

FakeStoreAPI (v2.1.11)

Download OpenAPI specification: [Download](#)

E-mail: support@fakestoreapi.com | URL: <https://fakestoreapi.com/docs>

A free fake API for testing and prototyping e-commerce applications.

Products

Get all products

Retrieve a list of all available products.

Responses

› 200 Success

– 400 Bad request

GET /products

Request samples

[JavaScript \(Browser\)](#)

[Node.js \(Axios\)](#)

[PHP](#)

[cURL](#)

[Python](#)

[Copy](#)

```
fetch('https://fakestoreapi.com/products')
  .then(response => response.json())
  .then(data => console.log(data));
```

Response samples

200

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
[  
  - {  
      "id": 0,  
      "title": "string",  
      "price": 0.1,  
      "description": "string",  
      "category": "string",  
      "image": "http://example.com"  
    }  
]
```

Add a new product

Create a new product.

REQUEST BODY SCHEMA: application/json
required

id	integer
----	---------

title	string
-------	--------

price	number <float>
-------	----------------

description	string
category	string
image	string <uri>

Responses

› 201 Product created successfully

– 400 Bad request

POST /products

Request samples

Payload

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Content type

application/json

Copy

```
{  
    "id": 0,  
    "title": "string",  
    "price": 0.1,  
    "description": "string",  
    "category": "string",  
    "image": "http://example.com"  
}
```

Response samples

201

Content type
application/json

[Copy](#)

```
{  
    "id": 0,  
    "title": "string",  
    "price": 0.1,  
    "description": "string",  
    "category": "string",  
    "image": "http://example.com"  
}
```

Get a single product

Retrieve details of a specific product by ID.

PATH PARAMETERS

id	integer
required	

Responses

➤ **200 Success**

– **400 Bad request**

GET /products/{id}

Request samples

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Copy

```
fetch('https://fakestoreapi.com/products/1')
  .then(response => response.json())
  .then(data => console.log(data));
```

Response samples

200

Content type

application/json

Copy

```
{
  "id": 0,
  "title": "string",
  "price": 0.1,
  "description": "string",
  "category": "string",
  "image": "http://example.com"
}
```

Update a product

Update an existing product by ID.

PATH PARAMETERS

id integer
required

REQUEST BODY SCHEMA: application/json
required

id integer

title string

price number <float>

description string

category string

image string <uri>

Responses

> **200** Product updated successfully

– **400** Bad request

PUT /products/{id}

Request samples

Payload

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Content type

application/json

[Copy](#)

```
{  
    "id": 0,  
    "title": "string",  
    "price": 0.1,  
    "description": "string",  
    "category": "string",  
    "image": "http://example.com"  
}
```

Response samples

[200](#)

Content type

application/json

[Copy](#)

```
{  
    "id": 0,  
    "title": "string",  
    "price": 0.1,  
    "description": "string",  
    "category": "string",  
    "image": "http://example.com"  
}
```

Delete a product

Delete a specific product by ID.

PATH PARAMETERS

id integer
required

Responses

– **200** Product deleted successfully

– **400** Bad request

DELETE /products/{id}

Request samples

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Copy

```
fetch('https://fakestoreapi.com/products/1', {
  method: 'DELETE'
})
.then(response => response.json())
.then(data => console.log(data));
```

Carts

Get all carts

Retrieve a list of all available carts.

Responses

> 200 Success

- 400 Bad request

GET /carts

Request samples

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Copy

```
fetch('https://fakestoreapi.com/carts')
  .then(response => response.json())
  .then(data => console.log(data));
```

Response samples

200

Content type

application/json

Copy Expand all Collapse all

```
[  
  - {  
      "id": 0,  
      "userId": 0,  
      + "products": [ ... ]  
    }  
]
```

Add a new cart

Create a new cart.

REQUEST BODY SCHEMA: application/json

required

id integer

userId integer

products > Array of objects (Product)

Responses

› **201** Cart created successfully

– **400** Bad request

POST /carts

Request samples

Payload

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Content type

application/json

Copy Expand all Collapse all

```
{  
  "id": 0,  
  "userId": 0,  
  -  "products": [  
    +  { ... }  
  ]}
```

}

Response samples

201**Content type**

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
    "id": 0,  
    "userId": 0,  
    - "products": [  
        + { ... }  
    ]  
}
```

Get a single cart

Retrieve details of a specific cart by ID.

PATH PARAMETERS

id	integer
required	

Responses

➤ **200 Success**

– **400 Bad request**

GET /carts/{id}

Request samples

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Copy

```
fetch('https://fakestoreapi.com/carts/1')
  .then(response => response.json())
  .then(data => console.log(data));
```

Response samples

200

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

{

"id": 0,

"userId": 0,

- "products":

+ { ... }

1

3

Update a cart

Update an existing cart by ID.

