



Guia de Integração da API de Produtos



URL Base da API

```
https://bancodeprodutos.abacusai.app
```



Autenticação

Todos os endpoints requerem API Key no header:

```
X-API-Key: sk_admin_master456
```



Endpoints Disponíveis

1. Listar Produtos (Público)

```
GET /api/products
```

Parâmetros de Query:

- `page` (opcional): Número da página (padrão: 1)
- `limit` (opcional): Itens por página (padrão: 50, máx: 100)
- `search` (opcional): Buscar por nome ou descrição
- `category` (opcional): Filtrar por categoria
- `sortBy` (opcional): Campo para ordenar (name, price, createdAt)
- `order` (opcional): asc ou desc

Exemplo de Resposta:

```
{
  "data": [
    {
      "id": 1,
      "name": "Xiaomi Redmi 9A",
      "description": "Smartphone semi novo...",
      "price": 499.90,
      "category": "Celulares",
      "categoryId": 1,
      "condition": "Semi Novo",
      "brand": "Xiaomi",
      "model": "Redmi 9A",
      "images": [
        {
          "id": 1,
          "url": "/uploads/products/1_1234567890_imagem_1.jpg"
        }
      ],
      "createdAt": "2025-11-23T20:30:59.811Z",
      "updatedAt": "2025-11-23T20:30:59.811Z"
    }
  ],
  "total": 2621,
  "page": 1,
  "limit": 50,
  "totalPages": 53
}
```

2. Buscar Produto por ID

```
GET /api/products/{id}
```

Exemplo:

```
GET /api/products/1
```

3. Listar Categorias

```
GET /api/categories
```

Resposta:

```
[
  {
    "id": 1,
    "name": "Celulares",
    "description": "Smartphones e acessórios",
    "_count": {
      "products": 150
    }
  }
]
```

Exemplos de Código

JavaScript / Node.js

```

// Usando Fetch API
const API_URL = 'https://bancodeprodutos.abacusai.app';
const API_KEY = 'sk_admin_master456';

// Listar produtos
async function getProducts(page = 1, limit = 50) {
  const response = await fetch(
    `${API_URL}/api/products?page=${page}&limit=${limit}`,
    {
      headers: {
        'X-API-Key': API_KEY
      }
    }
  );
  return await response.json();
}

// Buscar produto específico
async function getProduct(id) {
  const response = await fetch(
    `${API_URL}/api/products/${id}`,
    {
      headers: {
        'X-API-Key': API_KEY
      }
    }
  );
  return await response.json();
}

// Buscar produtos por categoria
async function getProductsByCategory(categoryName) {
  const response = await fetch(
    `${API_URL}/api/products?category=${encodeURIComponent(categoryName)}`,
    {
      headers: {
        'X-API-Key': API_KEY
      }
    }
  );
  return await response.json();
}

// Buscar produtos
async function searchProducts(query) {
  const response = await fetch(
    `${API_URL}/api/products?search=${encodeURIComponent(query)}`,
    {
      headers: {
        'X-API-Key': API_KEY
      }
    }
  );
  return await response.json();
}

// Uso:
const products = await getProducts(1, 20);
console.log(products.data);

```

React / Next.js

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

const API_URL = 'https://bancodeprodutos.abacusai.app';
const API_KEY = 'sk_admin_master456';

function ProductList() {
  const [products, setProducts] = useState([]);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    async function fetchProducts() {
      try {
        const response = await fetch(`${API_URL}/api/products?limit=20`, {
          headers: { 'X-API-Key': API_KEY }
        });
        const data = await response.json();
        setProducts(data.data);
      } catch (error) {
        console.error('Erro ao buscar produtos:', error);
      } finally {
        setLoading(false);
      }
    }

    fetchProducts();
  }, []);

  if (loading) return <div>Carregando...</div>;

  return (
    <div className="grid grid-cols-3 gap-4">
      {products.map(product => (
        <div key={product.id} className="border p-4 rounded">
          {product.images[0] && (
            <img
              src={`${API_URL}${product.images[0].url}`}
              alt={product.name}
              className="w-full h-48 object-cover"
            />
          )}
          <h3 className="font-bold mt-2">{product.name}</h3>
          <p className="text-gray-600">{product.category}</p>
          {product.price && (
            <p className="text-blue-600 font-bold">
              R$ {product.price.toFixed(2)}
            </p>
          )}
        </div>
      )}
    </div>
  );
}
```

Python

