



Guia de Uso - API de Produtos



Status: API FUNCIONANDO!

Sua API REST de produtos está rodando em **http://localhost:3000** e pronta para ser usada!



Suas API Keys

Você tem **2 API Keys** disponíveis:



1 API Key de Admin (para gerenciar produtos)

```
49e516cb-aeb1-44aa-9d76-f9341db7973a
```

Permissões: Criar, editar, deletar e consultar produtos



2 API Key Read-Only (para consultar produtos nos seus apps)

```
223bd275-39e4-4c35-998b-be537a5850f1
```

Permissões: Apenas consultar produtos (ideal para apps públicos)



Documentação Interativa

Acesse a documentação Swagger (já em português!) em:

http://localhost:3000/api-docs

Lá você pode **testar todos os endpoints** diretamente no navegador!



Endpoints Disponíveis



Endpoints Públicos (use a API Key Read-Only)

1. Listar Todos os Produtos

```
GET /api/products
```

```
Header: X-API-Key: 223bd275-39e4-4c35-998b-be537a5850f1
```

Query Parameters (opcionais):

- `page` - Número da página (padrão: 1)
- `limit` - Itens por página (padrão: 10)
- `category` - Filtrar por categoria
- `brand` - Filtrar por marca

- `condition` - Filtrar por condição (novo, semi-novo)
- `minPrice` - Preço mínimo
- `maxPrice` - Preço máximo

Exemplo: `GET /api/products?page=1&limit=20&category=iMac&condition=novo`

2. Buscar Produtos

```
GET /api/products/search?q=macbook
Header: X-API-Key: 223bd275-39e4-4c35-998b-be537a5850f1
```

Query Parameters:

- `q` - Termo de busca (busca no nome e descrição)
 - `category` - Filtrar por categoria
 - `brand` - Filtrar por marca
 - `condition` - Filtrar por condição
-

3. Obter Detalhes de Um Produto

```
GET /api/products/:id
Header: X-API-Key: 223bd275-39e4-4c35-998b-be537a5850f1
```

Exemplo: `GET /api/products/1`

4. Listar Categorias

```
GET /api/categories
Header: X-API-Key: 223bd275-39e4-4c35-998b-be537a5850f1
```

Endpoints Administrativos (use a API Key de Admin)

5. Criar Produto

```
POST /api/admin/products
Header: X-API-Key: 49e516cb-aeb1-44aa-9d76-f9341db7973a
Content-Type: application/json

Body:
{
  "name": "MacBook Pro M3 14\"",
  "description": "Notebook profissional...",
  "price": 15999.90,
  "category": "MacBook",
  "condition": "novo",
  "brand": "Apple",
  "model": "MacBook Pro 14",
  "specs": {
    "processor": "M3 Pro",
    "ram": "16GB",
    "storage": "512GB SSD"
  },
  "images": [
    { "url": "https://shop.arizona.edu/images/products/414973-apple-macbook-air-13-inch-m4-chip-16gb-memory-512gb-ssd-storage-laptop-media-midnight-1.jpg", "order": 1 }
  ]
}
```

6. Criar Múltiplos Produtos (Bulk)

```
POST /api/admin/products/bulk
Header: X-API-Key: 49e516cb-aeb1-44aa-9d76-f9341db7973a
Content-Type: application/json

Body:
{
  "products": [
    { /* produto 1 */ },
    { /* produto 2 */ }
  ]
}
```

7. Atualizar Produto

```
PATCH /api/admin/products/:id
Header: X-API-Key: 49e516cb-aeb1-44aa-9d76-f9341db7973a
Content-Type: application/json

Body:
{
  "price": 14999.90,
  "description": "Nova descrição..."
}
```

8. Deletar Produto

```
DELETE /api/admin/products/:id
Header: X-API-Key: 49e516cb-aeb1-44aa-9d76-f9341db7973a
```

9. Estatísticas

```
GET /api/admin/stats
Header: X-API-Key: 49e516cb-aeb1-44aa-9d76-f9341db7973a
```

Exemplos de Código

JavaScript (Node.js / Replit)

```
// Buscar todos os produtos
async function buscarProdutos() {
  const response = await fetch('http://localhost:3000/api/products', {
    headers: {
      'X-API-Key': '223bd275-39e4-4c35-998b-be537a5850f1'
    }
  });

  const dados = await response.json();

  if (dados.success) {
    console.log(`Total de produtos: ${dados.meta.total}`);
    dados.data.forEach(produto => {
      console.log(`${produto.name} - R$ ${produto.price}`);
    });
  }
}

buscarProdutos();
```

```
// Buscar produtos específicos
async function buscarMacBooks() {
  const response = await fetch(
    'http://localhost:3000/api/products/search?q=macbook&condition=novo',
    {
      headers: {
        'X-API-Key': '223bd275-39e4-4c35-998b-be537a5850f1'
      }
    }
  );

  const dados = await response.json();
  return dados.data;
}
```

```
// Criar um produto (admin)
async function criarProduto(produto) {
  const response = await fetch('http://localhost:3000/api/admin/products', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'X-API-Key': '49e516cb-aeb1-44aa-9d76-f9341db7973a'
    },
    body: JSON.stringify(produto)
  });

  return await response.json();
}
```

