API de Itens -1:1

Conceitos

Todos os robôs da Oystr que são utilizados através da API 1:1 possuem um nome e versão. Diferente do modelo em lote, onde existe o conceito de filas e execuções com múltiplos itens, na API 1:1 cada requisição representa a execução de um único item de forma isolada. **Não há o conceito de fila ou execução agrupada neste modelo**.

Cada chamada realizada na API 1:1 dispara uma execução individual de robô, responsável por processar os dados enviados e retornar um resultado específico para aquele item. Essas execuções são independentes entre si e não compartilham estado ou fila.

De maneira geral, para executar um robô utilizando a API 1:1, temos o seguinte fluxo:

Ordem	Tipo	Descrição
0.1	*Opcional*	*Validar as credenciais que serão utilizadas pelos robôs para acessar um sistema (tribunal, gestão jurídico, etc)*
1	**Obrigatório**	Criar execução do item e receber o identificador da execução (id assíncrono)
2	**Obrigatório**	Consultar o retorno/status da execução
3	**Obrigatório**	Obter o resultado final 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.oystr.com.br/v1/[método de retorno assíncrono] X-Oystr-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.oystr.com.br/v1/credentials/validate/sample/v1.0 X-Oystr-Auth:[token] username=zzz
password=zzz

HTTP/1.1 200 OK
Content-Type: application/json

{
   "id": "zqwG3hcEGxYl457T8A1d6qS77PCky71HgWdBDj4Z1sTmV6mqlp5VCcCs1ZutIEue"
}
```

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 | 204	Resultado não disponível ainda	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.oystr.com.br/v1/cached/[id] X-Oystr-Auth:[token]
HTTP/1.1 404 NOT FOUND // indica que o resultado não está pronto
// 3 segundos depois
http GET https://api4.oystr.com.br/v1/cached/[id] X-Oystr-Auth:[token]
HTTP/1.1 200 OK
Content-Type: application/json
{
   "Message"     : null,
   "notAllowed"  : null,
   "notAvailable": null,
   "Valid"       : true
}
```

0.1 - Validar Credenciais

```
POST: https://api4.oystr.com.br/credentials/validate/:botId/v3.0.0-dev
```

Usuário/Senha - UP

objeto ***"credentials"***, com par **usuário e senha**, será utilizado pelo robô para conseguir acesso ao sistema no qual ele irá executar as tarefas. Por exemplo, em um robô de tribunal onde a tarefa é obter a última movimentação processual, esse usuário/senha será utilizado para entrar no sistema do tribunal.

Quando informado na requisição (via arquivo ***payload*** ou especificado no ***body*** da requisição), é necessário seguir o formato abaixo:

```
{
  "username": "...",
  "password": "...",
  "credentialsOption": "..."
}
```

Parâmetros

Parâmetro	Tipo	Descrição
username	*String (obrigatório)*	Usuário utilizado para acessar o sistema em que o robô irá executar a tarefa.
password	*String (obrigatório)*	Senha correspondente ao usuário informado.

Parâmetro	Tipo	Descrição
credentialsOption	*String (opcional)*	Valor opcional que será informado indicando particularidades das opções informadas. Quando necessário, a documentação do robô irá apontar.

Certificado Digital A1 - CP

O objeto *“credentials”* com certificado A1 será utilizado pelo robô para conseguir acesso ao sistema no qual ele irá executar as tarefas. Por exemplo, em um robô unificado intimações onde a tarefa é obter a todas as intimações disponíveis para o advogado em diversos tribunais do Brasil, esse certificado A1 será utilizado para entrar no sistema de cada tribunal especificado.

Quando informado na requisição (via arquivo ***payload*** ou especificado no ***body*** da requisição), é necessário seguir o formato abaixo:

```
{
  "username": "...",
  "base64Cert": "...",
  "pin": "...",
  "credentialsOption": "..."
}
```

Parâmetros

Parâmetro	Tipo	Descrição
username	*String (obrigatório)*	Usuário utilizado para acessar o sistema em que o robô irá executar a tarefa.
base64Cert	*String (obrigatório)*	Conteúdo do certificado A1 codificado em Base64
pin	*String (obrigatório)*	PIN ou senha utilizada para utilizar o certificado.
pin	*String (opcional)*	Valor opcional que será informado indicando particularidades das opções informadas. Quando necessário, a documentação do robô irá apontar.

