[PT] Open Finance Brasil Financial-grade API Security Profile 1.0

Prefácio

The normative version in English

A Estrutura Inicial do Open Finance Brasil (EIOFB) é responsável por criar padrões e especificações necessárias para atender aos requisitos e obrigações da Legislação do Open Finance do Brasil, conforme originalmente delineado pelo Banco Central do Brasil. É possível

que alguns dos elementos deste documento estejam sujeitos a direitos autorais ou patenteados. O EIOFB não se responsabiliza pela identificação de qualquer ou todos esses direitos.

O Financial-grade API 1.0 do Open Finance Brasil consiste nas seguintes partes:

- Open Finance Brasil Financial-grade API Security Profile 1.0
- Open Finance Brasil Dynamic Client Registration Profile 1.0

Estas partes são destinados a serem usados com [RFC6749], [RFC6750], [RFC7636], [OIDC], [FAPI-1-Baseline] e [FAPI-1-Advanced]

Introdução

A Financial-grade API do Open Finance Brasil é um perfil OAuth altamente seguro que visa fornecer diretrizes de implementação específicas para segurança e interoperabilidade que podem ser aplicadas a APIs na área de Open Finance do Brasil que requerem um nível de privacidade superior ao fornecido pelo padrão [Financial-grade API Security Profile 1.0 - Part 2: Advanced][FAPI-1-Advanced]. Entre outras melhorias, esta especificação aborda considerações de privacidade identificadas em [FAPI-1-Advanced] que são relevantes nas especificações do Open Finance Brasil, mas não foram, até agora, exigidas por outras jurisdições.

Embora seja possível codificar um provedor de OpenID e parte de confiança a partir dos primeiros princípios usando esta especificação, o público principal para esta especificação são as partes que já possuem uma implementação certificada do [Financial-grade API Security Profile 1.0 - Part 2: Advanced][FAPI-1-Advanced] e deseja obter a certificação para o programa Brasil Open Finance.

Convenções Notacionais

As palavras-chave "*deve*" (shall), "*não deve*" (shall not), "*deveria*" (should), "*não deveria*" (should not) e "*pode*" (may) presentes nesse documento devem ser interpretadas conforme as diretrizes descritas em [ISO Directive Part 2][ISODIR2] observando seguinte equivalência:

- "deve" ⇒ equivalente ao termo "shall" e expressa um requerimento definido no documento (nas traduções é similar ao termo "must", que pode denotar um requerimento externo ao documento);
- "não deve" ⇒ equivalente ao termo "shall not" e também expressa um requerimento definido no documento;
- "deveria" e "não deveria"⇒ equivalente ao termo "should" e "should not" e expressa uma recomendação
- "pode" ⇒ equivalente ao termo "may" indica uma permissão

Estas palavras-chave não são usadas como termos de dicionário, de modo que qualquer ocorrência deles deve ser interpretada como palavras-chave e não devem ser interpretados com seus significados de linguagem natural.

Escopo

Este documento especifica o método para os aplicativos

- obterem de maneira segura os tokens OAuth necessários para acesso a dados críticos de acordo com os requisitos do Open Finance Brasil;
- utilizarem o OpenID Connect para identificação do usuário do Open Finance; e
- utilizarem o OpenID Connect para afirmar a identidade do cliente.

Este documento é aplicável a todos os participantes do Open Finance no Brasil.

Referências normativas

Os seguintes documentos referenciados são indispensáveis para a adoção das especificações deste documento. Para referências datadas, apenas a edição citada se aplica. Para referências não datadas, deve-se aplicar a última edição do documento referenciado (incluindo quaisquer emendas).

ISODIR2 - ISO/IEC Directives Part 2

RFC6749 - The OAuth 2.0 Authorization Framework

RFC6750 - The OAuth 2.0 Authorization Framework: Bearer Token Usage

RFC7636 - Proof Key for Code Exchange by OAuth Public Clients

RFC6819 - OAuth 2.0 Threat Model and Security Considerations

RFC7515 - JSON Web Signature (JWS)

RFC7519 - JSON Web Token (JWT)

RFC7591 - OAuth 2.0 Dynamic Client Registration Protocol

RFC7592 - OAuth 2.0 Dynamic Client Registration Management Protocol

BCP195 - Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)

