

# Documentação Completa - Nexi CRM Chat API v2.0

Esta documentação detalha todos os endpoints disponíveis na API do Nexi CRM Chat, incluindo métodos, parâmetros, exemplos de requisição e respostas.

## Informações Gerais

- **Base URL:** `https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy`
- **Formato de Resposta:** Todas as respostas são em JSON.

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true/false,
  "code": "CODIGO_RESPOSTA",
  "message": "Mensagem descritiva",
  "timestamp": "2025-10-17T10:30:00.000Z",
  "data": { /* dados específicos */ }
}
```

## Autenticação

Todas as requisições requerem o parâmetro `key`.

- **Chave padrão:** `1526`
- **Localização:** Query string (GET) ou body (POST)

## APIs GET

### 1. HELP - Documentação da API

- **Endpoint:** `GET /?action=help&key=1526`
- **Descrição:** Retorna informações sobre todos os endpoints disponíveis.

Parâmetro	Tipo	Obrigatório	Descrição	Padrão
key	string	✓	Chave de segurança	1526
action	string	✗	"help" ou vazio	help

### Exemplo de Requisição:

Bash

```
curl "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy?
action=help&key=1526"
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "HELP",
  "message": "API de Chat - Documentação",
  "timestamp": "2025-10-17T10:30:00.000Z",
  "data": {
    "version": "2.0",
    "endpoints": {
      "GET": [
        "help - Esta mensagem",
        "get_messages - Obter mensagens do chat",
        "get_chat - Obter detalhes do chat",
        "list_chats - Listar chats",
        "get_attachment - Obter anexo",
        "cleanup - Executar limpeza manual"
      ],
      "POST": [
        "create_chat - Criar chat",
        "add_message - Adicionar mensagem",
        "upload_attachment - Upload de anexo",
        "assign_attendant - Atribuir atendente",
        "update_status - Atualizar status",
        "add_observation - Adicionar observação"
      ]
    }
  }
}
```

```
}  
}
```

## 2. GET\_MESSAGES - Obter Mensagens do Chat

- **Endpoint:** GET `/?action=get_messages&key=1526&chat_id={id}&page=1&page_size=50`
- **Descrição:** Lista todas as mensagens de um chat específico com paginação.

Parâmetro	Tipo	Obrigatório	Descrição	Padrão
key	string	✓	Chave de segurança	1526
action	string	✓	"get_messages"	-
chat_id	string	✓	ID do chat	-
page	integer	✗	Número da página	1
page_size	integer	✗	Itens por página	50

### Exemplo de Requisição:

Bash

```
curl "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy?  
action=get_messages&key=1526&chat_id=chat_abc123&page=1&page_size=20"
```

### Resposta de Sucesso (200):

JSON

```
{  
  "appName": "Nexi CRM",  
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",  
  "success": true,  
  "code": "MESSAGES_RETRIEVED",  
  "message": "Mensagens obtidas",  
  "timestamp": "2025-10-17T10:30:00.000Z",  
  "data": {  
    "messages": [  
      {  
        "message_id": "msg_xyz789",  
        "chat_id": "chat_abc123",  
        "sender_user_id": "user_123",  
      }  
    ]  
  }  
}
```

```
    "sender_name": "João Silva",
    "message": "Olá, preciso de ajuda!",
    "has_attachment": false,
    "created_at": "2025-10-17T10:25:00.000Z"
  },
  ],
  "pagination": {
    "page": 1,
    "page_size": 20,
    "total": 45,
    "total_pages": 3
  }
}
```

### 3. GET\_CHAT - Obter Detalhes do Chat

- **Endpoint:** GET /?action=get\_chat&key=1526&chat\_id={id}
- **Descrição:** Retorna informações completas de um chat específico.

Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"get_chat"
chat_id	string	✓	ID do chat

#### Exemplo de Requisição:

Bash

