

i Você pode usar esta documentação para as seguintes unidades de negócio:

mercado
livre

mercado
shops

mercado
envios

Última atualização em 11/06/2025

Para começar a utilizar nossos recursos, você deve desenvolver os processos de Autenticação e Autorização. Assim, você poderá trabalhar com os recursos privados do usuário quando autorize seu aplicativo.

Enviar access token no header

i Por segurança, você deve enviar o token de acesso por header toda vez que fizer chamadas para a API. O header da autorização será:

```
curl -H 'Authorization: Bearer APP_USR-12345678-031820-X-12345678' \
```

Por exemplo, fazer um GET para o recurso /users/me seria:

```
curl -H 'Authorization: Bearer APP_USR-12345678-031820-X-12345678' \
https://api.mercadolibre.com/users/me
```

Saiba mais sobre [a segurança do seu desenvolvimento](#).

Autenticação

O processo de autenticação é utilizado para verificar a identidade de uma pessoa em função de um ou vários fatores, garantindo que os dados de quem os enviou sejam corretos. Ainda que existam diferentes métodos, em Mercado Livre utilizamos o baseado em senhas.

Autorização

A autorização é o processo por meio do qual permitimos acessar a recursos privados. Nesse processo deverá ser definido que recursos e operações podem ser realizados ("só leitura" ou "leitura e escrita").

Como obtemos a autorização?

Por meio do Protocolo OAuth 2.0, um dos mais utilizados em plataformas abertas (Twitter, Facebook, etc.) e método seguro para trabalhar com recursos privados.

Este protocolo nos oferece:

- Confidencialidade, o usuário nunca deverá revelar sua senha.
- Integridade, apenas poderão ver dados privados os aplicativos que tiverem permissão para fazê-lo.
- Disponibilidade, os dados sempre serão disponibilizados no momento em que forem necessários.

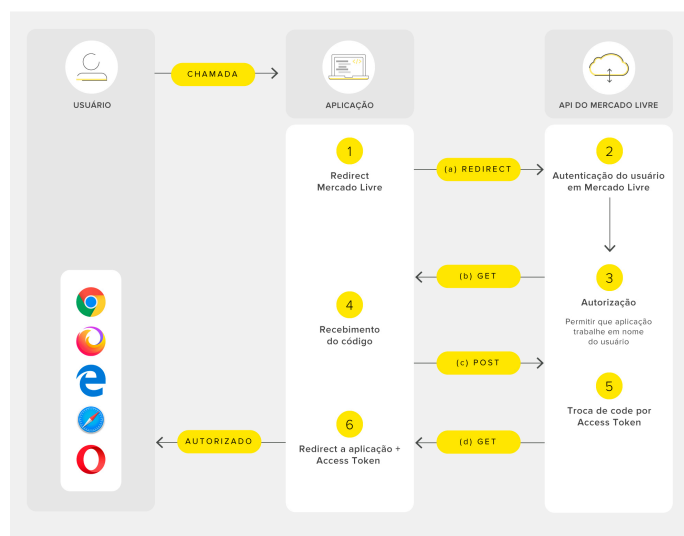
O protocolo de operação é chamado de Grant Types, e o utilizado é The Authorisation Code Grant Type (Server Side).

A seguir mostraremos a você como trabalhar com os recursos de Mercado Livre utilizando Implicit Grant Type.

Server side

O fluxo Server side é o mais adequado para os aplicativos que executam código do lado do server. Por exemplo, aplicativos desenvolvidos em linguagens como Java, Grails, Go, etc.

Em resumo, o processo que estará realizando é o seguinte:



- 1 Redireciona o aplicativo para Mercado Livre.
- 2 Não se preocupe com a autenticação dos usuários para Mercado Libre, nossa plataforma tomará conta disso!
- 3 Página de autorização.
- 4 POST para alterar o código de autorização por um access token.
- 5 O API de Mercado Libre altera o código de autorização por um token.

6 Já pode utilizar o access token para realizar chamadas ao nosso API e acessar os dados privados do usuário.

Passo a passo:

1. Realizando autorização

1.1. Conecte-se com seu usuário de Mercado Livre:

Notas:

- Você pode usar um [usuário de teste](#).
- Lembre que **o usuário que inicie sessão deve ser administrador**, para que o access token obtido tenha as permissões suficientes para realizar as consultas.
- Se o usuário for operador/colaborador, o grant será inválido e vai receber o erro **invalid_operator_user_id**.
- Os eventos a seguir podem invalidar um access token antes do tempo de expiração:
 - Alteração da senha pelo usuário.
 - Atualização do **Client Secret** por um aplicativo.
 - Revogação de permissões para seu aplicativo pelo usuário.
 - Se não utilizar a aplicação com alguma chamada em <https://api.mercadolivre.com/> durante 4 meses.