OIDC - OpenID Connect Core 1.0 incorporating errata set 1

FAPI-CIBA - Financial-grade API: Client Initiated Backchannel Authentication Profile

OIDD - OpenID Connect Discovery 1.0 incorporating errata set 1

OIDR - OpenID Connect Registration 1.0 incorporating errata set 1

RFC8705 - OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens

JARM - Financial-grade API: JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)

PAR - OAuth 2.0 Pushed Authorization Requests

JAR - OAuth 2.0 JWT Secured Authorization Request

FAPI-1-Baseline - Financial-grade API Security Profile 1.0 - Part 1: Baseline

FAPI-1-Advanced - Financial-grade API Security Profile 1.0 - Part 2: Advanced

FAPI-2-Baseline - Financial-grade API Security Profile 2.0 - Part 1: Baseline

LIWP - OIDF FAPI WG Lodging Intent Working Paper

LIWP - OIDF FAPI WG Lodging Intent Working Paper

OFB-FAPI-DCR - Open Finance Brasil Financial-grade API Dynamic Client Registration Profile 1.0

RFC4648 - The Base16, Base32, and Base64 Data Encodings

Termos e definições

Para efeitos deste documento, os termos definidos em RFC6749, RFC6750, RFC7636, OpenID Connect Core e ISO29100 se aplicam.

Símbolos e termos abreviados

- **API** - Application Programming Interface (Interface de programação de aplicativo)
- **CSRF** - Cross Site Request Forgery
- **DCR** - Registro de cliente dinâmico
- **EIOFB** - Estrutura Inicial do Open Finance Brasil
- **FAPI** - Financial-grade API
- **HTTP** - Protocolo de transferência de hipertexto
- **MFA** - Multi-Factor Authentication (Autenticação por Múltiplos Fatores)
- **OIDF** - OpenID Foundation
- **REST** - Representational State Transfer (Transferência de Estado Representacional)
- **TLS** - Transport Layer Security (Segurança da Camada de Transporte)

Profile de Segurança para o Open Finance Brasil

Introdução

O perfil de segurança do Open Finance Brasil especifica requisitos adicionais de segurança e de identificação para o acesso a API´s com recursos críticos protegidas pelo OAuth 2.0 Authorization Framework, que consiste em [RFC6749], [RFC6750], [RFC7636], [FAPI-1-Baseline], [FAPI-1-Advanced] e outras especificações.

Este perfil descreve as capacidades e os recursos de segurança que devem ser oferecidos por servidores e clientes que são necessários para o Programa do Open Finance Brasil, definindo as medidas para mitigar ou endereçar:

- ataques que abordam considerações de privacidade identificadas na cláusula 9.1 de [FAPI-1 Advanced].
- o requisito de concessão de acesso granular a recursos, com vistas à minimização de dados;
- o requisito de informar sobre o contexto da autenticação do usuário (claim Authentication Context Request - acr) que foi realizada por um Provedor OpenID, com vistas a favorecer o adequado gerenciamento do risco decorrente do acesso do usuário;
- o requisito para que os clientes de API declarem um relacionamento prévio com o usuário, afirmando em uma `claim` de identificação do usuário como parte do fluxo de autorização.

Disposições de segurança do Open Finance Brasil

Introdução

O Open Finance Brasil tem um requisito para endereçar considerações de privacidade que foram identificadas, mas não abordadas na especificação final FAPI-1-Advanced, sem impor requisitos adicionais aos Servidores de Autorização que estão sendo propostos em FAPI-2-Baseline.

Os participantes desse ecossistema precisam que os clientes de API solicitem a um provedor openid a confirmação dos valores das `claims` de identificação do usuário como parte de uma solicitação de autorização usando o mecanismo definido na cláusula 5.5.1 de [OIDC].

O uso do parâmetro `claims` para solicitar a validação de valores de identificação explícitos requer que os clientes de API protejam com criptografia o Request Object para evitar vazamento de informações. Este risco é identificado na cláusula 7.4.1 do [FAPI-1-Baseline].

Além disso, este perfil descreve o escopo específico, valores de `acr` e requisitos de gerenciamento de clientes necessários para dar suporte ao ecossistema Open Finance Brasil mais amplo.

