

API request

url: <https://fakestoreapi.com/products>

Pruebas con thunderClient

GetAllProducts

The screenshot shows the ThunderClient interface. The top bar displays the method 'GET' and the URL 'https://fakestoreapi.com/products'. The status is '200 OK', size is '10.37 KB', and time is '657 ms'. The 'Query' tab is selected, showing 'Query Parameters' with a table for parameters. The 'Response' tab is also visible, showing the JSON response for the first product.

GET <https://fakestoreapi.com/products> Send

Status: 200 OK Size: 10.37 KB Time: 657 ms

Query Headers 2 Auth Body Tests Pre Run

Query Parameters

parameter	value
-----------	-------

Response Headers 14 Cookies Results Docs {}

```
1 [
2   {
3     "id": 1,
4     "title": "Fjallraven - Foldsack No. 1
5       Backpack, Fits 15 Laptops",
6     "price": 109.95,
7     "description": "Your perfect pack for
8       everyday use and walks in the forest.
9       Stash your laptop (up to 15 inches) in
10      the padded sleeve, your everyday",
11    "category": "men's clothing",
12    "image": "https://fakestoreapi.com/img
13      /81fPKd-2AYL._AC_SL1500_.jpg",
14    "rating": {
15      "rate": 3.9,
16      "count": 120
17    }
18  }
19 ]
```

GetProduct

The screenshot shows the ThunderClient interface. The top bar displays the method 'GET' and the URL 'https://fakestoreapi.com/products/20'. The status is '200 OK', size is '438 Bytes', and time is '654 ms'. The 'Query' tab is selected, showing 'Query Parameters' with a table for parameters. The 'Response' tab is also visible, showing the JSON response for the product with ID 20.

GET <https://fakestoreapi.com/products/20> Send

Status: 200 OK Size: 438 Bytes Time: 654 ms

Query Headers 2 Auth Body Tests Pre Run

Query Parameters

parameter	value
-----------	-------

Response Headers 14 Cookies Results Docs {}

```
1 {
2   "id": 20,
3   "title": "DANVOUY Womens T Shirt Casual Cotton
4     Short",
5   "price": 12.99,
6   "description": "95%Cotton,5%Spandex, Features:
7     Casual, Short Sleeve, Letter Print,V-Neck
8     ,Fashion Tees, The fabric is soft and has
9     some stretch., Occasion: Casual/Office
10    /Beach/School/Home/Street. Season: Spring
11    ,Summer,Autumn,Winter.",
12   "category": "women's clothing",
13   "image": "https://fakestoreapi.com/img
14     /61pHAEJ4NML._AC_UX679_.jpg",
15   "rating": {
16     "rate": 3.6,
17     "count": 145
18   }
19 }
```

## AddProduct

POSThttps://fakestoreapi.com/productsSend

QueryHeaders 2AuthBody 1TestsPre Run

JSONXMLTextFormForm-encodeGraphQLBinary

JSON ContentFormat

```
1 {
2   "id": 21,
3   "title": "test mmproduct",
4   "price": 13.5,
5   "description": "lorem ipsum set",
6   "image": "https://i.pravatar.cc",
7   "category": "electronic"
8 }
9
```

Status: 200 OKSize: 135 BytesTime: 888 ms

ResponseHeaders 14CookiesResultsDocs{} ≡

```
1 {
2   "id": 21,
3   "title": "test mmproduct",
4   "price": 13.5,
5   "description": "lorem ipsum set",
6   "image": "https://i.pravatar.cc",
7   "category": "electronic"
8 }
```

## UpdateProduct

PUThttps://fakestoreapi.com/products/21Send

QueryHeaders 2AuthBody 1TestsPre Run

JSONXMLTextFormForm-encodeGraphQLBinary

JSON ContentFormat

```
1 {
2   "id": 21,
3   "title": "test product",
4   "price": 13.5,
5   "description": "Actualizacion",
6   "image": "https://i.pravatar.cc",
7   "category": "medicina"
8 }
```

Status: 200 OKSize: 129 BytesTime: 870 ms

ResponseHeaders 14CookiesResultsDocs{} ≡

```
1 {
2   "id": 21,
3   "title": "test product",
4   "price": 13.5,
5   "description": "Actualizacion",
6   "image": "https://i.pravatar.cc",
7   "category": "medicina"
8 }
```

