

ARTIGO: 55

MViral - Autenticação e acesso a API

Nesse artigo veremos como uma aplicação cliente deve autenticar-se para utilizar serviços da API do MViral. Sem autenticação não será possível utilizar nenhum serviço da API. Artigos posteriores explicarão quais os serviços disponíveis. Importante: O certificado SSL que permite a conexão segura a API (<https://www.mviral.com.br/api>) ainda não é reconhecido por uma autoridade certificadora(CA), ou seja, é um self-signed-certificate. Por enquanto, se necessário, configure sua biblioteca http para permitir comunicação através de certificados não reconhecidos por autoridade certificadora.

De forma muito simples:

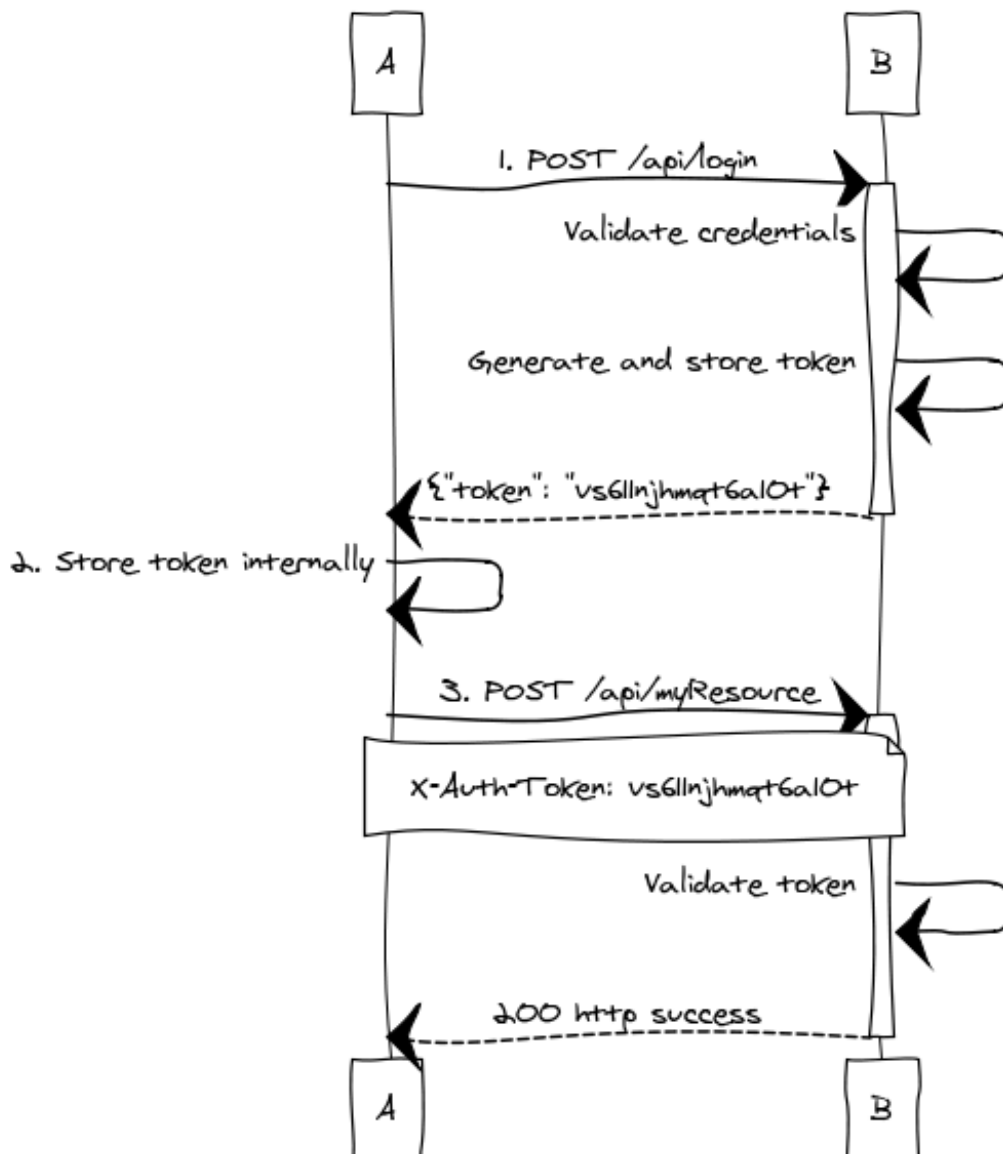
1. Aplicação cliente envia informações de login (username/password)
2. API valida informações e, se obtiver sucesso, retorna token
3. Aplicação cliente armazena token para enviar junto com requisições aos serviços
4. API verifica pela presença do token

