



Documentação da API - VEND

2.0

Versão: 1.0.1

Data: Dezembro 2025

Desenvolvedor: Daniel Clemente de Cayres Filho

Tecnologia: NestJS + Prisma + PostgreSQL



Índice

1. Introdução
2. Autenticação
3. Endpoints
 - 3.1 Auth (Autenticação)
 - 3.2 Users (Usuários)
 - 3.3 Segments (Segmentos)
 - 3.4 Lines (Linhas)
 - 3.5 Contacts (Contatos)
 - 3.6 Campaigns (Campanhas)
 - 3.7 Conversations (Conversas)
 - 3.8 Tabulations (Tabulações)
 - 3.9 Blocklist (Lista de Bloqueio)
 - 3.10 Evolution (Configurações Evolution API)
 - 3.11 Tags
 - 3.12 Templates (WhatsApp Cloud API)
 - 3.13 Reports (Relatórios)

- 3.14 Media (Mídias)
- 3.15 API Logs
- 3.16 API Messages (Mensagens Externas)
- 3.17 Webhooks
- 3.18 Health Check

4. Enums e Tipos

5. Modelos de Dados

6. Códigos de Erro

1. Introdução

O sistema NewVend é uma plataforma de gestão de atendimento via WhatsApp, integrando com a Evolution API para envio e recebimento de mensagens. Esta documentação descreve todos os endpoints disponíveis na API REST.

Base URL

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

Headers Padrão

```
Content-Type: application/json  
Authorization: Bearer <token>
```

2. Autenticação

A API utiliza autenticação JWT (JSON Web Token). Após realizar o login, você receberá um token que deve ser enviado no header `Authorization` de todas as requisições subsequentes.

Fluxo de Autenticação

1. Faça uma requisição POST para `/auth/login` com email e senha
2. Receba o token JWT na resposta
3. Inclua o token em todas as requisições: `Authorization: Bearer <token>`

Roles (Papéis)

Role	Descrição
<code>admin</code>	Acesso total ao sistema
<code>supervisor</code>	Gerenciamento de equipe e relatórios
<code>operator</code>	Atendimento ao cliente

3. Endpoints

3.1 Auth (Autenticação)

POST `/auth/login`

Realiza o login do usuário no sistema.

Autenticação: Não requer

Roles: Público

Request Body:

```
{
  "email": "usuario@email.com",
  "password": "senha123"
}
```

Campo	Tipo	Obrigatório	Descrição
email	string	✓	Email do usuário (formato válido)
password	string	✓	Senha do usuário

Response Success (200):

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "id": 1,
    "name": "Nome do Usuário",
    "email": "usuario@email.com",
    "role": "admin",
    "segment": 1,
    "line": 1,
    "status": "Online"
  }
}
```

Response Error (401):

```
{
  "statusCode": 401,
  "message": "Credenciais inválidas"
}
```

POST `/auth/logout`

Realiza o logout do usuário.

Autenticação: Requer JWT

Roles: Todos

Response Success (200):

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

GET `/auth/me`

Retorna os dados do usuário autenticado.

Autenticação: Requer JWT

Roles: Todos

Response Success (200):

```
{
  "id": 1,
  "name": "Nome do Usuário",
  "email": "usuario@email.com",
  "role": "admin",
  "segment": 1,
  "line": 1,
  "status": "Online",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

3.2 Users (Usuários)

POST `/users`

Cria um novo usuário.

Autenticação: Requer JWT

Roles: `admin`

Request Body:

```
{
  "name": "Nome do Usuário",
  "email": "usuario@email.com",
  "password": "senha123",
  "role": "operator",
  "segment": 1,
  "line": 1,
  "status": "Offline"
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	✓	Nome completo do usuário
email	string	✓	Email único do usuário
password	string	✓	Senha (mínimo 6 caracteres)
role	enum	✓	Papel: <code>admin</code> , <code>supervisor</code> , <code>operator</code>
segment	number	Condicional	ID do segmento (obrigatório para operator/supervisor)
line	number	✗	ID da linha vinculada
status	enum	✗	Status: <code>Online</code> , <code>Offline</code> (padrão: Offline)

Response Success (201):

```
{
  "id": 1,
  "name": "Nome do Usuário",
  "email": "usuario@email.com",
  "role": "operator",
  "segment": 1,
  "line": 1,
  "status": "Offline",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET `/users`

Lista todos os usuários.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Query Parameters:

Parâmetro	Tipo	Descrição
role	string	Filtrar por role
segment	number	Filtrar por segmento
status	string	Filtrar por status

Response Success (200):

```
[
  {
    "id": 1,
    "name": "Nome do Usuário",
    "email": "usuario@email.com",
    "role": "admin",
    "segment": null,
    "line": null,
    "status": "Online",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET `/users/online-operators`

Lista operadores online.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Query Parameters:

Parâmetro	Tipo	Descrição
segment	number	Filtrar por segmento

Response Success (200):

```
[
  {
    "id": 2,
```

```
[
  {
    "name": "Operador 1",
    "email": "operador1@email.com",
    "role": "operator",
    "segment": 1,
    "line": 1,
    "status": "Online"
  }
]
```

GET /users/:id

Busca um usuário pelo ID.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`

Response Success (200):

```
{
  "id": 1,
  "name": "Nome do Usuário",
  "email": "usuario@email.com",
  "role": "admin",
  "segment": null,
  "line": null,
  "status": "Online",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

PATCH /users/:id

Atualiza um usuário.

Autenticação: Requer JWT

Roles: `admin`

Request Body:

```
{
  "name": "Novo Nome",
  "email": "novoemail@email.com",
}
```



```
{
  "password": "novasenha",
  "role": "supervisor",
  "segment": 2,
  "line": 3,
  "status": "Online"
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	✗	Nome completo
email	string	✗	Email único
password	string	✗	Nova senha (mínimo 6 caracteres)
role	enum	✗	Papel: <code>admin</code> , <code>supervisor</code> , <code>operator</code>
segment	number/null	✗	ID do segmento
line	number/null	✗	ID da linha vinculada
status	enum	✗	Status: <code>Online</code> , <code>Offline</code>

Response Success (200):

```
{
  "id": 1,
  "name": "Novo Nome",
  "email": "novoemail@email.com",
  "role": "supervisor",
  "segment": 2,
  "line": 3,
  "status": "Online",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T12:00:00.000Z"
}
```

DELETE `/users/:id`

Remove um usuário.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

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

3.3 Segments (Segmentos)

POST `/segments`

Cria um novo segmento.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Request Body:

```
{
  "name": "Nome do Segmento"
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	<input checked="" type="checkbox"/>	Nome único do segmento

Response Success (201):

```
{
  "id": 1,
  "name": "Nome do Segmento",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET `/segments`

Lista todos os segmentos.

Autenticação: Requer JWT

Roles: Todos

Query Parameters:

Parâmetro	Tipo	Descrição
search	string	Buscar por nome

Response Success (200):

```
[
  {
    "id": 1,
    "name": "Segmento A",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET `/segments/:id`

Busca um segmento pelo ID.

