

Inclusão de Solicitação

Last updated by | Tiago Souza < tiago.souza@simply.com.br > | Mon, 19 Sep 2022 11:23:24 GMT

Inclusão de documentos

Utilize esse procedimento para enviar documentos ao Atomics.

POST no endpoint:

```
https://atomics-api.simply.com.br/api/v1/tipificacao
```

Esse *endpoint* irá tipificar todos os documentos enviados, de acordo com configurações, mapeamentos, parâmetros e perfis específicos de cada cliente.

Dados de Entrada

Realizar um **POST** informando na chave `Authorization` do header , o *token* recebido no [login](#) ↗

```
{
  "Authorization": "Bearer {token}"
}
```

Para enviar os documentos alguns parâmetros de entrada são necessários, sendo o **CodigoLegado** e **Arquivo** obrigatórios.

Parâmetro	Descrição	Tipo	
CodigoLegado	Campo que poderá ser utilizado para identificar a identidade do sistema legado.	string	Obrigatorio
Arquivo	Deve conter os arquivos que serão tipificados. Pelo menos um arquivo deve ser enviado para que a solicitação seja criada.	bytes	Obrigatorio

```
{
  "CodigoLegado": "{codigo}"
  "Arquivo": "{arquivos}"
}
```

Exemplo de retorno de sucesso

```
{
  99999999
}
```

Quando o envio é realizado com sucesso, você receberá um código referente ao número da solicitação, esse código devere ser utilizado posteriormente para consulta ao retorno da tipificação

Exemplo de retorno de erro

```
{
  "Message": "The request is invalid.",
  "ModelState": {
    "": [
      "O código da proposta no sistema legado deve ser informado."
    ]
  }
}
```

```
{
  "Message": "The request is invalid.",
  "ModelState": {
    "": [
      "Pelo menos um arquivo deve ser informado."
    ]
  }
}
```

Exemplo de Código (C#)

```
public static async Task IncluiSolicitacao(string[] arquivos, string Legado = "55")
{
    try
    {
        if (BearerToken == null)
            throw new Exception("Token não foi recuperado. Utilize o método de login.");
        using (var client = new HttpClient(new HttpClientHandler {
            AutomaticDecompression =
                DecompressionMethods.Deflate | DecompressionMethods.GZip}))
        {
            client.BaseAddress = new Uri(UrlBase);
            MultipartFormDataContent form = new MultipartFormDataContent();
            var camposEnviados = new Dictionary();

            //cabeçalho de autorização
            client.DefaultRequestHeaders.Add("Authorization", "Bearer " + BearerToken);

            //adiciona campos de entrada e
            form.Add(new StringContent(codigoLegado), "CodigoLegado");

            foreach (string arquivo in arquivos)
            {
                byte[] dados = File.ReadAllBytes(arquivo);
                form.Add(new ByteArrayContent(dados, 0, dados.Length),
                    "arquivo", Path.GetFileName(arquivo));
            }

            HttpResponseMessage resposta;
            resposta = await client.PostAsync("api/v1/tipificacao", form);

            //recupera código da solicitação
            resposta.EnsureSuccessStatusCode();
            string codigoSolicitacao = await resposta.Content.ReadAsStringAsync();
        }
    }
    catch (Exception e)
    {
        //Sua excessão
    }
}
```

Exemplo de Código (PHP)

```
$curlPost = curl_init();

$arqs = array();

$arqs[] = (new cURLFile("PATH ARQUIVO 1", "mime", "Arquivo 1"));
$arqs[] = (new cURLFile("PATH ARQUIVO 2", "mime", "Arquivo 2"));

curl_setopt_array($curlPost, array(

    CURLOPT_URL => "URL_BASE",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 0,
    CURLOPT_FOLLOWLOCATION => true,
    CURLOPT_SSL_VERIFYPEER => false,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_POSTFIELDS => array(
        'CodigoLegado' => "55555",
        'arquivo[0]' => $arqs[0],
        'arquivo[1]' => $arqs[1]
    ),
    CURLOPT_HTTPHEADER => array(
        "Authorization: Bearer {$token_simplify__}",
    ),
));

$responsePost = curl_exec($curlPost);

curl_close($curlPost);

var_dump($responsePost);
```

Exemplo de Código (Node)

```
var fs = require('fs');
var options = {
  'method': 'POST',
  'url': 'URL_BASE',
  'headers': {
    'Authorization': 'Bearer ' + tokenJSON.access_token,
    'Content-Type': 'multipart/form-data; boundary=123'
  },
  formData: {
    'CodigoLegado': '55555',
    'arquivo[0]': {
      'value': fs.createReadStream('PATH ARQUIVO 1'),
      'options': {
        'filename': 'arquivo1.jpg',
        'contentType': null
      }
    },
    'arquivo[1]': {
      'value': fs.createReadStream('PATH ARQUIVO 2'),
      'options': {
        'filename': 'arquivo2.jpg',
        'contentType': null
      }
    },
    'PerfilTipificacao': 'XXXXX',
    'ExtrairDados': 'true'
  }
};

request(options, function(error, response) {
  if (error) throw new Error(error);
  console.log(response.body);
});
```