

GUIA DE INTEGRAÇÃO - API DE PRODUTOS

OBJETIVO

Este guia ensina **EXATAMENTE** como fazer buscas **RÁPIDAS** e **EFICIENTES** na nossa API de produtos.

REGRA DE OURO

SEMPRE use `includeImages=false` para LISTAGENS e BUSCAS!

Por quê? Porque cada imagem tem ~500KB em Base64. Se você buscar 20 produtos com 4 imagens cada = **40MB de dados!** Isso demora 20-30 segundos.

CREDECIAIS

URL Base: <https://bancoodeprodutos.abacusai.app>

API Key: ed126afe-92a8-415f-b886-a1b0fed24ff5

Header: X-API-Key

3 CENÁRIOS DE USO

CENÁRIO 1: BUSCA DE PRODUTOS (Mais Comum)

Use Case: Usuário digitando no campo de busca, autocomplete, listagens.

FORMA CORRETA (0.2-0.5 segundos)

`GET /api/products/search?q=iphone&includeImages=false&limit=20`

Retorna: - Nome do produto - Preço - Descrição - Marca, modelo, categoria - **SEM imagens** (você busca depois se precisar)

FORMA ERRADA (15-30 segundos)

`GET /api/products/search?q=iphone`
Retorna TODAS as imagens em Base64 = LENTO!

CENÁRIO 2: LISTAGEM COM FOTO (Grid de Produtos)

Use Case: Página de resultados com foto pequena (thumbnail).

FORMA CORRETA (1-2 segundos)

`GET /api/products?imageLimit=1&limit=20`

Retorna: - Dados do produto - **Apenas 1 imagem** (primeira foto)

FORMA ERRADA (20-40 segundos)

```
GET /api/products?limit=20
# Retorna TODAS as imagens de cada produto = LENTO!
```

CENÁRIO 3: DETALHES COMPLETOS DO PRODUTO

Use Case: Usuário clicou no produto, quer ver todas as fotos.

FORMA CORRETA (2-3 segundos)

```
GET /api/products/123
Retorna: - Todos os dados - Todas as imagens (galeria completa)
```

EXEMPLOS PRÁTICOS POR LINGUAGEM

JavaScript / Fetch

```
// BUSCA RÁPIDA (sem imagens)
async function buscarProdutos(termo) {
  const response = await fetch(
    `https://bancoodeprodutos.abacusai.app/api/products/search?q=${termo}&includeImages=false&limit=20`,
    {
      headers: {
        'X-API-Key': 'ed126afe-92a8-415f-b886-a1b0fed24ff5'
      }
    }
  );

  const data = await response.json();
  return data.data; // Array de produtos
}

// BUSCAR 1 IMAGEM POR PRODUTO (listagem)
async function listarProdutos(pagina = 1) {
  const response = await fetch(
    `https://bancoodeprodutos.abacusai.app/api/products?page=${pagina}&limit=20&imageLimit=1`,
    {
      headers: {
        'X-API-Key': 'ed126afe-92a8-415f-b886-a1b0fed24ff5'
      }
    }
  );

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

// DETALHES COMPLETOS (todas imagens)
async function detalhesProduto(id) {
  const response = await fetch(
    `https://bancoodeprodutos.abacusai.app/api/products/${id}`,
    {
      headers: {
        'X-API-Key': 'ed126afe-92a8-415f-b886-a1b0fed24ff5'
      }
    }
  );
}
```

```

    {
      headers: {
        'X-API-Key': 'ed126afe-92a8-415f-b886-a1b0fed24ff5'
      }
    }
  );
}

return await response.json();
}