```
curl "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy?
action=get_chat&key=1526&chat_id=chat_abc123"
```

#### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "CHAT_RETRIEVED",
}
```

```
"message": "Chat obtido",
"timestamp": "2025-10-17T10:30:00.000Z",
"data": {
  "chat_id": "chat_abc123",
  "title": "Suporte Técnico - Sistema Lento",
  "creator_user_id": "user_123",
  "creator_email": "joao@empresa.com",
  "attendant_user_id": "attendant_456",
  "attendant_email": "maria@empresa.com",
  "observations": "[2025-10-17T09:00:00.000Z] Cliente reportou lentidão",
  "status": "em_atendimento",
  "send_email_summary": true,
  "created_at": "2025-10-17T09:00:00.000Z",
  "updated_at": "2025-10-17T10:30:00.000Z"
}
```

## 4. LIST\_CHATS - Listar Chats

- **Endpoint:** GET /?action=list\_chats&key=1526&status={status}&page=1
- **Descrição:** Lista todos os chats com filtros opcionais e paginação.

Parâmetro	Tipo	Obrigatório	Descrição	Padrão
key	string	✓	Chave de segurança	1526
action	string	✓	"list_chats"	-
status	string	✗	Filtrar por status	-
creator_user_id	string	✗	Filtrar por criador	-
attendant_user_id	string	✗	Filtrar por atendente	-
page	integer	✗	Número da página	1
page_size	integer	✗	Itens por página	50

**Status Válidos:** aberto , em\_atendimento , fechado , cancelado

### Exemplo de Requisição:

Bash

```
# Todos os chats abertos
curl "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy?"
```

```
action=list_chats&key=1526&status=aberto&page=1&page_size=10"
```

*# Chats de um usuário específico*

```
curl "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy?
action=list_chats&key=1526&creator_user_id=user_123"
```

## Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "CHATS_RETRIEVED",
  "message": "Chats obtidos",
  "timestamp": "2025-10-17T10:30:00.000Z",
  "data": {
    "chats": [
      {
        "chat_id": "chat_abc123",
        "title": "Suporte Técnico - Sistema Lento",
        "creator_user_id": "user_123",
        "creator_email": "joao@empresa.com",
        "attendant_user_id": "attendant_456",
        "attendant_email": "maria@empresa.com",
        "observations": "Cliente reportou lentidão",
        "status": "aberto",
        "send_email_summary": true,
        "created_at": "2025-10-17T09:00:00.000Z",
        "updated_at": "2025-10-17T10:30:00.000Z"
      }
    ],
    "pagination": {
      "page": 1,
      "page_size": 10,
      "total": 25,
      "total_pages": 3
    }
  }
}
```

## 5. GET\_ATTACHMENT - Obter Anexo

- **Endpoint:** GET /?action=get\_attachment&key=1526&attachment\_id={id}
- **Descrição:** Retorna informações de um anexo específico.

Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"get_attachment"
attachment_id	string	✓	ID do anexo

### Exemplo de Requisição:

Bash

```
curl "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy?
action=get_attachment&key=1526&attachment_id=att_xyz123"
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "ATTACHMENT_RETRIEVED",
  "message": "Anexo obtido",
  "timestamp": "2025-10-17T10:30:00.000Z",
  "data": {
    "attachment_id": "att_xyz123",
    "message_id": "msg_abc456",
    "chat_id": "chat_abc123",
    "file_name": "documento.pdf",
    "file_type": "application/pdf",
    "file_size": 2048576,
    "drive_file_id": "1ABC123XYZ",
    "public_url": "https://drive.google.com/file/d/1ABC123XYZ/view",
    "uploaded_by": "user_123",
    "created_at": "2025-10-17T10:25:00.000Z"
  }
}
```

## 6. CLEANUP - Limpeza Manual

- **Endpoint:** GET /?action=cleanup&key=1526

- **Descrição:** Executa limpeza manual de mensagens antigas (>90 dias).

Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"cleanup"

### Exemplo de Requisição:

Bash

```
curl "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy?
action=cleanup&key=1526"
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "CLEANUP_DONE",
  "message": "Limpeza executada. 127 mensagens removidas",
  "timestamp": "2025-10-17T10:30:00.000Z",
  "data": {
    "deleted": 127
  }
}
```



## APIs POST

### 1. CREATE\_CHAT - Criar Chat

- **Endpoint:** POST /
- **Descrição:** Cria um novo chat no sistema.
- **Content-Type:** application/json



Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"create_chat"
creator_user_id	string	✓	ID do usuário criador
title	string	✓	Título do chat
creator_email	string	✗	Email do criador
observations	string	✗	Observações iniciais
send_email_summary	boolean	✗	Enviar resumo por email

### Exemplo de Requisição:

Bash