1 - Criar Execução do Item 1:1

```
POST: https://api4.oystr.com.br/v1/execute/single
```

Código de retorno

Retorno	Descrição
202	inicia a execução de um ítem. Retona o id para consulta futur

Retorno	Descrição
5xx ou 4xx	Falha ao iniciar a execução do ítem. Nestes casos é necessário utilizar (manualmente via suporte) o header X-Oystr-TraceId retornado para verificar se o ítem foi recebido pela API e está em execução ou não.

Formato da Requisição (via HTTP)

```
{
   "dry": false,
   "bot": "",
   "version": "",
   "cid": "",
   "timeout": "",
   "deadline": "",
   "credentials": {
      "username": "",
      "password": ""
   },
   "data": {},
   "files": [{
      "bound": true,
      "property": "",
      "description": "",
      "name": "",
      "data": ""
   }]
}
```

Campo	Obrigatório	Formato	Default	Descrição
dry	Não	Boolean	false	Indica se é uma execução de teste
bot	Sim	String		Id do robô
version	Sim	String		Versão do robô
cid	Não	String	vazio	Identificador único da requisição[1]
force	Não	Boolean	vazio	Indica se um item único, discriminado pelo cid deve ser reexecutado ou não
timeout	Não	Duration	5 minutos	Tempo máximo aguardando o retorno de um item desde o início da execução[2]. Exemplo: 30s

Campo	Obrigatório	Formato	Default	Descrição
deadline	Não	ZonedDateTime	vazio	Data máxima para o início da execução do item[3] no formato ISO 8601. Exemplo 2022-01-01T00:00:00.0-03:00
credentials	Robô	json		Credenciais usadas na requisição
data	Sim	json		Dados passados diretamente para o robô. Cada robô possuí um formato específico.
file	Robô	json		Arquivos passados para o robô

1. ID usado para evitar requisições duplicadas. Caso o cid não seja único a resposta será do tipo Duplicate . Caso o cid seja inválido a requisição será rejeitada
2. . Após este período a resposta será do tipo Timeout
3. Caso a execução do ítem não tenha iniciado até esta data a resposta será do tipo Overdue.

2 - Consultar o retorno da execução - (assíncrono)

```
GET: https://api.oystr.com.br/v1/execute/single/:bot/:cached?format=latest
```

Código de retorno

Retorno	Descrição
404	id inválido ou ítem rejeitado. Nunca haverá uma resposta para este item.
202	Reposta indisponível, aguarde
200	Resposta disponível

3 - Interpretar e obter o resultado final da execução

Após uma resposta 200, teremos como payload da resposta o resultado final da execução. Os resultados ficam disponíveis no modelo do endereço abaixo de acordo com o header **X-Oystr-ReturnPath**:

```
GET: https://api.oystr.com.br/v1/execute/single/:bot/:cached?format=latest
```

O payload recebido depende do valor da variável forma:

Formato Legado - format=legacy

```
GET: https://api.oystr.com.br/v1/execute/single/:bot/:cached?format=latest
```

```
{
   "type": "",
   "payload": {}
}
```

Formato Recomendado - format=latest

O payload em json obtido quando a resposta está disponível segue o seguinte modelo:

```
{
   "account": 0,
   "user": 0,
   "bot": "",
   "version": "",
   "cid": "",
   "request": "",
   "started": "",
   "ended": "",
   "taskTime": "",
   "timeout": "",
   "finishedAs": "",
   "retry": "",
   "dry": false,
   "result": {}
}
```

Descrição dos campos:

Campo	Formato	Descrição
account	Long	Conta que realizou a requisição
user	Long	Usuário que realizou a requisição
bot	String	Robô
version	String	Versão do robô
cid	String	id de integração
request	String	Id da requisição

