pre> означает, что маршрут суммарно содержит в себе 15 километров передвижения за МКАД, оценочное время преодоления этого участка (участков) маршрута составит 20 минут.
suburbServe <pre>routeVariant.routeParams.suburbServe</pre>	Суммарное расстояние (в километрах) и оценочное время (в минутах), которое машина должна проехать за пределами МКАД, двигаясь к начальной точке маршрута из центра города. Используется для определения стоимости подачи машины по адресу. Ассоциативный массив. Пример: <pre>{km: 12, min: 18}</pre> означает, что для прибытия в начальную точку маршрута машине необходимо проехать 12 километров, двигаясь за пределами МКАД и оценочное время преодоления этого участка (участков) составляет 18 минут.

dateTime	Дата и время предполагаемой поездки. Если не указано, используется текущее время.
routeVariant.dateTime	
необязательный параметр	Пример: <pre>{"date": "07.02.2015", "time": "09:28" }</pre> означает, что необходимо рассчитать стоимость поездки по маршруту на 7 февраля 2015 года в 09:28.

Пример JSON-запроса

```
{
  "partnerId": 31,
  "routeVariant": {
    "waypoints": [{
      "coordinates": [55.542997, 37.540488],
      "geocode": "Россия, Москва, Южнобутовская улица, 17",
      "waypointType": "suburb"
    }, {
      "coordinates": [55.980704, 37.410313],
      "geocode": "Россия, Московская область, Химки, аэропорт Шереметьево, Шереметьево терминал С",
      "waypointType": "airport"
    }],
    "crossingPoint": [55.581817, 37.569917],
    "routeParams": {
      "total": {"km": 62, "min": 61},
      "ringRoad": {"km": 40, "min": 31},
      "suburb": {"km": 6, "min": 12},
      "suburbServe": {"km": 6, "min": 10}
    },
    "dateTime": {
      "date": "27.02.2015",
      "time": "09:28"
    }
  }
}
```

JSON-ответ

routeVariantId	Идентификатор созданного маршрута. Этот параметр требуется для запроса на получение телефона таксопарка.
results	Массив данных с информацией о стоимостях поездки во всех активных тарифах всех таксопарков. Также содержит вспомогательную информацию для осуществления дальнейших запросов.
routeVariantId.results	Возвращаемые значения всегда отсортированы по цене поездки по возрастанию.
fare	Содержит необходимую информацию о тарифе:

<code>routeVariant.results[].fare</code>	его идентификатор и класс.
<code>id</code> <code>routeVariant.results[].fare.id</code>	Уникальный идентификатор тарифа.
<code>fareClassName</code> <code>routeVariant.results[].fare.fareClassName</code>	Название класса тарифа. Требуется, для различения разных тарифов одного таксопарка. Примеры: "Эконом" "Стандарт"
<code>company</code> <code>routeVariant.results[].company</code>	Содержит информацию о таксопарке: идентификатор, название таксопарка и т.д.
<code>id</code> <code>routeVariant.results[].company.id</code>	Идентификатор таксопарка. Этот параметр требуется для запроса на получение телефона таксопарка.
<code>name</code> <code>routeVariant.results[].company.name</code>	Название таксопарка.
<code>acceptsOrders</code> <code>routeVariant.results[].company.acceptsOrders</code>	Параметр, указывающий, можно ли создать заказ для этого таксопарка. Пример: true
<code>rating</code> <code>routeVariant.results[].company.rating</code>	Рейтинг таксопарка, целое число Пример: 17
<code>price</code> <code>routeVariant.results[].price</code>	Содержит информацию о рассчитанной стоимости такси по маршруту.
<code>plain</code> <code>routeVariant.results[].price.plain</code>	Рассчитанная цена в данном тарифе данного таксопарка по заданному маршруту.
<code>crypted</code> <code>routeVariant.results[].price.crypted</code>	Зашифрованная цена. Этот параметр требуется для запроса на получение телефона таксопарка.
<code>components</code> <code>routeVariant.results[].price.components</code>	Объект с компонентами, из которых складывается стоимость поездки. Содержит информацию о всех тарифах и наценках, влияющих на стоимость поездки. Пример:

	<pre>"minimal": {"volume": 1, "rateType": "km", "cost": 250}</pre> Возможные ключи объекта components (в порядке, в котором их предпочтительно показывать пользователю): <pre>"fix" // фиксированная стоимость поездки по тарифу "minimal" // минимальная стоимость заказа "extension" // продление по истечению минимального объёма "extraExtension" // доп. плата за продление в км. (мин.) "town" // продление по городу (если за МКАД – новый тариф) "extraTown" // доп. плата по городу в км. (мин.) "ringRoad" // стоимость при движении по МКАД "suburb" // стоимость при движении за городом (за МКАД) "suburbServe" // стоимость подачи машины за город "jam" // стоимость простоя в пробках "extra" // наценка за доп. услуги (причина наценки указана в ключе "reason") "uberSurge" // наценка Uber-а при большом количестве заказов (volume – число с пл. точкой: коэффициент, на который домножается стоимость поездки, rateType отсутствует) "discount" // скидка для пользователей сервисов-партнеров Таксовика (volume и rateType отсутствуют, cost принимает отрицательное значение, размер скидки)</pre> Причины наценки (при наличии ключа "extra"): <pre>"night" // поездка ночью "railroad" // поездка на вокзал "airport" // поездка в аэропорт "meeting" // встреча (в аэропорту, на вокзале) "departure" // провозы (в аэропорт, на вокзал)</pre>
--	---

Пример JSON-ответа

```
{
  "routeVariantId": 115186,
  "results": [
    {
      "company": {"id": 172, "name": "Хорошее такси", "acceptsOrders": true, "rating": 8},
      "fare": {"id": 1727, "fareClassName": "Эконом"},
      "price": {
        "plain": 1040,
        "crypted": "МТА0МА==",
        "components": {
          "minimal": {"volume": 1, "rateType": "km", "cost": 500},
          "extension": {"volume": 28, "rateType": "km", "cost": 540}
        }
      }
    }, {
      "company": {"id": 172, "name": "Хорошее такси", "acceptsOrders": true, "rating": 8},
      "fare": {"id": 1728, "fareClassName": "Стандарт"},
      "price": {
```

```

{
  "plain":1040,
  "crypted": "MTA0MA==",
  "components": {
    "minimal": {"volume": 1, "rateType":"km", "cost": 500},
    "extension": {"volume": 28, "rateType":"km", "cost": 540}
  }
}, {
  "company": {"id":179, "name":"Доброе такси", "acceptsOrders":false, "rating":12},
  "fare":{"id":1844, "fareClassName": "Эконом"},
  "price":
  {
    "plain":1040,
    "crypted": "MTA0MA==",
    "components": {
      "minimal": {"volume": 1, "rateType":"km", "cost": 500},
      "extension": {"volume": 28, "rateType":"km", "cost": 540}
    }
  }
}
]
}

```

Обновление даты маршрута по его модификатору

URL https://taxovik.ru/api/route_variants/1
 Обратите внимание, что параметр **routeVariantId** (полученный при его создании идентификатор маршрута, который вы хотите обновить), передаётся в URL-запросе.

Метод **PUT**

JSON-запрос

partnerId	Уникальный идентификатор партнёра
routeVariant	Ключ для передачи новой информации о маршруте
dateTime routeVariant.dateTime	Дата и время предполагаемой поездки. Пример: <code>{"date":"09.06.2015", "time":"12:14" }</code> Означает, что необходимо рассчитать стоимость по тому же маршруту, но на 9 июня 2015 года в 12:14.
fareId	Необязательный параметр. Идентификатор тарифа таксопарка, полученный при создании маршрута. Этот параметр передаётся, если вам необходимо рассчитать стоимость поездки для нового времени только в одно таксопарке с использованием только одного тарифа. Пример:

	fareId: 10024
--	---------------

Пример JSON-запроса

```
{
  "partnerId": 31,
  "fareId": 811,
  "routeVariant": {
    "dateTime": {
      "date": "09.06.2015",
      "time": "12:15"
    }
  }
}
```

Пример JSON-запроса с помощью CURL

```
curl -v -H "Content-type: application/json" -X PUT -d '{ "partnerId": 1, "fareId": 1844,
"routeVariant": { "dateTime":{ "date":"09.06.2020", "time":"00:00"} } }'
https://devtaxovik.com/api/route_variants/1108033
```

JSON-ответ

Аналогичен [JSON-ответу](#) для создания маршрута с рассчитанными параметрами.