```

PHP / cURL

```

<?php
// BUSCA RÁPIDA (sem imagens)
function buscarProdutos($termo) {
  $url = "https://bancoodeprodutos.abacusai.app/api/products/search?" . http_build_query([
    'q' => $termo,
    'includeImages' => 'false',
    'limit' => 20
  ]);

  $ch = curl_init($url);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
  curl_setopt($ch, CURLOPT_HTTPHEADER, [
    'X-API-Key: ed126afe-92a8-415f-b886-a1b0fed24ff5'
  ]);

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

  $data = json_decode($response, true);
  return $data['data'];
}

// LISTAGEM COM 1 IMAGEM
function listarProdutos($pagina = 1) {
  $url = "https://bancoodeprodutos.abacusai.app/api/products?" . http_build_query([
    'page' => $pagina,
    'limit' => 20,
    'imageLimit' => 1
  ]);

  $ch = curl_init($url);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
  curl_setopt($ch, CURLOPT_HTTPHEADER, [
    'X-API-Key: ed126afe-92a8-415f-b886-a1b0fed24ff5'
  ]);

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

  $data = json_decode($response, true);
  return $data['data'];
}

```

?>

Python / Requests

```
import requests

API_URL = "https://bancoodeprodutos.abacusai.app"
API_KEY = "ed126afe-92a8-415f-b886-a1b0fed24ff5"

# BUSCA RÁPIDA (sem imagens)
def buscar_produtos(termo):
    response = requests.get(
        f"{API_URL}/api/products/search",
        params={
            'q': termo,
            'includeImages': 'false',
            'limit': 20
        },
        headers={'X-API-Key': API_KEY}
    )
    return response.json()['data']

# LISTAGEM COM 1 IMAGEM
def listar_produtos(pagina=1):
    response = requests.get(
        f"{API_URL}/api/products",
        params={
            'page': pagina,
            'limit': 20,
            'imageLimit': 1
        },
        headers={'X-API-Key': API_KEY}
    )
    return response.json()['data']

# DETALHES COMPLETOS
def detalhes_produto(id):
    response = requests.get(
        f"{API_URL}/api/products/{id}",
        headers={'X-API-Key': API_KEY}
    )
    return response.json()
```

PARÂMETROS DISPONÍVEIS

Endpoint: /api/products/search

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
q	string	SIM	-	Termo de busca
includeImages	boolean	Não	true	Use false para busca rápida!

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
imageLimit	number	Não	1	Quantas imagens retornar (0 = todas)
limit	number	Não	10	Produtos por página (máx: 100)
page	number	Não	1	Número da página
category	string	Não	-	Filtrar por categoria
brand	string	Não	-	Filtrar por marca
condition	string	Não	-	Filtrar por condição

Endpoint: /api/products

Parâmetro	Tipo	Obrigatório	Padrão	Descrição
includeImages	boolean	Não	true	Use false para listagem rápida!
imageLimit	number	Não	1	Quantas imagens retornar
limit	number	Não	10	Produtos por página
page	number	Não	1	Número da página
category	string	Não	-	Filtrar por categoria
brand	string	Não	-	Filtrar por marca
condition	string	Não	-	Filtrar por condição
minPrice	number	Não	-	Preço mínimo
maxPrice	number	Não	-	Preço máximo

EXEMPLOS COMPLETOS DE URLs

BUSCAS RÁPIDAS (Recomendado)

```
# Buscar "iphone" sem imagens (0.2s)
GET /api/products/search?q=iphone&includeImages=false

# Buscar "samsung" com 1 imagem (1s)
GET /api/products/search?q=samsung&imageLimit=1&limit=20

# Buscar "notebook" na categoria "Eletrônicos" sem imagens
GET /api/products/search?q=notebook&category=Eletrônicos&includeImages=false

# Buscar produtos entre R$ 1000 e R$ 5000
GET /api/products?minPrice=1000&maxPrice=5000&includeImages=false
```

BUSCAS LENTAS (Evite!)

```
# Vai retornar TODAS as imagens = 20-30s
GET /api/products/search?q=iphone

# Muitos produtos com todas imagens = TIMEOUT
GET /api/products?limit=100

# Sem limitar imagens = LENTO
GET /api/products/search?q=samsung&limit=50
```