Como um perfil do OAuth 2.0 Authorization Framework, este documento exige o seguinte para o perfil de segurança do Open Finance Brasil.

Servidor de Autorização

O Servidor de Autorização **deve** suportar as disposições especificadas na cláusula 5.2.2 de [Financial-grade API Security Profile 1.0 - Parte 2: Advanced] [FAPI-1-Advanced].

Além disso, ele deve:

1. deve suportar Request Objects JWE assinados e criptografados passados por valor ou deve exigir requisições do tipo "pushed authorization requests" [PAR];
2. deve publicar metadados de descoberta (incluindo a do endpoint de autorização) por meio do documento de metadado especificado em [OIDD] e [RFC8414] (".well-known");
3. deve suportar os parâmetros `claims` como definido no item 5.5 do [OpenID Connect Core][OIDC];
4. deve suportar o atributo `claim` padrão oidc "cpf" conforme definido no item 5.2.2.2 deste documento;
5. deve suportar o atributo `claim` padrão oidc "cnpj" conforme definido no item 5.2.2.3 deste documento, se a instituição for detentora de conta para pessoas jurídicas;
6. deve suportar o atributo `acr` "urn:brasil:openbanking:loa2" como definido no item 5.2.2.4 deste documento;
7. deveria suportar o atributo `acr` "urn:brasil:openbanking:loa3" como definido no item 5.2.2.4 deste documento;
8. deve implementar o endpoint "userinfo" como definido no item 5.3 do [OpenID Connect Core][OIDC];
9. deve suportar o escopo parametrizável ("parameterized OAuth 2.0 resource scope") *consent* como definido no item 6.3.1 de [OIDF FAPI WG Lodging Intent Pattern][LIWP];
10. pode suportar [Financial-grade API: Client Initiated Backchannel Authentication Profile] [FAPI-CIBA];
11. (requisito temporariamente retirado);
12. deve suportar `refresh tokens`;
13. deve emitir `access tokens` com o tempo de expiração entre 300 (mínimo) e 900 (máximo) segundos;
14. deve sempre incluir a claim `acr` no id_token;
15. deve suportar os valores `code` e `id_token` para o atributo `response_type`;

16. pode suportar o valor `code` para o atributo `response_type` em conjunto com o valor `jwt` para o atributo `response_mode`;
17. não deve permitir o recurso de rotação de `refresh tokens`;
18. deve garantir que em caso de compartilhamento do Servidor de Autorização para outros serviços, além do Open Finance, não divulgue e/ou possibilite o uso de métodos não certificados no ambiente do Open Finance;
19. deve garantir que as configurações divulgadas aos demais participantes através do `OpenID Discovery` (indicado pelo arquivo de `Well-Known` cadastrado no Diretório) sejam restritos aos modos de operação aos quais a instituição se certificou;
    a. deve manter em suas configurações os métodos para os quais ainda hajam clientes ativos;
    b. deve atualizar os cadastros que utilizem métodos não certificados, através de tratamento bilateral entre as instituições envolvidas;
20. deve recusar requisições, para o ambiente do Open Finance, que estejam fora dos modos de operação ao qual a instituição certificou seu Servidor de Autorização;
21. deve recusar requisições de autenticação que incluam um id_token_hint, visto que o id_token em posse do requisitante pode conter Informação de Identificação Pessoal, que poderia ser enviada descriptografada pelo cliente público;
22. o tempo mínimo de expiração do `request_uri` deve ser de 60 segundos;
23. deve recusar requisições que não apresentem o cabeçalho `x-fapi-interaction-id` em endpoints FAPI;

Token de ID como assinatura separada

O Servidor de Autorização *deve* suportar as disposições especificadas na cláusula 5.2.2.1 de [Financial-grade API Security Profile 1.0 - Parte 2: Advanced] [FAPI-1-Advanced]

Além disso, se o valor `response_type` `code id_token` for usado, o servidor de autorização:

1. **não deveria** retornar Informação de Identificação Pessoal (PII) confidenciais no token de ID na resposta de autorização, mas se for necessário, então ele **deve** criptografar o token de ID.
2. Informação de Identificação Pessoal pode incluir, mas não está restrito a:
    a. Claim `sub` caso use informação que possibilite a identificação da pessoa natural;
    b. As Claims padrões definidas na cláusula 5.1 [OIDC], que podem conter dados como data de nascimento, endereço ou telefone;
    c. A nova Claim CPF, definida na próxima seção.

