Agência para a Modernização Administrativa I.P.

Serviço de Assinatura de Fatura Sem Papel Eletrónicas

Documento de Integração

Versão 1.9

AGÊNCIA PARA A MODERNIZAÇÃO ADMINISTRATIVA

Referências a outros Documentos

Ref.	Descrição	Autor
-	-	-

Registo de Revisões

Data	Versão	Descrição	Autor
28-10-2022	1.0	Documento Inicial	AMA
02-11-2022	1.1	Documento com updates e correções	AMA
09-11-2022	1.2	• Adicionado fluxo típico; • Alteração ao fluxo criação de conta; • Alteração ao fluxo atualizar token.	AMA
15-11-2022	1.3	• Correções e atualizações	AMA
24-11-2022	1.4	• Alteração de parâmetros no fluxo Obter Cifra • Adicionada definição dos serviços	AMA
9-12-2022	1.5	• Adicionados códigos de erro	AMA
28-12-2022	1.6	• Alterado método de atualização de token • Correções	AMA

Data	Versão	Descrição	Autor
14-06-2023	1.7	• Adição campos necessários para envio de fatura • Especificação dos mecanismos de reenvio de faturas com mais de 48h • Partilha de estados de faturas para os softwares de faturação • Adesão simplificada dos comerciantes	AMA
04-10-2023	1.8	• Especificação do JWT necessário para a autenticação de um software na adesão simplificada	AMA
03-10-2024	1.9	• Alteração de urls dos ambientes	AMA

<

Índice

1 Introdução

O projeto Fatura Sem Papel (FSP) visa o desenvolvimento, implementação e suporte ao funcionamento de uma aplicação que permita a todos comerciantes, que reúnam as condições necessárias, enviar faturas eletronicamente, via e-mail, a todos os cidadãos/empresas que tenham demonstrado interesse em assim as receber ao aderir ao serviço.

Pretende-se assim contribuir para a desmaterialização das faturas, contribuindo não só para uma redução de custos evidente como também para a diminuição do consumo de papel, com claras vantagens e impactos positivos no contexto ambiental.

Este documento detalha os fluxos e especifica os serviços da FSP. Para além disso, aborda também outros temas importantes como o processo de integração.

1.1 Definições, Acrónimos e Abreviações

FSP – Fatura Sem Papel

SCAP – Sistema de Certificação de Atributos Profissionais

FA – Fornecedor de Autenticação

AMA – Agência para a Modernização Administrativa

CC – Cartão de Cidadão

CMD – Chave Móvel Digital

2 Arquitetura da Solução

A Fatura Sem Papel (FSP) está inserida no ecossistema Autenticacao.Gov (ver Figura 1), tirando proveito das funcionalidades de sistemas já existentes. Nomeadamente:

1. Fornecedor de Autenticação (FA) – responsável pela autenticação de cidadãos, podendo os cidadãos utilizar a Chave Móvel Digital (CMD) ou o Cartão de Cidadão (CC) para proceder à sua autenticação. Após correta autenticação, o FA comunica com a FSP para criação de conta de assinatura de faturas eletrónicas;
2. Sistema de Certificação de Atributos Profissionais (SCAP) – responsável pela gestão e obtenção de atributos, em particular, os empresariais de cidadãos. A FSP comunica com o SCAP para verificar se um cidadão tem o atributo necessário para criar uma conta de envio de faturas eletrónicas.

A FSP integra com estes dois sistemas no fluxo de criação de conta (ver 4.1.1). Este fluxo é iniciado pelo Software de Faturação, comunicando com o FA.

No que diz respeito aos fluxos de assinatura, são também iniciados pelo Software de Faturação, comunicando diretamente com a FSP (ver 4.2).

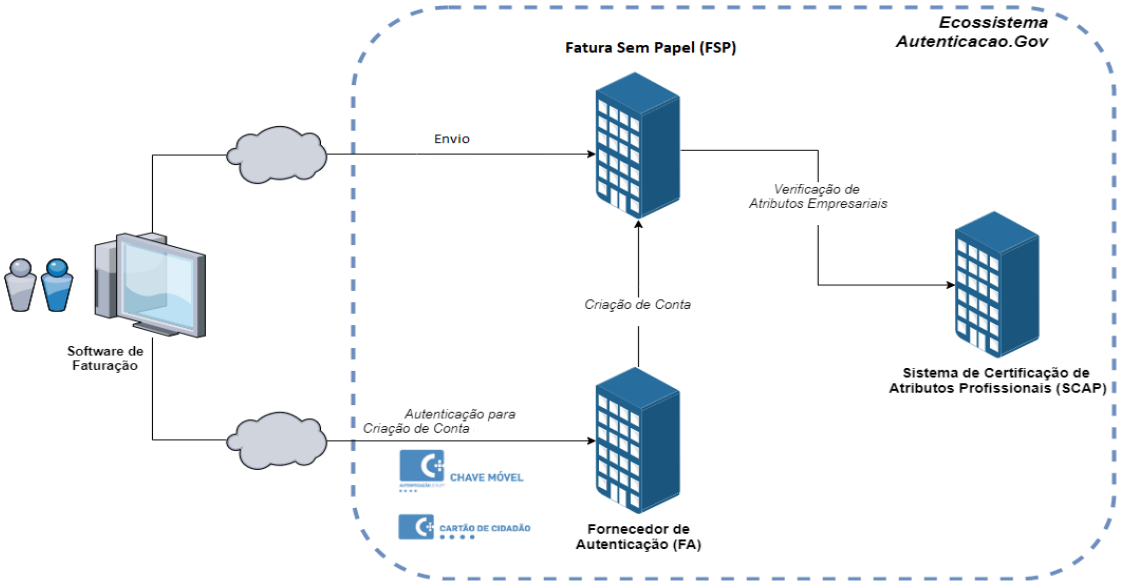


Figura 1. Ecossistema Autenticacao.Gov

3 Requisitos para utilização

3.1 Requisitos para a Empresa (ou Entidade)

- Acesso à Internet;
- Colaborador da empresa com poderes para emitir, que detenha:
    - Chave Móvel Digital + PIN de autenticação ou Cartão de Cidadão + PIN de autenticação + Leitor de Cartões;
    - Atributo “Emissão de faturas” ativo no SCAP na empresa para a qual pretende criar conta de envio de faturas.

