

- id (integer): Уникальный идентификатор рейса.
- point a (string): Начальная точка маршрута.

- point_b (string): Конечная точка маршрута.
- start_plan (string): Запланированное время начала поездки в формате HH:MM.
- end_plan (string): Запланированное время окончания поездки в формате HH:MM.
- status (string): Статус прибытия рейса. Возможные варианты: default - ожидается, canceled - отменен, arrived - прибыл, gone - ушел
- passengers_count (integer): Количество купленных билетов на поездку.
- baggage_count (integer): Количество купленных мест для багажа.
- contractor (string): Перевозчик.
- platform (string): Номер платформы.
- bus_number (string): Номерной знак автобуса.

По умолчанию эндпоинт возвращает рейсы на сегодняшний день. Чтобы получить рейсы на определенную дату, можно использовать параметр date. Дата указывается в формате DD.MM.YY

Пример запроса:

GET api/controller/trips/?start_plan=30.06.2024
Authorization: Bearer eyJhbGciOiJIUz...EdiDe6-x4fgdXTZc
Content-Type: application/json

Пример ответа:

HTTP 200 OK
Content-Type: application/json

```
[
  {
    "id": 1507696,
    "point_a": "Белгород",
    "point_b": "Воронеж",
    "start_plan": "06:00",
    "end_plan": "10:00",
    "status": "default",
    "passengers_count": 10,
    "baggage_count": 2,
    "contractor": "Федотова Е.П. ИП",
    "platform": "5",
    "bus_number": "A123BC12"
  },
  ...
]
```

Примеры ошибок:

401 - ошибка авторизации

HTTP 401 Unauthorized
Content-Type: application/json

```
{
  "detail": "Учетные данные не были предоставлены."
}
```

Информация о рейсе и список пассажиров

Эндпоинт: **api/controller/trips/**

Метод: GET

Возвращает объект со следующими полями:

- id (integer): Уникальный идентификатор рейса.
- point_a (string): Начальная точка маршрута.
- point_b (string): Конечная точка маршрута.
- start_plan (string): Запланированное время начала поездки в формате HH:MM.
- end_plan (string): Запланированное время окончания поездки в формате HH:MM.
- tickets (array): Массив купленных билетов рейса.

Объекты в массиве tickets имеют следующие поля:

- id (integer): Уникальный идентификатор билета.
- passenger_name (string): ФИО пассажира в формате "Иванов И. И."
- passenger_name_full (string): ФИО пассажира в формате "Иванов Иван Иванович"
- doc_type (string): Тип документа, удостоверяющего личность.
- doc_number (string): Серия и номер документа, удостоверяющего личность.
- ticket_number (string): Номер билета (может отличаться от id).
- ticket_price (float): Стоимость билета.
- point_b (string): Пункт прибытия.
- place_number (integer): Номер места.
- baggage_count (integer): Количество багажа.
- is_baggage (boolean): Является ли билет багажным.
- is_gone (boolean): Значение true, если пассажир вычеркнут или false, если нет.

Пример запроса:

GET api/controller/trips/1507696/
Authorization: Bearer eyJhbGciOiJIUz...EdiDe6-x4fgdXTZc
Content-Type: application/json

Пример ответа:

HTTP 200 OK
Content-Type: application/json

```
{
  "id": 1507477,
  "point_a": "Белгород",
  "point_b": "Москва",
  "start_plan": "06:40",
  "end_plan": "17:00",
  "tickets": [
    {
      "id": 12345,
      "passenger_name": "Иванов И. И.",
      "passenger_name_full": "Иванов Иван Иванович",
      "doc_type": "Паспорт гражданина РФ",
      "doc_number": "1234567890",
      "ticket_number": "67891",
      "ticket_price": 1020.0,
      "point_b": "Россошь АВ",
      "place_number": 4,
      "baggage_count": 0,
      "is_baggage": false,
      "is_gone": false
    }
  ],
  ...
}
```

Примеры ошибок:

401 - ошибка авторизации

HTTP 401 Unauthorized
Content-Type: application/json

```
{
  "detail": "Учетные данные не были предоставлены."
}
```

404 - объект не найден

HTTP 404 Not Found
Content-Type: application/json

```
{
  "detail": "Рейс с id 12345 не найден."
}
```

Вычеркивание пассажиров

Эндпоинт: **api/controller/trips//**

Метод: POST

Принимает массив объектов со следующими полями:

- id (integer): Уникальный идентификатор билета.
- is_gone (boolean): Значение true, если пассажир вычеркнут или false, если нет.