```
curl -X POST \
  "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \
  -H "Content-Type: application/json" \
  -d '{
    "action": "create_chat",
    "key": "1526",
    "creator_user_id": "user_123",
    "creator_email": "joao@empresa.com",
    "title": "Problema no Login",
    "send_email_summary": true
  }'
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "CHAT_CREATED",
  "message": "Chat criado com sucesso",
  "timestamp": "2025-10-17T10:35:00.000Z",
  "data": {
    "chat_id": "chat_abc123",
```

```
{
  "title": "Problema no Login",
  "creator_user_id": "user_123",
  "creator_email": "joao@empresa.com",
  "status": "aberto",
  "send_email_summary": true,
  "created_at": "2025-10-17T10:35:00.000Z"
}
```

## 2. ADD\_MESSAGE - Adicionar Mensagem

- **Endpoint:** POST /
- **Descrição:** Adiciona uma nova mensagem a um chat existente.
- **Content-Type:** application/json

Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"add_message"
chat_id	string	✓	ID do chat
sender_user_id	string	✓	ID do usuário remetente
sender_name	string	✓	Nome do remetente
message	string	✓	Texto da mensagem

### Exemplo de Requisição:

Bash

```
curl -X POST \
  "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \
  -H "Content-Type: application/json" \
  -d '{
    "action": "add_message",
    "key": "1526",
    "chat_id": "chat_abc123",
    "sender_user_id": "user_123",
    "sender_name": "João Silva",
    "message": "Oi, tudo bem?"
  }'
```

```
"message": "Não consigo fazer login no sistema"
}'
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "MESSAGE_ADDED",
  "message": "Mensagem adicionada com sucesso",
  "timestamp": "2025-10-17T10:38:00.000Z",
  "data": {
    "message_id": "msg_xyz789",
    "chat_id": "chat_abc123",
    "sender_user_id": "user_123",
    "sender_name": "João Silva",
    "message": "Não consigo fazer login no sistema",
    "has_attachment": false,
    "created_at": "2025-10-17T10:38:00.000Z"
  }
}
```

## 3. UPLOAD\_ATTACHMENT - Upload de Anexo

- **Endpoint:** POST /
- **Descrição:** Faz upload de um arquivo como anexo para uma mensagem específica.
- **Content-Type:** application/json

Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"upload_attachment"
chat_id	string	✓	ID do chat
message_id	string	✓	ID da mensagem
file_name	string	✓	Nome do arquivo
file_type	string	✓	Tipo MIME do arquivo
file_data	string	✓	Dados do arquivo em Base64
uploaded_by	string	✓	ID do usuário que fez o upload

### Exemplo de Requisição:

Bash

```
curl -X POST \
  "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \
  -H "Content-Type: application/json" \
  -d '{
    "action": "upload_attachment",
    "key": "1526",
    "chat_id": "chat_abc123def456",
    "message_id": "msg_xyz789abc123",
    "file_name": "screenshot_erro.png",
    "file_type": "image/png",
    "file_data":
    "iVBORw0KGgoAAAANSUgAAAAEAAAABCAAAAAfFcSJAAAADULEQVR42mNk+M9QDwADhgGAWjR9
    awAAAAABJRU5ErkJggg==",
    "uploaded_by": "user_123"
  }'
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
```