3. Caso seja solicitada alguma Claim contendo Informação de Identificação Pessoal:
    a. Se esta for marcada como essencial, em não se havendo chave de criptografia registrada para o Cliente, deverá falhar a requisição se for solicitada no Endpoint de Autorização. Não há impedimentos caso a solicitação seja feita pelo Cliente Confidencial através do Endpoint de Token;
    b. Se esta não for marcada como essencial, o Servidor de Autorização deverá omiti-la no Endpoint de Autorização, podendo ser respondida no Endpoint de Token chamado pelo Cliente Confidencial.
4. Para a criptografia do id_token deve ser utilizada chave disponível no `JWKS` informado no parâmetro `jwks_uri` durante o registro do cliente, indicada através do cabeçalho `kid` do documento JWT;
5. O uso de outros cabeçalhos para indicação da chave utilizada, como `x5u`, `x5c`, `jku` ou `jkw` é vetado conforme definido na cláusula 2 [OIDC].

Clarificações sobre a "sub" Claim

Este perfil usa a definição oficial encontrada em: https://github.com/OpenBanking-Brasil/specs-seguranca/tree/main/idtoken_review. Isso significa que o sub é um identificador nunca transferido ou alterado para o usuário final. O valor para um dado usuário nunca deve mudar dentro de uma instituição, mesmo em diferentes consentimentos.

Solicitando uma "claim" cpf

Este perfil define "cpf" como uma nova `claim` padrão de acordo com cláusula 5.1 [OIDC]

O número do **CPF** (Cadastro de Pessoas Físicas, [sepeˈɛfi]; português para "Registro de Pessoas Físicas") é o cadastro de pessoa física **brasileira**. Este número é atribuído pela Receita Federal **Brasileira** para brasileiros e estrangeiros residentes que, direta ou indiretamente, pagar impostos no **Brasil**.

No modelo de identidade do Open Finance Brasil, o cpf é uma string composta por números 11 caracteres de comprimento e podem começar com 0.

Se a Claim **cpf** for solicitada como essencial para constar no ID token ou na resposta ao endpoint de UserInfo e na solicitação constar no parâmetro `value` com determinado **CPF** exigido, o Authorization Server **DEVE** retornar no atributo **cpf** o valor que corresponda ao da solicitação.

Se a Claim **cpf** for solicitada como essencial para constar no ID Token ou na resposta no endpoint de UserInfo, o Authorization Server deve retornar no atributo **cpf** o valor com o **CPF** do usuário autenticado.

Se a Claim **cpf** indicada como essencial não puder ser preenchida ou não for compatível com o requisito, o Authorization Server deve tratar a solicitação como uma tentativa de autenticação com falha.

Se a Claim **cpf** indicada como essencial for solicitada no endpoint de Autorização, deverá seguir as regras definidas na seção 5.2.2.1.

Nome: cpf, Tipo: String, Regex: '^\d{11}$'

Solicitando a "claim" cnpj

Este perfil define "cnpj" como uma nova reivindicação padrão de acordo com cláusula 5.1 [OIDC]

CNPJ, abreviação de Cadastro Nacional de Pessoas Jurídicas, é um número de identificação de empresas **brasileiras** emitidas pelo Ministério da Fazenda **brasileira**, **na** "Secretaria da Receita Federal" ou "Ministério da Fazenda" do Brasil. No modelo de identidade do Open Finance Brasil, pessoas físicas podem se associar a 0 ou mais CNPJs. Um CNPJ é uma string que consiste em números de 14 dígitos e pode começar com 0, os primeiros oito dígitos identificam a empresa, os quatro dígitos após a barra identificam a filial ou subsidiária ("0001" padrão para a sede), e os dois últimos são dígitos de soma de verificação. Para este perfil, o pedido de cnpj deve ser solicitado e fornecido como o número de 14 dígitos.

Se a Claim **cnpj** for solicitada como essencial para constar no ID Token ou na resposta ao endpoint UserInfo e na solicitação constar, no parâmetro `value`, determinado **CNPJ** exigido, o Authorization Server **DEVE** retornar no atributo **cnpj** um **conjunto** de **CNPJs** relacionado com o usuário, um dos quais deve incluir valor que corresponda ao da solicitação.

