

Passo 1 - Obter o pacote de acesso

O *endpoint* da API de negócio implementa *Mutual SSL Authentication* (também conhecido como *Two-Way SSL Authentication*) para autenticar e OAuth 2.0 para autorizar sua aplicação.

Seguindo os passos a seguir é possível validar a comunicação entre a sua ponta e a B3.

Para implementar o padrão de segurança você irá precisar dos certificados digitais e senha, necessários na implementação do *Mutual SSL Authentication* e do *client_id* e *client_secret* (duas *strings* que funcionam como usuário e senha) para obtenção do token de autorização. Esse conjunto de arquivos será chamado nesse tutorial de "pacote de acesso". O pacote de acesso te identifica unicamente na B3, ele é seu passaporte e deve ser armazenado com segurança. Alguém com acesso a esses arquivos pode se passar por você.

No ambiente de CERTIFICAÇÃO você poderá obter o pacote de acessos através da API abaixo:

```
POST /api/acesso/autoservico/stvm HTTP/1.1
Host: apib3i-cert.b3.com.br
Content-Type: application/json
{
  "nome" : "Nome da Empresa",
  "documento": "42451170000132",
  "email": "email@emailcom.br"
}
```

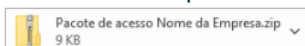
Onde:

Atributo	Tipo	Obrigatório	Pode ser vazio/nulo?	Restrições
"nome"	String	Sim	Não	Máximo 120 caracteres
"documento"	String	Sim	Não	CNPJ válido, apenas números
"email"	String	Sim	Não	Email válido (*@*.**)

Ao consumir essa API com sucesso, você receberá o retorno:

HTTP Status	200 OK
Corpo	{ "status": "Sucesso", "mensagem": "O pacote de acesso foi enviado para o email informado." }

E será enviado para o email informado no corpo da requisição, o seguinte email:



Olá Nome da Empresa,

Falta muito pouco para você acessar todas as funcionalidades que criamos para você. O pacote de acesso anexo é seu passaporte na B3, armazene-o com segurança.

Estamos felizes em fazer parte dessa sua jornada de investimentos.

B3. A Bolsa do Brasil

acesse B3.com.br



[Políticas e Termos](#)

