**API сервиса**

**Базовый URL АПИ:**

[*https://fe-diplom.herokuapp.com/*](https://fe-diplom.herokuapp.com/).

Пример запроса:

fetch( 'https://netology-trainbooking.herokuapp.com/routes/last' )

.then( response => response.json() )

.then( data => console.log(data) );

Ответы API в формате JSON.

**Основные функции API**

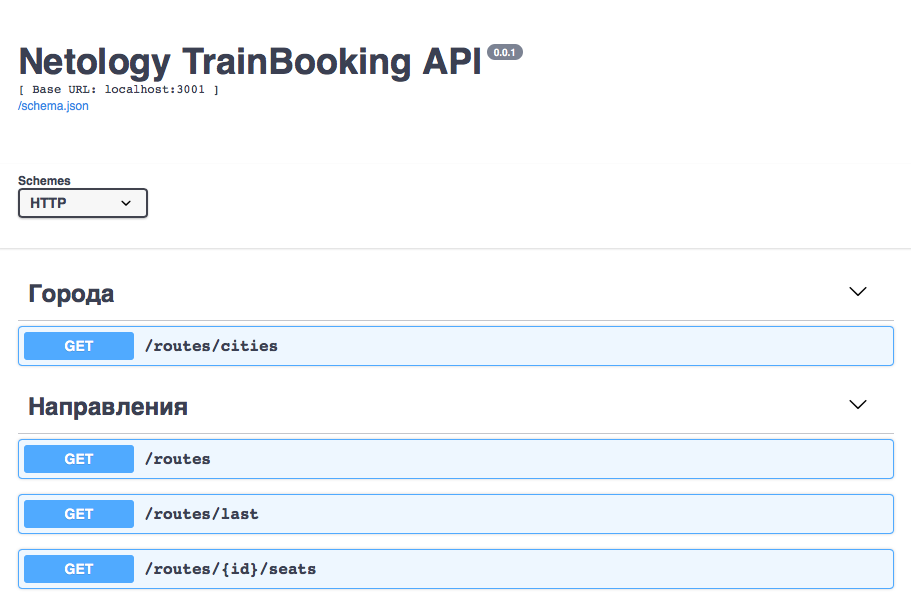
1. [Поиск городов](https://github.com/netology-code/fe-2-diplom/blob/master/api/cities.md)
2. [Поиск направлений](https://github.com/netology-code/fe-2-diplom/blob/master/api/routes.md)
3. [Последние направления](https://github.com/netology-code/fe-2-diplom/blob/master/api/last-routes.md)
4. [Карта поезда (свободные и занятые места в вагонах)](https://github.com/netology-code/fe-2-diplom/blob/master/api/seats.md)
5. [Заказ билетов](https://github.com/netology-code/fe-2-diplom/blob/master/api/order.md)
6. [Подписка на email-новости](https://github.com/netology-code/fe-2-diplom/blob/master/api/subscribe.md)

Не забывайте обрабатывать ошибки АПИ.