```
"success": true,
"code": "ATTACHMENT_UPLOADED",
"message": "Anexo enviado com sucesso",
"timestamp": "2025-10-17T10:40:00.000Z",
"data": {
  "attachment_id": "att_def456ghi789",
  "message_id": "msg_xyz789abc123",
  "chat_id": "chat_abc123def456",
  "file_name": "screenshot_erro.png",
  "file_type": "image/png",
  "file_size": 2048576,
  "drive_file_id": "1ABC123XYZ789DEF456",
  "public_url": "https://drive.google.com/file/d/1ABC123XYZ789DEF456/view",
  "download_url": "https://drive.google.com/uc?
export=download&id=1ABC123XYZ789DEF456",
  "uploaded_by": "user_123",
  "created_at": "2025-10-17T10:40:00.000Z"
}
}
```

## 4. ASSIGN\_ATTENDANT - Atribuir Atendente

- **Endpoint:** POST /
- **Descrição:** Atribui um atendente a um chat e muda o status para "em\_atendimento".
- **Content-Type:** application/json

Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"assign_attendant"
chat_id	string	✓	ID do chat
attendant_user_id	string	✓	ID do atendente
attendant_email	string	✗	Email do atendente

### Exemplo de Requisição:

Bash

```
curl -X POST \
  "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \
```

```
-H "Content-Type: application/json" \
-d '{
  "action": "assign_attendant",
  "key": "1526",
  "chat_id": "chat_abc123def456",
  "attendant_user_id": "attendant_789",
  "attendant_email": "maria@empresa.com"
}'
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "ATTENDANT_ASSIGNED",
  "message": "Atendente atribuído com sucesso",
  "timestamp": "2025-10-17T10:45:00.000Z",
  "data": {
    "chat_id": "chat_abc123def456",
    "title": "Suporte Técnico - Sistema Lento",
    "creator_user_id": "user_123",
    "creator_email": "joao@empresa.com",
    "attendant_user_id": "attendant_789",
    "attendant_email": "maria@empresa.com",
    "observations": "Cliente reportou lentidão no sistema desde ontem",
    "status": "em_atendimento",
    "send_email_summary": true,
    "created_at": "2025-10-17T10:30:00.000Z",
    "updated_at": "2025-10-17T10:45:00.000Z"
  }
}
```

## 5. UPDATE\_STATUS - Atualizar Status

- **Endpoint:** POST /
- **Descrição:** Atualiza o status de um chat. Quando fechado, envia email de resumo se configurado.
- **Content-Type:** application/json

Parâmetro	Tipo	Obrigatório	Descrição	Valores Válidos
key	string	✓	Chave de segurança	-
action	string	✓	"update_status"	-
chat_id	string	✓	ID do chat	-
status	string	✓	Novo status	aberto , em_atendimento , fechado , cancelado

### Exemplo de Requisição:

Bash

```
curl -X POST \
  "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \
  -H "Content-Type: application/json" \
  -d '{
    "action": "update_status",
    "key": "1526",
    "chat_id": "chat_abc123def456",
    "status": "fechado"
  }'
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "STATUS_UPDATED",
  "message": "Status atualizado com sucesso",
  "timestamp": "2025-10-17T11:00:00.000Z",
  "data": {
    "chat_id": "chat_abc123def456",
    "title": "Suporte Técnico - Sistema Lento",
    "creator_user_id": "user_123",
    "creator_email": "joao@empresa.com",
    "attendant_user_id": "attendant_789",
    "attendant_email": "maria@empresa.com",
    "observations": "Cliente reportou lentidão no sistema desde ontem",
    "status": "fechado",
    "send_email_summary": true,
  }
}
```

```
    "created_at": "2025-10-17T10:30:00.000Z",  
    "updated_at": "2025-10-17T11:00:00.000Z"  
  }  
}
```

## 6. ADD\_OBSERVATION - Adicionar Observação

- **Endpoint:** POST /
- **Descrição:** Adiciona uma nova observação ao chat com timestamp automático.
- **Content-Type:** application/json

Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"add_observation"
chat_id	string	✓	ID do chat
observation	string	✓	Texto da observação

### Exemplo de Requisição:

Bash

```
curl -X POST \  
  "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \  
  -H "Content-Type: application/json" \  
  -d '{  
    "action": "add_observation",  
    "key": "1526",  
    "chat_id": "chat_abc123def456",  
    "observation": "Verificado logs do servidor - identificado problema na  
base de dados"  
  }'
```

### Resposta de Sucesso (200):

JSON

```
{  
  "appName": "Nexi CRM",  
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",  
  "success": true,  
}
```