Autenticação: Requer JWT

Roles: Todos

Response Success (200):

```
{
  "id": 1,
  "name": "Segmento A",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

PATCH `/segments/:id`

Atualiza um segmento.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Request Body:

```
{
  "name": "Novo Nome do Segmento"
}
```

Response Success (200):

```
{
  "id": 1,
  "name": "Novo Nome do Segmento",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T12:00:00.000Z"
}
```

DELETE `/segments/:id`

Remove um segmento.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
{
  "message": "Segmento removido com sucesso"
}
```

3.4 Lines (Linhas)

POST `/lines`

Cria uma nova linha.

Autenticação: Requer JWT

Roles: `admin`

Request Body:

```
{
  "phone": "5511999999999",
  "evolutionName": "Evolution01",
  "segment": 1,
  "oficial": false,
  "lineStatus": "active",
  "linkedTo": 1,
  "token": "token_whatsapp_business",
  "businessID": "business_id",
  "numberId": "number_id"
}
```

Campo	Tipo	Obrigatório	Descrição
phone	string	✓	Número de telefone (formato: 5511999999999)
evolutionName	string	✓	Nome da instância Evolution
segment	number	✗	ID do segmento
oficial	boolean	✗	Se é linha oficial WhatsApp Business
lineStatus	enum	✗	Status: <code>active</code> , <code>ban</code> (padrão: <code>active</code>)
linkedTo	number	✗	ID do usuário vinculado
token	string	✗	Token WhatsApp Business
businessID	string	✗	ID do Business WhatsApp
numberId	string	✗	ID do número WhatsApp

Response Success (201):

```
{
  "id": 1,
  "phone": "5511999999999",
  "lineStatus": "active",
  "segment": 1,
  "linkedTo": null,
  "evolutionName": "Evolution01",
  "oficial": false,
```

```
    "token": null,  
    "businessID": null,  
    "numberId": null,  
    "createdAt": "2024-01-01T00:00:00.000Z",  
    "updatedAt": "2024-01-01T00:00:00.000Z"  
  }  
}
```

GET /lines

Lista todas as linhas.

Autenticação: Requer JWT

Roles: `admin`

Query Parameters:

Parâmetro	Tipo	Descrição
segment	number	Filtrar por segmento
lineStatus	string	Filtrar por status
evolutionName	string	Filtrar por instância Evolution

Response Success (200):

```
[  
  {  
    "id": 1,  
    "phone": "5511999999999",  
    "lineStatus": "active",  
    "segment": 1,  
    "linkedTo": 1,  
    "evolutionName": "Evolution01",  
    "oficial": false,  
    "token": null,  
    "businessID": null,  
    "numberId": null,  
    "createdAt": "2024-01-01T00:00:00.000Z",  
    "updatedAt": "2024-01-01T00:00:00.000Z"  
  }  
]
```

GET /lines/schema

Retorna a estrutura esperada para criar uma linha.

Autenticação: Requer JWT

Roles: admin

Response Success (200):

```
{
  "message": "Estrutura esperada para criar uma linha",
  "required": {
    "phone": "string (obrigatório) - Ex: \"5511999999999\"",
    "evolutionName": "string (obrigatório) - Ex: \"Evolution01\""
  },
  "optional": {
    "segment": "number (opcional) - ID do segmento",
    "oficial": "boolean (opcional) - Se é linha oficial",
    "lineStatus": "string (opcional) - \"active\" ou \"ban\"",
    "linkedTo": "number (opcional) - ID do usuário vinculado",
    "token": "string (opcional)",
    "businessID": "string (opcional)",
    "numberId": "string (opcional)"
  },
  "example": {
    "phone": "5511999999999",
    "evolutionName": "Evolution01",
    "segment": 1,
    "oficial": false
  }
}
```

GET /lines/evolutions

Lista todas as instâncias Evolution disponíveis.

Autenticação: Requer JWT

Roles: admin

Response Success (200):

```
[
  {
```

```
    "id": 1,  
    "evolutionName": "Evolution01",  
    "evolutionUrl": "https://api.evolution.com",  
    "evolutionKey": "key123"  
  }  
]
```

GET /lines/instances/:evolutionName

Lista instâncias de uma Evolution específica.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
[  
  {  
    "instanceName": "instancia1",  
    "status": "connected"  
  }  
]
```

GET /lines/available/:segment

Lista linhas disponíveis para um segmento.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
[  
  {  
    "id": 1,  
    "phone": "5511999999999",  
    "lineStatus": "active",  
    "segment": 1,  
    "linkedTo": null  
  }  
]
```

GET /lines/:id

Busca uma linha pelo ID.

Autenticação: Requer JWT

Roles: admin

Response Success (200):

```
{
  "id": 1,
  "phone": "5511999999999",
  "lineStatus": "active",
  "segment": 1,
  "linkedTo": 1,
  "evolutionName": "Evolution01",
  "oficial": false,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET /lines/:id/qrcode

Obtém o QR Code para conectar a linha.

Autenticação: Requer JWT

Roles: admin

Response Success (200):

```
{
  "qrcode": "data:image/png;base64,..."
}
```

PATCH /lines/:id

Atualiza uma linha.

Autenticação: Requer JWT

Roles: admin

Request Body:

```
{
  "phone": "5511999999999",
  "segment": 2,
  "lineStatus": "active",
  "linkedTo": 2
}
```

Response Success (200):

```
{
  "id": 1,
  "phone": "5511999999999",
  "lineStatus": "active",
  "segment": 2,
  "linkedTo": 2,
  "evolutionName": "Evolution01",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T12:00:00.000Z"
}
```

POST `/lines/:id/ban`

Marca uma linha como banida.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
{
  "message": "Linha marcada como banida"
}
```

DELETE `/lines/:id`

Remove uma linha.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
{
  "message": "Linha removida com sucesso"
}
```

3.5 Contacts (Contatos)

POST `/contacts`

Cria um novo contato.

Autenticação: Requer JWT

Roles: `admin` , `supervisor` , `operator`

Request Body:

```
{
  "name": "Nome do Contato",
  "phone": "5511999999999",
  "segment": 1,
  "cpf": "12345678901",
  "contract": "CONTRATO123"
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	✓	Nome do contato
phone	string	✓	Telefone do contato
segment	number	✗	ID do segmento
cpf	string	✗	CPF do contato
contract	string	✗	Número do contrato

Response Success (201):

```
{
  "id": 1,
  "name": "Nome do Contato",
  "phone": "5511999999999",
  "segment": 1,
```

```
    "cpf": "12345678901",
    "contract": "CONTRATO123",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
}
```

GET /contacts

Lista todos os contatos.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`, `operator`

Query Parameters:

Parâmetro	Tipo	Descrição
search	string	Buscar por nome ou telefone
segment	number	Filtrar por segmento

Response Success (200):

```
[
  {
    "id": 1,
    "name": "Nome do Contato",
    "phone": "5511999999999",
    "segment": 1,
    "cpf": "12345678901",
    "contract": "CONTRATO123",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET /contacts/:id

Busca um contato pelo ID.

Autenticação: Requer JWT

Roles: `admin` , `supervisor` , `operator`

Response Success (200):

```
{
  "id": 1,
  "name": "Nome do Contato",
  "phone": "5511999999999",
  "segment": 1,
  "cpf": "12345678901",
  "contract": "CONTRATO123",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

PATCH `/contacts/:id`

Atualiza um contato.