**Swagger-документация**

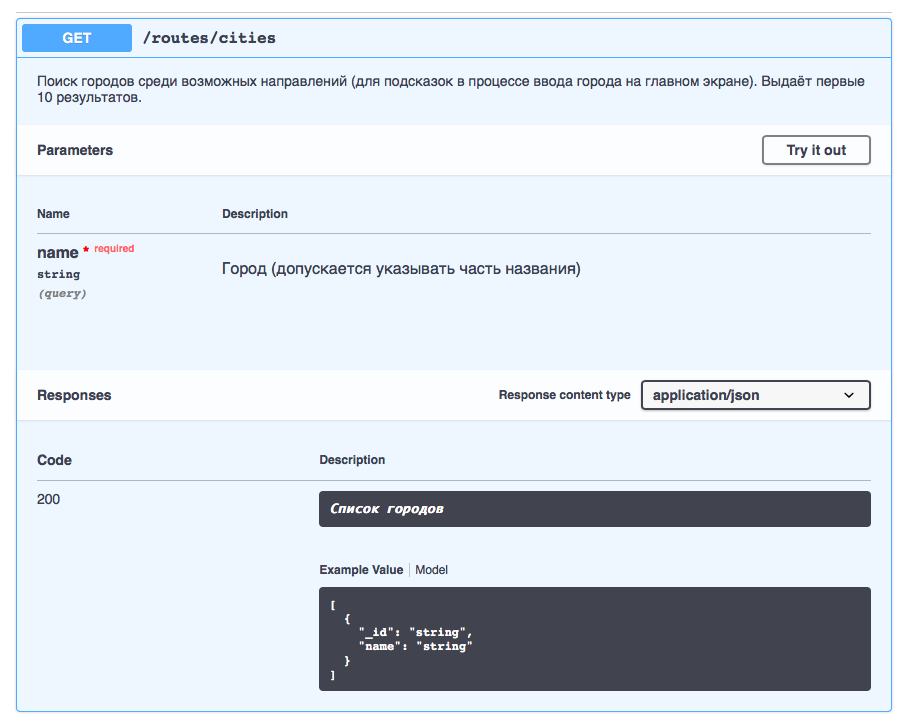
<https://fe-swagger-test.herokuapp.com/>

Вы увидите интерактивную страницу, на которой сможете протестировать работу сервиса.

[](https://github.com/netology-code/fe-2-diplom/blob/master/reference/images/swagger.png)

**Сведения о запросе и ответе**

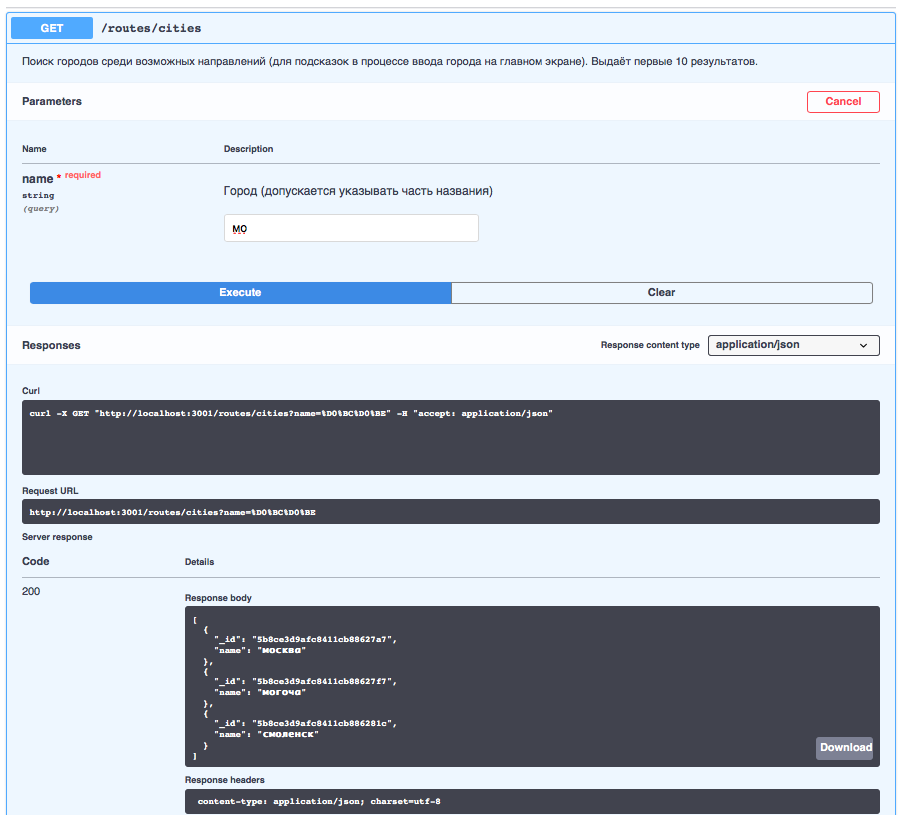
Для понимания, какие данные необходимо отправлять по соответствующим адресам и в каком формате, нажмите на интересующий вас цветной блок.

