

MANUAL DE INTEGRAÇÃO

The logo for Stelo, featuring the word "stelo" in a white, lowercase, sans-serif font. The letter "o" is stylized with a teal-colored circular element. The logo is centered within a dark blue, irregular, organic shape that has a teal-to-blue gradient. This shape is set against a background of soft, flowing waves in shades of teal and blue, which transition into a solid blue area at the bottom of the page.

stelo

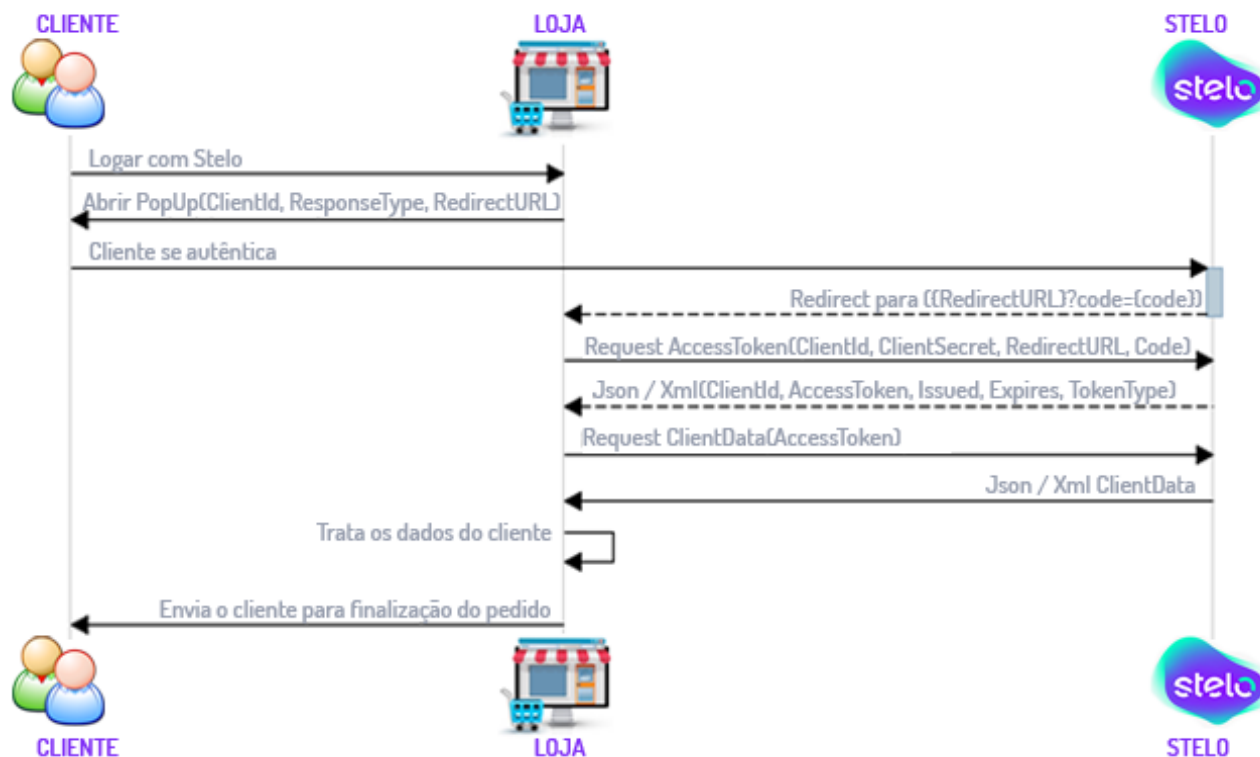
Índice

Integração com Login Stelo.....	3
1. Autorização de usuário.....	3
1.1. Loja redireciona o cliente	3
1.2. Retorno Stelo	4
1.3. Loja requisita AccessToken.....	4
1.4. Retorno Stelo	5
1.5. Recuperação de dados dos Clientes.....	6
1.6. Stelo retorna dados do cliente.....	6
2. Tratamento de dados dos clientes.....	8
2.1. Auto-preencher os campos na tela de cadastro	8
2.2. Auto-cadastrar e logar o cliente	8
3. Finalização da compra	8