Conforme diagrama de sequencia abaixo:

A = App. Cliente

B = API MViral

Authentication sequence



Passo 1, para autenticar dados de login, enviar uma requisição com as configurações abaixo:

- Formato: POST
- URL: **https://www.mviral.com.br/api/login**
- Tipo de dados: JSON (Content-type: application/json)
- Parâmetros obrigatórios: username, password

O serviço de autenticação espera uma requisição como essa:

```
{
  "username": "api@cliente.com.br",
  "password": "secret"
}
```

Passo 2, resposta da API, a aplicação cliente obterá uma resposta como abaixo:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
{
  "access_token":"3bicek1gc63oai6tfjkhog4kqn8ojd6a",
  "token_type":"Bearer",
  "username": "john.doe",
  "roles": [
    "ROLE_ADMIN",
    "ROLE_USER"
  ]
}
```

Caso contrário, uma resposta http com status 401(Unauthorized) será retornada, identificando que o usuário não foi encontrado ou não possui permissões.

Para testar essa etapa e obter a resposta acima, você pode utilizar o programa [curl](#):

```
curl -i --insecure -H "Content-Type: application/json" -d
'{"username":"api@mobilemind.com.br","password":"secret"}' https://www.mviral.com.br/api/login
```

Passo 3, a aplicação cliente deve extrair o token para enviá-lo como um header http quando realizar requisições http para os serviços.

Extraia o atributo 'token_type' recebido na resposta do passo 2 e armazene-o.

Passo 4, verifica pela presença do token quando cliente realizar as chamadas aos serviços

Configure um atributo cabeçalho http(header) com o nome '**X-Auth-Token**' com o valor do token recebido.

A presença desse token irá validar a aplicação cliente e permitir a utilização dos serviços.

Muito provavelmente você estará utilizando uma biblioteca http na linguagem de programação escolhida. Verifique na documentação de sua biblioteca como adicionar um header http ao realizar a requisição.

Para testar essa etapa, você pode utilizar o programa [curl](#):

```
curl --insecure -i -X POST -H "X-Auth-Token:c947h3e76f82v27g2gu15q4463l8d8op" -d
'numbers=99008894&text="Sms de teste"' https://www.mviral.com.br/api/dispatch
```

Será enviado uma resposta http 200(OK) caso a chamada tenha sido processada com sucesso e o texto 'success'.

```
HTTP/1.1 200 OK
```

```
Date: Wed, 27 Aug 2014 13:44:30 GMT
Server: Apache/2.4.6 (Ubuntu)
Content-Length: 7
Content-Type: text/html;charset=UTF-8
```

success

Será enviado uma resposta http 400(Bad Request) caso a chamada esteja mal formada(parâmetros incorretos/ausentes) e o texto descrevendo o erro.

```
HTTP/1.1 400 Bad Request
Date: Wed, 27 Aug 2014 14:09:49 GMT
Server: Apache/2.4.6 (Ubuntu)
Content-Length: 64
Connection: close
Content-Type: text/html;charset=UTF-8
```

Parameters missing. Make sure to send all of these: numbers,text

Esses 4 passos compõem o fluxo de autenticação que permite utilizar serviços da API do MViral.

Em caso de dúvidas ou dificuldades, por favor, entre em contato com o suporte através do e-mail suporte@mobilemind.com.br