Documentação da API - Sistema de Estoque para Peças de Carro

Versão: 1.0
Data: Junho 2025
Autor: Manus AI

Sumário

Introdução

A API do Sistema de Estoque para Peças de Carro é uma interface RESTful que permite integração completa com todas as funcionalidades do sistema. Esta API foi desenvolvida seguindo padrões modernos de desenvolvimento, oferecendo endpoints consistentes, documentação abrangente, e facilidade de integração.

URL Base

```
https://seu-dominio.com/api
```

Para desenvolvimento local:

```
http://localhost:5001/api
```

Formato de Dados

Todas as requisições e respostas utilizam formato JSON. Certifique-se de incluir o header `Content-Type: application/json` em todas as requisições POST e PUT.

Versionamento

A API utiliza versionamento através de headers. A versão atual é v1 e é definida automaticamente. Futuras versões serão especificadas através do header `API-Version`.

Autenticação

Sistema de Autenticação

A API utiliza autenticação baseada em sessão para simplicidade de integração. Para acessar endpoints protegidos, é necessário primeiro realizar login através do endpoint de autenticação.

Login

POST `/auth/login`

Realiza autenticação do usuário e estabelece sessão.

Parâmetros:

```
{
  "email": "usuario@empresa.com",
  "senha": "senha_do_usuario"
}
```

Resposta de Sucesso (200):

```
{
  "message": "Login realizado com sucesso",
  "user": {
    "id": 1,
    "nome": "Nome do Usuário",
    "email": "usuario@empresa.com",
    "role": "admin"
```

```
  }
}
```

Exemplo de Uso:

```
curl -X POST http://localhost:5001/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"email": "admin@empresa.com", "senha": "123456"}'
```

Logout

POST `/auth/logout`

Encerra a sessão atual do usuário.

Resposta de Sucesso (200):

```
{
  "message": "Logout realizado com sucesso"
}
```

Verificação de Sessão

GET `/auth/me`

Retorna informações do usuário autenticado atualmente.

Resposta de Sucesso (200):

```
{
  "id": 1,
  "nome": "Nome do Usuário",
  "email": "usuario@empresa.com",
  "role": "admin",
  "ativo": true,
  "data_cadastro": "2025-06-11T10:30:00"
}
```

Endpoints de Produtos

Listar Produtos

GET `/produtos`

Retorna lista paginada de todos os produtos cadastrados.

Parâmetros de Query: - `page` (opcional): Número da página (padrão: 1) - `per_page` (opcional): Itens por página (padrão: 50, máximo: 100) - `search` (opcional): Termo de busca para filtrar produtos - `categoria` (opcional): Filtrar por categoria específica - `estoque_baixo` (opcional): true para mostrar apenas produtos com estoque baixo

Resposta de Sucesso (200):


```
{
  "produtos": [
    {
      "id": 1,
      "nome": "Pastilha de Freio Dianteira",
      "descricao": "Pastilha de freio dianteira para Ford Focus
2012-2018",
      "categoria": "Freios",
      "codigo_referencia": "PF001",
      "preco_custo": 45.50,
      "preco_venda": 89.90,
      "quantidade_estoque": 25,
      "estoque_minimo": 5,
      "veiculos_compativeis": "Ford Focus 2012-2018 1.6/2.0",
      "ml_item_id": "MLB123456789",
      "ml_last_sync": "2025-06-11T15:30:00",
      "data_cadastro": "2025-06-10T09:15:00",
      "imagens": [
        {
          "id": 1,
          "url": "/static/images/produto_1_1.jpg",
          "principal": true
        }
      ]
    }
  ],
  "pagination": {
    "page": 1,
    "per_page": 50,
    "total": 150,
    "pages": 3
```

```
  }
}
```

Obter Produto Específico

GET `/produtos/{id}`

Retorna detalhes completos de um produto específico.

