

## Passo 1 – Envio do pacote de acesso

O anexo é um arquivo compactado (.zip) chamado "Pacote de acesso \*". Ele contém os arquivos:

- #.cer, #.key, #.p12 e #\_senha\_p12.txt (certificados digitais e senha)
- #\_client\_id\_secret.txt (client\_id e client\_secret)

(# se refere ao atributo "documento" informado na requisição e \* se refere ao atributo "nome" informado na requisição).

## Passo 2 - Obter token de autorização

Para obter o token de autorização à API, precisamos do client\_id e client\_secret que foram obtidos no passo anterior e consumir a seguinte API:

```
POST /aa5ac705-873b-4afc-a29d-f0adb89ccf5c/oauth2/v2.0/token HTTP/1.1
Host: login.microsoftonline.com
Content-Type: application/x-www-form-urlencoded
grant_type=client_credentials&client_id=0c991613-4d90-454d-8685-
d466a47669cb&client_secret=3OPOIdg6KDMeWUE-hLf0_b5T6__VFe82-u&scope=d659d836-2622-
4973-a3ce-e4b8a8af757f%2F.default
```

Onde:

Atributo	Tipo	Obrigatório	Pode ser vazio/nulo?	Restrições
"grant_type"	String	Sim	Não	Valor fixo "client_credentials"
"client_id"	String	Sim	Não	String "Client_ID" recebida no arquivo #_client_id_secret.txt
"client_secret"	String	Sim	Não	String "Secret" recebida no arquivo #_client_id_secret.txt
"scope"	String	Sim	Não	Valor fixo: <b>"d659d836-2622-4973-a3ce-e4b8a8af757f/.default"</b> .  Observação: <ul style="list-style-type: none"> <li>• Se o Content-Type for "application/x-www-form-urlencoded" então a "/" deve ser substituída por "%2F", isto é: <b>"d659d836-2622-4973-a3ce-e4b8a8af757f%2F.default"</b>.</li> <li>• Se o Content-Type for "multipart/form-data" então mantém o valor fixo original:</li> </ul>



Para consumir a API de validação vamos precisar apresentar do conjunto de certificados e senha obtidos no passo 1 e o token de acesso, obtido no passo anterior (atributo "access\_token").

O token deverá ser apresentado num cabeçalho "Authorization", com conteúdo "Bearer + espaço + access\_token".

```
GET /api/acesso/healthcheck HTTP/1.1
Host: investidor.b3.com.br:2443
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS1hFZyJ9.eyJhdWQiOiI5OGRkZjRiMC1mNjZkLTRjOTYtOTdlYS05ZTMwMzA2NTk5ZTciLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20vNGJlZTYzOWYtNTM4OC00NGM3LWJiYWVtY2I5MmE5MzIxMmU2L3YyLjAiLCJpYXQiOiJlY2Mjc2NTc4MTcslm5iZiI6MTYyNzY1NzgxNywiZXhwIjoxNjM3NjYxNzE3LCJhaW8iOiJFbmlpbnVUvol3FybGpvZkZ6NnczTEV4TmwxWFIRqVFBPSlsmF6cCI6IjBjOTkxNjEzLTRjOTAtNDU0ZC04Njg1LWQ0NjZhNDC2NjIjYiImlmF6cGFjciI6IjEiLCJvaWQiOiI0NTg5NDEzOC1jMmE2LTQ2MjEjEtYWUxNi1kNDgyYzI0OTcxYzkiLCJyaCI6IjAuQVJlY2M4YUUhVTNGhUeDBTN3JNdVNXVGtSNWhNV21ReVFURTFGaG9YVWpxUjJhY3NWQUFBLiIsInN1YiI6IjQ1ODk0MTM4LWMyYTYtNDYyMS1hZTE2LWQ0ODJmZg5NzFjOSIsInRpZCI6IjRiZWU2MzlmLTUzODgtNDRjNy1iYmFjLWNIOTJhOTM5MTEFNiIsInV0aSI6Ikhkb2xvbiE2tc2tITEF5a1pjcXcwQUeILCJ2ZXliOiIyLjAifQ.B-5n4jm3WaEdxoPt-iZ5A4FCWH5IyNaKdn8rTDgOILqr7PAxTevxmOSy4IEpfL3bEpGY78C2TBt1qWAMoJmUfHp5FMxjHj035N4SjgBVQTKiA1cFSeirlb1w7SKtJLyIC8ivT0srijdja4PVLdqDkQhd4qAtLtCnNEI7d7XjSRhy0ZCBTh-loa8GSvWFJqz5c_bTnK80IL_4LxGWNLiXfN6elgm2ISJsHP0PtI7JTty4HD3TrCbO6o6BpG4-twQQi2KcWJNkpdRsXkUnkC-M892MI42KgZ0f_pV0hmgisrx9FPf5Q9QKflqt9LrLABc8IQLhy5wZPYB3ZO5PGg1g'
```