Autenticação: Requer JWT

Roles: `admin` , `supervisor` , `operator`

Request Body:

```
{
  "name": "Novo Nome",
  "phone": "5511888888888",
  "segment": 2,
  "cpf": "98765432101",
  "contract": "CONTRATO456"
}
```

Response Success (200):

```
{
  "id": 1,
  "name": "Novo Nome",
  "phone": "5511888888888",
  "segment": 2,
  "cpf": "98765432101",
  "contract": "CONTRATO456",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

```
"updatedAt": "2024-01-01T12:00:00.000Z"
}
```

DELETE `/contacts/:id`

Remove um contato.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`

Response Success (200):

```
{
  "message": "Contato removido com sucesso"
}
```

3.6 Campaigns (Campanhas)

POST `/campaigns`

Cria uma nova campanha (suporta texto simples ou templates).

Autenticação: Requer JWT

Roles: `admin`, `supervisor`

Request Body:

```
{
  "name": "Nome da Campanha",
  "speed": "medium",
  "segment": "1",
  "useTemplate": true,
  "templateId": 1,
  "templateVariables": [
    { "key": "nome", "value": "João" },
    { "key": "valor", "value": "R$ 100,00" }
  ]
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	✓	Nome da campanha
speed	enum	✓	Velocidade: <code>fast</code> , <code>medium</code> , <code>slow</code>
segment	string	✓	ID do segmento
useTemplate	boolean	✗	Se deve usar template (padrão: <code>false</code>)
templateId	number	✗	ID do template (obrigatório se <code>useTemplate=true</code>)
templateVariables	array	✗	Variáveis para substituição no template

Response Success (201):

```
{
  "id": 1,
  "name": "Nome da Campanha",
  "speed": "medium",
  "segment": 1,
  "useTemplate": true,
  "templateId": 1,
  "createdAt": "2024-01-01T00:00:00.000Z"
}
```

POST `/campaigns/:id/upload`

Faz upload de contatos CSV para a campanha.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Content-Type: `multipart/form-data`

Request Body:

Campo	Tipo	Obrigatório	Descrição
file	file	✓	Arquivo CSV com contatos
message	string	✗	Mensagem para envio

Formato do CSV:

```
name,phone,segment
João Silva,5511999999999,1
Maria Santos,5511888888888,1
```

Response Success (200):

```
{
  "message": "Upload realizado com sucesso",
  "contactsAdded": 150
}
```

GET `/campaigns`

Lista todas as campanhas.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
[
  {
    "id": 1,
    "name": "Campanha 1",
    "contactName": "João",
    "contactPhone": "5511999999999",
    "contactSegment": 1,
    "dateTime": "2024-01-01T10:00:00.000Z",
    "lineReceptor": 1,
    "response": false,
    "speed": "medium",
    "retryCount": 0,
    "createdAt": "2024-01-01T00:00:00.000Z"
  }
]
```



```
}  
]
```

GET `/campaigns/:id`

Busca uma campanha pelo ID.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
{  
  "id": 1,  
  "name": "Campanha 1",  
  "contactName": "João",  
  "contactPhone": "5511999999999",  
  "contactSegment": 1,  
  "dateTime": "2024-01-01T10:00:00.000Z",  
  "lineReceptor": 1,  
  "response": false,  
  "speed": "medium",  
  "retryCount": 0,  
  "createdAt": "2024-01-01T00:00:00.000Z"  
}
```

GET `/campaigns/stats/:name`

Obtém estatísticas de uma campanha.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
{  
  "campaignName": "Campanha 1",  
  "totalContacts": 150,  
  "sent": 120,  
  "responses": 45,  
  "pending": 30,  
}
```

```
"responseRate": 37.5
}
```

DELETE /campaigns/:id

Remove uma campanha.

Autenticação: Requer JWT

Roles: admin , supervisor

Response Success (200):

```
{
  "message": "Campanha removida com sucesso"
}
```

3.7 Conversations (Conversas)

POST /conversations

Cria uma nova mensagem de conversa.

Autenticação: Requer JWT

Roles: admin , supervisor , operator

Request Body:

```
{
  "contactName": "Nome do Contato",
  "contactPhone": "5511999999999",
  "segment": 1,
  "userName": "Operador 1",
  "userLine": 1,
  "message": "Mensagem de texto",
  "sender": "operator",
  "tabulation": null,
  "messageType": "text",
  "mediaUrl": null
}
```

Campo	Tipo	Obrigatório	Descrição
contactName	string	✓	Nome do contato
contactPhone	string	✓	Telefone do contato
segment	number	✗	ID do segmento
userName	string	✗	Nome do operador
userLine	number	✗	ID da linha do operador
message	string	✓	Conteúdo da mensagem
sender	enum	✓	Remetente: <code>operator</code> , <code>contact</code>
tabulation	number	✗	ID da tabulação
messageType	string	✗	Tipo: <code>text</code> , <code>image</code> , <code>video</code> , <code>audio</code> , <code>document</code>
mediaUrl	string	✗	URL da mídia (se aplicável)

Response Success (201):

```
{
  "id": 1,
  "contactName": "Nome do Contato",
  "contactPhone": "5511999999999",
  "segment": 1,
  "userName": "Operador 1",
  "userLine": 1,
  "message": "Mensagem de texto",
  "sender": "operator",
  "datetime": "2024-01-01T10:00:00.000Z",
  "tabulation": null,
  "messageType": "text",
  "mediaUrl": null,
  "createdAt": "2024-01-01T10:00:00.000Z"
}
```

GET `/conversations`

Lista todas as conversas.

Autenticação: Requer JWT

Roles: `admin` , `supervisor` , `operator`

Query Parameters:

Parâmetro	Tipo	Descrição
segment	number	Filtrar por segmento
userLine	number	Filtrar por linha
contactPhone	string	Filtrar por telefone do contato

Response Success (200):

```
[
  {
    "id": 1,
    "contactName": "Nome do Contato",
    "contactPhone": "5511999999999",
    "segment": 1,
    "userName": "Operador 1",
    "userLine": 1,
    "message": "Mensagem de texto",
    "sender": "operator",
    "datetime": "2024-01-01T10:00:00.000Z",
    "tabulation": null,
    "messageType": "text",
    "mediaUrl": null
  }
]
```

GET `/conversations/active`

Lista conversas ativas do operador logado.

Autenticação: Requer JWT

Roles: `operator`

Response Success (200):

```
[
  {
```

```
[
  {
    "contactPhone": "5511999999999",
    "contactName": "Nome do Contato",
    "lastMessage": "Última mensagem",
    "lastMessageTime": "2024-01-01T10:00:00.000Z",
    "unreadCount": 3
  }
]
```

GET /conversations/segment/:segment

Lista conversas por segmento.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`

Query Parameters:

Parâmetro	Tipo	Descrição
tabulated	string	Filtrar: <code>true</code> = tabuladas, <code>false</code> = não tabuladas

Response Success (200):