3.2 Requisitos para o Software de Faturação

- Efetuar integração com FSP e FA;
- Passar pelo Processo de Integração e credenciação (consultar secção 6 Processo de Certificação).

4 Fluxos

Esta secção descreve os fluxos necessários para que Softwares de Faturação possam integrar com o FSP.

4.1 Fluxos de gestão de conta

4.1.1 Criação de Conta

A criação de uma conta para envio de faturas através da FSP é feita via Fornecedor de Autenticação (FA) que, após a devida autenticação do cidadão (colaborador da empresa com poderes para enviar faturas), encaminha o pedido de criação de conta de envio para a FSP. Assim, neste fluxo, o Software de Faturação comunica apenas com o cidadão e com o FA.

4.1.2 Identificador de conta

As contas para envio de faturas pela FSP são identificadas univocamente pelas seguintes componentes:

- Identificador do cidadão;
- NIPC da empresa;
- Email associado à conta
- Nome
- Identificador da instância (UUID – consultar secção 5 Geração de UUID)

4.1.3 Autenticação do Cidadão

O cidadão autentica-se perante o FA, recorrendo a um dos seguintes meios:

- Chave Móvel Digital (CMD);
- Cartão de Cidadão (CC).

Mais informação sobre autenticações com CMD e CC pode ser encontrada em:

- https://www.autenticacao.gov.pt/chave-movel-digital/autenticacao
- https://www.autenticacao.gov.pt/cartao-cidadao/autenticacao

4.1.4 Elegibilidade do Cidadão (Comerciante)

A validação da elegibilidade de um cidadão para criação de uma conta como representante de uma empresa é feita pelo Sistema de Certificação de Atributos Profissionais (SCAP), através da existência do atributo ativo "Emissão de faturas".

4.1.5 Fluxo de Criação de conta

O diagrama da Figura 2 ilustra o processo de criação de conta de envio na FSP. São em seguida descritos os passos deste fluxo:

1. Cidadão pede adesão ao serviço de envio de faturas, introduzindo os seguintes dados:
    1.1. NIPC da empresa associada à conta – obrigatório (9 dígitos);
    1.2. Email associado à conta – obrigatório;
    1.3. Nome – obrigatório;
    1.4. Identificador da instância – obrigatório (consultar secção 5 Geração de UUID).
2. Software de Faturação invoca FA para o cidadão se poder autenticar. Esta autenticação será feita através do protocolo OAuth2 e devem ser pedidos os seguintes atributos:
    2.1. http://interop.gov.pt/MDC/Cidadao/NIC (se cidadão português)
    2.2. http://interop.gov.pt/MDC/Cidadao/DocType1 (se cidadão estrangeiro)
    2.3. http://interop.gov.pt/MDC/Cidadao/DocNationality1 (se cidadão estrangeiro)
    2.4. http://interop.gov.pt/MDC/Cidadao/DocNumber1 (se cidadão estrangeiro)
    2.5. http://interop.gov.pt/MDC/Cidadao/NomeProprio
    2.6. http://interop.gov.pt/MDC/Cidadao/NomeApelido

2.7. http://interop.gov.pt/FSP/createFSPAccount?enterpriseNipc=<enterprise Nipc>$email=<email>$instanceId=<instanceId>$creationClientName=<*creationClientName (os valores entre <> devem ser substituídos pela informação introduzida no passo 1). No caso de alguma destas informações conter espaços em branco, os parâmetros do atributo (tudo o que vem depois do '?'), deve ser convertido numa string base64.*

3. FA mostra a página de autenticação no mecanismo utilizado pelo Software de Faturação (e.g. WebView ou Browser);
4. Cidadão efetua autenticação com CMD ou CC;
5. Página da autenticação envia dados para o FA;
6. FA valida a autenticação;
7. FA pede criação de conta de envio de fatura, enviando para a FSP os dados obtidos na autenticação;
8. FA devolve um token OAuth associado à autenticação efetuada;
9. Software de Faturação verifica token OAuth associado à autenticação efetuada;
10. Software de Faturação obtém token OAuth associado à autenticação efetuada;
11. FSP pede ao SCAP os atributos empresariais do cidadão na empresa para a qual pretende criar conta de envio;
12. SCAP devolve atributos empresariais do cidadão na empresa para a qual pretende criar conta de envio;
13. FSP valida se o cidadão tem o atributo "Emissão de Faturas" na empresa para a qual pretende criar conta de assinatura;
14. FSP cria conta de envio;
15. Software de Faturação invoca FA com o token OAuth obtido no passo 10, de forma a obter a informação sobre a conta de envio de fatura. Antes de fazer esta invocação, o Software de Faturação deve esperar **15 segundos**;
16. FA valida token OAuth recebido;
17. FSP devolve informação de conta de envio de fatura para o FA.
18. FA devolve informação de conta de envio fatura para o Software de Faturação

19. Software de Faturação guarda informação de conta de envio de fatura;
20. Software de Faturação mostra mensagem de sucesso ao cidadão.