A apresentação de certificado de cliente (Mutual TLS 1.2 Authentication) pode ser implementada de várias formas, de acordo com a linguagem de sua aplicação (JAVA, C#, Python etc) ou software de requisição HTTP (Postman, JMETER, cURL etc) que esteja usando. Para fins didáticos, mostraremos como executar essa requisição apresentando certificado de cliente utilizando cURL de uma máquina Linux:

```
curl \
--cert #.cer:SENHA \
--key #.key \
--k --request GET 'https://investidor.b3.com.br:2443/api/acesso/healthcheck' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS1hFZyJ9.eyJhdWQiOiI5OGRkZjRiMC1mNjZkLTRjOTYtOTdlYS05ZTMwMzA2NTk5ZTciLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20vNGJlZTYzOWYtNTM4OC00NGM3LWJiYWVtY2I5MmE5MzIxMmU2L3YyLjAiLCJpYXQiOiJlY2Mjc2NTc4MTcslm5iZiI6MTYyNzY1NzgxNywiZXhwIjoxNjM3NjYxNzE3LCJhaW8iOiJFbmlpbnVUvol3FybGpvZkZ6NnczTEV4TmwxWFIRqVFBPSlsmF6cCI6IjBjOTkxNjEzLTRjOTAtNDU0ZC04Njg1LWQ0NjZhNDC2NjIjYiImlmF6cGFjciI6IjEiLCJvaWQiOiI0NTg5NDEzOC1jMmE2LTQ2MjEjEtYWUxNi1kNDgyYzI0OTcxYzkiLCJyaCI6IjAuQVJlY2M4YUUhVTNGhUeDBTN3JNdVNXVGtSNWhNV21ReVFURTFGaG9YVWpxUjJhY3NWQUFBLiIsInN1YiI6IjQ1ODk0MTM4LWMyYTYtNDYyMS1hZTE2LWQ0ODJmZg5NzFjOSIsInRpZCI6IjRiZWU2MzlmLTUzODgtNDRjNy1iYmFjLWNIOTJhOTM5MTEFNiIsInV0aSI6Ikhkb2xvbiE2tc2tITEF5a1pjcXcwQUeILCJ2ZXliOiIyLjAifQ.B-5n4jm3WaEdxoPt-iZ5A4FCWH5IyNaKdn8rTDgOILqr7PAxTevxmOSy4IEpfL3bEpGY78C2TBt1qWAMoJmUfHp5FMxjHj035N4SjgBVQTKiA1cFSeirlb1w7SKtJLyIC8ivT0srijdja4PVLdqDkQhd4qAtLtCnNEI7d7XjSRhy0ZCBTh-loa8GSvWFJqz5c_bTnK80IL_4LxGWNLiXfN6elgm2ISJsHP0PtI7JTty4HD3TrCbO6o6BpG4-twQQi2KcWJNkpdRsXkUnkC-M892MI42KgZ0f_pV0hmgisrx9FPf5Q9QKflqt9LrLABc8IQLhy5wZPYB3ZO5PGg1g'
```