ESTRUTURA DA RESPOSTA

Resposta de Busca/Listagem

```
{  
  "data": [  
    {  
      "id": 123,  
      "name": "iPhone 14 Pro Max 256GB",  
      "description": "Smartphone Apple iPhone...",  
      "price": "7999.00",  
      "brand": "Apple",  
      "model": "iPhone 14 Pro Max",  
      "category": "Smartphones",  
      "condition": "novo",  
      "images": [  
        {  
          "id": 456,  
          "url": "...",  
          "order": 1  
        }  
      ],  
      "categoryRelation": {  
        "id": 1,  
        "name": "Smartphones",  
        "slug": "smartphones"  
      },  
      "createdAt": "2024-11-24T10:30:00.000Z",  
      "updatedAt": "2024-11-24T10:30:00.000Z"  
    }  
  ],  
  "meta": {  
    "page": 1,  
    "limit": 20,  
    "total": 150,  
    "totalPages": 8,  
    "query": "iphone",  
    "includeImages": true,  
    "imageLimit": 1  
  }  
}
```

Resposta de Produto Individual

```
{  
  "id": 123,  
  "name": "iPhone 14 Pro Max 256GB",  
  "description": "Smartphone Apple iPhone 14 Pro Max com 256GB...",  
  "price": "7999.00",  
  "brand": "Apple",  
  "model": "iPhone 14 Pro Max",  
  "category": "Smartphones",  
  "condition": "novo",
```

```

"images": [
  {
    "id": 456,
    "url": "...",
    "order": 1
  },
  {
    "id": 457,
    "url": "...",
    "order": 2
  }
],
"categoryRelation": {
  "id": 1,
  "name": "Smartphones",
  "slug": "smartphones"
}
}

```

COMPARAÇÃO DE PERFORMANCE

Requisição	Incluir Imagens?	Tempo Médio	Tamanho	Uso Recomendado
?includeImages=false	Não	0.2s	50KB	Buscas, autocomplete, listagens
?imageLimit=1	1 imagem	1-2s	800KB	Grid de produtos
?imageLimit=2	2 imagens	2-3s	1.5MB	Detalhes parciais
Padrão (sem params)	Todas	5-10s	5-10MB	Evite!

FLUXO RECOMENDADO

1 Usuário digita no campo de busca

```

// Busca SEM imagens (ultra-rápido)
const resultados = await buscarProdutos(termo, false);

// Mostra apenas:
// - Nome
// - Preço
// - Descrição curta

```

2 Usuário vê a lista de resultados

```

// Busca com 1 imagem (thumbnail)
const produtos = await listarProdutos(pagina, 1);

// Mostra cards com:
// - Foto pequena
// - Nome
// - Preço

```

3 Usuário clica em um produto

```
// Busca produto completo
const produto = await detalhesProduto(id);

// Mostra:
// - Galeria completa
// - Todas as informações
// - Descrição completa
```

TROUBLESHOOTING

Problema: “Busca muito lenta (15-30s)”

Causa: Você está retornando todas as imagens.

Solução:

```
# Errado
GET /api/products/search?q=termo

# Correto
GET /api/products/search?q=termo&includeImages=false
```

Problema: “Erro 401 - Unauthorized”

Causa: API Key incorreta ou faltando.

Solução:

```
// Sempre inclua o header
headers: {
  'X-API-Key': 'ed126afe-92a8-415f-b886-a1b0fed24ff5'
}
```

Problema: “Timeout / Sem resposta”

Causa: Muitos produtos com muitas imagens.

Solução:

```
# Evite limite muito alto
GET /api/products?limit=200

# Use paginação
GET /api/products?limit=20&page=1&imageLimit=1
```

Problema: “Imagens não aparecem”

Causa: As imagens estão em Base64, você precisa usar no HTML/CSS.

Solução:

```
<!-- Correto -->

```

EXEMPLOS DE INTEGRAÇÃO COMPLETA

Exemplo 1: Busca com Autocomplete

```
// Debounce para não fazer requisição a cada letra
let timeout;
const searchInput = document.getElementById('search');

searchInput.addEventListener('input', (e) => {
  clearTimeout(timeout);

  timeout = setTimeout(async () => {
    const termo = e.target.value;

    if (termo.length < 3) return; // Mínimo 3 caracteres

    // Busca SEM imagens (ultra-rápido)
    const response = await fetch(
      `https://bancoodeprodutos.abacusai.app/api/products/search?q=${termo}&includeImages=false&limit=10
    `
    headers: {
      'X-API-Key': 'ed126afe-92a8-415f-b886-a1b0fed24ff5'
    }
  );
  });

  const data = await response.json();
  mostrarSugestoes(data.data);
}, 300); // Aguarda 300ms após usuário parar de digitar
});
```