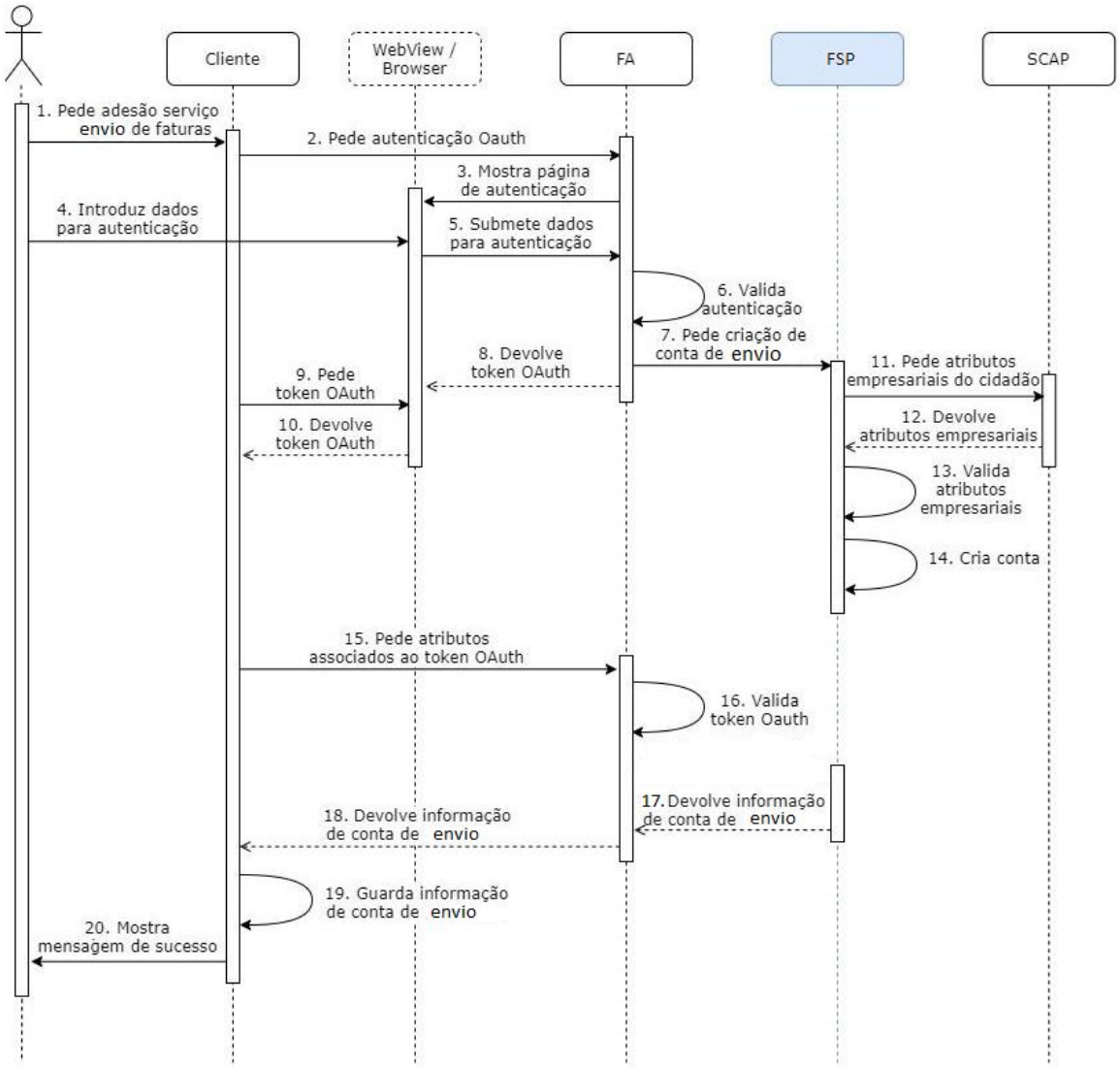


Figura 2. Fluxo de criação de conta

4.1.6 Estrutura da resposta de informação de conta

O FA devolve a informação de conta para o Software de Faturação. Esta informação é enviada em formato JSON como valor do atributo http://interop.gov.pt/FSP/createFSPAccount. Em caso de sucesso na criação de conta, é enviada uma string base64, cuja estrutura é um JSON com as seguintes propriedades

- Token de acesso às operações de assinatura (accessToken);
- Token para atualização de tokens (refreshToken);
- Data de expiração do refresh token (expirationDate).

```
{
  accessToken: string,
  refreshToken: string,
  expirationDate: DateTime (yyyy-MM-ddTHH:mm:ss.fffffffZ)
}
```

Em caso de erro na criação de conta, são enviados os atributos:

- Erro (error);
- Descrição do erro (error_description);

As causas possíveis para se obter um erro, são (error – error_description):

- Bad Request - Invalid parameter citizenDocId
- Bad Request - Missing parameter citizenDocId
- Bad Request - Invalid parameter citizenDocType
- Bad Request - Missing parameter citizenDocType
- Bad Request - Invalid parameter citizenDocCountry

- Bad Request - Missing parameter citizenDocCountry
- Bad Request - Invalid parameter enterpriseNipc
- Bad Request - Missing parameter enterpriseNipc
- Bad Request - Missing parameter citizenGivenName
- Bad Request - Missing parameter citizenLastName
- Bad Request - Invalid parameter email
- Bad Request - Missing parameter creationClientName
- Bad Request - Invalid parameter instanceId
- Bad Request – Missing parameter instanceId
- Bad Request - Client is not active
- Missing required enterprise attributes - The citizen attributes obtained are not valid
- Internal Server Error - error_description: Unexpected error while processing client request

4.1.7 Cancelamento

O pedido de cancelamento de conta deverá ser efectuado utilizando a seguinte especificação:

Endpoint: <url base da API>/Seller

Método HTTP: DELETE

Headers: Authorization : Bearer <JWT access token>

A Figura 3 ilustra o processo cancelamento de uma conta de assinatura:

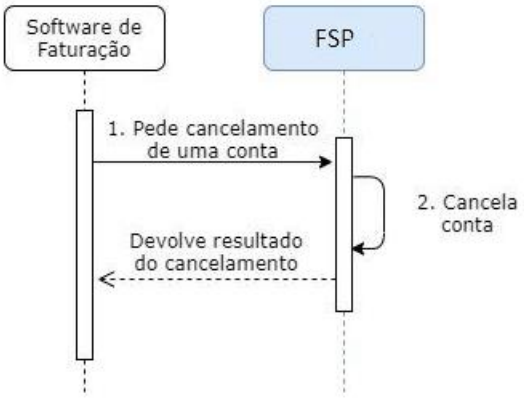


Figura 3 – Fluxo de Cancelamento de Conta.