Onde:

```
#.cer = Arquivo obtido no passo 1, dentro "Pacote de acesso *.zip".
#.key = Arquivo obtido no passo 1, dentro "Pacote de acesso *.zip".
SENHA = String obtida no passo 1, dentro do arquivo #_senha_p12.txt que está no "Pacote de acesso *.zip"
```

O retorno da API será:

<b>HTTP Status</b>	200 OK
<b>Corpo</b>	{ "status": "Sucesso", "mensagem": "Autenticação e autorização do usuário # validado com sucesso" }

\* atributo "nome" apresentado na requisição do passo 1.

# atributo "documento" apresentado na requisição do passo 1.

## Passo 4 – Configurações e Credenciais do Custodiante.

Para que o B3I possa obter o token de autorização à API do custodiante, o custodiante deve disponibilizar um serviço que retorne um token JWT válido para a requisição da chamada descrita no passo 5.

O "client\_id" e "client\_secret" deverá estar no corpo (body) da mensagem do tipo "POST" com os nomes de "cliente\_id" e "cliente\_secret". Também será necessário que o custodiante configure o certificado Mutual TLS 1.2.

Sendo assim, se faz necessário que nos informem os dados abaixo, assim que disponíveis:

- URL para o B3 Investidor obter o token;
- Client Id e Client Secret para obter o token\*;

\*sem utilização de qualquer tipo de encoding para mascaramento das informações de client id e secret id a serem enviadas na obtenção do token.

- Certificado para configuração do Mutual TLS 1.2 (padrão **.pfx** + senha, no formato **PKCS #12**);

Atenção: Esta integração não aceita o certificado em outro formato que não seja em **.pfx**

- URL do serviço para que o B3 Investidor envie a autorização.

formato: "{host\_custodiante}/api/request-authorizations/v1/investors"

Sendo:

{host\_custodiante} = Host do serviço do custodiante

Exemplo:

Para {host\_custodiante} = "https://exemplo.com.br"

A URL seria: "https://exemplo.com.br/api/request-authorizations/v1/investors"

Segue um exemplo de chamada e como os campos devem ser preenchidos:

```
curl POST 'https://{URL_Custodiante}\  
--header 'Content-Type: application/x-www-form-urlencoded' \  
--data-urlencode 'grant_type=client_credentials' \  
--data-urlencode 'client_id=0c991613-4d90-454d-8685-d466a47669cb' \  
'
```



	M892MI42KGzOf_pV0hmgjsrx9FPf5Q9QKflqt9LrLABc8lQLhy5wZPYB3ZO5PGg1g"}
--	---

Onde:

Atributo	Restrição
"token_type"	Exibe o tipo de token de segurança que foi emitido. Sempre será "Bearer".
"expires_in"	"ext_expires_in": Tempo de validade do token em segundos. Depois desse tempo NÃO será mais válido, sendo necessária a geração de um novo token. Sempre será "3599" segundos (1 hora).
"access_token"	Token de autorização. Que Será utilizado pelo B3I no cabeçalho da requisição da API do passo 5.

## Passo 5 – Receber Autenticação de Portabilidade do Investidor.

Para receber a autenticação de portabilidade do investidor (serviço fornecido pelo custodiante de origem a ser consumido pelo B3 Investidor), a rota deve estar no seguinte formato: [url\\_custodiante/api/request-authorizations/v1/investors](url_custodiante/api/request-authorizations/v1/investors).

A seguir, os dados transmitidos (payload) que o B3 Investidor envia para o custodiante.