```
"code": "OBSERVATION_ADDED",
"message": "Observação adicionada com sucesso",
"timestamp": "2025-10-17T10:50:00.000Z",
"data": {
  "chat_id": "chat_abc123def456",
  "title": "Suporte Técnico - Sistema Lento",
  "creator_user_id": "user_123",
  "creator_email": "joao@empresa.com",
  "attendant_user_id": "attendant_789",
  "attendant_email": "maria@empresa.com",
  "observations": "Cliente reportou lentidão no sistema desde ontem\n[2025-10-17T10:50:00.000Z] Verificado logs do servidor - identificado problema na base de dados",
  "status": "em_atendimento",
  "send_email_summary": true,
  "created_at": "2025-10-17T10:30:00.000Z",
  "updated_at": "2025-10-17T10:50:00.000Z"
}
```

## 7. DELETE\_ATTACHMENT - Deletar Anexo

- **Endpoint:** POST /
- **Descrição:** Remove um anexo do sistema e do Google Drive.
- **Content-Type:** application/json

Parâmetro	Tipo	Obrigatório	Descrição
key	string	✓	Chave de segurança
action	string	✓	"delete_attachment"
attachment_id	string	✓	ID do anexo

### Exemplo de Requisição:

Bash

```
curl -X POST \
  "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \
  -H "Content-Type: application/json" \
  -d '{
    "action": "delete_attachment",
    "key": "1526",
```

```
"attachment_id": "att_def456ghi789"
}'
```

### Resposta de Sucesso (200):

JSON

```
{
  "appName": "Nexi CRM",
  "appLogo": "https://i.ibb.co/j9ccZJbh/logo.png",
  "success": true,
  "code": "ATTACHMENT_DELETED",
  "message": "Anexo deletado com sucesso",
  "timestamp": "2025-10-17T11:05:00.000Z"
}
```



## Códigos de Resposta



### Códigos de Sucesso

Código	Descrição	Ação
HELP	Documentação retornada	GET help
CHAT_CREATED	Chat criado com sucesso	POST create_chat
MESSAGE_ADDED	Mensagem adicionada	POST add_message
ATTACHMENT_UPLOADED	Anexo enviado com sucesso	POST upload_attachment
ATTENDANT_ASSIGNED	Atendente atribuído	POST assign_attendant
STATUS_UPDATED	Status atualizado	POST update_status
OBSERVATION_ADDED	Observação adicionada	POST add_observation
ATTACHMENT_DELETED	Anexo deletado	POST delete_attachment
MESSAGES_RETRIEVED	Mensagens obtidas	GET get_messages
CHAT_RETRIEVED	Chat obtido	GET get_chat
CHATS_RETRIEVED	Lista de chats obtida	GET list_chats
ATTACHMENT_RETRIEVED	Anexo obtido	GET get_attachment
CLEANUP_DONE	Limpeza executada	GET cleanup

## ✖ Códigos de Erro

Código	Descrição	Causa Comum	HTTP Sta
INVALID_KEY	Chave de segurança inválida	Key incorreta ou ausente	401
MISSING_FIELDS	Campos obrigatórios ausentes	Parâmetros faltando	400
NOT_FOUND	Recurso não encontrado	ID inexistente	404
INVALID_STATUS	Status inválido	Status não permitido	400
INVALID_EMAIL	Email inválido	Formato de email incorreto	400
MISSING_CONTENT	Conteúdo ausente	Mensagem ou anexo obrigatório	400
CHAT_NOT_FOUND	Chat não encontrado	chat_id inexistente	404
UNKNOWN_ACTION	Ação desconhecida	Action não existe	400
INTERNAL_ERROR	Erro interno do servidor	Problema no processamento	500



## Fluxo de Trabalho Completo - Exemplo

Este exemplo ilustra um ciclo completo de atendimento usando as APIs:

Bash

```
# 1. CRIAR CHAT
curl -X POST "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \
  -H "Content-Type: application/json" \
  -d '{
    "action": "create_chat",
    "key": "1526",
    "creator_user_id": "user_123",
    "creator_email": "joao@empresa.com",
    "title": "Problema no Login",
    "send_email_summary": true
  }'
# Resposta: chat_id = "chat_abc123"

