




**Passo 2. [Segurança].** Clicar no Link "Segurança" para obter a lista de API's relacionadas com esta função.

← → ↻ 🔒 https://auto-smartapi.linx.com.br/product#product=s01-seguran-a

 Home Catálogo APIs

### S01 - Segurança

Produto de Segurança com EndPoints de segurança do sistema, como geração de tokens por exemplo

S01 - Segurança

#### Assinaturas

Name	Status
Segurança	Active

Your new product subscription name

#### APIs do Produto

🔍 Search APIs

Name	Description
Segurança	Pagina de documentação do módulo Segurança. Efetue a procura do endpoint desejado utilizando CTRL+F, expanda e clique em "Try it out!" ou acesse a documentação de outros módulos navegando entre os itens no canto superior direito.

**Passo 3. [Endpoint].** Selecionar o Endpoint que Gera o Token "/token".



Segurança

🔍 Search operations Group by tag ☒



Token

**POST** Gera o token para ser ut...

**POST** Gera um novo token atual...

API c

Pagina i  
docume

Ger

Gera ur  
token g

**Passo 4. [Endpoint].** Selecionar o Endpoint que gera um novo Token. Serão mostrados os detalhes sobre os dados de entrada que podem ser enviados na requisição para a API. Bem como um botão **Try It** que deve ser acionado para efetuar o teste.

API definition

Changelog

Página de documentação do módulo Segurança. Efetue a procura do endpoint desejado utilizando CTRL+F, expanda e clique em "Try it out!" ou acesse a documentação de outros módulos navegando entre os itens no canto superior direito.

Try It ▶

### Gera o token para ser utilizado no sistema.

Gera um novo token para ser utilizado em todas as requisições de API. Para garantir a segurança do sistema, todas as requisições de APIs, deverão enviar esse token gerado por esse EndPoint.

Token

#### Request

**POST** <https://auto-gwsmartapi.linx.com.br/api-seguranca/token>

#### Request body

multipart/form-data

Name	In	Required	Type	Description
grant_type	body	false	string	Credenciais do cliente ( client_credentials ).
username	body	true	string	Nome utilizado para fazer login no sistema. Tamanho do campo: 15.
password	body	true	string	Senha do usuário que realiza o login. Tamanho do campo: 15.
cnpjEmpresa	body	false	string	CNPJ da Revenda. Caso seja informado o valor vazio utilizará o CNPJ da empresa padrão do usuário. Pode ser informado também 8 dígitos se for necessário. Tamanho do campo: 14.

**Passo 5. [Teste].** Ao clicar em **Try It**, será apresentada a tela que permite informar os dados de entrada. Para o endpoint **"/token"** os dados obrigatórios são:

**Header:** **AMBIENTE**

Informar o nome do backend fornecido pela LINX nos dados de cadastro. Caso não seja informado, a API de geração de token não responderá corretamente, e será emitido um **erro de CORS (Cross-Origin Resource Sharing)**.

## BODY:

**username** [ USUÁRIO DO LINXDMS ]

**password** [ SENHA DO LINXDMS ]

Credenciais do LINXDMS. Estes são os dados mais sensíveis para utilização das API's. E após geração do token não são mais requeridos.

### Parameters

+ Add parameter

### Headers

Cache-Control	no-cache	Remove
Ocp-Apim-Subscr	.....	Remove
AMBIENTE	ambiente_backend	Remove

+ Add header

### Body

☒ Form-data ☐ Raw ☐ Binary

grant\_type (type: string)

username (type: string)

password (type: string)

cnpjEmpresa (type: string)

Send

**Passo 6.** [Retorno]. Quando os parâmetros enviados atendem plenamente os requisitos da API, o retorno enviado terá a estrutura json mostrada a seguir:

**Http json:**

```
{  
  "access_token": "string",  
  "token_type"   : "string",  
  "expires_in"   : 0  
}
```

**Exemplo 1 do retorno 200 recebido:**

Se todos os parâmetros foram enviados corretamente, a API devolverá um token com validade de 900 segundos. Este token deverá ser utilizado na requisição de todas as demais API's do sistema, para que seja atendida a nova regra de segurança e controle de acessos.

```
content-type: application/json; charset=utf-8
date: Fri, 24 Dec 2021 14:27:15 GMT
server: cloudflare
x-powered-by: ASP.NET
```

### Exemplo 2, retorno 400:

Neste exemplo solicitamos um token enviando a senha de usuário errada. O retorno neste caso contém uma STRING com as informações sobre o erro.

