Intimações

App de Intimações

Referência da API intimações, para automação com acionamento de hooks por API para notificação de eventos.

O APP de intimações é um serviço de agendamento para execução recorrente dos robôs de intimação da Oystr. Ele executa os robôs de intimação nos dias e horários agendados com as configurações definidas e, ao fim da execução, pode disparar um **webhook** contendo o endpoint para obtenção dos resultados para uma url definida.

O que são Webhooks e o seu papel no App de intimações?

Os Webhooks são uma forma de um software avisar outro software através de gatilhos e disparos. A melhor maneira de entender isso é pensar nos webhooks como notificações de eventos. Na nossa aplicação de intimações, sempre que um perfil configurado terminar de carregar ou abrir as intimações no tribunal, um disparo de evento será feito através da nossa API para o webhook configurado no software integrado. É desta maneira que as integrações serão feitas no App de Intimações Oystr.



Perfis de Abertura de Intimações

O App de Intimações opera a partir de perfis. Um perfil representa uma configuração de credenciais, opções, filtros, etc. usada para agendar e iniciar os robôs de intimação. Quando dia e horário especificados no perfil forem cumpridos, o App de intimações executará as seguintes ações:

1. `Listagem`: acionará o robô com as credenciais e opções definidas para obter uma lista das intimações disponíveis. Erros nessa etapa são marcados como `Erro ao listar` na interface do App e uma nova tentativa será realizada no próximo ciclo de 30 minutos contanto que o horário limite configurado não tenha sido atingido. Caso não tenha sido concluída em até 3 horas, será interrompida automaticamente. Para cada cliente são listados até 2 perfis simultâneos no momento.

2. `Filtragem`: aplicará os filtros automáticos definidos sobre a lista de intimações recebida para decidir quais deve abrir
3. `Disparo do webhook opcional`: se definido, enviará um webhook contendo a lista das intimações que serão abertas pelo robô e a lista das intimações que *não* serão abertas pelo robô. Caso ocorra erro ou seja recebido qualquer status de retorno fora da lista `200, 201, 202, 203, 204`, uma nova tentativa será feita 30 minutos depois com um limite de 10 tentativas. O estado atual de envio e consulta fica disponível na aba `Auditoria` seção `Integração` da interface do App.
4. `Abertura`: acionará o robô com as credenciais e opções definidas para realizar a captura das intimações escolhidas na `Filtragem`. As execuções criadas são disponibilizadas na tela inicial do console para a conta que tiver criado a chave de API cadastrada no App de intimações (a chave de API da integradora não influencia esse comportamento). **Somente as execuções criadas pelo App de intimações são passíveis de integração**. Não há limite de execuções simultâneas por parte do App de intimações. A conclusão das execuções é verificada a cada 5 minutos
5. `Obtenção do registro`: recebe o registro de execução do robô e realiza o download dos anexos para integração
6. `Processamento`: se definido, aplicará processamentos extras no registro de execução do robô. Se não especificado, aplicará a padronização dos campos de data com a máscara `DD/MM/YYYY`. A padronização é aplicada apenas aos campos que puderem ser identificados, caso contrário o conteúdo é deixado intacto
7. `Disparo do webhook principal`: enviará um webhook contendo o endpoint de consulta do registro. Caso ocorra erro ou seja recebido qualquer status de retorno fora da lista `200, 201, 202, 203, 204`, uma nova tentativa será feita 30 minutos depois com um limite de 10 tentativas. O estado atual de envio e consulta fica disponível na aba `Auditoria` da interface do App.

Configurações Gerais

IMPORTANTE: Antes de Definir o Primeiro Perfil

Antes de começar, é necessário realizar duas configurações/ajustes pela primeira (e única) vez:

* Será necessário criar uma chave de API, caso ainda não tenha sido feito. Veja neste link como fazer.
* Será necessário realizar uma configuração padrão que servirá como modelo base para novos perfis. A configuração padrão deve ser definida em **Perfis**->**Configurações**.

Configurações Gerais

Link para cadastrar perfis no painel de Intimações.

Link das configurações gerais.

