

# Документация API

## Получение данных об ингредиентах

GET <https://stellarburgers.education-services.ru/api/ingredients>

## Создание заказа

POST <https://stellarburgers.education-services.ru/api/orders>

Тело запроса:

```
{  
    "ingredients": ["60d3b41abdacab0026a733c6", "609646e4dc916e00276b2870"]  
}
```

Пример ответа:

```
{  
    "name": "Краторный метеоритный бургер",  
    "order": {  
        "number": 6257  
    },  
    "success": true  
}
```

Если в запросе передан невалидный хеш ингредиента, вернётся код ответа 500 Internal Server Error.

Если не передать ни один ингредиент, вернётся код ответа 400 Bad Request.

```
{  
    "success": false,  
    "message": "Ingredient ids must be provided"  
}
```

## Восстановление и сброс пароля

POST <https://stellarburgers.education-services.ru/api/password-reset>

Тело запроса:

```
{  
    "email": ""  
}
```

Тело успешного ответа:

```
{  
    "success": true,  
    "message": "Reset email sent"  
}
```

Если ответ успешный, пользователь направляется на маршрут `/reset-password`. На введённый имейл приходит инструкция с кодом для восстановления пароля.

На экране `/reset-password` пользователь вводит новый пароль и код из имейла, а после нажимает кнопку «Сохранить». Происходит POST-запрос к эндпоинту

<https://stellarburgers.education-services.ru/api/password-reset/reset>.

Тело запроса:

```
{  
    "password": "",  
    "token": ""  
}
```

Тело успешного ответа:

```
{  
    "success": true,  
    "message": "Password successfully reset"  
}
```

## Создание пользователя

`POST https://stellarburgers.education-services.ru/api/auth/register`

Пример тела запроса:

```
{  
    "email": "test-data@yandex.ru",
```

```
"password": "password",
"name": "Username"
}
```

Если пользователь существует, вернётся код ответа 403 Forbidden.

```
{
"success": false,
"message": "User already exists"
}
```

Если нет одного из полей, вернётся код ответа 403 Forbidden.

```
{
"success": false,
"message": "Email, password and name are required fields"
}
```

## Авторизация и регистрация

С сервера на запросы авторизации и регистрации к эндпоинтам `auth/login` и `auth/register` приходят `accessToken` и `refreshToken`. Первый токен используют в запросах к эндпоинту `auth/user`, чтобы получить, обновить и удалить данные пользователя. Срок жизни токена — 20 минут.

Второй токен — `refreshToken` — используется для выхода из системы и для получения нового `accessToken`, если последний перестал подходить и просрочился.

`POST https://stellarburgers.education-services.ru/api/auth/login` — эндпоинт для авторизации.

Тело ответа сервера при успешной авторизации:

```
{
"success": true,
"accessToken": "Bearer ...",
"refreshToken": "",
"user": {
  "email": "",
  "name": ""
}
}
```

Если логин или пароль неверные или нет одного из полей, вернётся код ответа 401 Unauthorized.

```
{  
  "success": false,  
  "message": "email or password are incorrect"  
}
```

`POST https://stellarburgers.education-services.ru/api/auth/register` — ЭНДПОИНТ ДЛЯ регистрации пользователя.

Формат тела запроса регистрации:

```
{  
  "email": "",  
  "password": "",  
  "name": ""  
}
```

Тело ответа сервера при успешной регистрации:

```
{  
  "success": true,  
  "user": {  
    "email": "",  
    "name": ""  
  },  
  "accessToken": "Bearer ...",  
  "refreshToken": ""  
}
```

`POST https://stellarburgers.education-services.ru/api/auth/logout` — ЭНДПОИНТ ДЛЯ ВЫХОДА ИЗ СИСТЕМЫ.

Формат тела запроса для выхода из системы:

```
{  
  "token": "{{refreshToken}}"  
}
```

Чтобы выйти из системы или обновить токен, используют `refreshToken`. Его можно получить после успешной регистрации или авторизации.

Тело ответа сервера при выходе из системы:

```
{  
  "success": true,  
  "message": "Successful logout"  
}
```

`POST https://stellarburgers.education-services.ru/api/auth/token` — эндпоинт обновления токена.