```
import requests

API_URL = 'https://bancodeprodutos.abacusai.app'
API_KEY = 'sk_admin_master456'

headers = {
    'X-API-Key': API_KEY
}

# Listar produtos
def get_products(page=1, limit=50):
    response = requests.get(
        f'{API_URL}/api/products',
        params={'page': page, 'limit': limit},
        headers=headers
    )
    return response.json()

# Buscar produto por ID
def get_product(product_id):
    response = requests.get(
        f'{API_URL}/api/products/{product_id}',
        headers=headers
    )
    return response.json()

# Buscar produtos
def search_products(query):
    response = requests.get(
        f'{API_URL}/api/products',
        params={'search': query},
        headers=headers
    )
    return response.json()

# Uso:
products = get_products(page=1, limit=20)
for product in products['data']:
    print(f"{product['name']} - R$ {product['price']}")
```

PHP

```
<?php

$API_URL = 'https://bancoedeprodutos.abacusai.app';
$API_KEY = 'sk_admin_master456';

// Listar produtos
function getProducts($page = 1, $limit = 50) {
    global $API_URL, $API_KEY;

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "$API_URL/api/products?page=$page&limit=$limit");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, [
        "X-API-Key: $API_KEY"
    ]);

    $response = curl_exec($ch);
    curl_close($ch);

    return json_decode($response, true);
}

// Buscar produto por ID
function getProduct($id) {
    global $API_URL, $API_KEY;

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, "$API_URL/api/products/$id");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, [
        "X-API-Key: $API_KEY"
    ]);

    $response = curl_exec($ch);
    curl_close($ch);

    return json_decode($response, true);
}

// Uso:
$products = getProducts(1, 20);
foreach ($products['data'] as $product) {
    echo $product['name'] . " - R$ " . $product['price'] . "\n";
}
?>
```

Flutter / Dart

```
import 'package:http/http.dart' as http;
import 'dart:convert';

class ProductService {
  static const String apiUrl = 'https://bancodeprodutos.abacusai.app';
  static const String apiKey = 'sk_admin_master456';

  static Future<Map<String, dynamic>> getProducts({
    int page = 1,
    int limit = 50,
  }) async {
    final response = await http.get(
      Uri.parse('$apiUrl/api/products?page=$page&limit=$limit'),
      headers: {'X-API-Key': apiKey},
    );

    if (response.statusCode == 200) {
      return json.decode(response.body);
    } else {
      throw Exception('Erro ao buscar produtos');
    }
  }

  static Future<Map<String, dynamic>> getProduct(int id) async {
    final response = await http.get(
      Uri.parse('$apiUrl/api/products/$id'),
      headers: {'X-API-Key': apiKey},
    );

    if (response.statusCode == 200) {
      return json.decode(response.body);
    } else {
      throw Exception('Erro ao buscar produto');
    }
  }
}
```



URLs das Imagens

As imagens são retornadas com URLs relativas. Para exibir, combine com a URL base:

```
// Imagem do produto
const imageUrl = product.images[0].url;
const fullImageUrl = `https://i.ytimg.com/vi/XB4aLwKtKoM/hq720.jpg?sqp=-oaymwE7CK4-
FEIIDSFryq4qpAy0IARUAAAAAGAEIAADIQj0AgKJD8AEB-AG-B4AC0AWKAgwIABABGD4gZShi-
MA8=&rs=A0n4CLCw9vWPkz0-Z8PJ62h0cvBTkHyuka`;

// Exemplo:
// /uploads/products/1_1234567890_imagem_1.jpg
// Vira:
// https://lh3.googleusercontent.com/LDM1EeL7nGX50JKYNv7iyD0nea9sKBmw48s-
fjd0fi-80XnJmhYU_1DQyiMK-7wQoCdZ2-h4fNfFFHgUf8HVHABa-WZ0v8k7QyTQvLLaUD5tRTxp-
G4YV0_KqDiguJQKNrGoNYlypXVpIv1Xdruh_bNI
```


Filtros e Busca

Buscar por texto

```
GET /api/products?search=xiaomi
```

Filtrar por categoria

```
GET /api/products?category=Celulares
```

Ordenar produtos

```
GET /api/products?sortBy=price&order=asc  
GET /api/products?sortBy=name&order=desc  
GET /api/products?sortBy=createdAt&order=desc
```

Paginação