Versão	Detalhes	Data
1.0	Versão inicial do manual.	18/08/2014
1.1	Alterado layout do documento	19/08/2014
1.2	Correção com base na API em funcionamento	22/04/2015

Integração com Login Stelo

A integração com o **Login Stelo** tem o objetivo de reduzir os passos para o checkout com foco no cadastro do cliente, com ela é possível recuperar os dados do cliente na **Stelo** e utilizar em sua loja. Caso deseje utilizar este modelo entre em contato com a equipe de integração da Stelo para solicitar suas credenciais.



1. Autorização de usuário

1.1. Loja redireciona o cliente

A loja deverá redirecionar o comprador para a url abaixo informando por método GET os seguintes parâmetros:

URL de Requisição:

<https://login.hml.stelo.com.br/sso/auth/v1/oauth2/authorize>

Query Parameters	Descrição	Tipo	Obrigatório
client_id	Chave Login Stelo da loja, diferente da chave ClientId utilizada nas integrações do meio de pagamento.	String	Sim
redirect_uri	URL da loja que será responsável por receber e gerenciar a resposta dessa requisição	String	Sim

response_type	Sempre informar "code"	String	Sim
State	Valor aleatório gerado pelo cliente para evitar <i>Cross-Site Request Forgery</i> (CSRF).	String	Sim
Scope	user_profile.all	String	Não

Exemplo Request

https://login.hml.stelo.com.br/sso/auth/v1/oauth2/authorize?response_type=code&client_id=f30e9903-efea-4bd9-83dd-7f0dc546909f&redirect_uri=https://172.31.0.34:9998/sso/auth/v1/oauth2/callback&scope=resource.READ&state=818e2198f

1.2.Retorno Stelo

Após o login do Cliente na Stelo, a Stelo retornará para a URL de Retorno informando por GET o valor do "code" na variável de mesmo nome (code).

Parameters	Descrição	Tipo	Obrigatório
code	Código de autorização retornado a partir da requisição inicial.	String	Sim
state	O valor deste parâmetro deve ser o mesmo código do parâmetro state enviado na requisição inicial.	String	Sim
error	Código de erro conforme RFC 6749, seção 4.1.2.1.	String	Sim, caso tenha ocorrido algum erro.
error_description	Descrição do erro.	String	Não
error_uri	Página de ajuda contendo a informações sobre o erro.	String	Não

RETORNO

<lojaexemplo.com.br/autenticacaoStelo?code=xxxxxxxxx&state=818e2198f>

1.3.Loja requisita AccessToken

A loja utiliza o "code" junto com seus identificadores para solicitar o token de acesso via requisição POST.

URL de Requisição:

<https://200.142.203.223/sso/auth/v1/oauth2/token>

Query Parameters	Descrição	Tipo	Obrigatório
client_id	Chave pública da Loja	String	Sim
client_secret	Chave secreta da Loja	String	Sim
redirect_uri	O valor deste parâmetro deve ser idêntico ao do parâmetro redirect_uri enviado na requisição inicial. Este parâmetro é usado somente para validação e conformidade com a especificação, o usuário não será redirecionado para o endereço.	String	Sim
code	Código de autorização retornado a partir da requisição inicial de autorização.	String	Sim
grant_type	Determina o tipo de autorização. Valor válido: <code>authorization_code</code>	String	Sim

Exemplo Request

Content-type: **application/x-www-form-urlencoded**

Encoding: utf-8

Body:

grant_type=authorization_code&**code**=XfBbCTR6R3DEHn2Cq25FLWig9kFv1b&**client_id**=f30e9903efea4bd983dd7f0dc546909f&**client_secret**=ff9738b5c066442e9c65bd7cd935ae7b&**redirect_uri**=https:// 172.22.0.34:9999/sso/auth/v1/oauth2/callback