Se a Claim **cnpj** for solicitada como essencial para constar no ID Token ou na resposta ao endpoint UserInfo, o Authorization Server deve incluir no ID Token ou na resposta ao endpoint UserInfo um **conjunto** que inclua um elemento com o número do **CNPJ** relacionado à conta utilizada na autenticação do usuário.

Se a Claim **cnpj** indicada como essencial não puder ser preenchida ou validada, o Authorization Server deve tratar a solicitação como uma tentativa de autenticação com falha.

Nome: cnpj, Tipo: Array of Strings, Array Element Regex: '^\d{14}$'

Solicitando o "urn:brasil:openbanking:loa2" ou "urn:brasil:openbanking:loa3" Solicitação de contexto de autenticação

Esse perfil define "urn:brasil:openbanking:loa2" e "urn:brasil:openbanking:loa3" como novas classes de "Authentication Context Request" (ACR)

- **LoA2**: mecanismo de autenticação com a adoção de um único fator
- **LoA3**: mecanismo de autenticação com múltiplos fatores de autenticação

A seguinte orientação deve ser observada para o mecanismo de autenticação das APIs do Open Finance Brasil:

- De acordo com o Art. 17 da Resolução Conjunta nº 01, as instituições devem adotar procedimentos e controles para autenticação de cliente **compatíveis com os aplicáveis ao acesso a seus canais de atendimento eletrônicos**.
- Em observância à regulação em vigor, sugere-se que:
  - **Para a autenticação do usuário em autorizações de acessos às APIs de compartilhamento de dados (Fase 2)**, os *Authorization Servers* **deveriam** adotar, no mínimo, método compatível com `LoA2`; e
  - **Para a autenticação do usuário em autorizações de acessos às API´s das fases subsequentes**, os *Authorization Servers* **deveriam** adotar método de autenticação compatível com `LoA3` ou superior.

Em todos os casos, a adoção de mecanismo de autenticação mais rigoroso (`LoA3` ou superior) fica a critério da instituição transmissora ou detentora de conta, de acordo com sua avaliação de riscos e de forma compatível com os mecanismos habitualmente utilizados.

Esclarecimentos adicionais sobre fatores de autenticação

São fatores de autenticação:

- Aquilo que **você conhece**, como uma senha ou frase secreta
- Aquilo que **você tem**, como um token, smartcard ou dispositivo
- Aquilo que **"você é"**, ou seja, autenticação condicionada a apresentação de uma característica física exclusivamente sua, como a validação por biometria

Para realizar autenticação por múltiplos fatores (MFA) é necessário que o usuário apresente, ao menos, dois diferentes fatores dos listados acima. Um mesmo fator usado mais de uma vez - por exemplo, a apresentação de suas senhas que ele conhece - não pode ser aceito como MFA.

Cliente confidencial

Um cliente confidencial deve apoiar as disposições especificadas na cláusula 5.2.3 de [Financial-grade API Security Profile 1.0 - Part 2: Advanced][FAPI-1-Advanced],

Além disso, o cliente confidencial

1. deve suportar objetos de solicitação *encrypted*;
2. deve suportar solicitações de autorização push (pushed authorization requests) [PAR];
3. deve usar objetos de solicitação *encrypted* se não usar [PAR];

4. deve suportar o escopo de recurso OAuth 2.0 parametrizado *consent* conforme definido na cláusula 6.3.1 [OIDF FAPI WG Lodging Intent Pattern][LIWP];
5. deve suportar `refresh tokens`;
6. não deve incluir um valor específico na *claim* `acr`;
7. deve definir a *claim* `acr` como *essential*;
8. deve suportar todos os métodos de autenticação especificados no item 14 da seção 5.2.2 da [Financial-grade API Security Profile 1.0 - Part 2: Advanced][FAPI-1-Advanced] incluindo as diferentes combinações de métodos de encaminhamento dos Requests Objects (usando ou não [PAR] - item 11);
9. não deve permitir o recurso de rotação de `refresh tokens`;
10. não deve solicitar requisições de autenticação que incluam um id_token_hint, visto que o id_token a ser utilizado pode conter Informação de Identificação Pessoal, que poderia ser enviada descriptografada através do cliente público;
11. deve enviar o cabeçalho `x-fapi-interaction-id` em endpoints FAPI;