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 /4bee639f-5388-44c7-bbac-cb92a93911e6/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=98ddf4b0-f66d-
4c96-97ea-9e30306599e7%2F.default
```

Onde:

Atributo	Tipo	Obrigatório	Pode ser vazio/nulo?	Restrições
"grant_type"	String	Sim	Não	Valor fixo "client_credentials"

Atributo	Tipo	Obrigatório	Pode ser vazio/nulo?	Restrições
"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 "98ddf4b0-f66d-4c96-97ea-9e30306599e7/.default" necessário ser URL encoded, então o / fica "%2F".

Ao consumir essa API com sucesso, você receberá o retorno:

HTTP Status	200 OK
Corpo	<pre>{ "token_type": "Bearer", "expires_in": 3599, "ext_expires_in": 3599, "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS1hfZyJ9.eyJhdWQiOiI5OGRkZjRiMC1mNjZkLTRjOTYtOTdlYS05ZTMwMzA2NTk5ZTciLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20vNGJIZTYzOWYtNTM4OC00NGM3LWJiYWMyY2I5MmE5MzcxMWU2L3YyLjAiLCJpYXQiOiJlMjc2NTc4MTc5MTYyYzY1NzgxNywiZXhwIjozNjYxNzE3LCJhaW8iOiJFMlpuWUVoL3FybGpvZkZ6NnczTEV4TmwxWFJrQVFBPSIsImF6cCI6IjBjOTkxNjEzLTRjOTAtNDU0ZC04Njg1LWQ0NjZhNDc2NjYiLSImF6cGFjci6iIjEiLCJvYWQiOiI0NTg5NDEzOC1jMmE2LTQ2MjEtYWUxNi1kNDgyYzYzOTc5YzkiLCJyaCI6IjAuQVJvQW4yUHVTNGhUeDBTN3JNdVNXVGVtSNWhNV21ReVFURTFGaG9YVWpxUjJhY3NWQUFBLiIsInN1YiI6IjQ1ODk0MTM4LWMyYTYtNDYyMS1hZTE2LWQ0ODJjMzg5NzFjOSIsInRpZCI6IjRiZWU2MzlmLTUzODgtNDJRjNy1iYmFjLWNiOTJhOTM5MTFNilslInV0aSI6Ikhtb2xvbmEztc2tITEF5a1pjcXcwQUEiLCJ2ZXIIOiIyLjAifQ.B-5n4jm3WaEdxoPt-iZ5A4FCWH5IyNaKDN8rTDgOILjqr7PAxTevxmOSy4IEpfl3bEpGY78C2TBt1qWAMoJmUfHp5FMxjHj035N4SjgBVQtkiA1cfSeir1b1w7SKtJLYlC8ivT0sridja4PVLdqDkQhd4qAtLtCnNEI7d7XjSRhyOZCBTh-1oa8GSvWFJqz5c_bTnK80IL_4LxGWNLiXfN6elgm2ISJsHP0Pt7JTty4HD3TrCbO6o6BpG4-twQQi2KcWJNkpdRsXkUnkC-M892MI42KGz0f_pV0hmgjsrx9FPf5Q9Qkflqt9LrLABc8lQLhy5wZPYB3ZO5PGg1g" }</pre>

Onde:

Atributo	Restrição
"token_type"	Exibe o tipo de token de segurança que foi emitido. Sempre será "Bearer".

Atributo	Restrição
"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. Será necessário informá-lo no cabeçalho da requisição da API que deseja consumir.

Passo 3 - Chamar API para validar a autenticação e autorização.

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/aceso/healthcheck HTTP/1.1
Host: apib3i-cert.b3.com.br:2443
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS1hFZyJ9.eyJhdWQiOiIiOGRkZjRiMC1mNjZkLTRjOTYtOTdlYS05ZTMwMzA2NTk5ZTciLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20vNGJlZTYzOWYtNTM4OC00NGM3LWJiYWVtY2I5MmE5MzZkMWU2L3YyLjAiLCJpYXQiOiJlY2Mjc2NTc4MTcslm5iZiI6MTYyNzY1NzgxNywiZXhwIjoxNjI3NjYxNzE3LCJhaW8iOiJFbmlpbnWUVOl3FybGpvZkZ6NnczTEV4TmwxWFirQVFBSlsmF6cCI6IjBjOTkxNjEzLTRjOTAtNDU0ZC04Njg1LWQ0NjZhNDc2NjlyYiIsImF6cGFjciI6IjEiLCJvaWQiOiI0NTg5NDEzOC1jMmE2LTQ2MjEtYWUxNi1kNDgyYzY4OTcxYzkiLCJyaCI6IjAuQVJlVjQwY4yUHVtNGhUeDBTN3JndVNxVGtSNWhNV21ReVFURTFGaG9YVWpxUjJhY3NWQUFBLiIsInN1YiI6IjQ1ODk0MTM4LWMyYTYtNDYyMS1hZTE2LWQ0ODJjMzg5NzFjOSIsInRpZCI6IjRiZWU2MzlmLTUzODgtNDRjNy1iYmFjLWVhOTJhOTM5MTFInlInV0aSI6Ikhkb2xvbiE2Ztc2tTEF5a1pjXcWQUEiLCJZXXliOiIyLjAifQ.B-5n4jm3WaEdxoPt-iZ5A4FCWH5IyNaKdn8rTDgOILq7PAxTevxmOsy4IEpfL3bEpgY78C2TBt1qWAMoJmUfHp5FMxjHj035N4SjgBVQTKiA1cFSeirlb1w7SKtJLyIC8ivT0sjridja4PVLdqDkQhd4qAtLtCnNEI7d7XjSRhy0ZCBTh-loa8GSvWFJqz5c_bTnK80IL_4LxGWNLiXfN6elgm2ISJsHP0Pti7JTty4HD3TrCbO6o6BpG4-twQQi2KcWJNkpdRsXkUnkC-M892MI42KgZof_pV0hmgisrx9FPf5Q9QKflqt9LrLABc8lQLhy5wZPYB3ZO5PGg1g
```