Exemplo 2: Grid de Produtos com Paginação

```
async function carregarProdutos(pagina = 1) {
  // Busca COM 1 imagem (thumbnail)
  const response = await fetch(
    `https://bancoodeprodutos.abacusai.app/api/products?page=${pagina}&limit=20&imageLimit=1`,
  {
    headers: {
      'X-API-Key': 'ed126afe-92a8-415f-b886-a1b0fed24ff5'
    }
  );
  );

  const data = await response.json();

  // Renderizar produtos
  const grid = document.getElementById('products-grid');
  grid.innerHTML = data.data.map(produto => `
    <div class="product-card" onclick="verDetalhes(${produto.id})">
      ${produto.images && produto.images[0]
        ? `` :
        '<div class="no-image">Sem foto</div>'}
    <h3>${produto.name}</h3>
  
```

```

        <p class="price">R$ ${parseFloat(produto.price).toFixed(2)}</p>
    </div>
`).join('');

// Renderizar paginação
renderizarPaginacao(data.meta);
}

```

Exemplo 3: Página de Detalhes

```

async function verDetalhes(id) {
    // Busca produto completo (todas imagens)
    const response = await fetch(
        `https://bancoodeprodutos.abacusai.app/api/products/${id}`,
        {
            headers: {
                'X-API-Key': 'ed126afe-92a8-415f-b886-a1b0fed24ff5'
            }
        }
    );

    const produto = await response.json();

    // Renderizar galeria
    const galeria = document.getElementById('gallery');
    galeria.innerHTML = produto.images.map((img, i) => `
        
    `).join('');

    // Renderizar informações
    document.getElementById('product-name').textContent = produto.name;
    document.getElementById('product-price').textContent =
        `R$ ${parseFloat(produto.price).toFixed(2)}`;
    document.getElementById('product-description').textContent =
        produto.description;
}

```

CHECKLIST DE INTEGRAÇÃO

Antes de ir para produção, verifique:

- Estou usando `includeImages=false` em buscas/autocomplete?
 - Estou usando `imageLimit=1` em listagens?
 - Estou usando paginação (`limit 50`)?
 - Estou incluindo o header `X-API-Key` em todas requisições?
 - Implementei tratamento de erros (try/catch)?
 - Testei com vários termos de busca?
 - Verifiquei o tempo de resposta no DevTools?
 - Implementei debounce em campos de busca?
-

RESUMO - COPIE E COLE

```
// COPIE E COLE ISTO NO SEU CÓDIGO

const API_URL = 'https://bancoodeprodutos.abacusai.app';
const API_KEY = 'ed126afe-92a8-415f-b886-a1b0fed24ff5';

// 1. BUSCA RÁPIDA (sem imagens)
async function buscar(termo) {
  const res = await fetch(
    `${API_URL}/api/products/search?q=${termo}&includeImages=false&limit=20`,
    { headers: { 'X-API-Key': API_KEY } }
  );
  return (await res.json()).data;
}

// 2. LISTAGEM (com 1 foto)
async function listar(pagina = 1) {
  const res = await fetch(
    `${API_URL}/api/products?page=${pagina}&limit=20&imageLimit=1`,
    { headers: { 'X-API-Key': API_KEY } }
  );
  return (await res.json()).data;
}

// 3. DETALHES (completo)
async function detalhes(id) {
  const res = await fetch(
    `${API_URL}/api/products/${id}`,
    { headers: { 'X-API-Key': API_KEY } }
  );
  return await res.json();
}
```

SUPORTE

Dúvidas? Entre em contato: - Email: seu-email@exemplo.com - Documentação: <https://bancoodeprodutos.abacusai.app/api-docs>

Última atualização: 24/11/2024

Versão: 2.0

Status: TESTADO E FUNCIONANDO