```
GET /api/products?page=2&limit=30
```

Combinar filtros





```
GET /api/products?category=Celulares&search=samsung&sortBy=price&order=asc&page=1&limit=20
```

Dicas de Performance

1. **Use paginação:** Não busque todos os produtos de uma vez
2. **Cache local:** Guarde os produtos em cache para não refazer requisições
3. **Lazy loading:** Carregue imagens sob demanda
4. **Debounce na busca:** Aguarde o usuário parar de digitar antes de buscar

Segurança da API Key

IMPORTANTE:

-  Use a API Key em requisições server-side quando possível
 -  Configure CORS no seu backend para limitar origens
 -  Se usar no frontend, a API Key ficará exposta (considere criar uma proxy)
 -  Nunca commite a API Key em repositórios públicos
-



Exemplo de App Completo (React)

```

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

const API_URL = 'https://bancodeprodutos.abacusai.app';
const API_KEY = 'sk_admin_master456';

function App() {
  const [products, setProducts] = useState([]);
  const [categories, setCategories] = useState([]);
  const [selectedCategory, setSelectedCategory] = useState('');
  const [searchQuery, setSearchQuery] = useState('');
  const [page, setPage] = useState(1);
  const [loading, setLoading] = useState(false);

  // Buscar produtos
  useEffect(() => {
    async function fetchProducts() {
      setLoading(true);
      try {
        const params = new URLSearchParams({
          page: page,
          limit: 20,
          ...(selectedCategory && { category: selectedCategory }),
          ...(searchQuery && { search: searchQuery })
        });

        const response = await fetch(
          `${API_URL}/api/products?${params}`,
          { headers: { 'X-API-Key': API_KEY } }
        );
        const data = await response.json();
        setProducts(data.data);
      } catch (error) {
        console.error('Erro:', error);
      } finally {
        setLoading(false);
      }
    }

    fetchProducts();
  }, [page, selectedCategory, searchQuery]);

  // Buscar categorias
  useEffect(() => {
    async function fetchCategories() {
      const response = await fetch(
        `${API_URL}/api/categories`,
        { headers: { 'X-API-Key': API_KEY } }
      );
      const data = await response.json();
      setCategories(data);
    }

    fetchCategories();
  }, []);

  return (
    <div className="container mx-auto p-4">
      <h1 className="text-3xl font-bold mb-6">Produtos</h1>

      <div className="mb-6 flex gap-4">
        <input

```

```

    type="text"
    placeholder="Buscar produtos..."
    value={searchQuery}
    onChange={(e) => setSearchQuery(e.target.value)}
    className="flex-1 px-4 py-2 border rounded"
  />

  <select
    value={selectedCategory}
    onChange={(e) => setSelectedCategory(e.target.value)}
    className="px-4 py-2 border rounded"
  >
    <option value="">Todas as categorias</option>
    {categories.map(cat => (
      <option key={cat.id} value={cat.name}>
        {cat.name} ({cat._count.products})
      </option>
    ))}
  </select>
</div>

{/* Lista de produtos */}
{loading ? (
  <div className="text-center py-8">Carregando...</div>
) : (
  <div className="grid grid-cols-1 md:grid-cols-3 lg:grid-cols-4 gap-4">
    {products.map(product => (
      <div key={product.id} className="border rounded-lg overflow-hidden">
        {product.images[0] && (
          <img
            src={` ${API_URL} ${product.images[0].url} `}
            alt={product.name}
            className="w-full h-48 object-cover"
          />
        )}
        <div className="p-4">
          <h3 className="font-semibold mb-2 truncate">{product.name}</h3>
          <p className="text-sm text-gray-600 mb-2">{product.category}</p>
          {product.price && (
            <p className="text-lg font-bold text-blue-600">
              R$ {product.price.toFixed(2)}
            </p>
          )}
        </div>
      </div>
    ))}
  </div>
)}



{/* Paginação */}
<div className="mt-6 flex justify-center gap-2">
  <button
    onClick={() => setPage(p => Math.max(1, p - 1))}
    disabled={page === 1}
    className="px-4 py-2 bg-blue-500 text-white rounded disabled:bg-gray-300"
  >
    Anterior
  </button>
  <span className="px-4 py-2">Página {page}</span>
  <button
    onClick={() => setPage(p => p + 1)}
    className="px-4 py-2 bg-blue-500 text-white rounded"
  >
    Próximo
  </button>
</div>

```

```
        Próxima
      </button>
    </div>
  </div>
);
}

export default App;
```

Suporte

-  **Documentação Swagger:** <https://bancodeprodutos.abacusai.app/api-docs>
 -  **Admin Dashboard:** <https://bancodeprodutos.abacusai.app/admin>
-

✅ Pronto! Agora você pode integrar a API em qualquer aplicação! 🚀