Resposta de Sucesso (200):

```
{
  "id": 1,
  "nome": "Pastilha de Freio Dianteira",
  "descricao": "Pastilha de freio dianteira para Ford Focus
2012-2018",
  "categoria": "Freios",
  "codigo_referencia": "PF001",
  "preco_custo": 45.50,
  "preco_venda": 89.90,
  "quantidade_estoque": 25,
  "estoque_minimo": 5,
  "veiculos_compativeis": "Ford Focus 2012-2018 1.6/2.0",
  "ml_item_id": "MLB123456789",
  "ml_last_sync": "2025-06-11T15:30:00",
  "data_cadastro": "2025-06-10T09:15:00",
  "imagens": [
    {
      "id": 1,
      "url": "/static/images/produto_1_1.jpg",
      "principal": true
    }
  ]
}
```

Criar Produto

POST `/produtos`

Cria um novo produto no sistema.

Parâmetros:

```
{
  "nome": "Nome do Produto",
  "descricao": "Descrição detalhada do produto",
  "categoria": "Categoria do Produto",
```

```
  "codigo_referencia": "REF001",
  "preco_custo": 45.50,
  "preco_venda": 89.90,
  "quantidade_estoque": 25,
  "estoque_minimo": 5,
  "veiculos_compativeis": "Veículos compatíveis",
  "ml_item_id": "MLB123456789"
}
```

Resposta de Sucesso (201):

```
{
  "message": "Produto criado com sucesso",
  "produto": {
    "id": 2,
    "nome": "Nome do Produto",
    // ... outros campos
  }
}
```

Atualizar Produto

PUT `/produtos/{id}`

Atualiza um produto existente.

Parâmetros: Mesmos do endpoint de criação

Resposta de Sucesso (200):

```
{
  "message": "Produto atualizado com sucesso",
  "produto": {
    "id": 1,
    // ... campos atualizados
  }
}
```

Excluir Produto

DELETE `/produtos/{id}`

Remove um produto do sistema.

Resposta de Sucesso (200):

```
{
  "message": "Produto excluído com sucesso"
}
```

Endpoints Auxiliares de Produtos

GET `/produtos/categorias`

Retorna lista de todas as categorias de produtos cadastradas.

Resposta de Sucesso (200):

```
{
  "categorias": [
    "Freios",
    "Suspensão",
    "Motor",
    "Elétrica",
    "Carroceria"
  ]
}
```

GET `/produtos/estoque-baixo`

Retorna produtos com estoque abaixo do mínimo configurado.

Resposta de Sucesso (200):

```
{
  "produtos": [
    {
      "id": 1,
      "nome": "Produto com Estoque Baixo",
      "quantidade_estoque": 2,
      "estoque_minimo": 5,
      "diferenca": -3
    }
  ]
}
```

Endpoints de Pedidos

Listar Pedidos

GET `/pedidos`

Retorna lista de pedidos recebidos do Mercado Livre.

Parâmetros de Query: - `page` (opcional): Número da página - `per_page` (opcional): Itens por página - `status` (opcional): Filtrar por status (paid, pending, cancelled) - `data_inicio` (opcional): Data inicial (formato: YYYY-MM-DD) - `data_fim` (opcional): Data final (formato: YYYY-MM-DD)

Resposta de Sucesso (200):


```
{
  "pedidos": [
    {
      "id": 123456789,
      "data_criacao": "2025-06-11T10:30:00",
      "data_fechamento": "2025-06-11T10:35:00",
      "total_amount": 179.80,
      "paid_amount": 179.80,
      "currency_id": "BRL",
      "status": "paid",
      "comprador_nickname": "COMPRADOR123",
      "comprador_email": "comprador@email.com",
      "itens": [
        {
          "ml_item_id": "MLB123456789",
          "titulo_item": "Pastilha de Freio Dianteira",
          "quantidade": 2,
          "preco_unitario": 89.90
        }
      ],
      "pagamentos": [
        {
          "id": 987654321,
          "metodo_pagamento": "credit_card",
          "status": "approved",
          "transaction_amount": 179.80
        }
      ]
    }
  ],
  "pagination": {
    "page": 1,
    "per_page": 50,
```

```
    "total": 25,
    "pages": 1
  }
}
```

Obter Pedido Específico

GET `/pedidos/{id}`

Retorna detalhes completos de um pedido específico.