Campo	Formato	Descrição
started	ZonedDateTime	Data de início da execução do item no formato ISO-8601
ended	ZonedDateTime	Data de início da execução do item no formato ISO-8601
taskTime	Duration	Tempo de execução do item
timeout	Duration	Timeout efetivamente usado
retry	Enum	indica se é possível re-executar o item
dry	Json	Indica se é uma execução DE TESTE
finishedAs	Enum	Tipo de resultado (tabela abaixo)
result	Json	Payload com o resultado

Tipos de retorno

Os tipos de resultados possíveis:

Resposta	Reexecução Recomendada?	Origem	Descrição
Response	Não	Bot	Resposta válida
PartialResponse	Não	Bot	
NotAllowed	Sim	Bot	Não é permitida a execução do item
NotFound	Não	Bot	Item não encontrado
NotConsistent	Não	Bot	Dados inconsistentes
NotAvailable	Sim	Bot	Item não disponivel
ProxyError	Sim	Bot	Erro de proxy
CaptchaError	Sim	Bot	Erro de captcha
Forbidden	Sim	Bot	Execução do item não é permitida
BotError	Sim	Bot	Erro no robô

Resposta	Reexecução Recomendada?	Origem	Descrição
Other	Não	Bot	Outros
RequestHasViolations	Não	API	Pedido contém informações inválidas. O item não foi executado.
NotAndApiRequest	Não	API	Formato inválido de requisição. O item não foi executado
NotAccepted	Não	API	Execução do item nõa foi aceita. O item pode ou não ter sido executado
Timeout	Não	API	Timeout durante a execução do item. O item pode ou não ter sido executado
Overdue	Não	API	Execução do item ocorreria apís o deadline. Item não foi executado
Duplicate	Não	API	Item duplicado baseado no cid. Item não foi executado
UnexpectedError	Não	API	Erro inesperado. O item pode ou não ter sido executado
Unknown	Não	API	Desconhecido. O item pode ou nõa ter sido executado

Requisições Duplicadas

A Oystr não testa os dados enviados para a execução com a exceção dos atributos cid e force . Caso o cid esteja presente na requisição, a Oystr fará a verificação se este cid foi executado nos últimos 15 dias para esta conta e robô. Caso o cid seja "duplicado", o ítem será executado apenas se o parâmetro force estiver presente com o valor true , caso contrário a resposta do ítem será do tipo Duplicate .

A Oystr emprega uma estratégia estatística para verifica ser o ítem já foi executado. Esta estratégia permite falsos positivos, ou seja:

1. ítem não foi executado (com certeza) para esta conta e robô nos últimos 15 dias, ou2
2. . ítem possivelmente já foi executado antes (false positivo)

Um cid , se presente, deve ter entre 1 e 50 carateres alphanuméricos ou '-' de acordo com a seguinte expressão regular: [0-9a-zA-Z\\-]{1,50} . Caso o cid seja inválido o ítem não será executado, a requisição será ignorada e ficará sem resposta, retornando 404 em consultas futuras.

Reexecução

Em alguns casos é possível que um ítem seja reexecutado em caso de erro, mesmo que um cid tenha sido utilizado. Os critérios que definem se um ítem pode ou não ser reexecutado ficam a cargo do cliente da API que deve observar os seguintes critérios:

1. Se houve uma execução anterior "do mesmo" íten, esta deve ter retornado como insucesso, com o parâmetro retry = SAFE . Caso contrário não é seguro reexecutar o ítem.
2. . Caso um cid seja utilizado, ele deve ser único ou o parâmetro force = true deve ser utilziado para ignorar o teste de duplicidade.

A Oystr não verifica se a requisição anterior retornou com retry = SAFE durante a nova execução, mesmo que o cid seja o mesmo. Esta verificação fica a cargo do cliente/usuário da api.

Revision #6
Created 25 March 2024 03:17:51 by Luan Melo
Updated 15 April 2026 16:50:13 by Jonas Pacheco