Просмотр телефона

URL `https://taxovik.ru/api/companies/1/show_phone`
Обратите внимание, что параметр **companyId** таксопарка, для которого вы хотите посмотреть номер, передаётся в URL-запроса.

Метод **GET**

JSON-запрос

partnerId	Уникальный идентификатор партнёра
routeVariantId	Уникальный идентификатор маршрута, необходимо использовать значение, полученное после создания маршрута .
cryptedException	Зашифрованная цена, необходимо использовать значение, полученное в качестве ответа при создании маршрута . routeVariant.results[#].price.cryptedException где # — номер элемента массива с ценами и тарифами, для которого необходимо показать телефон.
ip	IP-адрес пользователя, который запрашивает информацию о телефоне. Используется как

необязательный параметр	вспомогательный параметр проверки клика на стороне Таксовика.
-------------------------	---

Пример JSON-запроса

```
{
  "partnerId":1,
  "routeVariantId":115186,
  "cryptedPrice":"MTAwMW==",
  "ip":"247.8.8.8"
}
```

Пример JSON-запроса с помощью CURL

```
curl -v -H "Accept: application/json" -H "Content-type: application/json" -X PUT -d '{
"partnerId":1, "routeVariantId":"373124", "cryptedPrice":"MzA5MA==", "ip":"127.0.0.1" }'
https://taxovik.ru/api/companies/172/show_phone
```

JSON-ответ

phone	Телефон таксопарка
-------	--------------------

Пример JSON-ответа

```
{
  "phone":"+7 495 222-222-2"
}
```

Создание заказа

URL <https://taxovik.ru/api/orders>

Метод **POST**

JSON-запрос

partnerId	Уникальный идентификатор партнёра
order	Содержит необходимую для создания заказа информацию
fareId order.fareId	Уникальный идентификатор тарифа, полученный в ответ на запрос на создание маршрута . По этому тарифу будет рассчитана стоимость заказа.
routeVariantId order.routeVariantId	Уникальный идентификатор варианта маршрута, полученный, полученный в ответ на запрос на создание маршрута .
clientName order.clientName	Имя клиента для обращения к нему представителем таксопарка. Примеры: "Василий" "Иван Иванов"
clientPhone	Телефон клиента с кодом страны, можно также

<code>order.clientPhone</code>	передать телефон клиента без плюса и с восьмёркой, если номер обслуживает на территории РФ (должен содержать 10 цифр, кроме восьмёрки). Пунктуация кроме начинающего плюса игнорируется. Примеры: <pre>+7 910 100-20-80" "8 (916) 100-20-80" "+380987654321"</pre>
<code>comment</code> <code>order.comment</code>	Комментарий к заказу. Может содержать любую текстовую информацию: необходимость подачи автомобиля с детским креслом, номер подъезда, номер рейса, номер поезда и так далее. Примеры: <pre>"Въезд со двора. Подъезд 3"</pre>

Пример JSON-запроса

```
{
  "partnerId": 10,
  "order": {
    "fareId": 1,
    "routeVariantId": 1,
    "clientName": "Василий Иванов",
    "clientPhone": "+7 910 200-30-40",
    "comment": "Рейс DP 436"
  }
}
```

Пример JSON-запроса с помощью CURL

```
curl -v -H "Content-type: application/json" -X POST -d '{ "partnerId": 172, "order": {
"fareId": 1912, "routeVariantId": 332394, "clientName": "Василий Иванов", "clientPhone":
"+79260792064", "comment": "Рейс DP 436" } }' https://taxovik.ru/api/orders
```