```
[
  {
    "contactPhone": "5511999999999",
    "contactName": "Nome do Contato",
    "messages": []
  }
]
```

GET /conversations/contact/:phone

Lista conversas de um contato específico.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`, `operator`

Query Parameters:

Parâmetro	Tipo	Descrição
tabulated	string	Filtrar: <code>true</code> = tabuladas, <code>false</code> = não tabuladas

Response Success (200):

```
[
  {
    "id": 1,
    "contactName": "Nome do Contato",
    "contactPhone": "5511999999999",
    "message": "Mensagem",
    "sender": "contact",
    "datetime": "2024-01-01T10:00:00.000Z",
    "messageType": "text"
  }
]
```

GET `/conversations/:id`

Busca uma conversa pelo ID.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`, `operator`

Response Success (200):

```
{
  "id": 1,
  "contactName": "Nome do Contato",
  "contactPhone": "5511999999999",
  "segment": 1,
  "userName": "Operador 1",
  "userLine": 1,
  "message": "Mensagem de texto",
  "sender": "operator",
  "datetime": "2024-01-01T10:00:00.000Z",
  "tabulation": null,
  "messageType": "text",
  "mediaUrl": null
}
```

PATCH /conversations/:id

Atualiza uma conversa.

Autenticação: Requer JWT

Roles: admin , supervisor , operator

Request Body:

```
{
  "tabulation": 1
}
```

Response Success (200):

```
{
  "id": 1,
  "contactName": "Nome do Contato",
  "contactPhone": "5511999999999",
  "message": "Mensagem",
  "tabulation": 1
}
```

POST /conversations/tabulate/:phone


Tabula todas as conversas de um contato.

Autenticação: Requer JWT

Roles: operator

Request Body:

```
{
  "tabulationId": 1
}
```

Campo	Tipo	Obrigatório	Descrição
tabulationId	number		ID da tabulação

Response Success (200):

```
{
  "message": "Conversas tabuladas com sucesso",
  "count": 5
}
```

DELETE `/conversations/:id`

Remove uma conversa.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`

Response Success (200):

```
{
  "message": "Conversa removida com sucesso"
}
```

3.8 Tabulations (Tabulações)

POST `/tabulations`

Cria uma nova tabulação.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`

Request Body:

```
{
  "name": "Nome da Tabulação",
  "isCPC": true
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	✓	Nome da tabulação
isCPC	boolean	✗	Se é CPC (Contato com a Pessoa Certa)

Response Success (201):

```
{
  "id": 1,
  "name": "Nome da Tabulação",
  "isCPC": true,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET /tabulations

Lista todas as tabulações.

Autenticação: Requer JWT

Roles: `admin` , `supervisor` , `operator`

Query Parameters:

Parâmetro	Tipo	Descrição
search	string	Buscar por nome

Response Success (200):

```
[
  {
    "id": 1,
    "name": "CPC - Sucesso",
    "isCPC": true,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  },
  {
    "id": 2,
    "name": "Não atendeu",
    "isCPC": false,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET /tabulations/:id

Busca uma tabulação pelo ID.

Autenticação: Requer JWT

Roles: `admin` , `supervisor` , `operator`

Response Success (200):

```
{
  "id": 1,
  "name": "CPC - Sucesso",
  "isCPC": true,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

PATCH /tabulations/:id

Atualiza uma tabulação.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Request Body:

```
{
  "name": "Novo Nome",
  "isCPC": false
}
```

Response Success (200):

```
{
  "id": 1,
  "name": "Novo Nome",
  "isCPC": false,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T12:00:00.000Z"
}
```

DELETE /tabulations/:id

Remove uma tabulação.

Autenticação: Requer JWT

Roles: admin , supervisor

Response Success (200):

```
{
  "message": "Tabulação removida com sucesso"
}
```

3.9 Blocklist (Lista de Bloqueio)

POST /blocklist

Adiciona um contato à blocklist.

Autenticação: Requer JWT

Roles: admin , supervisor

Request Body:

```
{
  "name": "Nome do Contato",
  "phone": "5511999999999",
  "cpf": "12345678901"
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	×	Nome do contato
phone	string	×	Telefone do contato
cpf	string	×	CPF do contato

Nota: Pelo menos um dos campos (phone ou cpf) deve ser informado.

Response Success (201):

```
{
  "id": 1,
  "name": "Nome do Contato",
  "phone": "5511999999999",
  "cpf": "12345678901",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET /blocklist

Lista todos os contatos bloqueados.

Autenticação: Requer JWT

Roles: Todos

Query Parameters:

Parâmetro	Tipo	Descrição
search	string	Buscar por nome, telefone ou CPF

Response Success (200):

```
[
  {
    "id": 1,
    "name": "Nome do Contato",
    "phone": "5511999999999",
    "cpf": "12345678901",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET /blocklist/check

Verifica se um contato está bloqueado.

Autenticação: Requer JWT

Roles: Todos

Query Parameters:

Parâmetro	Tipo	Descrição
phone	string	Telefone a verificar
cpf	string	CPF a verificar

Response Success (200):

```
{
  "blocked": true
}
```

GET `/blocklist/:id`

Busca um registro da blocklist pelo ID.

Autenticação: Requer JWT

Roles: Todos

Response Success (200):

```
{
  "id": 1,
  "name": "Nome do Contato",
  "phone": "5511999999999",
  "cpf": "12345678901",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

PATCH `/blocklist/:id`

Atualiza um registro da blocklist.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Request Body:

```
{
  "name": "Novo Nome",
  "phone": "5511888888888"
}
```

Response Success (200):

```
{
  "id": 1,
  "name": "Novo Nome",
  "phone": "5511888888888",
  "cpf": "12345678901",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T12:00:00.000Z"
}
```

DELETE /blocklist/:id

Remove um contato da blocklist.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`

Response Success (200):

```
{
  "message": "Contato removido da blocklist"
}
```

3.10 Evolution (Configurações Evolution API)

POST /evolution

Cria uma nova configuração de Evolution API.

Autenticação: Requer JWT

Roles: `admin`

Request Body:

```
{
  "evolutionName": "Evolution01",
  "evolutionUrl": "https://api.evolution.example.com",
  "evolutionKey": "sua-api-key"
}
```

Campo	Tipo	Obrigatório	Descrição
evolutionName	string	✔	Nome único da instância
evolutionUrl	string	✔	URL da API Evolution (formato URL válido)
evolutionKey	string	✔	Chave de API da Evolution

Response Success (201):

```
{
  "id": 1,
  "evolutionName": "Evolution01",
  "evolutionUrl": "https://api.evolution.example.com",
  "evolutionKey": "sua-api-key",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET /evolution

Lista todas as configurações Evolution.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
[
  {
    "id": 1,
    "evolutionName": "Evolution01",
    "evolutionUrl": "https://api.evolution.example.com",
    "evolutionKey": "sua-api-key",
    "createdAt": "2024-01-01T00:00:00.000Z",
  }
]
```

```
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET /*evolution*/*id*

Busca uma configuração Evolution pelo ID.