[](https://github.com/netology-code/fe-2-diplom/blob/master/reference/images/swagger-details.png)

**Тестирование сервиса**

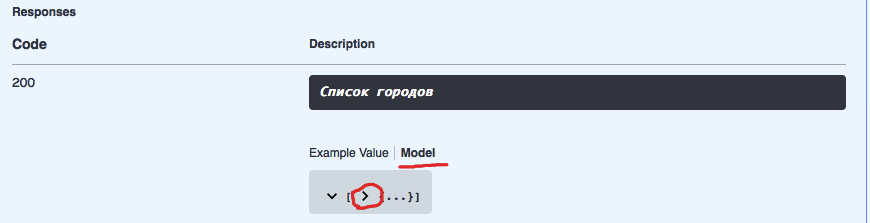
Вы можете отправить на сервер какие-то реальные данные и посмотреть, какой результат будет, нажав на кнопку «Try it out».

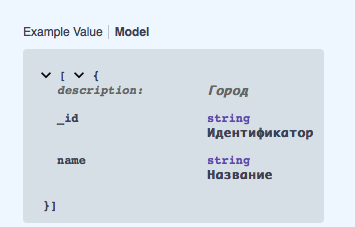
Введите необходимые данные и нажмите «Execute» для отправки данных на сервер. В колонке Results вы увидите полученную информацию.

[](https://github.com/netology-code/fe-2-diplom/blob/master/reference/images/swagger-try.png)

**Разбор результатов**

Для того, чтобы понять, что означает каждое поле результата, нажмите на пункт «Model» колонки «Description». Для того, чтобы развернуть описание, нажмите на стрелочку «>»:

[](https://github.com/netology-code/fe-2-diplom/blob/master/reference/images/swagger-model.png)

[](https://github.com/netology-code/fe-2-diplom/blob/master/reference/images/swagger-model-2.png)

**Поиск городов**

GET /routes/cities

Поиск городов среди возможных направлений (для подсказок в процессе ввода города на главном экране). Выдаёт первые 10 результатов.

**Параметры**

* **name** - Город (допускается указывать часть названия)

**Пример**

fetch( 'https://fe-diplom.herokuapp.com/routes/cities?name=мос' )

.then( response => response.json())

.then( data => console.log( data ));

**Результат**

Массив объектов со свойствами:

* **\_id** - идентификатор города
* **name** - имя города

**Поиск направлений**

GET <https://netology-trainbooking.herokuapp.com/routes>

**Пример запроса:**

fetch( 'https://netology-trainbooking.herokuapp.com/routes?from\_city\_id=5b9a2fa7f83e028786ea5672&to\_city\_id=5b9a2fa8f83e028786ea567b' )

.then( response => response.json()

.then( data => { console.log( 'routes', data ) })

);

**Возможные параметры запроса:**

* **from\_city\_id** - Идентификатор города, откуда планируется путешествие (обязательный)
* **to\_city\_id** - Идентификатор города, куда планируется путешествие (обязательный)
* **date\_start** - Дата отбытия туда (в формате YYYY-DD-MM; например 2030-03-01)
* **date\_end** - Дата отбытия обратно (в формате YYYY-DD-MM; например 2030-03-01)
* **date\_start\_arrival** - Дата прибытия туда (в формате YYYY-DD-MM; например 2030-03-01)
* **date\_end\_arrival** - Дата прибытия обратно (в формате YYYY-DD-MM; например 2030-03-01)
* **have\_first\_class** - Люкс (true/false)
* **have\_second\_class** - Купе (true/false)
* **have\_third\_class** - Плацкарт (true/false)
* **have\_fourth\_class** - Сидячее место (true/false)
* **have\_wifi** - Имеется WiFi (true/false)
* **have\_air\_conditioning** - Имеется кондиционер (true/false)
* **have\_express** - Экспресс (true/false)
* **price\_from** - Цена от
* **price\_to** - Цена до
* **start\_departure\_hour\_from** - Час отбытия от (число)
* **start\_departure\_hour\_to** - Час отбытия до (число)
* **start\_arrival\_hour\_from** - Час прибытия от (число)
* **start\_arrival\_hour\_to** - Час прибытия до (число)
* **end\_departure\_hour\_from** - Час отбытия назад от (число)
* **end\_departure\_hour\_to** - Час отбытия назад до (число)
* **end\_arrival\_hour\_from** - Час прибытия назад от (работает при установленном параметре date\_end)
* **end\_arrival\_hour\_to** - Час прибытия назад до (работает при установленном параметре date\_end)
* **limit** - Количество результатов на странице
* **offset** - Количество результатов, которое необходимо пропустить в выдаче
* **sort** - Сортировка результатов (date, price, duration)