# 2. ADICIONAR MENSAGEM DO CLIENTE
curl -X POST "https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy" \
  -H "Content-Type: application/json" \
  -d '{
```

```
    "action": "add_message",
    "key": "1526",
    "chat_id": "chat_abc123",
    "sender_user_id": "user_123",
    "sender_name": "João Silva",
    "message": "Não consigo fazer login no sistema"
  }'
```

```
# Resposta: message_id = "msg_xyz789"
```

### # 3. ADICIONAR ANEXO (SCREENSHOT)

```
curl -X POST "https://astounding-donut-
e0ea90.netlify.app/.netlify/functions/proxy" \
-H "Content-Type: application/json" \
-d '{
  "action": "upload_attachment",
  "key": "1526",
  "chat_id": "chat_abc123",
  "message_id": "msg_xyz789",
  "file_name": "erro_login.png",
  "file_type": "image/png",
  "file_data": "iVBORw0KGgoAAAA...",
  "uploaded_by": "user_123"
}'
```

### # 4. ATRIBUIR ATENDENTE

```
curl -X POST "https://astounding-donut-
e0ea90.netlify.app/.netlify/functions/proxy" \
-H "Content-Type: application/json" \
-d '{
  "action": "assign_attendant",
  "key": "1526",
  "chat_id": "chat_abc123",
  "attendant_user_id": "attendant_456",
  "attendant_email": "maria@empresa.com"
}'
```

### # 5. ADICIONAR OBSERVAÇÃO

```
curl -X POST "https://astounding-donut-
e0ea90.netlify.app/.netlify/functions/proxy" \
-H "Content-Type: application/json" \
-d '{
  "action": "add_observation",
  "key": "1526",
  "chat_id": "chat_abc123",
  "observation": "Resetado senha do usuário"
}'
```

### # 6. MENSAGEM DO ATENDENTE

```
curl -X POST "https://astounding-donut-
e0ea90.netlify.app/.netlify/functions/proxy" \
-H "Content-Type: application/json" \
-d '{
  "action": "add_message",
  "key": "1526",
  "chat_id": "chat_abc123",
  "sender_user_id": "attendant_456",
  "sender_name": "Maria Atendente",
  "message": "Olá! Resetei sua senha. Tente novamente com a senha
temporária: Temp@123"
}'
```

*# 7. FECHAR CHAT (ENVIA EMAIL AUTOMÁTICO)*

```
curl -X POST "https://astounding-donut-
e0ea90.netlify.app/.netlify/functions/proxy" \
-H "Content-Type: application/json" \
-d '{
  "action": "update_status",
  "key": "1526",
  "chat_id": "chat_abc123",
  "status": "fechado"
}'
```



## Exemplos Práticos por Linguagem

### JavaScript (Fetch API)

JavaScript

```
// Base URL da API original
const BASE_URL = 'https://astounding-donut-
e0ea90.netlify.app/.netlify/functions/proxy';
const API_KEY = '1526';

// Função auxiliar para fazer requisições
async function makeRequest(method, action, params = {}, body = null) {
  let url = `${BASE_URL}?action=${action}&key=${API_KEY}`;
  if (method === 'GET') {
    for (const key in params) {
      url += `&${key}=${params[key]}`;
    }
  }

  const options = {
    method: method,
```

```

    headers: { 'Content-Type': 'application/json' }
  };

  if (body) {
    options.body = JSON.stringify({ ...body, action, key: API_KEY }); //
    Inclui action e key no body para POST
  }

  const response = await fetch(url, options);
  return response.json();
}

// Criar Chat
async function criarChat(userId, email, titulo, sendEmailSummary = true) {
  return makeRequest('POST', 'create_chat', {}, {
    creator_user_id: userId,
    creator_email: email,
    title: titulo,
    send_email_summary: sendEmailSummary
  });
}

// Listar Chats Abertos
async function listarChats(status = 'aberto', page = 1, pageSize = 50) {
  return makeRequest('GET', 'list_chats', { status, page, page_size:
    pageSize });
}

// Adicionar Mensagem
async function adicionarMensagem(chatId, userId, nome, mensagem) {
  return makeRequest('POST', 'add_message', {}, {
    chat_id: chatId,
    sender_user_id: userId,
    sender_name: nome,
    message: mensagem
  });
}