Autenticação: Requer JWT

Roles: *admin*

Response Success (200):

```
{
  "id": 1,
  "evolutionName": "Evolution01",
  "evolutionUrl": "https://api.evolution.example.com",
  "evolutionKey": "sua-api-key",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

PATCH /*evolution*/*id*

Atualiza uma configuração Evolution.

Autenticação: Requer JWT

Roles: *admin*

Request Body:

```
{
  "evolutionUrl": "https://new-api.evolution.example.com",
  "evolutionKey": "nova-api-key"
}
```

Response Success (200):

```
{
  "id": 1,
  "evolutionName": "Evolution01",
  "evolutionUrl": "https://new-api.evolution.example.com",
}
```



```
"evolutionKey": "nova-api-key",
"createdAt": "2024-01-01T00:00:00.000Z",
"updatedAt": "2024-01-01T12:00:00.000Z"
}
```

DELETE `/evolution/:id`

Remove uma configuração Evolution.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
{
  "message": "Configuração Evolution removida com sucesso"
}
```

GET `/evolution/test/:name`

Testa conexão com uma Evolution API.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
{
  "status": "connected",
  "message": "Conexão com Evolution API estabelecida com sucesso"
}
```

3.11 Tags

POST `/tags`

Cria uma nova tag.

Autenticação: Requer JWT

Roles: `admin`

Request Body:

```
{
  "name": "Nome da Tag",
  "description": "Descrição da tag",
  "segment": 1
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	✓	Nome único da tag
description	string	✗	Descrição da tag
segment	number	✗	ID do segmento associado

Response Success (201):

```
{
  "id": 1,
  "name": "Nome da Tag",
  "description": "Descrição da tag",
  "segment": 1,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET /tags

Lista todas as tags.

Autenticação: Requer JWT

Roles: admin

Query Parameters:

Parâmetro	Tipo	Descrição
search	string	Buscar por nome
segment	number	Filtrar por segmento

Response Success (200):

```
[
  {
    "id": 1,
    "name": "Tag 1",
    "description": "Descrição da tag 1",
    "segment": 1,
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET /tags/:id

Busca uma tag pelo ID.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
{
  "id": 1,
  "name": "Tag 1",
  "description": "Descrição da tag 1",
  "segment": 1,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

PATCH /tags/:id

Atualiza uma tag.

Autenticação: Requer JWT

Roles: `admin`

Request Body:

```
{
  "name": "Novo Nome",
}
```

```
"description": "Nova descrição"
}
```

Response Success (200):

```
{
  "id": 1,
  "name": "Novo Nome",
  "description": "Nova descrição",
  "segment": 1,
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T12:00:00.000Z"
}
```

DELETE /tags/:id

Remove uma tag.

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
{
  "message": "Tag removida com sucesso"
}
```

3.12 Templates (WhatsApp Cloud API)

Esta seção descreve os endpoints para gerenciamento de templates do WhatsApp Cloud API.

POST /templates

Cria um novo template.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Request Body:

```
{
  "name": "template_boas_vindas",
  "language": "pt_BR",
  "category": "MARKETING",
  "lineId": 1,
  "namespace": "namespace_opcional",
  "headerType": "TEXT",
  "headerContent": "Olá 1!",
  "bodyText": "Bem-vindo à nossa plataforma, 1! Seu código é 2.",
  "footerText": "Responda SAIR para não receber mais mensagens",
  "buttons": [
    { "type": "QUICK_REPLY", "text": "Confirmar" },
    { "type": "URL", "text": "Acessar site", "url": "https://exer"
  ],
  "variables": ["nome", "codigo"]
}
```

Campo	Tipo	Obrigatório	Descrição
name	string	✓	Nome único do template
language	string	✗	Idioma (padrão: pt_BR)
category	string	✗	Categoria: MARKETING, UTILITY, AUTHENTICATION
lineId	number	✓	ID da linha oficial vinculada
namespace	string	✗	Namespace do template na Meta
headerType	string	✗	Tipo: TEXT, IMAGE, VIDEO, DOCUMENT
headerContent	string	✗	Conteúdo do header
bodyText	string	✓	Texto do corpo (suporta variáveis)
footerText	string	✗	Texto do rodapé
buttons	array	✗	Botões do template
variables	array	✗	Lista de variáveis esperadas

Response Success (201):

```
{
  "id": 1,
  "name": "template_boas_vindas",
  "language": "pt_BR",
  "category": "MARKETING",
  "lineId": 1,
  "status": "PENDING",
  "bodyText": "Bem-vindo à nossa plataforma, 1! Seu código é 2.",
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET /templates

Lista todos os templates.

Autenticação: Requer JWT

Roles: `admin` , `supervisor` , `operator`

Query Parameters:

Parâmetro	Tipo	Descrição
search	string	Buscar por nome ou texto
lineId	number	Filtrar por linha
status	string	Filtrar por status

Response Success (200):

```
[
  {
    "id": 1,
    "name": "template_boas_vindas",
    "language": "pt_BR",
    "category": "MARKETING",
    "lineId": 1,
    "status": "APPROVED",
    "bodyText": "Bem-vindo à nossa plataforma, 1!",
    "buttons": [
      { "type": "QUICK_REPLY", "text": "Confirmar" }
    ]
  }
]
```

```
    ],
    "variables": ["nome"],
    "createdAt": "2024-01-01T00:00:00.000Z"
  }
]
```

GET /templates/:id

Busca um template pelo ID.

Autenticação: Requer JWT

Roles: admin , supervisor , operator

Response Success (200):

```
{
  "id": 1,
  "name": "template_boas_vindas",
  "language": "pt_BR",
  "category": "MARKETING",
  "lineId": 1,
  "namespace": "business_namespace",
  "status": "APPROVED",
  "headerType": "TEXT",
  "headerContent": "Olá 1!",
  "bodyText": "Bem-vindo à nossa plataforma, 1! Seu código é 2.",
  "footerText": "Responda SAIR para não receber mais mensagens",
  "buttons": [
    { "type": "QUICK_REPLY", "text": "Confirmar" }
  ],
  "variables": ["nome", "codigo"],
  "createdAt": "2024-01-01T00:00:00.000Z",
  "updatedAt": "2024-01-01T00:00:00.000Z"
}
```

GET /templates/line/:lineId

Lista templates de uma linha específica.

Autenticação: Requer JWT

Roles: admin , supervisor , operator

Response Success (200):

```
[
  {
    "id": 1,
    "name": "template_boas_vindas",
    "status": "APPROVED",
    "bodyText": "Bem-vindo!"
  }
]
```

PATCH /templates/:id

Atualiza um template.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Request Body:

```
{
  "bodyText": "Novo texto do template 1",
  "status": "APPROVED"
}
```

Response Success (200):

```
{
  "id": 1,
  "name": "template_boas_vindas",
  "bodyText": "Novo texto do template 1",
  "status": "APPROVED",
  "updatedAt": "2024-01-01T12:00:00.000Z"
}
```

DELETE /templates/:id

Remove um template.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
{
  "message": "Template removido com sucesso"
}
```

POST `/templates/:id/sync`

Sincroniza template com WhatsApp Cloud API (Meta).

Autenticação: Requer JWT

Roles: `admin`

Response Success (200):

```
{
  "success": true,
  "message": "Template enviado para aprovação",
  "templateId": "meta_template_id_123"
}
```

Response Error (400):