A apresentação de certificado de cliente (Mutual SSL 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://apib3i-cert.b3.com.br:2443/api/aceso/healthcheck' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS1hFZyJ9.eyJhdWQiOiIiOGRkZjRiMC1mNjZkLTRjOTYtOTdlYS05ZTMwMzA2NTk5ZTciLCJpc3MiOiJodHRwczovL2xvZ2luLm1pY3Jvc29mdG9ubGluZS5jb20vNGJlZTYzOWYtNTM4OC00NGM3LWJiYWVtY2I5MmE5MzZkMWU2L3YyLjAiLCJpYXQiOiJlY2Mjc2NTc4MTcslm5iZiI6MTYyNzY1NzgxNywiZXhwIjoxNjI3NjYxNzE3LCJhaW8iOiJFbmlpbnWUVOl3FybGpvZkZ6NnczTEV4TmwxWFirQVFBSlsmF6cCI6IjBjOTkxNjEzLTRjOTAtNDU0ZC04Njg1LWQ0NjZhNDc2NjlyYiIsImF6cGFjciI6IjEiLCJvaWQiOiI0NTg5NDEzOC1jMmE2LTQ2MjEtYWUxNi1kNDgyYzY4OTcxYzkiLCJyaCI6IjAuQVJlVjQwY4yUHVtNGhUeDBTN3JndVNxVGtSNWhNV21ReVFURTFGaG9YVWpxUjJhY3NWQUFBLiIsInN1YiI6IjQ1ODk0MTM4LWMyYTYtNDYyMS1hZTE2LWQ0ODJjMzg5NzFjOSIsInRpZCI6IjRiZWU2MzlmLTUzODgtNDRjNy1iYmFjLWVhOTJhOTM5MTFInlInV0aSI6Ikhkb2xvbiE2Ztc2tTEF5a1pjXcWQUEiLCJZXXliOiIyLjAifQ.B-5n4jm3WaEdxoPt-iZ5A4FCWH5IyNaKdn8rTDgOILq7PAxTevxmOsy4IEpfL3bEpgY78C2TBt1qWAMoJmUfHp5FMxjHj035N4SjgBVQTKiA1cFSeirlb1w7SKtJLyIC8ivT0sjridja4PVLdqDkQhd4qAtLtCnNEI7d7XjSRhy0ZCBTh-loa8GSvWFJqz5c_bTnK80IL_4LxGWNLiXfN6elgm2ISJsHP0Pti7JTty4HD3TrCbO6o6BpG4-twQQi2KcWJNkpdRsXkUnkC-M892MI42KgZof_pV0hmgisrx9FPf5Q9QKflqt9LrLABc8lQLhy5wZPYB3ZO5PGg1g
```

```
00DJmZg5NzFjOSIsInRpZCI6lRiZWU2MzlmLTUzODgtNDRjNy1iYmFjLWNiOTJhOTM5MTFINiIsInV0aSI6Ikhtb2xvbE
Ztc2tITEF5a1pjcXcwQUEiLCJ2ZXliOiIyLjAifQ.B-5n4jm3WaEdxoPt-
iZ5A4FCWH5IyNaKDN8rTDgOILqR7PAxTevxmOSy4IEpfL3bEpGY78C2TBt1qWAMoJmUfHp5FMxjHj035N4SjgjBVQT
kiA1cFSeirlb1w7SKtJLyIC8ivT0sjridja4PVLdqDkQhd4qAtLtCnNEI7d7XjSRhy0ZCBTh-
loa8GSvWFJqz5c_bTnK80IL_4LxGWNLiXfN6eIgm2ISJsHP0Pti7JTty4HD3TrCbO6o6BpG4-
twQQi2KcWJNkpdRsXkUnkC-M892MI42KGz0f_pV0hmgisrx9FPf5Q9QKflqt9LrLABc8lQLhy5wZPYB3ZO5PGg1g'
```

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:

HTTP Status	200 OK
Corpo	{ "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 – 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](#).

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

```
{
  "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",
"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.

Atributo	Tipo	Descrição
"custodianCodeDestination"	String	Código de identificação do custodiante de destino.
"assetCode"	String	Código do ativo.
"assetQuantity"	Decimal	Quantidade do ativo a transferir.
"eventValue"	Decimal	Valor do evento provisionado a transferir. Observação: Se o ativo for um evento provisionado, enviamos o atributo eventValue . Caso contrário, enviamos o atributo assetQuantity .

Passo 5 – 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://apib3i-cert.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.

```
{
  "data": {
    "requesterTransactionId": "B3i-202103010001",
    "requestedTransactionId": "BTG-909103090009",
    "documentNumber": "9999999999",
    "authorizationToken": "155082098e134fa14279e7baa4cf8270",
    "transactionDateTime": "2021-03-30T14:00:00.001Z"
  },
  "errors": [
    {
      "code": null,
      "title": null,
      "detail": null
    }
  ]
}
```

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.

Passo 6 – Configurações e Credenciais do Custodiante.

Para o funcionamento da integração é necessário que o custodiante configure o Mutual SSL e disponibilize um serviço que retorne um token JWT válido para a requisição da chamada descrita no passo 4.

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*;*
- *URL do serviço para que o B3 Investidor envie a autorização (formato `url_custodiante/api/request-authorizations/v1/investors`);*
- *Certificado para configuração do mutual SSL (padrão .pfx + senha, no formato PKCS #12);*

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