PATH PARAMETERS

id integer
required

REQUEST BODY SCHEMA: application/json

required

id integer

userId integer

products > Array of objects (Product)

Responses

› **200** Cart updated successfully– **400** Bad request

PUT /carts/{id}

Request samples

[Payload](#)[JavaScript \(Browser\)](#)[Node.js \(Axios\)](#)[PHP](#)[cURL](#)[Python](#)

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#){
 "id": 0,
 "userId": 0,
 "products": [
 { ... }
]
}

Response samples

200

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
    "id": 0,  
    "userId": 0,  
    - "products": [  
        + { ... }  
    ]  
}
```

Delete a cart

Delete a specific cart by ID.

PATH PARAMETERS

id integer
required

Responses

– **200** Cart deleted successfully

– **400** Bad request

DELETE /carts/{id}

Request samples

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Copy

```
fetch('https://fakestoreapi.com/carts/1', {
  method: 'DELETE'
})
.then(response => response.json())
.then(data => console.log(data));
```

👤 Users

Get all users

Retrieve a list of all users.

Responses

➤ 200 Success

– 400 Bad request

GET /users

Request samples

[JavaScript \(Browser\)](#)[Node.js \(Axios\)](#)[PHP](#)[cURL](#)[Python](#)[Copy](#)

```
fetch('https://fakestoreapi.com/users')
  .then(response => response.json())
  .then(data => console.log(data));
```

Response samples

[200](#)

Content type

application/json

[Copy](#)[Expand all](#)[Collapse all](#)

```
[  
  - {  
      "id": 0,  
      "username": "string",  
      "email": "string",  
      "password": "string"  
    }  
]
```

Add a new user

Create a new user.

REQUEST BODY SCHEMA: application/json
required

id	integer
username	string
email	string
password	string

Responses

> **201** User created successfully

– **400** Bad request

POST /users

Request samples

Payload

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Content type

application/json

Copy

```
{  
    "id": 0,  
    "username": "string",  
    "email": "string",  
    "password": "string"  
}
```

Response samples

201

Content type
application/json

[Copy](#)

```
{  
    "id": 0,  
    "username": "string",  
    "email": "string",  
    "password": "string"  
}
```

Get a single user

Retrieve details of a specific user by ID.

PATH PARAMETERS

id
required integer

Responses

➤ **200 Success**

– **400 Bad request**

GET /users/{id}

Request samples

[JavaScript \(Browser\)](#)[Node.js \(Axios\)](#)[PHP](#)[cURL](#)[Python](#)[Copy](#)

```
fetch('https://fakestoreapi.com/users/1')
  .then(response => response.json())
  .then(data => console.log(data));
```

Response samples

[200](#)

Content type

application/json

[Copy](#)

```
{
  "id": 0,
  "username": "string",
  "email": "string",
  "password": "string"
}
```

Update a user

Update an existing user by ID.

PATH PARAMETERS

id integer
required

REQUEST BODY SCHEMA: application/json
required

id integer

username	string
----------	--------

email	string
-------	--------

password	string
----------	--------

Responses

➤ 200 User updated successfully

– 400 Bad request

PUT /users/{id}

Request samples

Payload

JavaScript (Browser)

Node.js (Axios)

PHP

cURL

Python

Content type

application/json

Copy

```
{  
  "id": 0,  
  "username": "string",  
  "email": "string",  
  "password": "string"  
}
```

Response samples

200

Content type
application/json

```
{  
    "id": 0,  
    "username": "string",  
    "email": "string",  
    "password": "string"  
}
```

Copy

Delete a user

Delete a specific user by ID.

PATH PARAMETERS

id integer
required

Responses

– 200 User deleted successfully

– 400 Bad request

DELETE /users/{id}

Request samples

[JavaScript \(Browser\)](#)[Node.js \(Axios\)](#)[PHP](#)[cURL](#)[Python](#)[Copy](#)

```
fetch('https://fakestoreapi.com/users/1', {
  method: 'DELETE'
})
.then(response => response.json())
.then(data => console.log(data));
```

Auth

Login

Authenticate a user.

REQUEST BODY SCHEMA: application/json

required

username string

password string

Responses

> **200** Login successful

– **400** Bad request

POST /auth/login

Request samples

[Payload](#)[JavaScript \(Browser\)](#)[Node.js \(Axios\)](#)[PHP](#)[cURL](#)[Python](#)

Content type

application/json

[Copy](#)

```
{  
    "username": "string",  
    "password": "string"  
}
```

Response samples

[200](#)

Content type

application/json

[Copy](#)

```
{  
    "token": "string"  
}
```