Importante:

A `redirect_uri` deve corresponder exatamente ao que está registrado nas configurações do seu aplicativo para evitar erros de acesso; a url não pode conter informações variáveis.

1.2. Coloque o seguinte URL na janela de seu navegador para obter a autorização:

```
https://auth.mercadolivre.com.br/authorization?response_type=code&client_id=$APP_ID&redirect_uri=$YOUR_URL&code_challenge=$CODE_CHALLENGE&code_challenge_method=$CODE_METHOD
```

No exemplo, utilizamos a URL para Brasil ([mercadolivre.com.br](https://api.mercadolivre.com.br/)), porém, se estiver trabalhando em outros países, lembre-se de alterar pelo domínio do país correspondente. Por exemplo, Uruguay: [mercadolivre.com.uy](https://api.mercadolivre.com.uy/). Ou Argentina: [mercadolivre.com.ar](https://api.mercadolivre.com.ar/). [Veja os países em que operamos](#).

Parâmetros

response_type: enviando o valor "**code**" será obtido um access token que permitirá ao aplicativo interagir com Mercado Livre.

redirect_URI: o atributo `YOUR_URL` é completado com o valor adicionado quando **quando o aplicativo for criado**. Deve ser exatamente igual ao que você configurou e não pode ter informações variáveis.

Autenticação e segurança

URI de redirect *

<https://localhost.com/redirect>



client_id: uma vez criado o aplicativo, será identificado como APP ID.

State: para aumentar a segurança, recomendamos que você inclua o parâmetro de estado na URL de autorização para garantir que a resposta pertença a uma solicitação iniciada por seu aplicativo. Caso você não tenha um identificador aleatório seguro, você pode criá-lo usando SecureRandom e deve ser exclusivo para cada tentativa de chamada. Portanto, a URL de redirecionamento será:

```
https://auth.mercadolivre.com.br/authorization?response_type=code&client_id=1620218256833906&redirect_uri=https://localhost.com/redirect&state=$12345
```

Um uso adequado para o parâmetro **state** é enviar um estado que você precisará saber quando a URL definida no `redirect_uri` é chamada. Lembre-se que o `redirect_uri` deve ser uma URL estática então se você está pensando em enviar parâmetros nesta URL use o parâmetro `state` para enviar esta informação, caso contrário a requisição irá falhar pois o `redirect_uri` não corresponde exatamente ao configurado em sua aplicação.

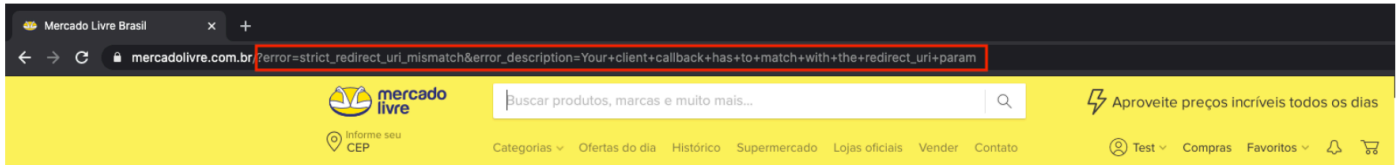
Os parâmetros a seguir são opcionais e só se aplicam se o aplicativo tiver o fluxo de **PKCE** (Proof Key for Code Exchange) habilitado. Entretanto ao ser ativada esta opção, o envio do campo se torna obrigatório.

code_challenge: código de verificação gerado a partir de `code_verifier` e cifrado com `code_challenge_method`.

code_challenge_method: método usado para gerar o `code_challenge`. Os seguintes valores são suportados atualmente:

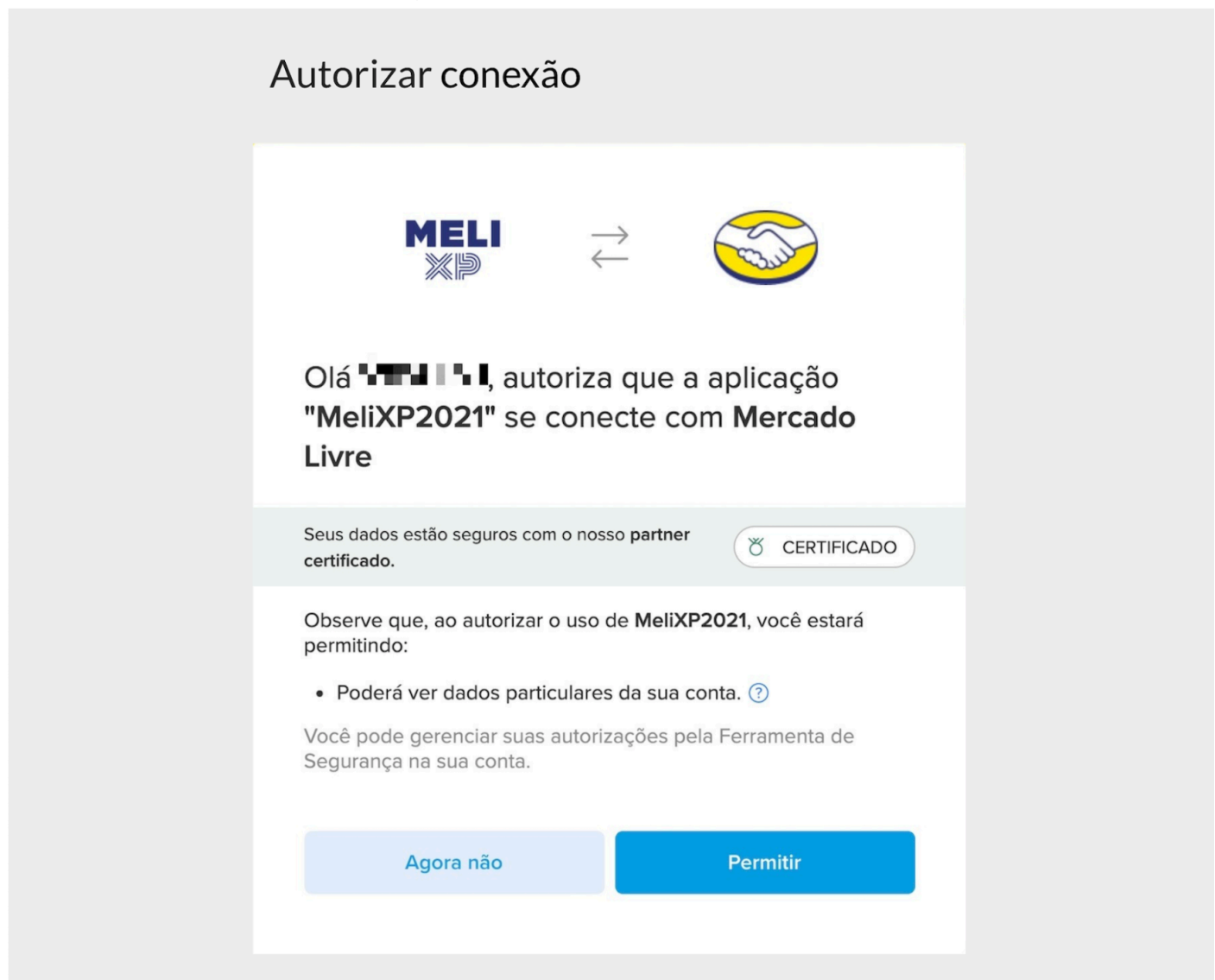
- S256: especifica que o `code_challenge` encontra-se usando o algoritmo de cifrado SHA-256.
- plain: o mesmo `code_verifier` é enviado como `code_challenge`. Por razões de segurança, não é recomendado usar este método.

O `redirect_uri` tem que corresponder **exatamente** ao inserido quando o aplicativo foi criado para evitar o seguinte erro, dessa forma, não pode conter informações variáveis:



Descrição: your client callback has to match with the redirect_uri param.

1.3. Como último passo do usuário, ele será redirecionado para a tela seguinte, onde lhe será requerido que autorize o aplicativo à sua conta.



Notas:

Adicionamos informações do DPP (nível integrador) informando ao vendedor se o aplicativo é certificado ou não.

Conferindo a URL, se pode observar que o parâmetro CODE foi adicionado.

```
https://YOUR_REDIRECT_URI?code=$SERVER_GENERATED_AUTHORIZATION_CODE&state=$RANDOM_ID
```

Exemplo:

```
https://localhost.com/redirect?code=TG-61828b7ffcc9a001b4bc890-314029626&state=ABC1234
```

Este CODE será utilizado para gerar um access token, que permitirá acessar a API.