Формат тела запроса авторизации:

```
{  
  "email": "",  
  "password": ""  
}
```

Только авторизованные пользователи могут делать заказы. Структура эндпоинтов не меняется, но нужно предоставлять токен при запросе к серверу в поле `Authorization`.

Когда пользователь нажимает «Оформить заказ», неавторизованный пользователь переадресовывается на маршрут `/login`.

После успешной авторизации пользователь переадресовывается на маршрут `/`, а структура конструктора не сбрасывается. Все ингредиенты, которые уже добавили, отображаются в конструкторе.

Все остальные действия доступны и неавторизованному пользователю.

## Получение и обновление информации о пользователе

`GET https://stellarburgers.education-services.ru/api/auth/user` — эндпоинт получения данных о пользователе.

Для получения данных о пользователе необходимо передать серверу токен в поле `authorization`. Сервер вернёт такой ответ:

```
{  
    "success": true,  
    "user": {  
        "email": "",  
        "name": ""  
    }  
}
```

`PATCH https://stellarburgers.education-services.ru/api/auth/user` — эндпоинт обновления данных о пользователе.

Если пользователь отредактирует информацию на экране профиля и нажмёт «Сохранить», серверу понадобится поле `authorization` и поля с отредактированной информацией. Если всё хорошо, сервер вернёт обновлённого пользователя:

```
{  
    "success": true,  
    "user": {  
        "email": "",  
        "name": ""  
    }  
}
```

Если выполнить запрос без авторизации, вернётся код ответа 401 Unauthorized.

```
{  
    "success": false,  
    "message": "You should be authorised"  
}
```

Если передать почту, которая уже используется, вернётся код ответа 403 Forbidden.

```
{  
    "success": false,  
    "message": "User with such email already exists"  
}
```

## Удаление пользователя

```
DELETE https://stellarburgers.education-services.ru/api/auth/user
```

## Получить все заказы

```
GET https://stellarburgers.education-services.ru/api/orders/all
```

При подключении бэкенд приложения вернёт максимум 50 последних заказов. Они сортируются по времени обновления.

Пример успешного ответа:

```
{
  "success": true,
  "orders": [
    {
      "ingredients": [
        "60d3463f7034a000269f45e7",
        "60d3463f7034a000269f45e9",
        "60d3463f7034a000269f45e8",
        "60d3463f7034a000269f45ea"
      ],
      "_id": "",
      "status": "done",
      "number": 0,
      "createdAt": "2021-06-23T14:43:22.587Z",
      "updatedAt": "2021-06-23T14:43:22.603Z"
    }
  ],
  "total": 1,
  "totalToday": 1
}
```

Бэкенд не возвращает итоговую стоимость заказа.

Количество всех заказов содержится в поле `total`, а количество заказов за день — в поле `totalToday`.

## Получить заказы конкретного пользователя

```
GET https://stellarburgers.education-services.ru/api/orders
```

При успешном подключении бэкенд вернёт максимум 50 последних заказов пользователя. Они сортируются по времени обновления.

Пример успешного ответа:

```
{
  "success": true,
```

```
"orders": [
  {
    "ingredients": [
      "60d3463f7034a000269f45e9",
      "60d3463f7034a000269f45e7"
    ],
    "_id": "",
    "status": "done",
    "number": 1,
    "createdAt": "2021-06-23T20:11:01.403Z",
    "updatedAt": "2021-06-23T20:11:01.406Z"
  },
  {
    "ingredients": [
      "60d3463f7034a000269f45e9"
    ],
    "_id": "",
    "status": "done",
    "number": 3,
    "createdAt": "2021-06-23T20:13:23.654Z",
    "updatedAt": "2021-06-23T20:13:23.657Z"
  }
],
"total": 2,
"totalToday": 2
}
```

Бэкенд не возвращает итоговую стоимость заказа. Статусы «Отменён», «Готовится» и «Выполнен» формируются на основании поля `status` каждого заказа.

Сервер не возвращает поля `owner`: при сокет-соединении с персональной лентой заказов нужно предоставить серверу авторизационный токен.

Если выполнить запрос без авторизации, вернётся код ответа 401 Unauthorized.

```
{
  "success": false,
  "message": "You should be authorised"
}
```