Em caso de sucesso no pedido de cancelamento da conta, é devolvido um JSON com as seguintes propriedades:

- Resultado da operação(result);

Exemplo:

```
{
  result: string
}
```

Em caso de erro na criação de conta, são enviados os atributos:

- Erro (error);
- Descrição do erro (error_description).

As causas possíveis de erro são (error – error_description):

Unauthorized - Invalid token

- Internal Server Error - error_description: Unexpected error while processing client request

4.1.8 Alteração de dados do comerciante

O pedido para alteração de dados da conta do comerciante (nome e/ou e-mail) deverá ser efectuado utilizando a seguinte especificação:

Endpoint: <*url base da API*>/Seller

Método HTTP: *PUT*

Headers: Authorization : Bearer <JWT access token>

Payload: <JSON contendo os parâmetros de envio>

Os parâmetros necessários deverão ser enviado no corpo do pedido HTTP, em formato JSON e são os seguintes:

- name – nova designação pretendida para o comerciante *(opcional, se email estiver preenchido)*
- email – novo e-mail do comerciante *(opcional, se name estiver preenchido)*

Exemplo:

```
{
  name: string,
  email: string
}
```

A Figura 7 ilustra o pedido de envio.

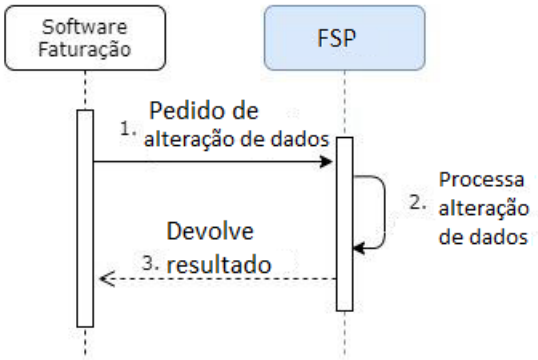


Figura 4 - Fluxo de Alteração de Conta

Em caso de sucesso no pedido de alteração de dados de conta do comerciante, é devolvido um JSON com as seguintes propriedades:

- Resultado da operação(result);

Exemplo:

{

result: string

}

Em caso de erro na criação de conta, são enviados os atributos:

- Erro (error);
- Código do erro (code)
- Descrição do erro (error_description)

As causas possíveis de erro são (*error* – code - *error_description*):

a) Unauthorized - Invalid token

b) Bad Request – 416 – Missing parameter update

c) Bad Request – 415 – Provider not found

d) Internal Server Error - error_description: Unexpected error while processing client request

4.2 Tokens

4.2.1 Access Tokens

Token necessário para efetuar operações de envio no FSP. Este token é do tipo Bearer e deve ser passado no header Authorization dos pedidos.

Por uma questão de segurança, o AccessToken tem uma validade reduzida (24 horas), definida pela FSP. Sempre que for invocado um método do FSP com um AccessToken expirado, a FSP retorna um erro HTTP 400 Bad Request, com a mensagem de erro “The access or refresh token is expired or has been revoked”. Nestes casos, o Software de Faturação deve invocar o método /Token, de modo a ser gerado um novo accessToken. Estes novos tokens devem ser utilizados nas invocações futuras ao serviço.

4.2.2 Refresh Tokens

Token necessário para invocar o método /Token. Este token é do tipo Bearer e deve ser passado no header Authorization dos pedidos. O resultado da invocação do método /Token é a geração de um novo accessToken. Estes novos tokens devem ser utilizados nas invocações futuras ao serviço.

Sempre que for invocado um método do FSP com um RefreshToken expirado, a FSP retorna um erro HTTP 400 Bad Request, com a mensagem de erro “The access or refresh token is expired or has been revoked”. Nestes casos, o Software de Faturação deve voltar a obter novos tokens através do fluxo de criação de conta.

4.2.3 Atualizar Token

Método que retorna um novo *AccessToken* e um novo *RefreshToken* para uma conta de envio. Estes novos tokens devem ser utilizados nas invocações futuras aos serviços.

Este método deve ser invocado sempre que o sistema retorne o erro HTTP **400 Bad Request**, com a mensagem de erro “*The access or refresh token is expired or has been revoked*”. A Figura 4 ilustra o processo de atualização de tokens.

O pedido deverá ser efectuado utilizando a seguinte especificação:

Endpoint: <*url base da API*>/token?access_token={access_token}&refresh_token={refresh_token}

Método HTTP: *PUT*

Os parâmetros necessários são:

- access_token - token de acesso actual do comerciante *(obrigatório)*
- refresh_token- refresh token do comerciante *(obrigatório)*

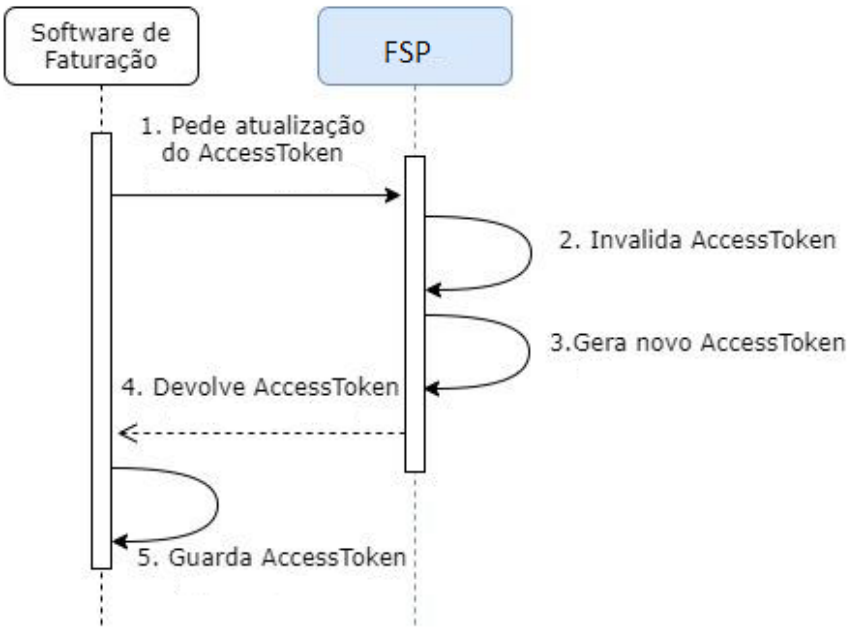


Figura 5 - Fluxo de Atualização de Token

Em caso de sucesso no pedido de atualização , é devolvido um JSON com as seguintes propriedades:

- Novo token de acesso (access_token);
- Refresh (refresh_token);
- Data expiração do token (expires_in)

Exemplo:

```
{
  access_token: string,
  refresh_token: string
  expires_in: DateTime
}
```

Em caso de erro são enviados os atributos:

- Erro (error);
- Código de erro (code)
- Descrição do erro (error_description).

As causas possíveis de erro são (error – error_description):

- Unauthorized - Invalid token
- Bad Request – 417 - Invalid parameter accessToken
- Bad Request – 418 - Invalid parameter refreshToken
- Bad Request – 419 – Expired refresh token
- Bad Request – 420 – Mismatch between tokens
-
- Internal Server Error - error_description: Unexpected error while processing client request

4.3 Fluxos de Envio

4.3.1 Obter Cifra

O pedido para obter a cifra de um cidadão , que será utilizada para proteger os pdf que serão enviados pelo software de faturação, deverá ser efetuado utilizando a seguinte especificação:

Endpoint: <*url base da API*>/Cypher?id={citizen_doc_id}&type={citizen_doc_type}

Método HTTP: *GET*

Headers: Authorization : Bearer <JWT access token>

Os parâmetros necessários são:

- nif – número de contribuinte do cidadão *(obrigatório)*

A Figura 5 ilustra o pedido para obter a cifra de um cidadão.

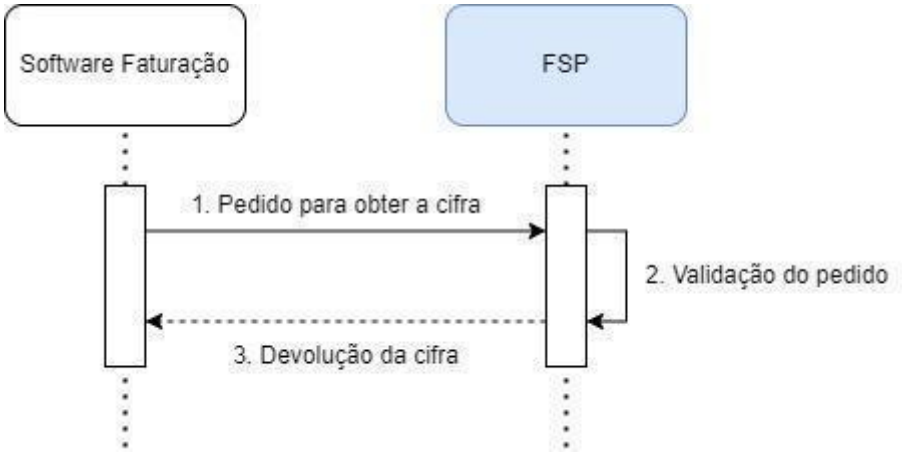


Figura 6 - Fluxo de pedido de cifra.

Em caso de sucesso no pedido de obtenção da cifra, é devolvido um JSON com as seguintes propriedades:

- Identificador da instância (instanceId);
- Cifra (cypher);

No caso em que o cidadão/empresa não definiu nenhuma cifra no FSP este campo deverá vir a null.

Exemplo:

```
{
  instanceId: uuid,
  cypher: string
}
```

Em caso de erro na criação de conta, são enviados os atributos:

- Erro (error);
- Código de erro (code)
- Descrição do erro (error_description)

As causas possíveis de erro são (error – code - error_description):

- Unauthorized - Invalid token
- Bad Request – 413 - Invalid parameter nif
- Bad Request – 414 - Missing parameter nif
- Bad Request – 415 - Provider not found
- Internal Server Error - error_description: Unexpected error while processing client request

4.3.2 Envio

O pedido para envio de faturas deverá ser efectuado utilizando a seguinte especificação:

Endpoint: <*url base da API*>/Invoice

Método HTTP: *POST*

Headers: Authorization : Bearer <JWT access token>

Payload: <JSON contendo os parâmetros de envio>

Os parâmetros necessários deverão ser enviado no corpo do pedido HTTP, em formato JSON e são os seguintes:

- clientId - id do documento de identificação fiscal do cidadão *(obrigatório)*
- enterpriseNipc - id do documento de identificação do comerciante *(obrigatório)*
- invoice - PDF da fatura (no formato Base64) *(obrigatório)*
- fileName - descrição da fatura *(obrigatório, máximo de 255 caracteres)*
- localId - Identificador da fatura por parte do Software de Faturação *(obrigatório)*
- EmissionDate - Data da emissão da Fatura no Software de Faturação *(obrigatório)*
- collaboratorId - id do documento de identificação fiscal do colaborador de uma empresa *(facultativo)*

Exemplo:

```
{
  clientId: string,
  enterpriseNipc: string,
  invoice: base64,
 description: string,
 localId: string,
 emissionDate: timestamp,
 collaboratorId: string
}
```

NOTA: O tamanho máximo suportado da fatura é de 30Mb, pelo qual deverá ser sempre enviado faturas abaixo do tamanho indicado.

Quando uma fatura é emitida a uma empresa e um colaborador solicitar receber uma cópia da fatura no seu email, basta indicar também o seu NIF. Para que essa cópia seja enviada ao colaborador, o NIF do mesmo deverá ser enviado no campo “collaboratorId”.

Caso o colaborador não seja aderente ao FSP, a fatura será enviada apenas para o NIF da fatura indicado.

A Figura 6 ilustra o pedido de envio.