**Результат:**

Результатом выполнения запроса является объект вида: { total\_count: 1, items: [object1, object2, object3]},

где total\_count - количество найденных записей,

items - массив объектов ,

а objectN - объект, содержащий следующие свойства:

* **have\_first\_class** - В поезде есть вагон класса «Люкс» (СВ)
* **have\_second\_class** - В поезде есть вагон класса «Купе»
* **have\_third\_class** - В поезде есть вагон класса «Плацкарт»
* **have\_fourth\_class** - В поезде есть вагон с сидячими местами
* **have\_wifi** - Есть Wi-Fi? (на всём направлении)
* **have\_air\_conditioning** - Есть кондиционер? (на всём направлении)
* **is\_express** - Экспресс-маршрут (на всём направлении)
* **min\_price** - Минимальная цена поездки (на 1 взрослого)
* **arrival** - информация об отбытии
* **departure** - информация о прибытии
* **total\_avaliable\_seats** - количество свободных мест

**arrival/departure**

Отбытие и прибытие - это, обычно, два разных поезда. Соответственно, у них могут быть разные характеристики.

Объект со свойствами:

* **have\_first\_class** - В поезде есть вагон класса «Люкс» (СВ)
* **have\_second\_class** - В поезде есть вагон класса «Купе»
* **have\_third\_class** - В поезде есть вагон класса «Плацкарт»
* **have\_fourth\_class** - В поезде есть вагон с сидячими местами
* **have\_wifi** - Есть Wi-Fi? (на всём направлении)
* **have\_air\_conditioning** - Есть кондиционер? (на всём направлении)
* **is\_express** - Экспресс-маршрут (на всём направлении)
* **min\_price** - Минимальная цена поездки (на 1 взрослого)
* **train** - Информация о поезде
* **from** - Информация об отправлении
* **to** - Информация о прибытии
* **duration** - Длительность поездки (в секундах)
* **price\_info** - Информация о ценах в вагонах разного класса
* **seats\_info** - Информация о количестве свободных мест в каждом типе вагонов

**arrival.from/arrival.to и departure.from/departure.to**

Информация об отбытии и прибытии

* **datetime** - Дата в секундах
* **railway\_station\_name** - Информация о вокзале
* **city** - информация о городе

# Последние направления

GET /routes/last

## Пример

fetch( 'https://fe-diplom.herokuapp.com//routes/last' )

.then( response => response.json())

.then( data => {

data.forEach( el => {

console.log('el', el)

})

});

**Посадочные места**

Информация о посадочных местах определённого направления. Параметры поиска позволят отфильтровать только нужные вагоны в составе

GET /routes/{id}/seats

Параметры в запросе позволяют исключить лишние вагоны в поезде

**Параметры:**

* **id** - Идентификатор направления (обязательный)
* **have\_first\_class** - Люкс (true/false)
* **have\_second\_class** - Купе (true/false)
* **have\_third\_class** - Плацкарт (true/false)
* **have\_fourth\_class** - Сидячее место (true/false)
* **have\_wifi** - Имеется WiFi (true/false)
* **have\_air\_conditioning** - Имеется кондиционер (true/false)
* **have\_express** - Экспресс (true/false)