Python

```
import requests

# Buscar produtos
def buscar_produtos():
    headers = {'X-API-Key': '223bd275-39e4-4c35-998b-be537a5850f1'}
    response = requests.get('http://localhost:3000/api/products', headers=headers)
    dados = response.json()

    if dados['success']:
        print(f"Total de produtos: {dados['meta']['total']}")
        for produto in dados['data']:
            print(f"{produto['name']} - R$ {produto['price']}")

buscar_produtos()
```

PHP

```
<?php

// Buscar produtos
function buscarProdutos() {
    $headers = [
        'X-API-Key: 223bd275-39e4-4c35-998b-be537a5850f1'
    ];

    $context = stream_context_create([
        'http' => [
            'header' => implode("\r\n", $headers)
        ]
    ]);

    $response = file_get_contents('http://localhost:3000/api/products', false, $context);
    $dados = json_decode($response, true);

    if ($dados['success']) {
        foreach ($dados['data'] as $produto) {
            echo "{$produto['name']} - R$ {$produto['price']}\n";
        }
    }
}

buscarProdutos();
?>
```

Como Usar no Seu App do Replit

1. **Copie a API Key Read-Only** (para consultar produtos)
2. **Faça requisições HTTP** para a URL da API
3. **Use nos seus projetos** (pode ser qualquer framework: React, Vue, Angular, etc.)

Exemplo com React:

```
import { useState, useEffect } from 'react';

function ListaProdutos() {
  const [produtos, setProdutos] = useState([]);

  useEffect(() => {
    fetch('http://localhost:3000/api/products', {
      headers: {
        'X-API-Key': '223bd275-39e4-4c35-998b-be537a5850f1'
      }
    })
    .then(res => res.json())
    .then(data => {
      if (data.success) {
        setProdutos(data.data);
      }
    });
  }, []);

  return (
    <div>
      <h1>Meus Produtos</h1>
      {produtos.map(produto => (
        <div key={produto.id}>
          <h3>{produto.name}</h3>
          <p>R$ {produto.price}</p>
          {produto.images.map(img => (
            <img key={img.id} src={img.url} alt={produto.name} />
          ))}
        </div>
      ))}
    </div>
  );
}
```



Formato das Respostas

Todas as respostas seguem este padrão:

Sucesso:

```
{
  "success": true,
  "data": { /* seus dados aqui */ },
  "meta": { /* informações de paginação */ }
}
```

Erro:

```
{  
  "success": false,  
  "error": "Mensagem de erro",  
  "statusCode": 400  
}
```

Próximos Passos

1. Deploy da API (quando estiver pronto)

- Você vai receber uma URL pública (ex: `https://sua-api.abacus.ai`)
- Substitua `localhost:3000` pela URL pública nos seus apps
- A API ficará disponível 24/7

2. Web App de Gerenciamento (próxima etapa)

Vou criar uma interface web moderna onde você poderá:

- ☒ Fazer upload de arquivos ZIP com produtos
- ☒ Ver todos os produtos em uma tabela bonita
- ☒ Adicionar/editar/deletar produtos individualmente
- ☒ Gerenciar imagens
- ☒ Ver estatísticas

? Perguntas Frequentes

Q: A API funciona em produção?

A: Sim! Quando você fizer o deploy, ela ficará disponível 24/7 em uma URL pública.

Q: Posso usar em múltiplos apps?

A: Sim! A mesma API pode ser consumida por quantos apps você quiser.

Q: As imagens ficam hospedadas onde?

A: No próximo passo vamos configurar cloud storage (AWS S3, Cloudinary, etc) para hospedar as imagens.

Q: Como adiciono produtos do arquivo ZIP?

A: Vou criar uma interface web onde você faz upload do ZIP e ele processa automaticamente!

Tudo Pronto!

Sua API está funcionando perfeitamente! Agora você pode:

1. ☒ Testar os endpoints na documentação: `http://localhost:3000/api-docs`
2. ☒ Integrar no seu app do Replit usando os exemplos acima
3. ☒ Aguardar eu criar o Web App de gerenciamento

Quer que eu crie o Web App de gerenciamento agora? 