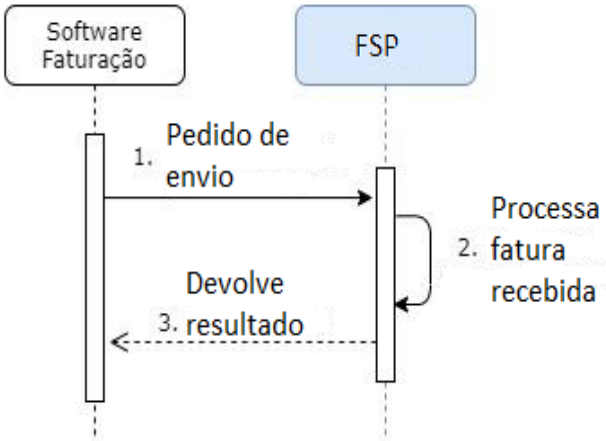


Figura 7 - Fluxo de Pedido de Envio

Em caso de sucesso no pedido de envio da fatura, é devolvido um JSON com as seguintes propriedades:

- Identificador da fatura no sistema FSP (id);
- Resultado da operação(result);

Exemplo:

{

id: uuid,

result: string

}

Em caso de erro na criação de conta, são enviados os atributos:

- Erro (error);
- Código do erro (code)
- Descrição do erro (error_description)

As causas possíveis de erro são (*error* – code - *error_description*):

e) Unauthorized - Invalid token

f) Bad Request – 411 – Invoice already submited

g) Bad Request – 402 - Invalid parameter clientId

h) Bad Request – 407 - Missing parameter clientId

i) Bad Request – 403 - Invalid parameter enterpriseNipc

j) Bad Request – 408 - Missing parameter enterpriseNipc

k) Bad Request – 404 - Invalid parameter invoice

l) Bad Request – 406 - Missing parameter emissionDateBad Request – 409 - Missing parameter invoice

m) Bad Request – 405 - Invalid parameter fileName

n) Bad Request – 410 - Missing parameter fileName

o) Bad Request – 412 – Generic error

p) Bad Request – 422 – *Invalid parameter collaboratorId*

q) Internal Server Error - error_description: Unexpected error while processing client request

4.4 Fluxo de reenvio de Faturas com mais de 48h

Para realizar o reenvio de faturas com mais de 48h, em primeiro lugar é necessário que o SW tenha a informação da/s fatura/s a reenviar. Posteriormente, será feito um pedido de reenvio à semelhança do pedido de envio de faturas.

Para que o SW obtenha a informação das faturas a reenviar, existem duas formas:

- O SW invoca um pedido para obter a listagem de faturas a reenviar (Método Pull);
- É enviado uma notificação ao SW que uma determinada fatura terá de ser reenviada (Método Push).

4.4.1 Método Pull

O pedido para obter a lista de faturas a reenviar deverá ser efetuado da seguinte forma:

Endpoint: <*url base da API*>/Invoice

Método HTTP: *GET*

Headers: Authorization : Bearer <JWT access token>

Payload: <JSON contendo os parâmetros de envio>

Em caso de sucesso no pedido de obter as faturas a reenviar, é devolvido uma lista de objetos JSON com as seguintes propriedades:

- Identificador da fatura (*localId*);
- Data de emissão da fatura (*emissionDate*);
- Estado da fatura(*state*);

Exemplo:

[{

"localId": "FS_Serie1_44",

"emissionDate": "2023-04-05T15:17:00+00:00",

"state": "pending_resend"

}]

4.4.2 Método Push

Em desenvolvimento.

4.4.3 Reenvio

O processo de reenvio de uma fatura é muito semelhante ao envio já explicado no ponto anterior.

O pedido para o reenvio de faturas deverá ser efectuado utilizando a seguinte especificação:

Endpoint: <*url base da API*>/Invoice/Resend

Método HTTP: *POST*

Headers: Authorization : Bearer <JWT access token>

Payload: <JSON contendo os parâmetros de envio>

Os parâmetros necessários deverão ser enviado no corpo do pedido HTTP, em formato JSON e são os seguintes:

- *clientId - id do documento de identificação fiscal do cidadão (obrigatório)*
- *enterpriseNipc - id do documento de identificação do comerciante (obrigatório)*
- *invoice - PDF da fatura (no formato Base64) (obrigatório)*
- *fileName* - nome do ficheiro *(obrigatório)*
- *localId - Identificador da fatura por parte do Software de Faturação (obrigatório)*
- *emissionDate - Data da emissão da Fatura no Software de Faturação (obrigatório)*

4.5 Partilha do estado de Faturas

Para a partilha dos estados das faturas com SW, existem duas formas:

- O SW invoca um pedido para obter a listagem de faturas e respetivos estados (Método Pull);
- É enviado uma notificação ao SW que uma determinada fatura possui um determinado estado (Método Push).

4.5.1 Método Pull

O pedido para obter a lista de faturas com o respetivo estado deverá ser efetuado da seguinte forma:

Endpoint: <*url base da API*>/Invoice/list

Método HTTP: *GET*

Headers: Authorization : Bearer <JWT access token>

Payload: <JSON contendo os parâmetros de envio>

É possível realizar filtragens para obter os estados das faturas. Para tal, os parâmetros necessários deverão ser enviado no corpo do pedido HTTP, em formato JSON e são os seguintes:

- *InvoiceState* - Estado atual da fatura no FSP
- *EmissionDate - Data da emissão (caso preenchido, serão filtradas todas as faturas com data de emissão posterior à indicada).*
- *PageNumber - Nº da página a apresentar a listagem (para paginação, por omissão é 1).*
- *PageSize* - Nº de itens por página a apresentar (para paginação, por omissão é 5).

Exemplo:

{

InvoiceState: string,

EmissionDate : string,

PageNumber: int,

PageSize: int

}

Tipos de estados possíveis para filtrar:

sent - Fatura enviada

sendPending - Fatura pendente de envio.

sendUnsuccessful - Falha no envio da Fatura.

resendPending - Fatura pendente de reenvio.

resendUnsuccessful - Falha no reenvio da Fatura.

resent - Fatura reenviada.