Considerações de segurança

Os participantes devem apoiar todas as considerações de segurança especificadas na cláusula 8
[Financial-grade API Security Profile 1.0 - Parte 2: Advanced] [FAPI-1-Advanced] e o [Manual de Segurança de Banco Central do Brasil]
(https://www.bcb.gov.br/estabilidadefinanceira/exibenormativo?tipo=Instru%C3%A7%C3%A3o%20Normativa%20BCB&numero=134).
O ICP Brasil emite certificados RSA x509 somente, portanto, para simplificar, a seção remove o suporte para algoritmos EC
e exige que apenas algoritmos de criptografia recomendados pela IANA sejam usados.

Considerações sobre assinatura do conteúdo de mensagens (JWS)

1. Para garantir a integridade e o não-repúdio das informações tramitadas em **API´s sensíveis e que indicam essa necessidade na sua documentação**, deve ser adotado a estrutura no padrão JWS definida na [RFC7515] e que inclui:
   - Cabeçalho (*JSON Object Signing and Encryption – JOSE Header*), onde se define o algoritmo utilizado e inclui informações sobre a chave pública ou certificado que podem ser utilizadas para validar a assinatura;
   - Payload (*JWS Payload*): conteúdo propriamente dito e detalhado na especificação da API;

- Assinatura digital (*JWS Signature*): assinatura digital, realizada conforme parâmetros do cabeçalho.

2. Cada elemento acima deve ser codificado utilizando o padrão Base64url RFC4648 e, feito isso, os elementos devem ser concatenados com "." (método JWS Compact Serialization, conforme definido na [RFC7515]).
3. O payload das mensagens (requisição *JWT* e resposta *JWT*) assinadas devem incluir as seguintes `claims` presentes na [RFC7519] (JWT):
   - **aud** (na requisição *JWT*): o Provedor do Recurso (p. ex. a instituição detentora da conta) deverá validar se o valor do campo **aud** coincide com o endpoint sendo acionado;
   - **aud** (na resposta *JWT*): o cliente da API (p. e. instituição iniciadora) deverá validar se o valor do campo **aud** coincide com o seu próprio `organisationId` listado no diretório;
   - **iss** (na requisição *JWT* e na resposta *JWT*): o receptor da mensagem deverá validar se o valor do campo **iss** coincide com o `organisationId` do emissor;
   - **jti** (na requisição *JWT* e na resposta *JWT*): o valor do campo **jti** deverá ser preenchido com o UUID definido pela instituição de acordo com a [RFC4122] usando o versão 4;
   - **iat** (na requisição *JWT* e na resposta *JWT*): o valor do campo **iat** deverá ser preenchido com horário da geração da mensagem e de acordo com o padrão estabelecido na RFC7519 para o formato *NumericDate*.
   - **cty** (na requisição *JWT* e na resposta *JWT*): o valor do campo **cty** deverá ser preenchido caso as operações de assinatura ou criptografia aninhadas não sejam empregadas, o uso desse parâmetro de cabeçalho não é recomendado. No caso de assinatura aninhada ou criptografia ser empregada, este parâmetro de cabeçalho deve estar presente; neste caso, o valor deve ser "JWT", para indicar que um JWT aninhado é transportado neste JWT. Embora os nomes dos tipos de mídia não diferenciem maiúsculas de minúsculas, é recomendável que "JWT" sempre seja escrito com caracteres maiúsculos para compatibilidade com implementações herdadas.
4. O content-type HTTP das requisições e respostas com mensagens JWS deve ser definido como: "application/jwt".
5. No cabeçalho JOSE deve constar os seguintes atributos:
   - **alg** - deve ser preenchido com o valor `PS256` ";
   - **kid** - deve ser obrigatoriamente preenchido com o valor do identificador da chave utilizado para a assinatura;
   - **typ** - deve ser preenchido com o valor `JWT`.