Resposta de Sucesso (200):

```
{
  "id": 123456789,
  "data_criacao": "2025-06-11T10:30:00",
  "data_fechamento": "2025-06-11T10:35:00",
  "ultima_atualizacao": "2025-06-11T10:35:00",
  "total_amount": 179.80,
  "paid_amount": 179.80,
  "currency_id": "BRL",
  "status": "paid",
  "status_detalhe": "accredited",
  "comprador_id": 123456,
  "comprador_nickname": "COMPRADOR123",
  "comprador_email": "comprador@email.com",
  "shipping_id": 456789123,
  "itens": [
    {
      "ml_item_id": "MLB123456789",
      "titulo_item": "Pastilha de Freio Dianteira",
      "quantidade": 2,
      "preco_unitario": 89.90,
      "produto_interno_id": 1
    }
  ],
  "pagamentos": [
    {
      "id": 987654321,
      "payer_id": 123456,
      "metodo_pagamento": "credit_card",
      "status": "approved",
      "status_detalhe": "accredited",
      "transaction_amount": 179.80,
      "data_aprovacao": "2025-06-11T10:35:00",
      "data_criacao": "2025-06-11T10:30:00"
    }
  ]
}
```

Estatísticas de Pedidos

GET `/pedidos/estatisticas`

Retorna estatísticas consolidadas de pedidos e vendas.

Parâmetros de Query: - `periodo` (opcional): today, week, month, year (padrão: month)

Resposta de Sucesso (200):

```
{
  "total_pedidos": 25,
  "total_vendas": 4500.00,
  "ticket_medio": 180.00,
  "produtos_vendidos": 45,
  "pedidos_por_status": {
    "paid": 20,
    "pending": 3,
    "cancelled": 2
  },
  "vendas_por_dia": [
    {
      "data": "2025-06-11",
      "pedidos": 5,
      "valor": 900.00
    }
  ]
}
```

Endpoints de Usuários

Listar Usuários

GET `/usuarios`

Retorna lista de usuários do sistema. Requer permissão de administrador.

Resposta de Sucesso (200):

```
{
  "usuarios": [
    {
      "id": 1,
      "nome": "Administrador",
```

```
      "email": "admin@empresa.com",
      "role": "admin",
      "ativo": true,
      "data_cadastro": "2025-06-10T09:00:00"
    }
  ]
}
```

Criar Usuário

POST `/usuarios`

Cria um novo usuário no sistema. Requer permissão de administrador.

Parâmetros:

```
{
  "nome": "Nome do Usuário",
  "email": "usuario@empresa.com",
  "senha": "senha_segura",
  "role": "estoque"
}
```

Resposta de Sucesso (201):

```
{
  "message": "Usuário criado com sucesso",
  "usuario": {
    "id": 2,
    "nome": "Nome do Usuário",
    "email": "usuario@empresa.com",
    "role": "estoque",
    "ativo": true
  }
}
```

Atualizar Usuário

PUT `/usuarios/{id}`

Atualiza informações de um usuário. Requer permissão de administrador.

Parâmetros:

```
{
  "nome": "Nome Atualizado",
```

```
  "email": "novo_email@empresa.com",
  "role": "admin",
  "senha": "nova_senha"
}
```

Resposta de Sucesso (200):

```
{
  "message": "Usuário atualizado com sucesso",
  "usuario": {
    "id": 2,
    "nome": "Nome Atualizado",
    "email": "novo_email@empresa.com",
    "role": "admin"
  }
}
```

Desativar Usuário

DELETE `/usuarios/{id}`

Desativa um usuário do sistema. Requer permissão de administrador.

Resposta de Sucesso (200):

```
{
  "message": "Usuário desativado com sucesso"
}
```

Endpoints de Integração Mercado Livre

Status da Integração

GET `/ml/status`

Retorna informações sobre o status da integração com Mercado Livre.

Resposta de Sucesso (200):

```
{
  "has_access_token": true,
  "has_refresh_token": true,
  "user_id": "123456789",
```

```
  "client_id": "1234567890...",
  "produtos_vinculados_ml": 15
}
```

URL de Autenticação

GET `/ml/auth`

Retorna URL para iniciar processo de autenticação OAuth com Mercado Livre.

Resposta de Sucesso (200):

```
{
  "auth_url": "https://auth.mercadolivre.com.br/authorization?
response_type=code&client_id=..."
}
```

Sincronizar Estoque

POST `/ml/sync-stock/{produto_id}`

Sincroniza o estoque de um produto específico com o Mercado Livre.