Em caso de sucesso no pedido de obter a listagem das faturas com o respetivo estado, é devolvido um objeto JSON com as seguintes propriedades:

- Nº total de faturas (*count*);
- Lista de faturas (*items*);
    - Identificador da fatura (*localId*);
    - Estado da Fatura (*state*);
    - Data do último envio da fatura (*sendDate*);
    - Nº de tentativas de envio da fatura (*totalResendAttempts*)

Exemplo:

{

"count": 51,

"items": [

{

"localId": "FT_Serie1_52",

"state": "sent",

"sendDate": "2023-04-18T17:16:32.940823+00:00",

"totalResendAttempts": 1

}]}

4.5.2 Método Push

Em desenvolvimento.

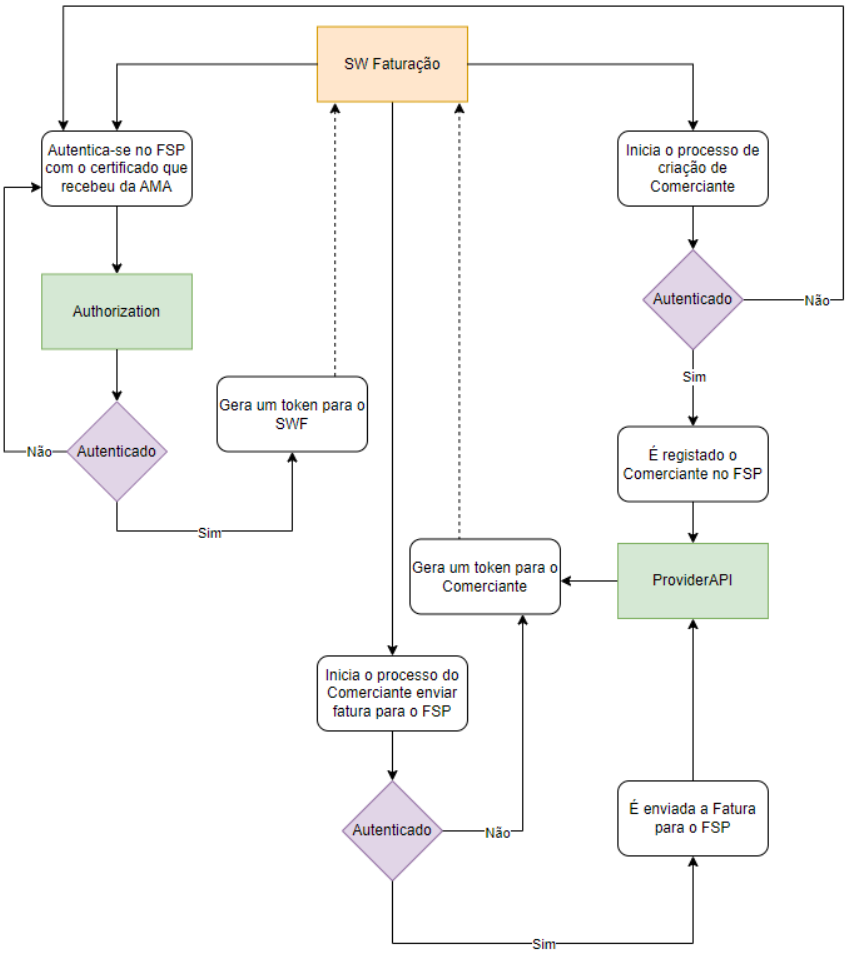
4.6 Adesão Simplificada de Comerciantes

Para que seja possível realizar a adesão simplificada de comerciantes, em primeiro lugar, será necessário solicitar à AMA um certificado de autenticação, para que posteriormente o Software de Faturação possa efetuar pedidos ao FSP de uma forma segura.

Depois de solicitado, o certificado será enviado pela AMA. É com este certificado que o Software de Faturação se irá autenticar para que de seguida possa registar comerciantes. Para além disso, a AMA realizará um registo no FSP de alguns dados relevantes do Software de Faturação, tais como: ***Nome, Instância, NIPC, Email e Certificado***.

Do ponto de vista do Comerciante, este não terá de se registar nem autenticar, visto que o processo está salvaguardado pelo Software de Faturação.

De modo a facilitar a compreensão deste fluxo, o mesmo segue ilustrado abaixo:



4.6.1 Autenticar Software de Faturação

O pedido para o SW de Faturação se autenticar deverá ser efetuado da seguinte forma:

Endpoint: <*url base da API*>/Sw/auth

Método HTTP: *POST*

Headers: CertThumbPrint: <Thumbprint do certificado>

CertToken: <JWT com as claims (instanceId e Nipc), assinado com a chave privada do certificado>

A claims necessárias para enviar no token JWT são as seguintes:

- *InstanceId - Instância do Software de Faturação.*
- *Nipc - NIPC associado ao Software de Faturação.*
-