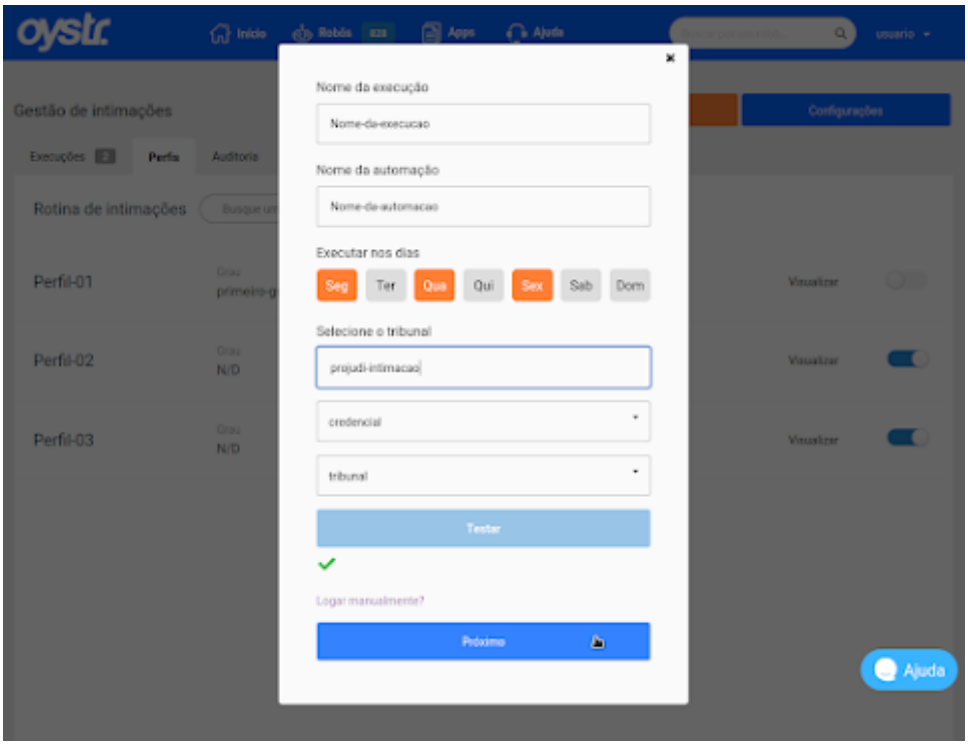
Formulário das configurações gerais da conta.

CAMPO	DESCRIÇÃO
Webhook	Endpoint de envio do hook principal. Se essa configuração ficar em branco, não haverá disparo no webhook.
JSON Header Webhook	Headers adicionais a ser incluídos na requisição do hook principal em formato JSON. Isso é usado normalmente para realizar autenticações e garantir a segurança da sua aplicação.
WebhookProto	Endpoint de envio do hook opcional com a listagem dos itens que serão abertos. Se essa configuração ficar em branco, não haverá disparo no webhook.
JSON Header WebhookProto	Headers adicionais a ser incluídos na requisição do hook opcional em formato JSON.
Enviar Protohook	Habilita/desabilita o envio do hook opcional.
Agendado para	Horário de agendamento geral, quando os perfis irão iniciar suas execuções.

Criando um Perfil

Criando um Perfil de Execução de Intimações

Após realizar as configurações acima, os perfis podem ser criados em **Perfis**->**Criar novo perfil**



Formulário de novo perfil.

CAMPO	DESCRIÇÃO
Nome da execução	Prefixo que será atribuído as execuções que forem criadas a partir desse perfil. **ATENÇÃO:** Evite usar espaços, caracteres especiais e símbolos.
Nome da automação	Nome que será atribuído ao perfil/automação que está sendo criado. **ATENÇÃO:** Evite usar espaços, caracteres especiais e símbolos.
Executar nos dias	Dias da semana onde o perfil deve ser executado.
Selecione o tribunal	Configurações do robô e credenciais.
Rotina	Filtro de data que será aplicado às intimações listadas. **Veja na tabela a baixo** como estes valores se aplicam.
Horário limite	Com essa configuração, o App não iniciará perfis após o horário determinado. O valor padrão é 21h00 do horário de Brasília.
Formatação dos campos de data	Com essa configuração, o App aplicará um processamento no registro do robô para definir um formato padronizado. A formatação é aplicada aos campos `postDate`, `date`, `assignment`, `periodStart` e `periodEnd`. Caso não especificado, será aplicada a máscara `DD/MM/YYYY` por padrão nos campos que puderem ser identificados.