```
{
  "statusCode": 400,
  "message": "Erro ao sincronizar template: Invalid template form"
}
```

POST `/templates/send`

Envia um template para um contato (1x1).

Autenticação: Requer JWT

Roles: `admin` , `supervisor` , `operator`

Request Body:

```
{
  "templateId": 1,
  "phone": "5511999999999",
  "contactName": "João Silva",
}
```

```
"variables": [
  { "key": "nome", "value": "João" },
  { "key": "codigo", "value": "ABC123" }
],
"lineId": 1
}
```

Campo	Tipo	Obrigatório	Descrição
templateId	number	✓	ID do template
phone	string	✓	Telefone do destinatário
contactName	string	✗	Nome do contato
variables	array	✗	Variáveis para substituição
lineId	number	✗	ID da linha (usa a do template se não informado)

Response Success (200):

```
{
  "success": true,
  "messageId": "wamid.abc123...",
  "templateMessageId": 1,
  "error": null
}
```

POST /templates/send/massive

Envia template para múltiplos contatos (disparo massivo).

Autenticação: Requer JWT

Roles: admin , supervisor

Request Body:

```
{
  "templateId": 1,
  "recipients": [
    {
      "phone": "5511999999999",
```

```
    "contactName": "João",
    "variables": [
      { "key": "nome", "value": "João" }
    ]
  },
  {
    "phone": "55118888888888",
    "contactName": "Maria",
    "variables": [
      { "key": "nome", "value": "Maria" }
    ]
  }
],
"lineId": 1
}
```

Campo	Tipo	Obrigatório	Descrição
templateId	number	✓	ID do template
recipients	array	✓	Lista de destinatários
recipients[].phone	string	✓	Telefone do destinatário
recipients[].contactName	string	✗	Nome do contato
recipients[].variables	array	✗	Variáveis específicas do contato
lineId	number	✗	ID da linha

Response Success (200):

```
{
  "status": "partial",
  "total": 2,
  "successful": 1,
  "failed": 1,
  "results": [
    {
      "phone": "55119999999999",
      "success": true,
      "messageId": "wamid.abc123..."
    },
  ],
}
```

```
{
  "phone": "55118888888888",
  "success": false,
  "error": "Número na blocklist"
}
]
```

GET /templates/:id/history

Obtém histórico de envios de um template.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Query Parameters:

Parâmetro	Tipo	Descrição
startDate	string	Data inicial (ISO 8601)
endDate	string	Data final (ISO 8601)
status	string	Filtrar por status

Response Success (200):

```
[
  {
    "id": 1,
    "templateId": 1,
    "contactPhone": "55119999999999",
    "contactName": "João",
    "lineId": 1,
    "status": "DELIVERED",
    "messageId": "wamid.abc123...",
    "variables": "[{\"key\":\"nome\",\"value\":\"João\"}]",
    "createdAt": "2024-01-01T10:00:00.000Z"
  }
]
```

GET `/templates/:id/stats`

Obtém estatísticas de um template.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
{
  "total": 1000,
  "sent": 950,
  "delivered": 900,
  "read": 750,
  "failed": 50,
  "deliveryRate": "94.74",
  "readRate": "83.33"
}
```

3.13 Reports (Relatórios)

Todos os endpoints de relatórios aceitam os mesmos parâmetros de filtro:

Query Parameters (comum a todos os relatórios):

Parâmetro	Tipo	Descrição
startDate	string	Data inicial (formato ISO 8601)
endDate	string	Data final (formato ISO 8601)
segment	number	Filtrar por segmento

GET `/reports/op-sintetico`

Relatório sintético de operadores.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
{
  "operators": [
    {
      "operatorId": 1,
      "operatorName": "Operador 1",
      "totalConversations": 150,
      "totalMessages": 450,
      "avgResponseTime": 30
    }
  ]
}
```

GET `/reports/kpi`

Relatório de KPIs (Key Performance Indicators).

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
{
  "totalMessages": 1500,
  "totalConversations": 500,
  "responseRate": 85.5,
  "avgResponseTime": 25,
  "cpcRate": 45.2
}
```

GET `/reports/hsm`

Relatório de mensagens HSM (High Structured Messages).

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Response Success (200):

```
{
  "totalSent": 1000,
  "delivered": 950,
```

```
"read": 800,  
"replied": 300,  
"deliveryRate": 95,  
"readRate": 84.2,  
"replyRate": 31.5  
}
```

GET /reports/line-status

Relatório de status das linhas.

Autenticação: Requer JWT

Roles: admin , supervisor

Response Success (200):

```
{  
  "totalLines": 20,  
  "activeLines": 18,  
  "bannedLines": 2,  
  "lines": [  
    {  
      "id": 1,  
      "phone": "5511999999999",  
      "status": "active",  
      "messagesSent": 150,  
      "lastActivity": "2024-01-01T10:00:00.000Z"  
    }  
  ]  
}
```

GET /reports/envios

Relatório de envios de mensagens.

Autenticação: Requer JWT

Roles: admin , supervisor

GET /reports/indicadores

Relatório de indicadores gerais.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

GET `/reports/tempos`

Relatório de tempos de atendimento.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

GET `/reports/templates`

Relatório de templates utilizados.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

GET `/reports/completo-csv`

Relatório completo para exportação CSV.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

GET `/reports/equipe`

Relatório de performance da equipe.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

GET `/reports/dados-transacionados`

Relatório de dados transacionados.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

GET /reports/detalhado-conversas

Relatório detalhado de conversas.

Autenticação: Requer JWT

Roles: admin , supervisor

GET /reports/linhas

Relatório de linhas.

Autenticação: Requer JWT

Roles: admin , supervisor

GET /reports/resumo-atendimentos

Relatório resumido de atendimentos.

Autenticação: Requer JWT

Roles: admin , supervisor

GET /reports/hiper-personalizado

Relatório hiper personalizado.

Autenticação: Requer JWT

Roles: admin , supervisor

GET /reports/consolidado

Retorna todos os relatórios consolidados.

Autenticação: Requer JWT

Roles: admin , supervisor

Response Success (200):

```
{
  "periodo": {
    "inicio": "2024-01-01",
```

```
    "fim": "2024-01-31"
  },
  "segmento": 1,
  "relatorios": {
    "opSintetico": { ... },
    "kpi": { ... },
    "hsm": { ... },
    "lineStatus": { ... },
    "envios": { ... },
    "indicadores": { ... },
    "tempos": { ... },
    "templates": { ... },
    "completoCsv": { ... },
    "equipe": { ... },
    "dadosTransacionados": { ... },
    "detalhadoConversas": { ... },
    "linhas": { ... },
    "resumoAtendimentos": { ... },
    "hiperPersonalizado": { ... }
  }
}
```

3.14 Media (Mídias)

POST `/media/upload`


Faz upload de um arquivo de mídia.

Autenticação: Requer JWT

Roles: Todos

Content-Type: `multipart/form-data`

Request Body:

Campo	Tipo	Obrigatório	Descrição
file	file		Arquivo de mídia

Tipos de arquivo permitidos:

- Imagens: `jpeg` , `png` , `gif` , `webp`

- Vídeos: `mp4` , `mpeg`
- Áudios: `mpeg` , `ogg` , `mp4`
- Documentos: `pdf` , `doc` , `docx` , `xls` , `xlsx`

Response Success (200):

```
{
  "success": true,
  "mediaUrl": "/media/abc123.jpg",
  "fileName": "abc123.jpg",
  "originalName": "foto.jpg",
  "mimeType": "image/jpeg",
  "size": 102400
}
```

GET `/media/:filename`

Obtém/visualiza um arquivo de mídia.

