

# Antes de avançar

## Autenticação na API

Todas as requisições http feitas aos servidores da Oystr devem conter, no header da requisição, o parâmetro **X-Oystr-Auth**, com o seu **token** de autenticação.

Todas as requisições para nossa API demandam o uso do **header X-Oystr-Auth**, ele é obrigatório. Caso o token não seja informado no header da requisição ou caso o token seja inválido, o retorno do servidor, nas demais requisições, será **HTTP/1.1 403 Forbidden**, indicando a ausência de permissões para efetuar a requisição.

O método /ping é utilizado apenas para realização de autenticação na API. Para saber como obter a o seu token, acesse o link abaixo e siga as instruções ali definidas.

## Como obter a chave de API

Antes de mais nada, a primeira coisa que você precisa fazer é obter a sua **chave de API (API Key)**. Sem ela, você não conseguirá se autenticar na API e, conseqüentemente, não conseguirá consumir seus métodos.

Abaixo, segue um passo a passo de como obter a sua chave de API:

- Acesse o Console/Painel da Oystr (<https://console4.oystr.com.br>);

Sua chave será criada e exibida no console. Para consumir a API, basta copiar a chave criada e passar no **HEADER**, conforme indicado no método de **Autenticação na API**.

## Conceitos

Todos os robôs da Oystr tem um nome e versão. Os robôs da Oystr trabalham e operam em execuções. Uma execução contém itens que fazem parte de uma fila. Cada item da fila é consumida por uma execução de robô. As execuções podem ter vários robôs trabalhando ao mesmo tempo. Após a execução é possível você ter acesso aos dados de relatório e saída do robô.

De maneira geral, para se criar uma execução e rodar um robô, temos que realizar as seguintes tarefas na ordem descrita:

Ordem	Tipo	Descrição
-------	------	-----------

*0.1*	**Opcional**	*Validar as credenciais que serão utilizadas pelos robôs para acessar um sistema (tribunal, gestão jurídico, etc)*
*0.2*	**Opcional**	*Executar uma proto fila para obter itens*
1	**Obrigatório** *	Criar uma fila de itens
2	**Obrigatório** *	Anexar arquivos a fila de itens (apenas para os robôs que fazem upload de arquivos)
3	**Obrigatório** *	Iniciar a execução
4	**Obrigatório** *	Consultar o status da execução
5	**Obrigatório** *	Consultar o informações da execução
6	**Obrigatório** *	Validar as respostas/dados/journal da execução
7	**Obrigatório** *	Obter o relatório/resultado da execução

## Respostas Assíncronas da API

Antes de continuar, é importante entender o conceito de que muitas das chamadas na **API da Oystr** retornam respostas de maneira **assíncrona**.

Isso quer dizer que a resposta da chamada **não estará pronta de imediato** e você receberá um id para consultar o resultado da chamada posteriormente.

O design da API leva em consideração **o tempo que os robôs demoram para executar uma determinada tarefa**. Lembre-se, algumas tarefas podem demorar horas para finalizar. Isso significa que seu código de integração precisa estar preparado para lidar com essas respostas assíncronas.

A seguir, veja um exemplo de chamada que recebe um id de consulta assíncrona.

```
http POST https://api4.oystre.com.br/v1/[método de retorno assíncrono] X-Oystre-Auth:[token]
```

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
```

```
"id": "[id]"
}
```

O retorno de um método assíncrono é um **json** com um único campo: **id**. Precisamos fazer uma segunda chamada na API, usando o **id retornado na chamada anterior** para consultar se o resultado já está pronto. Considere uma chamada para validação de credencial no robô de testes “sample”, usando o usuário “zzz” e a senha “zzz”. Segue exemplo:

```
http POST https://api4.oyster.com.br/v1/credentials/validate/sample/v1.0 X-Oyster-Auth:[token] username=zzz
password=zzz
```

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
```

```
{
  "id": "zqwG3hcEGxYl457T8A1d6qS77PCky71HgWdBDj4Z1sTmV6mqIp5VCcCs1ZutIEue"
}
```

Após esta chamada, agora temos em mãos o **id de consulta do resultado da nossa chamada assíncrona**. O recomendado é que se consulte no **mínimo a cada 10 segundos e no máximo a cada 120 segundos**.

A partir de agora, precisamos consultar continuamente a API, até que tenhamos a confirmação de conclusão da execução.

Neste momento, podemos ter 3 tipos de código de resposta: 200, 404 ou 500. Abaixo, segue descrição de cada tipo de resposta.

Código	Tipo	Descrição
200	OK - Resultado pronto	Com o payload da resposta, você receberá os dados relacionados ao seu resultado.
404	Não encontrado - Resultado não disponível	Neste caso indica-se que os dados ainda não estão prontos.
500	Erro interno	Neste caso entrar em contato conosco e informar o ID da chamada.

Segue um exemplo de consulta, com resposta:

```
http GET https://api4.oyster.com.br/v1/cached/[id] X-Oyster-Auth:[token]
HTTP/1.1 404 NOT FOUND // indica que o resultado não está pronto
// 3 segundos depois
http GET https://api4.oyster.com.br/v1/cached/[id] X-Oyster-Auth:[token]
```

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "Message"    : null,  
  "notAllowed" : null,  
  "notAvailable": null,  
  "Valid"      : true  
}
```

---

Revision #4

Created 24 March 2024 23:04:34 by Luan Melo

Updated 24 March 2024 23:14:01 by Luan Melo