Resposta de Sucesso (200):

```
{
  "message": "Estoque sincronizado com sucesso"
}
```

Sincronizar Todos os Estoques

POST `/ml/sync-all-stock`

Sincroniza o estoque de todos os produtos com ML Item ID configurado.

Resposta de Sucesso (200):

```
{
  "message": "Sincronização concluída",
  "success_count": 12,
  "error_count": 1,
  "total_products": 13
}
```

Buscar Detalhes de Pedido

GET `/ml/order/{order_id}`

Busca detalhes de um pedido específico diretamente do Mercado Livre.

Resposta de Sucesso (200):


```
{
  "id": 123456789,
  "status": "paid",
  "date_created": "2025-06-11T10:30:00.000-03:00",
  "total_amount": 179.80,
  "buyer": {
    "id": 123456,
    "nickname": "COMPRADOR123"
  },
  "order_items": [
    {
      "item": {
        "id": "MLB123456789",
        "title": "Pastilha de Freio Dianteira"
      },
      "quantity": 2,
      "unit_price": 89.90
    }
  ]
}
```


Webhook do Mercado Livre

POST `/ml/webhook`

Endpoint para receber notificações automáticas do Mercado Livre. Este endpoint é configurado automaticamente e não deve ser chamado manualmente.

Parâmetros de Query: - `user_id` : ID do usuário no Mercado Livre - `topic` : Tópico da notificação (orders_v2, payments)

Resposta de Sucesso (200):


```
{
  "message": "Notificação processada com sucesso"
}
```

Testar Conexão

GET `/ml/test-connection`

Testa a conectividade com a API do Mercado Livre.

Resposta de Sucesso (200):

```
{
  "message": "Conexão com ML estabelecida com sucesso",
  "user_id": "123456789",
  "nickname": "LOJA_TESTE",
  "email": "loja@teste.com"
}
```

Códigos de Resposta

Códigos de Sucesso

- **200 OK**: Requisição processada com sucesso
- **201 Created**: Recurso criado com sucesso
- **204 No Content**: Requisição processada, sem conteúdo de resposta

Códigos de Erro

- **400 Bad Request**: Dados da requisição inválidos
- **401 Unauthorized**: Autenticação necessária
- **403 Forbidden**: Permissão insuficiente
- **404 Not Found**: Recurso não encontrado
- **409 Conflict**: Conflito com estado atual do recurso
- **422 Unprocessable Entity**: Dados válidos mas não processáveis
- **500 Internal Server Error**: Erro interno do servidor

Formato de Resposta de Erro

```
{
  "error": "Descrição do erro",
  "code": "ERROR_CODE",
  "details": {
    "field": "Campo específico com erro",
    "message": "Mensagem detalhada"
```

```
  }
}
```

Exemplos de Uso

Exemplo Completo: Criar Produto e Sincronizar

```
# 1. Fazer login
curl -X POST http://localhost:5001/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"email": "admin@empresa.com", "senha": "123456"}' \
  -c cookies.txt

# 2. Criar produto
curl -X POST http://localhost:5001/api/produtos \
  -H "Content-Type: application/json" \
  -b cookies.txt \
  -d '{
    "nome": "Filtro de Óleo",
    "descricao": "Filtro de óleo para motores 1.0 e 1.6",
    "categoria": "Motor",
    "codigo_referencia": "FO001",
    "preco_custo": 15.00,
    "preco_venda": 29.90,
    "quantidade_estoque": 50,
    "estoque_minimo": 10,
    "veiculos_compativeis": "Ford Ka, Fiesta 1.0/1.6",
    "ml_item_id": "MLB987654321"
  }'

# 3. Sincronizar estoque com ML
curl -X POST http://localhost:5001/api/ml/sync-stock/2 \
  -b cookies.txt
```

Exemplo: Buscar Produtos com Filtros

```
# Buscar produtos da categoria "Freios" com estoque baixo
curl -X GET "http://localhost:5001/api/produtos?
categoria=Freios&estoque_baixo=true" \
  -b cookies.txt
```