## DeleteProduct

DELETEhttps://fakestoreapi.com/products/20Send

QueryHeaders 2AuthBodyTestsPre Run

Query Parameters

☐ parametervalue

Status: 200 OKSize: 438 BytesTime: 835 ms

ResponseHeaders 14CookiesResultsDocs{} ≡

```
1 {
2   "id": 20,
3   "title": "DANVOUY Womens T Shirt Casual Cotton
4     Short",
5   "price": 12.99,
6   "description": "95%Cotton,5%Spandex, Features:
7     Casual, Short Sleeve, Letter Print,V-Neck
8     ,Fashion Tees, The fabric is soft and has
9     some stretch., Occasion: Casual/Office
10    /Beach/School/Home/Street. Season: Spring
11    ,Summer,Autumn,Winter.",
12   "category": "women's clothing",
13   "image": "https://fakestoreapi.com/img
14     /61pHAEJ4NML._AC_UX679_.jpg",
15   "rating": {
16     "rate": 3.6,
17     "count": 145
18   }
19 }
```

Pruebas con código

GetAllProducts

```
def GetAllProducts():
    url = 'https://fakestoreapi.com/products'
    try:
        response = requests.get(url)
        response.raise_for_status()
        Respuesta = response.json()
        print("Listado de productos")
        print('-----')
        print(json.dumps(Respuesta, indent=4, ensure_ascii=False))
    except requests.exceptions.RequestException as e:
        print("Error al obtener la lista de productos:", e)
        if response is not None:
            print("Respuesta del servidor:", response.text)
```

Selecciona una opción (1-6): 1

Listado de productos

```
-----
[
  {
    "id": 1,
    "title": "Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops",
    "price": 109.95,
    "description": "Your perfect pack for everyday use and walks in the forest. Stash your laptop (up to 15 inches) in the padded sleeve, your everyday",
    "category": "men's clothing",
    "image": "https://fakestoreapi.com/img/81fPKd-2AYL. AC SL1500 .jpg",
    "rating": {
      "rate": 3.9,
      "count": 120
    }
  },
  {
    "id": 2,
    "title": "Mens Casual Premium Slim Fit T-Shirts ",
    "price": 22.3,
    "description": "Slim-fitting style, contrast raglan long sleeve, three-button henley placket, light weight & soft fabric for breathable and comfortable wearing. And Solid stitched shirts with round neck made for durability and a great fit for casual fashion wear and diehard baseball fans. The H
```

## GetProduct

```
def GetProduct():
    id_product = input('ID del producto: \n')
    url = 'https://fakestoreapi.com/products/' + id_product
    try:
        response = requests.get(url)
        response.raise_for_status()
        Respuesta = response.json()
        print("Búsqueda de producto")
        print(json.dumps(Respuesta, indent=4, ensure_ascii=False))
    except requests.exceptions.RequestException as e:
        print("Error al buscar el producto:", e)
        if response is not None:
            print("Respuesta del servidor:", response.text)
```

```
07-04-2021
Selecciona una opción (1-6): 2
ID del producto:
12
Búsqueda de producto
{
  "id": 12,
  "title": "WD 4TB Gaming Drive Works with Playstation 4 Portable External Hard Drive",
  "price": 114,
  "description": "Expand your PS4 gaming experience, Play anywhere Fast and easy, setup Sleek design with high capacity, 3-year manufacturer's limited warranty",
  "category": "electronics",
  "image": "https://fakestoreapi.com/img/6lmtL65D4cL. AC SX679 .jpg",
  "rating": {
    "rate": 4.8,
    "count": 400
  }
}
```