ROTINA	DESCRIÇÃO
Últimos 3 dias	Apenas as intimações postadas nos últimos 3 dias serão filtradas (selecionadas). Por exemplo, no dia 10/11/2021, apenas as intimações postadas no dia 09/11/2021, 08/11/2021 e 07/11/2021 serão filtradas.
Ontem	Apenas as intimações postadas no dia anterior (ontem) serão filtradas. Por exemplo, no dia 10/11/2021, apenas as intimações postadas no dia 09/11/2021 serão filtradas.

ROTINA	DESCRIÇÃO
Ontem e antes	Apenas as intimações postadas nos dias posteriores (ontem para trás) serão filtradas. Por exemplo, no dia 10/11/2021, as intimações postadas no dia 09/11/2021 para trás serão filtradas.
Últimos X dias	Apenas as intimações postadas nos últimos X dias serão selecionadas. Por exemplo, no dia 10/11/2021 com X = 2, apenas as intimações postadas nos dias 09/11/2021 e 08/11/2021 serão filtradas.
X dias e antes	Apenas as intimações postadas há X dias ou mais serão selecionadas. Por exemplo, no dia 10/11/2021 com X = 2, apenas as intimações postadas antes do dia 08/11/2021 serão filtradas.
Todos	O filtro irá pegar tudo, não levando em consideração a data da postagem.

A inicialização do robô pode atrasar em relação ao horário agendado em situações onde uma grande quantidade de perfis está agendada para o mesmo horário ou perfis iniciados anteriormente ainda não tenham finalizado a execução. O disparo dos perfis ocorre num modelo de fila por cliente, sendo executados no máximo 2 perfis por vez.

Webhooks

Um webhook é uma requisição `HTTP` enviada automaticamente quando um evento acontece. Isso permite que o próprio servidor notifique o cliente do evento em vez de o cliente perguntar repetidamente. No App de intimações, webhooks são usados para notificar o cliente em dois momentos: após a filtragem da lista de intimações e após o término do processamento dos resultados.

Para receber webhooks do App de intimações é necessário disponibilizar um endpoint para receber os disparos, que são requisições `POST`.

Webhook Principal (conteúdo das intimações)

Para o hook principal (retorno das intimações), o `body` terá o seguinte formato:

```
{
   "configName": "nome-perfil",
   "executionId": "id-execucao",
   "bot": "pje-tj-intimation",
   "credentials": {
      "username": "username",
      "credentialType": "password"
   },
   "resultEndpoint": "/results/id-execucao"
}
```

CAMPO	DESCRIÇÃO

configName	Nome da configuração.
executionId	ID da execução.
bot	ID do robô para conferência no painel da Oystr.
credentials	Resumo das credenciais utilizadas.
resultEndpoint	Endpoint para consulta dos resultados.

O endpoint deve responder com um dos status da lista `200`, `201`, `202`, `203`, `204` em caso de sucesso. Em caso de falha, por qualquer razão, serão realizadas mais 9 tentativas com 30 minutos de espera entre elas. Caso seja preciso reenviar algum webhook basta entrar em contato com o suporte.

É importante verificar se o endpoint é acessível pelo servidor do App, é comum que requisições partindo de faixas de IP de datacenter sejam bloqueadas.

Webhook Opcional (listagem das intimações)

Já para o opcional (listagem das intimações), o **body** terá o formato:

```
{
   "configName": "nome-perfil",
   "bot": "projudi-intimacao",
   "credentials": {
      "username": "username",
      "credentialType": "password"
   },
   "total": 1,
   "filtered": 1,
   "remaining": 0,
   "filteredItems": [{
      "valid": true,
      "data": [...],
      "files": null
   }],
   "remainingItems": [{
      "valid": true,
      "data": [...],
      "files": null
   }]
}
```

CAMPO	DESCRIÇÃO
configName	Nome da configuração
bot	ID do robô para conferência no painel da Oystr
credentials	Resumo das credenciais utilizadas
filteredItems	Itens filtrados de acordo com o filtro selecionado na configuração do perfil. Esses são os itens que serão executados pela automação
remainingItems	Itens remanescentes após o filtro ser aplicado. Esses itens não serão executados pela automação

O endpoint deve responder com um dos status da lista `200`, `201`, `202`, `203`, `204` em caso de sucesso. Em caso de falha, por qualquer razão, serão realizadas mais 9 tentativas com 30 minutos de espera entre elas. Caso seja preciso reenviar algum webhook basta entrar em contato com o suporte.