1.4.Retorno Stelo

Stelo retorna via JSON o AccessToken

Parâmetro	Descrição	Tipo
access_token	Token de acesso	String
token_type	Tipo do token	String
expires_in	Tempo de vida, em segundos, do token de acesso	Integer
scope	Valor aleatório gerado pelo cliente para evitar <i>Cross-Site Request Forgery</i> (CSRF).	String
state		String

Response

```
Content-type: application/json
{
  "access_token": "PDCcxTmkKjDM7jKhV8Yz2cq9TCHpbliN2oI7Jb",
  "token_type": "Bearer",
  "expires_in": 3599,
  "scope": "resource.READ",
  "state": "818e2198f"
}
```

1.5. Recuperação de dados dos Clientes

Deve ser realizado um Request via GET para a URL abaixo informando apenas o AccessToken no header, como no exemplo abaixo, para a Stelo retornar os dados dos clientes.

URL de Requisição: <http://200.142.203.223/sso/auth/v1/oauth2/customer>

```
Content-type: application/json
Authorization: bearer {AccessToken}
```

1.6. Stelo retorna dados do cliente

Stelo retornará um XML/JSON com os dados abaixo:

Parâmetro	Descrição	Tipo
Nome	Nome do comprador	String
Email	Email cadastrado	String
CPF	CPF	String
RG	RG	varchar
BirthDate	Data de Nascimento no formato	DateTime
Gender	Sexo ("f" e "m")	char
Phones	Dados do telefone	PhoneCollection
PhoneNumber	Número do telefone com código de área	String
PhoneType	Tipo de número de telefone: 0 – residencial 1- Comercial 2 – Celular	Number
Alias	Apelido do endereço	String

Zipcode	Código Postal (CEP)	String
Street	Logradouro	String
Number	Número	String
Complement	Complemento	String
Neighborhood	Bairro	String
City	Cidade	String
State	Estado	String

Response

Content-type: application/json

```
{
  "name": "{full_name}",
  "email": "{email}",
  "cpf": "{cpf}",
  "rg": "{rg}",
  "birthDate": "{birth_date}",
  "gender": "{gender}",
  "phones":
    [
      {
        "number": "{number}",
        "type": "{type}"
      },
      {
        "number": "{number}",
        "type": "{type}"
      }
    ],
  "address": {
    "alias": "{alias}",
    "zipCode": "{zip_code}",
    "street": "{street}",
    "number": "{number}",
    "complement": "{complement}",
    "neighborhood": "{neighborhood}",
    "city": "{city}",
    "state": "{state}"
  }
}
```

```
}  
}
```

2. Tratamento de dados dos clientes

A loja tem duas opções com os dados recebidos:

- a. Auto-preencher os campos na tela de cadastro
- b. Auto-cadastrar e Logar o cliente

Para ambos os casos a loja deve verificar se os dados retornados do cliente, que são chaves primárias na loja, já existem, por exemplo **email**, para evitar erros na autenticação do cliente.

Caso existam, a loja pode definir o processo, algumas opções são:

- a. Poderá autentica-lo com os dados cadastrados na loja;
- b. Substituir as informações da loja pelas da Stelo;
- c. Solicitar ao cliente que escolha quais dados quer utilizar.

2.1. Auto-preencher os campos na tela de cadastro

Este modelo perde um pouco na usabilidade mas é possível solicitar ao cliente que complete o cadastro com dados que a Stelo não possui como por exemplo: um petshop pode solicitar dados do cachorro como obrigatórios no cadastro, neste caso é melhor optar por este modelo

2.2. Auto-cadastrar e logar o cliente

Este modelo diminui muito os passos de compra e uma maneira de utilizá-lo é: após criar a conta do cliente automaticamente, gerar uma senha randômica para que o mesmo possa finalizar a compra e depois disparar um e-mail de troca de senha ao cliente logado pelo Login Stelo.

3. Finalização da compra

Quando o cliente finalizar o processo de cadastro utilizando o Login Stelo ele não precisará fornecer nenhuma outra informação para pagar com a carteira Stelo e já estará autenticado com o Login anterior.