// Exemplo de uso
(async () => {
  console.log('--- Testando APIs com JavaScript ---');

  // Criar Chat
  const newChat = await criarChat('js_user_001', 'js.user@example.com',
'Suporte JS');
  console.log('Chat Criado:', newChat);
  const chatId = newChat.data.chat_id;

```

```
// Adicionar Mensagem
const newMessage = await adicionarMensagem(chatId, 'js_user_001', 'Usuário JS', 'Olá do JavaScript!');
console.log('Mensagem Adicionada:', newMessage);

// Listar Chats
const chats = await listarChats('aberto');
console.log('Chats Abertos:', chats.data.chats);
})();
```

## Python (Requests)

Python

```
import requests
import base64
import mimetypes

BASE_URL = 'https://astounding-donut-e0ea90.netlify.app/.netlify/functions/proxy'
API_KEY = '1526'

def make_request(method, action, params=None, json_body=None):
    if params is None:
        params = {}
    params['action'] = action
    params['key'] = API_KEY

    if method == 'GET':
        response = requests.get(BASE_URL, params=params)
    elif method == 'POST':
        # Para POST, a API original espera action e key no body, junto com outros parâmetros
        if json_body is None:
            json_body = {}
        json_body['action'] = action
        json_body['key'] = API_KEY
        response = requests.post(BASE_URL, json=json_body)
    else:
        raise ValueError("Método HTTP não suportado")

    return response.json()

# Criar Chat
def criar_chat(user_id, email, titulo, send_email_summary=True):
    return make_request('POST', 'create_chat', json_body={
        'creator_user_id': user_id,
```



```

        'creator_email': email,
        'title': titulo,
        'send_email_summary': send_email_summary
    })

# Listar Chats
def listar_chats(status='aberto', page=1):
    return make_request('GET', 'list_chats', params={'status': status,
'page': page})

# Adicionar Mensagem
def adicionar_mensagem(chat_id, user_id, nome, mensagem):
    return make_request('POST', 'add_message', json_body={
        'chat_id': chat_id,
        'sender_user_id': user_id,
        'sender_name': nome,
        'message': mensagem
    })

# Upload de Arquivo
def upload_arquivo(chat_id, message_id, caminho_arquivo, uploaded_by):
    with open(caminho_arquivo, 'rb') as f:
        file_data = base64.b64encode(f.read()).decode('utf-8')

    file_name = mimetypes.guess_type(caminho_arquivo)[0]
    if file_name is None:
        file_name = 'application/octet-stream' # Fallback

    return make_request('POST', 'upload_attachment', json_body={
        'chat_id': chat_id,
        'message_id': message_id,
        'file_name': file_name,
        'file_type': mimetypes.guess_type(caminho_arquivo)[0] or
'application/octet-stream',
        'file_data': file_data,
        'uploaded_by': uploaded_by
    })

# Exemplo de uso
if __name__ == '__main__':
    print("--- Testando APIs com Python ---")

    # Criar chat
    chat = criar_chat('py_user_001', 'py.user@example.com', 'Suporte Python')
    print(f"Chat criado: {chat.get('data', {}).get('chat_id')}")
    chat_id = chat.get('data', {}).get('chat_id')

    if chat_id:

```

```
# Adicionar mensagem
msg = adicionar_mensagem(
    chat_id,
    'py_user_001',
    'Usuário Python',
    'Olá do Python!'
)
print(f"Mensagem enviada: {msg.get('data', {}).get('message_id')}")

# Listar chats abertos
chats = listar_chats(status='aberto')
print(f"Total de chats abertos: {chats.get('data',
{}).get('pagination', {}).get('total')}")

# Exemplo de upload de arquivo (crie um arquivo dummy para testar)
# with open("test_file.txt", "w") as f:
#     f.write("Conteúdo de teste")
# upload_response = upload_arquivo(chat_id, msg.get('data',
{}).get('message_id'), "test_file.txt", "py_user_001")
# print("Upload de arquivo:", upload_response)
```

## Referências

- [Documentação Original Nexi CRM Chat API v2.0](#)