Autenticação: Requer JWT

Roles: Todos

Response: Arquivo binário com Content-Type apropriado

3.15 API Logs

GET `/api-logs`

Lista os logs de chamadas da API.

Autenticação: Requer JWT

Roles: `admin` , `supervisor`

Query Parameters:

Parâmetro	Tipo	Descrição
startDate	string	Data inicial (formato ISO 8601)
endDate	string	Data final (formato ISO 8601)
endpoint	string	Filtrar por endpoint
method	string	Filtrar por método HTTP
statusCode	number	Filtrar por código de status

Response Success (200):

```
[
  {
    "id": 1,
    "endpoint": "/api/messages/massivocpc",
    "method": "POST",
    "requestPayload": "{...}",
    "responsePayload": "{...}",
    "statusCode": 200,
    "ipAddress": "192.168.1.1",
    "userAgent": "Mozilla/5.0...",
    "createdAt": "2024-01-01T10:00:00.000Z"
  }
]
```

GET `/api-logs/:id`

Busca um log específico pelo ID.

Autenticação: Requer JWT

Roles: `admin`, `supervisor`

Response Success (200):

```
{
  "id": 1,
  "endpoint": "/api/messages/massivocpc",
  "method": "POST",
  "requestPayload": "{...}",
  "responsePayload": "{...}",
```

```
"statusCode": 200,  
"ipAddress": "192.168.1.1",  
"userAgent": "Mozilla/5.0...",  
"createdAt": "2024-01-01T10:00:00.000Z"  
}
```

3.16 API Messages (Mensagens Externas)

Esta API é destinada para integrações externas, utilizando autenticação via API Key.

POST `/api/messages/massivocpc`

Envia mensagens em massa (disparo CPC). Suporta envio de texto simples ou templates oficiais.

Autenticação: API Key (header `x-api-key`)

Roles: Externo

Request Body:

```
{  
  "campaign": "Nome da Campanha",  
  "idAccount": "ID_CONTA_OPCIONAL",  
  "tag": "TAG_SEGMENTO",  
  "useOfficialTemplate": true,  
  "defaultTemplateId": 1,  
  "messages": [  
    {  
      "phone": "5511999999999",  
      "idMessage": 12345,  
      "clientId": "CLIENTE_001",  
      "contract": "CONTRATO_123",  
      "closeTicket": false,  
      "specialistCode": "operador01",  
      "mainTemplate": "Texto da mensagem (usado se useOfficialTer",  
      "retryTemplate": "template_retry",  
      "lastTemplate": "template_final",  
      "useOfficialTemplate": true,  
      "templateId": 1,  
      "templateVariables": [  
        { "key": "nome", "value": "João" },  
        { "key": "valor", "value": "R$ 100,00" }  
      ]  
    }  
  ]  
}
```

```
}  
]  
}
```

Campo	Tipo	Obrigatório	Descrição
campaign	string	✓	Nome da campanha
idAccount	string	✗	ID da conta
tag	string	✓	Tag do segmento
useOfficialTemplate	boolean	✗	Usar template oficial (global)
defaultTemplateId	number	✗	ID do template padrão (global)
messages	array	✓	Lista de mensagens

Estrutura de cada mensagem:

Campo	Tipo	Obrigatório	Descrição
phone	string	✓	Telefone do destinatário
idMessage	number	✗	ID da mensagem (referência externa)
clientId	string	✗	ID do cliente
contract	string	✗	Número do contrato
closeTicket	boolean	✓	Se deve fechar o ticket após envio
specialistCode	string	✓	Código do especialista (email antes do @)
mainTemplate	string	✓	Template/mensagem principal
retryTemplate	string	✗	Template para retry
lastTemplate	string	✗	Template final
useOfficialTemplate	boolean	✗	Usar template oficial (por mensagem)
templateId	number	✗	ID do template oficial
templateVariables	array	✗	Variáveis do template

Response Success (200):

```
{
  "status": "success",
  "message": "Mensagens enviadas com sucesso",
  "processed": 10,
  "errors": []
}
```

Response Partial (207):

```
{
  "status": "partial",
  "message": "8 mensagens processadas, 2 com erro",
  "processed": 8,
  "errors": [
```

```
{ "phone": "5511999999999", "reason": "Número na blocklist" }
{ "phone": "5511888888888", "reason": "Bloqueado por regra CF"
}
```

POST /api/messages/template

Envia um template para um contato específico (1x1) via API externa.

Autenticação: API Key (header `x-api-key`)

Roles: Externo

Request Body:

```
{
  "phone": "5511999999999",
  "templateId": 1,
  "contactName": "João Silva",
  "specialistCode": "operador01",
  "variables": [
    { "key": "nome", "value": "João" },
    { "key": "codigo", "value": "ABC123" }
  ],
  "tag": "COBRANCA"
}
```

Campo	Tipo	Obrigatório	Descrição
phone	string	✓	Telefone do destinatário
templateId	number	✓	ID do template
contactName	string	✗	Nome do contato
specialistCode	string	✓	Código do especialista
variables	array	✗	Variáveis do template
tag	string	✗	Tag para obter segmento

Response Success (200):


```
{
  "success": true,
  "message": "Template enviado com sucesso",
  "templateMessageId": 1,
  "templateName": "template_boas_vindas"
}
```

Response Error (400):

```
{
  "success": false,
  "message": "Número está na lista de bloqueio"
}
```

3.17 Webhooks

POST `/webhooks/evolution`

Recebe webhooks da Evolution API.

Autenticação: Não requer (webhook público)

Roles: Público

Request Body: Payload variável conforme evento da Evolution API

Tipos de eventos suportados:

- `messages.upsert` - Nova mensagem recebida
- `messages.update` - Atualização de mensagem
- `connection.update` - Atualização de conexão

Response Success (200):

```
{
  "received": true
}
```

3.18 Health Check

GET /health

Verifica o status de saúde da aplicação.