JSON-ответ

<code>order</code>	Содержит информацию о созданном заказе
<code>id</code> <code>order.id</code>	Уникальный идентификатор созданного заказа
<code>price</code> <code>order.price</code>	Конечная стоимость заказа
<code>routeVariant</code> <code>order.routeVariant</code>	Содержит информацию о маршруте заказа

id	Уникальный идентификатор маршрута созданного заказа
waypoints order.routeVariant.waypoints	Массив путевых точек маршрута, представленны в виде уникальных географических идентификаторов Пример: {'geocode': 'Россия, Москва, Симферопольский бульвар, 30к2'}
dateTime order.routeVariant.dateTime	Данные о дате и времени созданного заказа. Пример: {'date': '13.02.2015', 'time': '21:33'}

Пример JSON-ответа

```
{
  "order": {
    "id": 15,
    "routeVariant": {
      "id": 111,
      "waypoints": [{
        "geocode": "Россия, Московская область, Химки, аэропорт Шереметьево",
      }, {
        "geocode": "Россия, Москва, Хабаровская улица"
      }],
      "dateTime": {
        "date": "13.02.2015",
        "time": "21:33"
      }
    },
    "price": 350
  }
}
```

Авторизация

Запрос токена

Пример JSON-запроса с помощью CURL

```
curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d '{ "phone": "+79261234567" }' https://taxovik.ru/api/sms_authentications
```

Подтверждение токена

Пример JSON-запроса с помощью CURL

```
curl -H "Accept: application/json" -H "Content-type: application/json" -X PUT -d '{ "sms_authentication": { "sms_code": "216758" } } ' https://devtaxovik.com/api/sms_authentications/token
```

Получение истории заказов

Пример JSON-запроса с помощью CURL

```
curl -H "Accept: application/json" -H "Content-type: application/json" -X GET 'https://devtaxovik.com/api/orders?token=token&partnerId=1'
```

Отмена заказа пользователем

Пример JSON-запроса с помощью CURL

```
curl -H "Accept: application/json" -H "Content-type: application/json" -X PUT -d '{ "partnerId": 1, "token": "token", "order": { "cancellationReason": "reason" } } ' https://devtaxovik.com/api/orders/#{order.id}/cancel
```

it 'cancel order in normal flow' do

 expect do

 put "/api/orders/**#{order.id}**/cancel",

 { token: 'token', partnerId: partner.id, order: { cancellationReason: 'reason' } }, format: :json

 end.to change { order.reload.status }.to('cancelled')

 expect(response).to be_success

end

Получение списка городов

Пример JSON-запроса с помощью CURL

```
curl -H "Accept: application/json" -H "Content-type: application/json" -X GET 'https://taxovik.ru/api/cities/'
```

Получение параметров отдельного города

Пример JSON-запроса с помощью CURL

```
curl -H "Accept: application/json" -H "Content-type: application/json" -X GET 'https://taxovik.ru/api/cities/1'
```

```
{ "city": { "id": 1, "centerCoordinates": [55.76, 37.64], "boundedArea": [55.576307, 37.348506, 55.92947, 37.827785], "airportsList": ["Шереметьево", "Домодедово", "Внуково"], "nameCases": ["Москва", "Москву", "Москвой", "Москве", "Москве"], "cityPolygon": [[55.5952254001276, 37.5066375732422], [55.6630630471355, 37.4269866943359], [55.6895846699919, 37.4094772338867], [55.7100927134436, 37.3837280273438], [55.7516561766793, 37.3631286621094], [55.7875777143167, 37.3669052124023], [55.8045613172881, 37.3809814453125], [55.8371567883636, 37.3923110961914], [55.848529842406, 37.3875045776367], [55.8701102170047, 37.4015808105469], [55.8851323263444, 37.4465560913086], [55.8860950838436, 37.4647521972656], [55.9086167924671, 37.5268936157227], [55.9140046643974, 37.5807952880859], [55.8999561440681, 37.6443099975586], [55.8966838402608, 37.697868347168], [55.8279016795274, 37.8369140625], [55.8205731562555, 37.8417205810547], [55.7711660648286, 37.8482437133789], [55.7135741426627, 37.844123840332], [55.6948094383187, 37.833137512207], [55.6572529924794, 37.8458404541016], [55.5907633848853, 37.7366638183594], [55.5698045279833, 37.6738357543945], [55.5748512013123, 37.588005065918]], "sampleWaypoints": ["Южнобутовская, 17", "метро Фрунзенская", "Шереметьево", "Казанский вокзал"], "timeZone": null }}
```