É importante verificar se o endpoint é acessível pelo servidor do App, é comum que requisições partindo de faixas de IP de datacenter sejam bloqueadas.

Consulta dos resultados

```
GET: https://console4.oystr.com.br/api/v1/service/intimacoes/results/id-execucao
```

* **Lembrando que é necessário fazer essa requisição autenticada com o header X-Oystr-Auth**, veja neste link como fazer.

Ao consultar os resultados através do endpoint enviado no webhook, o retorno esperado é o seguinte:

```
{
   credentialsOption: "UF",
   data: {
      entries: [
         [{
            "offset": [posicao do item],
            "sent": [indicador de o item foi processado],
            "received": [indicador se o item teve resposta],
            "type": [TIPO DA RESPOSTA],
            "start": [data em millis],
```

```
            "duration": [timestamp],
            "request": {
               /** DADOS DE ENTRADA DO ROBÔ **/
            },
            "response": {
               /** DADOS DE SAÍDA DO ROBÔ **/
            }
         } /** ... N entradas **/ ]
      ],
   }
}
```

"type": "RES"

Abaixo o modelo dos dados contendo uma intimação:

```
{
   credentialsOption: "UF",
   data:
   {
      entries:
      [
         [{
            "offset": 0,
            "sent": true,
            "received": true,
            "type": "RES",
            "start": 1613484436618,
            "duration": 19831,
            "request":
            {
               "id": "[string]",
               "integration": "[string]",
               "process": "[número do processo]",
               "type": "[string]",
               "withText": "1",
               "removeDigitalCopy": "1",
               "grau": "[string]",
               "url": "[string]",
```

```
    "date": "[data da intimação]",
    "readerDate": "[string]",
    "tribunal": "[string]",
    "party": "[parte intimada]",
    "event": "[evento/tipo do movimento]",
    "postDate": "[string]",
    "lastDate": "[string]",
    "typeReader": "01-1",
    "owner": "[id]",
    "codAdv": "[advogado]",
    "autor": "[string]",
    "reu": "[string]",
    "idPendencia": "[string]",
    "codigoPendencia": "[string]",
    "abrirIntimacao": "[string]",
    "dataHora": "[string]",
    "erroComarca": "[string]",
    "meioComunicacao": "[string]",
    "leitor": "[string]"
},
"response":
{
    "notices":
    [{
        "id":"",                        // Identificação única no site do tribunal, se houver
        "processNumber":"[string]",     // Número do processo
        "origin":"[string]",            // Diário Eletronico de origem
        "author":"[string]",            // Autor
        "defendant":"[string]",         // Réu
        "postDate":"[string]",          // Data de postagem
        "automatic":"[string]",         // Data de leitura automatica
        "date":"[string]",              // Data da intimação
        "period":"[string]",            // prazo
        "processType":"[string]",       // Classe processual
        "periodStart":"[string]",       // Primeiro dia de prazo
        "periodEnd":"[string]",         // Último dia de prazo
        "type":"[string]",              // Tipo do diário
        "assignment":"[string]",        // Data da distribuição
        "personal":[boolean],           // Pessoal
        "juizo":"[string]",             // Juízo
```

```
                "fulfillmentDate":"[string]",   // Data de cumprimento
                "reader":""[string],            // Leitor
                "status":"[string]",            // Situação/Status
                "document":"[string]",          // Título documento da publicação
                "update":"[string]",            // Movimento/Publicação
                "lawyer":"[string]",            // Advogado
                "link":"[string]",              // Link
                "attachments":                  // Array de anexos,
                [{
                    "error":false,          // Indicação de erro
                    "name":"file-vWz.pdf",  // Nome do arquivo
                    "data":"[string]",      // Conteúdo do arquivo em base64
                    "text":"[string]",      // Texto extraído do arquivo
                    "description":null      // Descrição do arquivo
                }],
                "singleFile":null      // Uso interno, ignorar
            }],
            "total": 1,
            "withText": "1",
            "notFound": false,
            "error": false,
            "errorMessage": null,
            "exp": null
        }
      }]
    ],
  }
}
```


CAMPO	DESCRIÇÃO
data->entries[][]	Array contendo todos os itens de intimação que o robô tentou capturar, com todas as informações para recuperar a entrada (intimação a capturar) e saída do robô (resultado)

CAMPO	DESCRIÇÃO
data->entries[x][x]->type	Tipo da resposta que estará contido no objeto response deste item. Para cada tipo, campos diferentes serão trazidos: - RES - Resposta padrão com os dados de uma intimaçao. **Todos os outros tipos de resposta representam erros (e subsequente ausência do objeto `response` ****, sendo os mais comuns ERR e REQ*** - ERR - Resposta com erro, onde o robô não conseguiu capturar a intimação. Neste caso o "response" será nulo. Existirá um objeto "error" com as informações tecnicas sobre o erro. - REQ - Item sem resposta por parte do robô, neste caso o objeto `response` será nulo.