Nota:

- Considere que se o usuário for operador/colaborador, NÃO será possível realizar o grant para a aplicação. Vai retornar o erro invalid_operator_user_id.

```
https://YOUR_REDIRECT_URI/?error=invalid_operator_user_id&error_description=The+operator_user_id+is+not+allow+to+authorize
```



Não é possível acessar esse site

Não foi possível o endereço de IP do servidor de YOUR_REDIRECT_URI.

DNS_PROBE_FINISHED_NXDOMAIN

Cargar de nuevo

Lembre-se de verificar esse valor para certificar-se de que a resposta pertence a uma solicitação iniciada por seu aplicativo, pois o Mercado Livre não valida este campo.

1.4 Se você receber a mensagem de erro: **Sorry, the application cannot connect to your account.** (Desculpe, o aplicativo não pode se conectar à sua conta), as seguintes considerações devem ser feitas:



Lo sentimos, la aplicación no puede conectarse a
tu cuenta

Ir al Inicio

1. A redirect_uri deve corresponder exatamente ao que está registrado nas configurações do seu aplicativo para evitar erros de acesso; a url não pode conter informações variáveis.
2. Valide se o token e a concessão do appid são válidos.
3. Verifique se o vendedor está fazendo login com a conta principal e não com um colaborador.
4. Verifique se o vendedor ou owner da aplicação possuem [dados pendentes de validação](#) , ou alguma inabilitação na conta.

2. Trocando o code por um token

Para que o código de autorização seja trocado por um access token, você deve realizar um POST enviando os parâmetros por BODY:

```
curl -X POST \
-H 'accept: application/json' \
-H 'content-type: application/x-www-form-urlencoded' \
'https://api.mercadolibre.com/oauth/token' \
-d 'grant_type=authorization_code' \
-d 'client_id=$APP_ID' \
-d 'client_secret=$SECRET_KEY' \
-d 'code=$SERVER_GENERATED_AUTHORIZATION_CODE' \
-d 'redirect_uri=$REDIRECT_URI' \
-d 'code_verifier=$CODE_VERIFIER'
```

Parâmetros

grant_type: authorization_code indica que a operação desejada é mudar o "code" por um access token.

client_id: é o APP ID do aplicativo que foi criado.

client_secret: é a Secret Key que foi gerado ao criar o aplicativo.

code: o código de autorização obtido no passo anterior.

redirect_uri: o redirect URI configurado para seu aplicativo não pode conter informações variáveis.

O seguinte parâmetro e opcionais só se aplicam se o aplicativo tiver o fluxo de **PKCE** (Proof Key for Code Exchange).

code_verifier: sequência de caracteres aleatória com a qual o code_challenge foi gerado. Isso será usado para verificar e validar a solicitação.

Resposta:

```
{
  "access_token": "APP_USR-123456-090515-8cc4448aac10d5105474e1351-1234567",
  "token_type": "bearer",
  "expires_in": 21600,
  "scope": "offline_access read write",
  "user_id": 1234567,
  "refresh_token": "TG-5b9032b4e23464aed1f959f-1234567"
}
```

Pronto! Você já pode usar o access token para fazer chamadas a nossa API e acessar os recursos privados do usuário.

3. Refresh token

Considere que o access token gerado expirará após 6 horas, desde solicitado. Por isso, para garantir que possa trabalhar por um tempo prolongado e não seja necessário solicitar constantemente ao usuário que volte a se logar para gerar um token novo, oferecemos a solução de trabalhar com um refresh token. Além disso, lembre-se de que o refresh_token é de **utilização única e que será devolvido um novo refresh_token em cada processo de atualização executado**.

Importante:

Para isso, o aplicativo deve ter selecionada a opção **offline_access**

Cada vez que fizer a chamada que muda o code por um access token, também terá o dado de um refresh_token, que deverá guardar para trocá-lo por um access token quando expirado. Para renovar seu access token deverá realizar a chamada seguinte:

```
curl -X POST \
-H 'accept: application/json' \
-H 'content-type: application/x-www-form-urlencoded' \
'https://api.mercadolivre.com/oauth/token' \
-d 'grant_type=refresh_token' \
-d 'client_id=$APP_ID' \
-d 'client_secret=$SECRET_KEY' \
-d 'refresh_token=$REFRESH_TOKEN'
```

Parâmetros

grant_type: refresh_token Indica que a operação desejada é atualizar um token.

refresh_token: o refresh token do passo de aprovação guardado previamente.

client_id: é o APP ID do aplicativo que foi criado.

client_secret: é o APP ID do aplicativo que foi criado.