## HTTP response

HTTP/1.1 400 Bad Request

content-type: application/json; charset=utf-8

date: Mon, 27 Dec 2021 14:51:11 GMT

server: cloudflare

x-powered-by: ASP.NET

```
[{"O nome de usuário ou senha está incorreta."}, {"O nome de usuário ou senha está incorreta."}, {"Passo 0 - Método: LinxDMS.Seguranca.Token ValidaLoginToken(LinxDMS.DTO.Models.Seguranca.DadosUsuarioLoginModel) - Linha: D:\\a\\1\\s\\Library\\LinxDMS.Geral\\LinxDMS.Geral\\LoginToken.cs:39\\nPasso 1 - Método: Microsoft.AspNetCore.Mvc.ActionResult`1[LinxDMS.Seguranca.Token] ValidaUsuario(LinxDMS.DTO.Models.Seguranca.DadosUsuarioLoginModel) - Linha: D:\\a\\1\\s\\WebServices\\LinxDMS\\LinxDMS\\Controllers\\Seguranca\\TokenController.cs:47\\n"}]
```

### Exemplo 3, retorno com erro de CORS:

Um erro comum na requisição de geração de TOKEN é a falta do parâmetro AMBIENTE nos dados de Logon. Neste caso haverá uma falha crítica na resposta, com erro de CORS.

Unable to complete the request.

Since the browser initiates the request, it requires Cross-Origin Resource Sharing (CORS) enabled on the server. [Learn more](#)

## 2. PASSOS PARA RENOVAR A VALIDADE DO TOKEN:

Considerando que a validade de um token é 900 segundos ou 15 minutos, haverá necessidade de renovar a validade dele quando se aproxima o momento de expirar.

### 2.1. Renovação do Token:

Razões para efetuar um refresh do Token:

**Prorrogar o uso do Token por mais 15 minutos sem requerer login.**

Apesar de a validade de um Token ser limitada em 900 segundos, é possível renovar a validade do Token sem requerer um novo login, pela simples execução de um refresh antes de expirar completar o tempo.

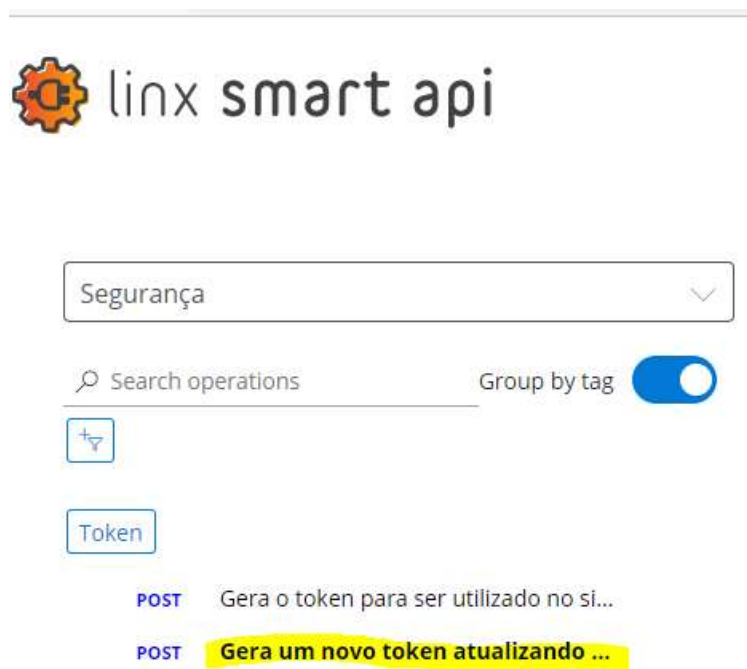
Este processo pode ser repetido indefinidamente, caso seja sempre feito dentro do prazo.

**Ao mudar a Sessão/Revenda:**

Um novo Token será gerado e retornado na operação TrocarRevendaSessão(). O qual deve ser enviado nas próximas requisições.