Exemplo: Obter Estatísticas de Vendas

```
# Estatísticas do mês atual
curl -X GET "http://localhost:5001/api/pedidos/estatisticas?
periodo=month" \
  -b cookies.txt
```

Exemplo em Python

```
import requests

# Configuração
base_url = "http://localhost:5001/api"
session = requests.Session()

# Login
login_data = {
    "email": "admin@empresa.com",
    "senha": "123456"
}
response = session.post(f"{base_url}/auth/login",
json=login_data)
print(f"Login: {response.status_code}")

# Listar produtos
response = session.get(f"{base_url}/produtos")
produtos = response.json()
print(f"Total de produtos: {len(produtos['produtos'])}")

# Criar novo produto
novo_produto = {
    "nome": "Vela de Ignição",
    "descricao": "Vela de ignição NGK",
    "categoria": "Motor",
    "codigo_referencia": "VI001",
    "preco_custo": 8.50,
    "preco_venda": 16.90,
    "quantidade_estoque": 100,
    "estoque_minimo": 20,
    "veiculos_compativeis": "Universal"
}
response = session.post(f"{base_url}/produtos",
json=novo_produto)
print(f"Produto criado: {response.status_code}")

# Sincronizar com ML (se tiver ML Item ID)
produto_id = response.json()['produto']['id']
response = session.post(f"{base_url}/ml/sync-stock/
```

```
{produto_id}")
print(f"Sincronização: {response.status_code}")
```

Exemplo em JavaScript

```
// Configuração
const baseUrl = 'http://localhost:5001/api';

// Função para fazer login
async function login(email, senha) {
  const response = await fetch(`${baseUrl}/auth/login`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    credentials: 'include',
    body: JSON.stringify({ email, senha })
  });
  return response.json();
}

// Função para listar produtos
async function listarProdutos(filtros = {}) {
  const params = new URLSearchParams(filtros);
  const response = await fetch(`${baseUrl}/produtos?${params}`,
{
    credentials: 'include'
  });
  return response.json();
}

// Função para criar produto
async function criarProduto(produto) {
  const response = await fetch(`${baseUrl}/produtos`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    credentials: 'include',
    body: JSON.stringify(produto)
  });
  return response.json();
}

// Uso
(async () => {
  await login('admin@empresa.com', '123456');

  const produtos = await listarProdutos({ categoria: 'Motor' });
  console.log(`Produtos encontrados: ${produtos.produtos.length}
```

```
`);

  const novoProduto = {
    nome: 'Correia Dentada',
    categoria: 'Motor',
    preco_venda: 45.90,
    quantidade_estoque: 30
  };

  const resultado = await criarProduto(novoProduto);
  console.log('Produto criado:', resultado.message);
})();
```

SDKs e Bibliotecas

SDK Python (Exemplo)

```python
class EstoqueAPI:
    def __init__(self, base_url, email=None, senha=None):
        self.base_url = base_url
        self.session = requests.Session()
        if email and senha:
            self.login(email, senha)

    def login(self, email, senha):
        response = self.session.post(
            f"{self.base_url}/auth/login",
            json={"email": email, "senha": senha}
        )
        response.raise_for_status()
        return response.json()

    def listar_produtos(self, **filtros):
        response = self.session.get(
            f"{self.base_url}/produtos",
            params=filtros
        )
        response.raise_for_status()
        return response.json()

    def criar_produto(self, produto):
        response = self.session.post(
            f"{self.base_url}/produtos",
            json=produto
        )
        response.raise_for_status()
        return response.json()
```

```
    def sincronizar_estoque(self, produto_id):
        response = self.session.post(
            f"{self.base_url}/ml/sync-stock/{produto_id}"
        )
        response.raise_for_status()
        return response.json()

# Uso do SDK
api = EstoqueAPI("http://localhost:5001/api",
"admin@empresa.com", "123456")
produtos = api.listar_produtos(categoria="Motor")
```

Esta documentação fornece uma referência completa para integração com a API do sistema. Para dúvidas específicas ou funcionalidades não documentadas, consulte o código fonte ou entre em contato com o suporte técnico.

Versão da API: 1.0
Última Atualização: Junho 2025
Desenvolvido por: Manus AI