- Em caso de erro na validação da assinatura pelo `Provedor do Recurso` a API deve retornar mensagem de erro HTTP com `status code` **400** e a resposta deve incluir na propriedade `code` do objeto de resposta de erro especificado na API (`ResponseError`) a indicação da falha com o conteúdo `BAD_SIGNATURE` .
- Erros na validação da mensagem recebida pela aplicação cliente (p. ex. iniciador de pagamento) devem ser registrados e o `Provedor do Recurso` (p. ex. instituição detentora de conta) deve ser notificado.

6. O receptor deve validar a consistência da assinatura digital da mensagem JWS **exclusivamente com base nas informações obtidas do diretório**, ou seja, com base nas chaves publicadas no JWKS da instituição no diretório.
7. As assinaturas devem ser realizadas com uso do certificado digital de assinatura especificado no Padrão de Certificados Open Finance Brasil;
8. A claim *iat* deve ser numérica no formato Unix Time GMT+0 com tolerância de +/- 60 segundos;
9. A claim do *jti* deve ser única para um *clientId* dentro de um intervalo de tempo de 86.400 segundos (24h), não podendo ser reutilizada neste período. Em caso de reutilização, deverá ser retornado o código de erro HTTP 403. Demais casos seguir instrução da RFC 6749, item 5.2;

Considerações sobre algoritmos de assinatura

Para JWS, clientes e servidores de autorização

1. devem usar o algoritmo PS256;

Considerações de algoritmo de criptografia

Para JWE, clientes e servidores de autorização

1. devem usar RSA-OAEP com A256GCM

Considerações sobre o uso seguro do Transport Layer Security

Para TLS, endpoints do Servidor de Autenticação e endpoints do Servidor de Recursos usados diretamente pelo cliente:

1. devem suportar `TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256`
2. devem suportar `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`
3. As funcionalidades "TLS Session Resumption" e "TLS Renegotiation" devem ser desabilitadas

Considerações sobre compartilhamento de dados

Mecanismo de Autorização

Introdução

Os mecanismos existentes para gerenciar adequadamente o acesso aos recursos definidos em [RFC6749] são insuficientes para atender aos requisitos de um ecossistema de compartilhamento de dados moderno. Aproveitar strings de escopo estático não fornece aos consumidores controle de granularidade suficiente para compartilhar com terceiros. O Open Finance Brasil optou por implementar uma API de consentimento como um recurso protegido OAuth 2.0 que pode ser usado para gerenciar o acesso granular aos recursos. A referência ao recurso de consentimento será transmitida como parte de um escopo de recurso dinâmico OAuth 2.0.

Definição de Escopo de Consentimento Dinâmico

Este perfil define o escopo dinâmico do OAuth 2.0 "consentimento" da seguinte maneira:

- string 'consent'; e
- delimitador de dois pontos ":"; e
- Consent API REST Resource Id retornado por uma criação bem-sucedida de Open Finance Consent Resource;

Adicionalmente:

- Consent Resource Id deve incluir caracteres seguros para url;
- Consent Resource Id deve ser "namespaced";
- Consent Resource Id deve ter propriedades de um `nonce` Nonce;

Exemplo de escopo de consentimento dinâmico

consent:urn:bancoex:C1DD33123

Ciclo de vida da autorização

Introdução

O recurso de consentimento tem um ciclo de vida gerenciado separada e distintamente da estrutura de autorização OAuth 2.0. As transições de estado e comportamentos esperados e condições de erro esperados dos Recursos REST protegidos com este perfil são definidos nas especificações funcionais da API publicadas pelo Open Finance Brasil.

Servidor de autorização

Além dos requisitos descritos nas disposições de segurança do Open Finance Brasil, o Servidor de Autorização