#### 2.1.1 Exemplificando a Renovação do Token na Linx Smart API:

**Passo 1. [Endpoint].** Ainda na SmartApi, catálogo segurança, selecionar o Endpoint `"/RefreshToken "`.





**Passo 2.** [Endpoint]. Todos os detalhes sobre a documentação do Endpoint "/RefreshToken " são apresentados. Para seguir com o teste, deve clicar no botão **Try It**.

## Segurança

API definition

[Changelog](#)

Página de documentação do módulo Segurança. Efetue a procura do endpoint desejado utilizando CTRL+F, expanda e clique em "Try it out!" ou acesse a documentação de outros módulos navegando entre os itens no canto superior direito.

### Gera um novo token atualizando seu tempo de expiração.

**Try It**

Utilizando o token gerado anteriormente, realiza a atualização do tempo de validade para que esse token não seja expirado. Dessa forma o tempo de expiração do token é renovado e é retornado um novo token para ser utilizado.

Token

### Request

**POST** [https://auto-gwsmartapi.linx.com.br/api-seguranca/RefreshToken\[?token\]](https://auto-gwsmartapi.linx.com.br/api-seguranca/RefreshToken[?token])

### Request parameters

Name	In	Required	Type	Description
token	query	false	string	Token gerando anter de expiração atualiza

**Passo 3.** [Teste]. Neste ponto, para enviar um token para a renovação, é preciso informar o parâmetro "token" com a string do token obtida na geração.

**Parâmetro GET:** token = "string token"

**Header:** AMBIENTE

Informar o nome do ambiente backend que atende ao usuário. Caso não seja informado, a API de geração de token não responderá corretamente, e será emitido um **erro** de CORS (Cross-Origin Resource Sharing).

## Tela de entrada de Header e Parâmetros

Segurança / Gera um novo token atualizando seu tempo de expiração. ✕

POST /RefreshToken

### Authorization ^

Subscription key Primary: Segurança ▼

### Parameters

token	eyJhbGciOiJIUzI1NiIsInR	Remove
-------	-------------------------	--------

+ Add parameter

### Headers

Cache-Control	no-cache	Remove
Ocp-Apim-Subscr	..... <span>🔍</span>	Remove
Ambiente	ambiente_backend	Remove

+ Add header

**Passo 4. [Retorno].** Quando os parâmetros enviados atendem plenamente os requisitos da API, o retorno recebido terá a estrutura json mostrada a seguir:

**Exemplo 1 do retorno 200 recebido:**

## HTTP response

HTTP/1.1 200 OK

```
content-type: application/json; charset=utf-8
date: Mon, 27 Dec 2021 19:00:33 GMT
server: cloudflare
x-powered-by: ASP.NET
```

[illegible]

### Exemplo 2, retorno 400:

O retorno 400 será emitido pela API quando for requisitado o Refresh de um token inválido, ou que já expirou o prazo de 900 segundos.

#### HTTP response

HTTP/1.1 400 Bad Request

content-type: application/json; charset=utf-8

date: Mon, 27 Dec 2021 18:59:03 GMT

server: cloudflare

x-powered-by: ASP.NET

```
[{"Token: Erro ao identificar o usuario."}, {"Token: Erro ao identificar o usuario."}, {"Passo 0 - Método: LinxDMS.Seguranca.Token.RefreshToken(System.String) - Linha: D:\\a\\1\\s\\Library\\LinxDMS.Seguranca\\LinxDMS.Seguranca\\TokenProvider.cs:60\\n"}, {"Passo 1 - Método: Microsoft.AspNetCore.Mvc.ActionResult.RefreshToken(System.String) - Linha: D:\\a\\1\\s\\WebServices\\LinxDMS\\LinxDMS\\Controllers\\Seguranca\\TokenController.cs:79\\n"}]
```

### Exemplo 3, retorno com erro de CORS:

Pelo mesmo motivo da geração do token, o erro de CORS ocorre pela falta do HEADER AMBIENTE nos dados da requisição.

Unable to complete the request

Since the browser initiates the request, it requires Cross-Origin Resource Sharing (CORS) enabled on the server. [Learn more](#)