Modelo de objeto resposta com erros/exceções

IMPORTANTE: Estes cenários onde a resposta não é **RES** podem, ou não, representar problemas para o cliente dependendo do erro e da configuração dos seus perfis (se houve ciência dada na intimação, qual configuração de filtro e agendamento o perfil possui, se mais perfis podem vir a abrir a intimação, etc.), **sendo recomendada a verificação com o suporte para determinar se a situação representa algum risco de perda de prazo**. Dentro do objeto *request* você sempre irá encontrar as informações preliminares da intimação, sendo possível assim saber qual foi o processo, evento e data da intimação com erro no robô.

"type": "ERR"

Quando o type for "**ERR**", o robô encontrou um erro ao tentar ler/abrir a intimação indicada. **Veja que o retorno não terá o objeto `response` mas sim terá um objeto `error`**.


```
{
  "offset": 10,
  "sent": true,
  "received": true,
  "type": "ERR",
  "start": 1626715329826,
  "duration": 25537,
  "request": {
    "id": null,
    "integration": null,
    "process": "[número do processo]",
    "type": "novas",
```

```
      "withText": "1",
      "removeDigitalCopy": "1",
      "grau": "primeiro-grau",
      "url": null,
      "date": "[data da intimação]",
      "readerDate": null,
      "tribunal": "JFRS",
      "party": "[parte intimada]",
      "event": "Expedida/certificada a intimação eletrônica",
      "postDate": "Abrir Prazo",
      "lastDate": "",
      "typeReader": "01-1",
      "owner": "[id]",
      "codAdv": "[advogado]",
      "autor": null,
      "reu": null,
      "idPendencia": null,
      "codigoPendencia": null,
      "abrirIntimacao": null,
      "dataHora": null,
      "erroComarca": null,
      "meioComunicacao": null,
      "leitor": null
    },
    "error": {
      "message": "curl error: 16",
      "stackTrace": "xingu.http.client.HttpException: curl error: 16\n\tat
xingu.http.client.impl.SimpleHttpRequest.exec(SimpleHttpRequest.java:63)\n\tat
xingu.http.client.impl.StatefullRequest.exec(StatefullRequest.java:35)\n\tat
oystr.eproc.intimacao.controllers.EprocController.pesquisarProcesso(EprocController.java:283)\n\tat
oystr.eproc.intimacao.controllers.EprocPendentesWithoutReaderController.execute(EprocPendentesWithoutRead
erController.java:57)\n\tat oystr.eproc.intimacao.EprocIntimacaoBot.execute(EprocIntimacaoBot.java:152)\n\tat
oystr.eproc.intimacao.EprocIntimacaoBot.execute(EprocIntimacaoBot.java:37)\n\tat
oystr.broker.client.MessageHandler.onRequest(MessageHandler.java:233)\n\tat
oystr.broker.client.MessageHandler.processAsync(MessageHandler.java:139)\n\tat
oystr.broker.client.MessageHandler.access$000(MessageHandler.java:50)\n\tat
oystr.broker.client.MessageHandler$1.run(MessageHandler.java:95)\n\tat
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)\n\tat
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)\n\tat
java.lang.Thread.run(Thread.java:748)\n"
```

```
  }
}
```

"type": "REQ"