Autenticação: Não requer

Roles: Público

Response Success (200):

```
{
  "status": "ok",
  "timestamp": "2024-01-01T10:00:00.000Z",
  "uptime": 86400,
  "database": "connected"
}
```

Response Error (500):

```
{
  "status": "error",
  "timestamp": "2024-01-01T10:00:00.000Z",
  "uptime": 86400,
  "database": "disconnected",
  "error": "Connection refused"
}
```

4. Enums e Tipos

Role (Papel do Usuário)

Valor	Descrição
admin	Administrador com acesso total
supervisor	Supervisor de equipe
operator	Operador de atendimento

Status (Status do Usuário)

Valor	Descrição
Online	Usuário conectado
Offline	Usuário desconectado

LineStatus (Status da Linha)

Valor	Descrição
active	Linha ativa e funcionando
ban	Linha banida/bloqueada

Sender (Remetente da Mensagem)

Valor	Descrição
operator	Mensagem enviada pelo operador
contact	Mensagem recebida do contato

Speed (Velocidade de Envio)

Valor	Descrição
fast	Envio rápido
medium	Envio médio
slow	Envio lento

5. Modelos de Dados

User (Usuário)

```
{
  id: number;           // ID único
```

```
name: string;           // Nome completo
email: string;          // Email (único)
password: string;       // Senha (hash)
role: Role;             // Papel
segment?: number;       // ID do segmento
line?: number;          // ID da linha vinculada
status: Status;         // Status atual
createdAt: DateTime;    // Data de criação
updatedAt: DateTime;    // Data de atualização
}
```

Segment (Segmento)

```
{
  id: number;           // ID único
  name: string;         // Nome (único)
  createdAt: DateTime;  // Data de criação
  updatedAt: DateTime;  // Data de atualização
}
```

Contact (Contato)

```
{
  id: number;           // ID único
  name: string;         // Nome do contato
  phone: string;        // Telefone
  segment?: number;     // ID do segmento
  cpf?: string;         // CPF
  contract?: string;    // Número do contrato
  createdAt: DateTime;  // Data de criação
  updatedAt: DateTime;  // Data de atualização
}
```

LinesStock (Linha)

```
{
  id: number;           // ID único
  phone: string;        // Número do telefone (único)
  lineStatus: LineStatus; // Status da linha
}
```

```
segment?: number; // ID do segmento
linkedTo?: number; // ID do usuário vinculado
evolutionName: string; // Nome da instância Evolution
oficial: boolean; // Se é linha oficial
token?: string; // Token WhatsApp Business
businessID?: string; // ID Business
numberId?: string; // ID do número
createdAt: DateTime; // Data de criação
updatedAt: DateTime; // Data de atualização
}
```

Campaign (Campanha)

```
{
  id: number; // ID único
  name: string; // Nome da campanha
  contactName: string; // Nome do contato
  contactPhone: string; // Telefone do contato
  contactSegment?: number; // Segmento do contato
  dateTime: DateTime; // Data/hora do envio
  lineReceptor?: number; // ID da linha receptora
  response: boolean; // Se houve resposta
  speed: Speed; // Velocidade de envio
  retryCount: number; // Contador de tentativas
  useTemplate: boolean; // Se usa template oficial
  templateId?: number; // ID do template
  templateVariables?: string; // JSON com variáveis do template
  createdAt: DateTime; // Data de criação
  updatedAt: DateTime; // Data de atualização
}
```

Conversation (Conversa)

```
{
  id: number; // ID único
  contactName: string; // Nome do contato
  contactPhone: string; // Telefone do contato
  segment?: number; // ID do segmento
  userName?: string; // Nome do operador
  userLine?: number; // ID da linha do operador
  message: string; // Conteúdo da mensagem
}
```

```
sender: Sender;           // Remetente
datetime: DateTime;       // Data/hora da mensagem
tabulation?: number;      // ID da tabulação
messageType: string;      // Tipo da mensagem
mediaUrl?: string;        // URL da mídia
createdAt: DateTime;      // Data de criação
updatedAt: DateTime;      // Data de atualização
}
```

Tabulation (Tabulação)

```
{
  id: number;              // ID único
  name: string;            // Nome da tabulação
  isCPC: boolean;          // Se é CPC
  createdAt: DateTime;     // Data de criação
  updatedAt: DateTime;     // Data de atualização
}
```

BlockList (Lista de Bloqueio)

```
{
  id: number;              // ID único
  name?: string;           // Nome do contato
  phone?: string;          // Telefone
  cpf?: string;            // CPF
  createdAt: DateTime;     // Data de criação
  updatedAt: DateTime;     // Data de atualização
}
```

Evolution

```
{
  id: number;              // ID único
  evolutionName: string;   // Nome da instância (único)
  evolutionUrl: string;    // URL da API
  evolutionKey: string;    // Chave de API
  createdAt: DateTime;     // Data de criação
}
```

```
    updatedAt: DateTime; // Data de atualização
}
```

Tag

```
{
  id: number;           // ID único
  name: string;          // Nome (único)
  description?: string; // Descrição
  segment?: number;      // ID do segmento
  createdAt: DateTime;   // Data de criação
  updatedAt: DateTime;   // Data de atualização
}
```

Template

```
{
  id: number;           // ID único
  name: string;          // Nome do template
  language: string;      // Idioma (ex: pt_BR)
  category: string;      // Categoria (MARKETING, UTILITY, AUTHENT
  lineId: number;        // ID da linha vinculada
  namespace?: string;    // Namespace na Meta
  status: string;        // Status (PENDING, SUBMITTED, APPROVED,
  headerType?: string;   // Tipo do header (TEXT, IMAGE, VIDEO, DC
  headerContent?: string; // Conteúdo do header
  bodyText: string;      // Texto do corpo
  footerText?: string;   // Texto do rodapé
  buttons?: string;       // JSON com botões
  variables?: string;     // JSON com variáveis
  createdAt: DateTime;    // Data de criação
  updatedAt: DateTime;    // Data de atualização
}
```

TemplateMessage

```
{
  id: number;           // ID único
  templateId: number;   // ID do template
}
```

```
contactPhone: string; // Telefone do destinatário
contactName?: string; // Nome do contato
lineId: number;        // ID da linha
status: string;        // Status (SENT, DELIVERED, READ, FAILED)
messageId?: string;    // ID da mensagem no WhatsApp
variables?: string;    // JSON com variáveis usadas
errorMessage?: string; // Mensagem de erro (se falhou)
campaignId?: number;   // ID da campanha (se enviado via campaih
createdAt: DateTime;   // Data de criação
updatedAt: DateTime;   // Data de atualização
}
```

ApiLog

```
{
  id: number;           // ID único
  endpoint: string;     // Endpoint chamado
  method: string;       // Método HTTP
  requestPayload: string; // Payload da requisição
  responsePayload: string; // Payload da resposta
  statusCode: number;   // Código HTTP
  ipAddress?: string;   // IP do cliente
  userAgent?: string;   // User-Agent
  createdAt: DateTime;  // Data de criação
}
```

6. Códigos de Erro

Códigos HTTP Padrão

Código	Significado	Descrição
200	OK	Requisição bem-sucedida
201	Created	Recurso criado com sucesso
400	Bad Request	Requisição inválida (dados faltando ou incorretos)
401	Unauthorized	Não autenticado (token inválido ou ausente)
403	Forbidden	Sem permissão para acessar o recurso
404	Not Found	Recurso não encontrado
409	Conflict	Conflito (ex: email já cadastrado)
422	Unprocessable Entity	Dados inválidos (validação falhou)
500	Internal Server Error	Erro interno do servidor

Formato de Erro Padrão

```
{
  "statusCode": 400,
  "message": "Descrição do erro",
  "error": "Bad Request"
}
```

Erros de Validação

```
{
  "statusCode": 400,
  "message": [
    "email must be an email",
    "password must be longer than or equal to 6 characters"
  ],
  "error": "Bad Request"
}
```

Contato e Suporte

Para dúvidas ou suporte técnico, entre em contato com a equipe de desenvolvimento.

© 2024 NewVend - Todos os direitos reservados