Quadro 1 – Dados que o B3i envia para o custodiante
<pre>{   "data":   {     "requestTransactionId": "B3i-202103010001",     "transactionTypeCode": 1,     "investorName": "André",     "accountNumber": 12345,     "document":     {       "documentNumber": "99999999999",       "documentTypeName": "CPF"     },     "transactionDateTime": "2021-10-21T14:00:00.001Z",     "transactionExpirationDateTime": "2021-10-23T14:00:00.001Z",   } }</pre>

```

"custodianCode": "9999",
"custodianCodeDestination": "0000",
"assets":
[
{
"assetCode": "PETR4",
"assetQuantity": 5000,
"eventValue": 150.5
}
]
}
}

```

Onde:

Atributo	Tipo	Descrição
"requestTransactionId"	String	Código de transação gerado pelo B3 Investidor.
"transactionTypeCode"	Integer	Código do tipo de transação. Para STVM será sempre 1.
"investorName"	String	Nome do investidor.
"accountNumber"	String	Conta de origem escolhida para a transferência.
"documentNumber"	String	Documento do investidor.
"documentTypeName"	String	Documento de identificação do investidor:
"transactionDateTime"	String	Data e hora em que a transação ocorreu. String com data e hora de acordo com a especificação RFC-3339, sempre utilizando o fuso horário UTC (formato de hora UTC).
"transactionExpirationDateTime"	String	Data e hora de expiração da transação.
"custodianCode"	String	Código de identificação do custodiante de origem.
"custodianCodeDestination"	String	Código de identificação do custodiante de destino.
"assetCode"	String	Código do ativo.

Atributo	Tipo	Descrição
"assetQuantity"	Decimal	Quantidade do ativo a transferir.
"eventValue"	Decimal	Valor do evento provisionado a transferir.  <b>Observação:</b> Se o ativo for um evento provisionado, enviamos o atributo <b>eventValue</b> . Caso contrário, enviamos o atributo <b>assetQuantity</b> .

Em caso de sucesso retornar o objeto enviado no quadro 1.

## Passo 6 – Confirmar/Negar Autenticação de Portabilidade do Investidor.

Para confirmar ou negar a autenticação de portabilidade do investidor (serviço fornecido pela B3 a ser consumido pelo custodiante de origem) utilizar o serviço: <https://investidor.b3.com.br:2443/api/response-authorizations/v1/tokens>

**Importante:** para esta requisição é necessário enviar o token de autorização obtido no passo 3.

A seguir, os dados transmitidos (payload) que o B3 Investidor espera receber do custodiante.

### Quadro 2 - Autorizar Portabilidade

```
{
  "data":
  {
    "requesterTransactionId": "B3i-202103010001",
    "requestedTransactionId": "BTG-909103090009",
    "documentNumber": "99999999999",
    "authorizationToken": "155082098e134fa14279e7baa4cf8270",
    "transactionDateTime": "2021-03-30T14:00:00.001Z"
  },
  "errors":
  [
  ]
}
```

### Quadro 3 - Negar Portabilidade

```
{
  "data":
  {
```



```

"requesterTransactionId": "B3i-202103010001",
"requestedTransactionId": "BTG-909103090009",
"documentNumber": "99999999999",
"authorizationToken": "155082098e134fa14279e7baa4cf8270",
"transactionDateTime": "2021-03-30T14:00:00.001Z"
},
"errors":
[
{
"code": "0001",
"title": "Não autorizado" ,
"detail": "Investidor não autorizou a portabilidade."
}
]
}

```

Onde:

Atributo	Tipo	Descrição
"requesterTransactionId"	String	Código de transação enviado pelo B3 Investidor.
"requestedTransactionId"	String	Código de transação gerado pelo custodiante.
"documentNumber"	String	Documento do investidor enviado pelo B3 Investidor.
"authorizationToken"	String	Token de autorização da portabilidade de ativos.
"transactionDateTime"	String	Data e hora da resposta.
"code"	String	Código do erro.
"title"	String	Breve mensagem do erro.
"detail"	String	Informações complementares sobre erro.

#### Observações:

- O atributo **"erros"** deve ser preenchido somente se o investidor **negar** a portabilidade. Caso contrário, enviá-lo vazio, exemplo: "error": [].
- Considerar o **sucesso** da confirmação somente se o B3 Investidor retornar o **status 200**. Caso contrário, apresentar uma mensagem para o usuário e reenviar a confirmação.