Пример запроса:

POST api/controller/trips/1507696/
Authorization: Bearer eyJhbGciOiJIUz...EdiDe6-x4fgdXTZc
Content-Type: application/json

```
[
  {
    "id": 9877211,
    "is_gone": true
  },
  {
    "id": 9877212,
    "is_gone": false
  }
]
```

Пример ответа:

HTTP 200 OK
Content-Type: application/json

```
[
  {
    "id": 9877211,
    "passenger_name": "Петров П. П.",
    "passenger_name_full": "Петров Пётр Петрович",
    "doc_type": "Паспорт гражданина РФ",
    "doc_number": "1111222222",
    "ticket_number": "9877211",
    "ticket_price": 1020.0,
    "point_b": "Россошь АВ",
    "place_number": 4,
    "baggage_count": 0,
    "is_baggage": false,
    "is_gone": true
  },
  {
    "id": 9877212,
    "passenger_name": "Иванов И. И.",
    "passenger_name_full": "Иванов Иван Иванович",
    "doc_type": "Паспорт гражданина РФ",
    "doc_number": "2222111111",
    "ticket_number": "9877212",
    "ticket_price": 1020.0,
    "point_b": "Россошь АВ",
    "place_number": 11,
    "baggage_count": 0,
    "is_baggage": false,
    "is_gone": false
  }
]
```

Примеры ошибок:

400 - данные невалидны

```
{
  "non_field_errors": [
    "Ожидался list со значениями, но был получен \"dict\"."
  ]
}
```

или

```
[
  {},
  {
    "id": [
      "Обязательное поле."
    ]
  }
]
```

Во втором случае возвращается массив с информацией по ошибкам для всех полученных объектов в том порядке, в каком они были отправлены. Для объектов без ошибок валидации вернется {}.

401 - ошибка авторизации

HTTP 401 Unauthorized
Content-Type: application/json

```
{
  "detail": "Учетные данные не были предоставлены."
}
```

404 - объект не найден

HTTP 404 Not Found
Content-Type: application/json

```
{
  "detail": "Билет на рейс 1507696 с id 12345 не найден."
}
```

Информация пассажира

Эндпоинт (получение через id): **api/controller/trips//ticket//**

Эндпоинт (получение через штрихкод): **api/controller/trips//barcode//**

Метод: GET

Эндпоинты возвращают объект билета.

Пример запроса:

```
GET api/controller/trip/1507696/ticket/12345/
Authorization: Bearer eyJhbGciOiJIUz...EdiDe6-x4fgdXTZc
Content-Type: application/json
```

Пример ответа:

```
HTTP 200 OK
Content-Type: application/json
```

```
{
  "id": 12345,
  "passenger_name": "Иванов И. И.",
  "passenger_name_full": "Иванов Иван Иванович",
  "doc_type": "Паспорт гражданина РФ",
  "doc_number": "1234567890",
  "ticket_number": "67891",
  "ticket_price": 1020.0,
  "point_b": "Россошь АВ",
  "place_number": 4,
  "baggage_count": 0,
  "is_baggage": false,
  "is_gone": false,
  "point_b": "Россошь АВ"
}
```

Примеры ошибок:

401 - ошибка авторизации

```
HTTP 401 Unauthorized
Content-Type: application/json
```

```
{
  "detail": "Учетные данные не были предоставлены."
}
```

404 - объект не найден

```
HTTP 404 Not Found
Content-Type: application/json
```

```
{
  "detail": "Билет на рейс 1507696 с id 12345 не найден."
}
```

или

```
HTTP 404 Not Found
Content-Type: application/json
```

```
{
  "detail": "Билет на рейс 1507696 со штрихкодом 0000099328467 не найден."
}
```

Профиль контролера

Эндпоинт: **api/controller/user/**

Метод: GET

Возвращает данные текущего пользователя:

- full_name (string): ФИО или username пользователя.
- roles (string): Список ролей пользователя.

Пример запроса:

```
GET api/controller/trip/1507696/ticket/12345/
Authorization: Bearer eyJhbGciOiJIUz...EdiDe6-x4fgdXTZc
Content-Type: application/json
```

Пример ответа:

```
HTTP 200 OK
Content-Type: application/json
```

```
{
  "full_name": "Иванов И.",
  "roles": "Полные права, Контролер"
}
```

Примеры ошибок:

401 - ошибка авторизации

```
HTTP 401 Unauthorized
Content-Type: application/json
```

```
{
  "detail": "Учетные данные не были предоставлены."
}
```