## AddProduct

```
def AddProduct():
    product_id = int(input("Ingrese el ID del producto: "))
    title = input("Ingrese el título del producto: ")
    price = float(input("Ingrese el precio del producto: "))
    description = input("Ingrese la descripción del producto: ")
    category = input("Ingrese la categoría del producto: ")
    image = input("Ingrese la URL de la imagen del producto: ")
    rating_rate = float(input("Ingrese la calificación del producto: "))
    rating_count = int(input("Ingrese el número de calificaciones del producto: "))

    datos = {
        "id": product_id,
        "title": title,
        "price": price,
        "description": description,
        "category": category,
        "image": image,
        "rating": {
            "rate": rating_rate,
            "count": rating_count
        }
    }
```

```
url = 'https://fakestoreapi.com/products'

try:
    response = requests.post(url, json=datos, headers={'Content-Type': 'application/json'})
    response.raise_for_status()
    added_product = response.json()
    print("Producto agregado exitosamente:", added_product)
except requests.exceptions.RequestException as e:
    print("Error al agregar el producto:", e)
    if response is not None:
        print("Respuesta del servidor:", response.text)
```

```
Selecciona una opción (1-6): 3
Ingrese el ID del producto: 12
Ingrese el título del producto: ddd
Ingrese el precio del producto: 12
Ingrese la descripción del producto: dd
Ingrese la categoría del producto: d
Ingrese la URL de la imagen del producto:
Ingrese la calificación del producto: 12
Ingrese el número de calificaciones del producto: 1
Producto agregado exitosamente: {'id': 21, 'title': 'ddd', 'price': 12, 'description': 'dd', 'image':
'', 'category': 'd'}
```

## UpdateProduct

```
def UpdateProduct():
    product_id = input("Ingrese el ID del producto a modificar: ")
    url = 'https://fakestoreapi.com/products/' + product_id

    title = input("Ingrese el nuevo título del producto (o presione Enter para omitir): ")
    price = input("Ingrese el nuevo precio del producto (o presione Enter para omitir): ")
    description = input("Ingrese la nueva descripción del producto (o presione Enter para omitir): ")
    category = input("Ingrese la nueva categoría del producto (o presione Enter para omitir): ")
    image = input("Ingrese la nueva URL de la imagen del producto (o presione Enter para omitir): ")
    rating_rate = input("Ingrese la nueva calificación del producto (o presione Enter para omitir): ")
    rating_count = input("Ingrese el nuevo número de calificaciones del producto (o presione Enter para omitir): ")

    datos = {}
    if title:
        datos['title'] = title
    if price:
        datos['price'] = float(price)
    if description:
        datos['description'] = description
    if category:
        datos['category'] = category
    if image:
        datos['image'] = image
    if rating_rate:
        if 'rating' not in datos:
            datos['rating'] = {}
        datos['rating']['rate'] = float(rating_rate)
    if rating_count:
        if 'rating' not in datos:
            datos['rating'] = {}
        datos['rating']['count'] = int(rating_count)
```

```
    if rating_rate:
        if 'rating' not in datos:
            datos['rating'] = {}
        datos['rating']['rate'] = float(rating_rate)
    if rating_count:
        if 'rating' not in datos:
            datos['rating'] = {}
        datos['rating']['count'] = int(rating_count)

    try:
        response = requests.put(url, json=datos, headers={'Content-Type': 'application/json'})
        response.raise_for_status()
        updated_product = response.json()
        print("Producto modificado exitosamente:", updated_product)
    except requests.exceptions.RequestException as e:
        print("Error al modificar el producto:", e)
        if response is not None:
            print("Respuesta del servidor:", response.text)
```

```
Selecciona una opción (1-6): 4
Ingrese el ID del producto a modificar: 12
Ingrese el nuevo título del producto (o presione Enter para omitir):
Ingrese el nuevo precio del producto (o presione Enter para omitir):
Ingrese la nueva descripción del producto (o presione Enter para omitir):
Ingrese la nueva categoría del producto (o presione Enter para omitir):
Ingrese la nueva URL de la imagen del producto (o presione Enter para omitir):
Ingrese la nueva calificación del producto (o presione Enter para omitir):
Ingrese el nuevo número de calificaciones del producto (o presione Enter para omitir):

Producto modificado exitosamente: {'id': 12}
```

## DeleteProduct

```
def DeleteProduct():
    product_id = input("Ingrese el ID del producto a eliminar: ")
    url = 'https://fakestoreapi.com/products/' + product_id

    try:
        response = requests.delete(url, headers={'Content-Type': 'application/json'})
        response.raise_for_status()
        print("Producto eliminado exitosamente.")
    except requests.exceptions.RequestException as e:
        print("Error al eliminar el producto:", e)
        if response is not None:
            print("Respuesta del servidor:", response.text)
```

```
Administración de Productos:
1. Consultar todos los productos
2. Consultar un producto en específico
3. Agregar un nuevo producto
4. Modificar producto en específico
5. Eliminar un producto
6. Salir
Selecciona una opción (1-6): 5
Ingrese el ID del producto a eliminar: 20
Producto eliminado exitosamente.
```