1. deve apenas emitir *refresh_tokens* quando vinculados a um consentimento ativo e válido;
    a. Não deve emitir refresh_token quando o consentimento estiver no status “CONSUMED” (para fase 3);
    b. Deve emitir um access_token por meio de um grant_type do tipo client credentials no status “CONSUMED” (para fase 3).
2. só deve compartilhar o acesso aos recursos quando apresentado *access_token* vinculado a um consentimento ativo e com o status "AUTHORIZED". Para tokens gerados com o scope: payments, o status do consentimento não será validado;
    a. No cenário de recebimento de token inválido, deve ser retornado status code 401.
3. deve revogar os *refresh tokens* e, quando aplicável, os *access tokens* quando o Consentimento (Consent Resource) relacionado for apagado;
4. deve garantir que os *access tokens* são emitidos com os *scopes* necessários para permitir acesso aos dados especificados em elemento *Permission* do Consentimento (Consent Resource Object) relacionado;
5. não deve rejeitar pedido de autorização com *scopes* além do necessário para permitir acesso a dados definidos em elemento *Permission* do Consentimento (Consent Resource Object) relacionado;
6. pode reduzir o escopo solicitado para um nível que seja suficiente para permitir o acesso aos dados definidos em elemento *Permission* do Consentimento (Consent Resource Object) relacionado;
7. deve manter registros sobre o histórico dos consentimento para permitir a adequada formação de trilhas de auditoria em conformidade com a regulação em vigor;
8. deve retornar falha na autenticação e o código de retorno *access_denied* no parâmetro *erro* (como especificado na seção 4.1.2.1 da [RFC6749]) caso o CPF do usuário autenticado não seja o mesmo indicado no elemento *loggedUser* do Consentimento (Consent Resource Object);
9. deve retornar falha na autenticação e o código de retorno *access_denied* no parâmetro *erro* (como especificado na seção 4.1.2.1 da [RFC6749]) caso o elemento *businessEntity* não tenha sido preenchido no Consentimento (Consent Resource Object) relacionado e o usuário tenha selecionado ou se autenticado por meio de credencial relacionada à conta do tipo Pessoa Jurídica (PJ);
10. deve condicionar a autenticação ou seleção de contas do tipo PJ à consistência entre o CNPJ relacionado à(s) conta(s) e o valor presente no elemento *businessEntity* do Consentimento (Consent Resource Object). Em caso de divergência deve retornar falha na

autenticação e o código de retorno *access_denied* no parâmetro *erro* (como especificado na seção 4.1.2.1 da [RFC6749]);

11. deve emitir *refresh_token* com validade não inferior à validade do consentimento ao qual está relacionado, respeitado os demais critérios acima.

Cliente confidencial

Além dos requisitos descritos nas disposições de segurança do Open Finance Brasil, o Cliente Confidencial

1. deve, sempre que possível, revogar e cessar o uso de *refresh* e de *access tokens* vinculados a um consentimento (Consent Resource Object) que foi excluído;
2. deve excluir consentimentos (Consent Resource Objects) que estão expirados;

Reconhecimentos

Agradecemos a todos que estabeleceram as bases para o compartilhamento seguro e seguro de dados por meio da formação do Grupo de Trabalho FAPI da OpenID Foundation, o GT de Segurança do Open Finance Brasil e aos pioneiros que ficarão em seus ombros.

As seguintes pessoas contribuíram para este documento:

- Alexandre Siqueira (Mercado Pago)
- Joseph Heenan (Authlete)
- Marcos Rodrigues (Itaú)
- Mário Ginglass (BNDES)
- Nic Marcondes (Quanto)
- Ralph Bragg (Raidiam)

Avisos

Open Finance Brasil e outros. Embora a Estrutura Inicial do Open Finance Brasil tenha tomado medidas para ajudar a garantir que a tecnologia esteja disponível para distribuição, ela não toma posição quanto à validade ou escopo de qualquer propriedade intelectual ou outros direitos que possam ser reivindicados como pertencentes à implementação ou uso do tecnologia descrita nesta especificação ou até que ponto qualquer licença sob tais direitos pode ou não estar disponível; nem representa que fez qualquer esforço independente para identificar tais direitos. A Estrutura Inicial do Open Finance Brasil e os contribuidores desta especificação não oferecem (e por meio deste expressamente se isentam de quaisquer) garantias (expressas, implícitas ou de outra forma), incluindo garantias implícitas de comercialização, não violação, adequação a uma finalidade específica ou título, relacionados a esta especificação, e todo o risco quanto à implementação desta especificação é assumido pelo implementador. A política de Direitos de Propriedade Intelectual do Open Finance Brasil exige que os contribuidores ofereçam uma promessa de patente de não fazer valer certas reivindicações de patentes contra outros contribuidores e implementadores. A Estrutura Inicial do Open Finance Brasil convida qualquer parte interessada a trazer à sua atenção quaisquer direitos autorais, patentes, pedidos de patentes ou outros direitos de propriedade que possam abranger a tecnologia que possa ser necessária para praticar esta especificação.