Resposta:

```
{
  "access_token": "APP_USR-5387223166827464-090515-b0ad156bce700509ef81b273466faa15-8035443",
  "token_type": "bearer",
  "expires_in": 21600,
  "scope": "offline_access read write",
  "user_id": 8035443,
  "refresh_token": "TG-5b9032b4e4b0714aed1f959f-8035443"
}
```

A resposta inclui um novo access token válido por mais 6 horas e um novo REFRESH_TOKEN que deverá guardar para utilizá-lo cada vez que expirar.

Importante:

- Permitimos usar apenas o último REFRESH_TOKEN gerado para fazer o intercâmbio.
- O REFRESH_TOKEN só pode ser usado uma vez e somente pelo client_id ao qual está associado, depois de ser usado ele se tornará inválido.
- Para otimizar os processos de seu desenvolvimento, sugerimos que renove seu access token somente quando perder validade.

Access Token Auth y Auto



Referencia de códigos de erro

1. invalid_client: o client_id e/ou client_secret do seu aplicativo fornecido é inválido.

2. invalid_grant: os motivos são vários: pode ser porque o authorization_code ou refresh_token são inválidos, **expiraram ou foram revogados**, foram enviados em um fluxo incorreto, pertencem a outro cliente ou o redirect_uri usado no fluxo de autorização não corresponde ao que tem configurado seu aplicativo, ou o usuário (vendedor) possui a pendência de incluir dados e/ou documentos.

- 3. invalid_scope:** o alcance solicitado é inválido, desconhecido ou foi criado no formato errado. Os valores permitidos para o parâmetro alcance são: "offline_access","write" e "read".
- 4. invalid_request:** a solicitação não inclui um parâmetro obrigatório, inclui um parâmetro ou valor de parâmetro não suportado, tem algum valor dobrado ou está mal formado.
- 5. unsupported_grant_type:** os valores permitidos para grant_type são "authorization_code" ou "refresh_token".
- 6. forbidden (403):** a chamada não autoriza o acesso, possivelmente está sendo usado o token de outro usuário, [ou o IP esta bloqueado ou faltam scopes](#). Para o caso de grant o usuário não tem acesso à URL de Mercado Livre de seu país (.ar, .br, .mx, etc) e deve verificar que sua conexão ou navegador funcione corretamente para os domínios do MELI
- 7. local_rate_limited (429):** por excessivas requisições, são bloqueadas temporariamente as chamadas. Volte a tentar em alguns segundos.
- 8. unauthorized_client:** a aplicação não tem grant com o usuário ou as permissões (scopes) que tem o aplicativo com esse usuári. Não permitem criar um token.
- 9. unauthorized_application:** [a aplicação está bloqueada](#), e por isso não poderá operar até resolver o problema.

Erro Invalid Grant

Durante o fluxo obter o refresh token ou authorization code, é possível obter o erro invalid_grant com a mensagem "Error validating grant. *Your authorization code or refresh token may be expired or it was already used*"

```
{
  "error_description": "Error validating grant. Your authorization code or refresh token may be expired or it was already used",
  "error": "invalid_grant",
  "status": 400,
  "cause": []
}
```

Essa mensagem indica que o authorization_code ou refresh_token não existem, ou foram excluídos. Alguns dos motivos são:

- **Tempo de Expiração:** passado o tempo de duração do [refresh_token](#) (6 meses), vai expirar automaticamente e será necessário fazer de novo o fluxo para obter um novo refresh_token.
- **Revogação da autorização:** ao revogar a autorização entre a conta do seller e seu aplicativo (seja por parte do integrador ou do vendedor), os access_token e refresh_token serão invalidados. É possível verificar os usuários que não tem grant com sua aplicação desde a opção "Administrar Permissões" (no painel Meus Aplicativos), ou utilizando a chamada para acessar aos [usuários que outorgaram permissões ao seu aplicativo](#).
- **Revogação interna:** existem alguns fluxos internos que causam a exclusão das credenciais dos usuários, impedindo que os integradores possam continuar trabalhando em nome dos vendedores; nesses casos, é necessário completar de novo o fluxo de autorização/autenticação. Esses fluxos são disparados principalmente por exclusão das seções dos usuários. Os motivos são vários, mas os mais comuns são alteração de senha, desvinculação de dispositivos ou fraude. Saiba como [revogar a autorização de um usuário para sua aplicação](#).

Importante:

Considere que para esse último fluxo, apenas detalhamos alguns exemplos, não todos os casos disponíveis.

Seguinte: [Consulta API Docs](#).