Quando o type for "**REQ**", o robô não executou o item em questão. Estes casos server para apontar problemas de configuração ou comunicação com o tribunal. Caso esse problema seja recorrente, indica-se que seja verificado a situação da configuração/perfil no App para evitar perda de prazos. **Veja que o retorno não terá o objeto `response` e que o valor de `received` é `false`.**


```
{
  "offset": 8,
  "sent": true,
  "received": false,
  "type": "REQ",
  "start": 1626775895441,
  "request": {
    "id": null,
    "integration": null,
    "process": "[número do processo]",
    "type": "novas",
    "withText": "1",
    "removeDigitalCopy": "1",
    "grau": "primeiro-grau",
    "url": "[url...]",
    "date": "[data da intimação]",
    "readerDate": null,
    "tribunal": "PR",
    "party": "[parte intimada]",
    "event": "JUNTADA DE ATO ORDINATÓRIO (10 dias úteis)",
    "postDate": "-",
    "lastDate": "-",
    "typeReader": "novas",
    "owner": null,
    "codAdv": null,
    "autor": null,
    "reu": null,
    "idPendencia": null,
    "codigoPendencia": null,
```

```
    "abrirIntimacao": null,
    "dataHora": null,
    "erroComarca": null,
    "meioComunicacao": null,
    "leitor": null
  }
}
```


Outras observações

- Dependendo das configurações dos perfis do cliente pode haver a abertura da mesma intimação múltiplas vezes. O cliente pode acionar o suporte para fazer a identificação da(s) causa(s) e as mudanças necessárias mas, caso seja necessária a implementação de um mecanismo de controle por parte da integradora, recomendamos que a intimação seja identificada através do hash de certas informações do elemento do array `notices`.
- O registro da execução do robô é sempre disponibilizado para integração, incluindo o envio do webhook, ainda que a execução tenha 100% de erro. A integradora pode notificar o cliente das ocorrências de erro (recomendamos a inclusão do id de execução na notificação, caso ocorra) mas o ideal é que o cliente acompanhe o painel da Oystr com frequência para o acompanhamento desses casos e acionamento do suporte caso necessário.
- Os registros das execuções ficam disponíveis mesmo após a consulta. O App de intimações armazena o status de envio e consulta dos registros mas **não** impede o acesso após o registro da consulta. Caso um registro esteja indisponível ele pode ter sido movido para armazenamento frio e podemos recuperar esses registros caso necessário.

Novo formato de anexos (beta)

Devido à inclusão de anexos inteiros em base64 no documento de integração o tamanho do arquivo pode atingir valores exorbitantes. Para contornar esse problema, um novo formato de anexos foi implementado em que são fornecidos endpoints para download posterior dos documentos, permitindo separar o processamento desses itens para um momento oportuno.

Todos os documentos seguem tendo o mesmo formato mas com pequenas mudanças. O conteúdo do webhook passará a ter as seguintes informações:

```
{
   "configName": "nome-perfil",
   "executionId": "id-execucao",
   "bot": "pje-tj-intimation",
   "credentials": {
      "username": "username",
      "credentialType": "password"
   },
   "resultEndpoint": "/results/id-execucao", // Ainda contém o arquivo no modelo anterior sem nenhuma mudança
   "testingResultEndpoint": "/results/testing-id-execucao" // Campo temporário com o journal no novo modelo, mais adiante passaremos a disponibilizar em resultEndpoint
}
```

A integradora pode então realizar a requisição do documento no endpoint em `testingResultEndpoint`. Caso haja qualquer problema, o arquivo no modelo anterior estará disponível no endpoint em `resultEndpoint`. Já os anexos em `attachments` passarão a ter o seguinte modelo:

```
[{
   "type": "link|raw",      // será "raw" caso o arquivo esteja presente em base64 no campo "data"; ou "link" caso o endpoint de download esteja presente no campo "link"
   "error":false,           // Indicação de erro
   "name":"file-vWz.pdf",   // Nome final do arquivo (pode ser diferente da extensão indicada no link pois, caso sejam arquivos html, convertidos para pdf antes do download)
   "data":"[string]|null",  // Conteúdo do arquivo em base64, somente presente caso "type" seja "raw", do contrário será null
   "text":"[string]",       // Texto extraído do arquivo
   "link": "/document/id-execucao/0/file-vWz.html|null", // Endpoint de download, somente presente caso "type" seja "link", do contrário será null
   "description":null       // Descrição do arquivo
}],
```

Caso o campo `type` seja `raw`, o anexo é compatível com o formato anterior e o documento completo estará em `data` em base64. Caso seja `link`, `data` será `null` e o documento em si deverá ser obtido posteriormente no endpoint em `link`.

Revision #3
Created 25 March 2024 03:05:49 by Luan Melo
Updated 15 April 2026 17:14:12 by Luan Melo