**Пример:**

fetch( 'https://fe-diplom.herokuapp.com/routes/5b9a35bcf83e028786ea74ef/seats?have\_wifi=false' )

.then( response => response.json())

.then( data => console.log( data ));

**Результат:**

Информация о местах в вагонах. Массив объектов со свойствами:

* **\_id** - Идентификатор
* **name** - Идентификатор
* **class\_type** - Идентификатор
* **have\_first\_class** - Люкс (true/false)
* **have\_second\_class** - Купе (true/false)
* **have\_third\_class** - Плацкарт (true/false)
* **have\_fourth\_class** - Сидячее место (true/false)
* **have\_wifi** - Имеется WiFi (true/false)
* **have\_air\_conditioning** - Имеется кондиционер (true/false)
* **have\_express** - Экспресс (true/false)
* **price** - Цена за место (Люкс)
* **top\_price** - Цена верхнего места
* **bottom\_price** - Цена нижнего места
* **side\_price** - Цена бокового места
* **linens\_price** - Цена постельного белья
* **wifi\_price** - Цена услуги Wi-Fi
* **avaliable\_seats** - Количество свободных мест в вагоне
* **is\_linens\_included** - Стоимость белья включена в стоимость билета и не может быть исключена (true/false)
* **seats** - информация о посадочных местах

**seats**

Массив объектов со свойствами:

* **index** - Числовой номер места в вагоне
* **available** - Место доступно для бронирования

**Заказ билетов**

**Посадочные места**

Информация о посадочных местах определённого направления. Параметры поиска позволят отфильтровать только нужные вагоны в составе

POST /routes/order

**Пример:**

fetch( 'https://fe-diplom.herokuapp.com/order', {

method: 'POST',

body: JSON.stringify({

"user": {

"first\_name": "Иван",

"last\_name": "Смирнов",

"patronymic": "Олегович",

"phone": "8900123123",

"email": "string@string.ru",

"payment\_method": "cash" // или online

},

"departure": {

"route\_direction\_id": "123431",

"seats": [

{

"coach\_id": "12341",

"person\_info": {

"is\_adult": true,

"first\_name": "Ivan",

"last\_name": "Popov",

"patronymic": "Popovich",

"gender": true,

"birthday": "1980-01-01",

"document\_type": "паспорт",

"document\_data": "45 6790195"

},

"seat\_number": 10,

"is\_child": true,

"include\_children\_seat": true

}

]

}

})

})

.then( response => response.json())

.then( console.log );

**Параметры:**

* **user** - пользователь в системе
* **departure** - билеты на отбытие
* **arrival** - билеты на прибытие

**user**

* **first\_name** - имя
* **last\_name** - фамилия
* **patronymic** - отчество
* **phone** - телефон
* **email** - E-mail
* **payment\_method** - метод оплаты (cash или online)

**departure/arrival**

Объект с данными заказа о прибытии и отбытии

* **route\_direction\_id** - id направления
* **seats** - массив с информацией о билетах

**departure.seats и arrival.seats**

Массив заказываемых билетов:

* **coach\_id** - Идентификатор вагона
* **person\_info** - объект с информацией о пассажире
* **seat\_number** - Номер места в вагоне
* **is\_child** - Детский билет
* **include\_children\_seat** - Необходимо ли доп. место для билета «Детский без места»? Доступно только для взрослого места.

**person\_info**

* **is\_adult** - Взрослый/детский
* **first\_name** - Имя
* **last\_name** - Фамилия
* **patronymic** - Отчество
* **gender** - Пол (true - мужской)
* **birthday** - День рождения (в формате YYYY-MM-DD)
* **document\_type** - Тип документа
* **document\_data** - Информация о документе

# Подписка

Подписка на рассылку

Принимает только один параметр - email-адрес

## Пример:

fetch( 'https://fe-diplom.herokuapp.com/subscribe?email=hello@kitty.com' )

.then( response => response.json())

.then( data => console.log( data ));