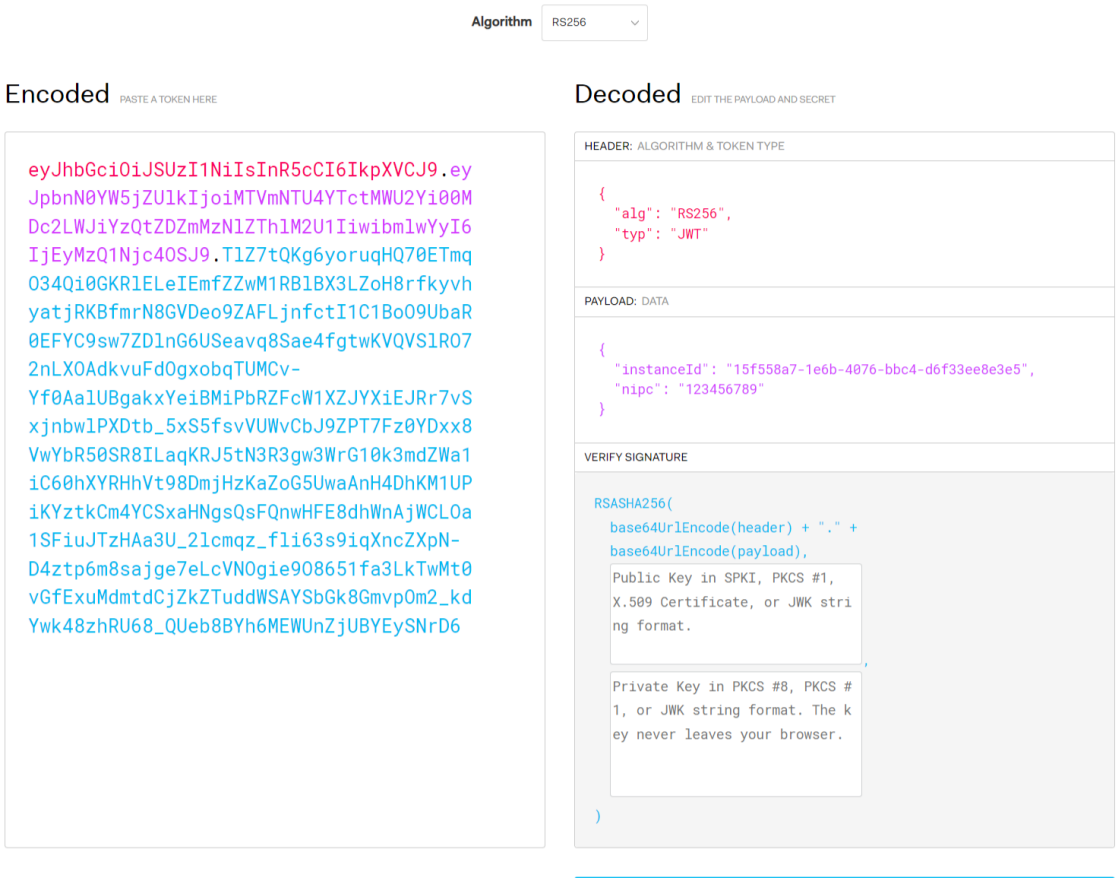
Para que o SW se autentique, estes atributos mencionados acima, terão de corresponder com os que a AMA registou aquando da criação da ficha de SW de Faturação.

O token JWT deverá ser criado utilizando o algoritmo RS256 e é formado por:

base64UrlEncode(header).base64UrlEncode(payload).base64UrlEncode(assinatura)

A assinatura é gerada com base na string correspondente: base64UrlEncode(header) + "." + base64UrlEncode(payload) assinada com chave privada do certificado.

Exemplo de geração de um JWT:



Exemplo dos headers necessários:

CertThumbPrint	B47F47F5DD4A43ADF9794E301E75D579743B6F5D
CertToken	eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpbnN0YW5jZUlkIjoiMTVmNTU4YTktMWU2Yi00MDc2LWJiYzQtZDZmMzNlZThlM2U1IiwibmlwYyI6IjEyMzQ1Njc4OSJ9.TGGntBeJCaGYC1v4RgMsiJbjwx0niZ8p8n6eiQm9-0p9c36om7-Du1r5jGUaiOBTNoswNElcpqwo_fMMOQo_WTkvMuA_FvaNleLszieH82V0Lr2SSW7pJylr7zK7jCPiQ7sWiPaYu8czbSjw1Dp-5G1AwVWGFPbvJxSUQH5FoJAPsufLeFubyliC6l_Kcjw2h334l3MFwlHp-VPO-pdElF3P2m5lyxeM9mVPM5xlQHFZB-SbjtyDK7aAWVSrhcyYi5mnqFlX1bHw6EkUVObXtNouKc3idVtweiJTKEgtajwH_4YqDJkBvoYFcZ0rpiibTlyY2bV3MOkGHYqB8OFVVQ
Key	

Em caso de sucesso no pedido de autenticação, é devolvido uma resposta OK com o código 200 contendo no corpo um token. **É com este token que o SW de Faturação irá poder registar Comerciantes no FSP.**

As causas possíveis de erro são (*error – code – error_description*):

Bad Request - 424 - Invalid certificate

Bad Request - 423 - Software provider not found

Bad Request - 412 - Unable to create software token

Internal Server Error - 500 - error_description: Unexpected error while processing client request.

Sempre que um token de Software expirar, poderá sempre ser renovado chamando o mesmo método de Atualizar Tokens mencionado nos capítulos anteriores relativos aos Tokens.

4.6.2 Registar comerciantes associados ao Software de Faturação

O pedido para o SW de Faturação registar comerciantes deverá ser efetuado da seguinte forma:

Endpoint: <*url base da API*>/Sw/seller

Método HTTP: *POST*

Headers: Authorization : Bearer <JWT access token do Software de Faturação>

Payload: <JSON contendo os parâmetros de envio>

Os parâmetros necessários deverão ser enviados no corpo do pedido HTTP, em formato JSON e são os seguintes:

- *InstanceId - Instância do Software de Faturação.*
- *EnterpriseNipc - NIPC do Comerciante a registar.*
- *ClientName - Nome do Comerciante.*
- *Email - Email do Comerciante.*

Exemplo:

{

InstanceId : Guid,

EnterpriseNipc : string,

ClientName : string,

Email : string

}

Em caso de sucesso no pedido de registo de comerciante, é devolvido uma resposta OK com o código 200 contendo no corpo um token. **É este token do Comerciante que o SW de Faturação terá de utilizar para enviar as faturas via FSP.**

As causas possíveis de erro são (*error – code – error_description*):

Unauthorized - 401 - Invalid token

Bad Request - 412 - Unexpect error creating seller

Bad Request - 412 - Unable to create seller's token

Internal Server Error - 500 - error_description: Unexpected error while processing client request

Sempre que um token de Comerciante expirar, poderá sempre ser renovado chamando o mesmo método de Atualizar Tokens mencionado nos capítulos anteriores relativos aos Tokens.

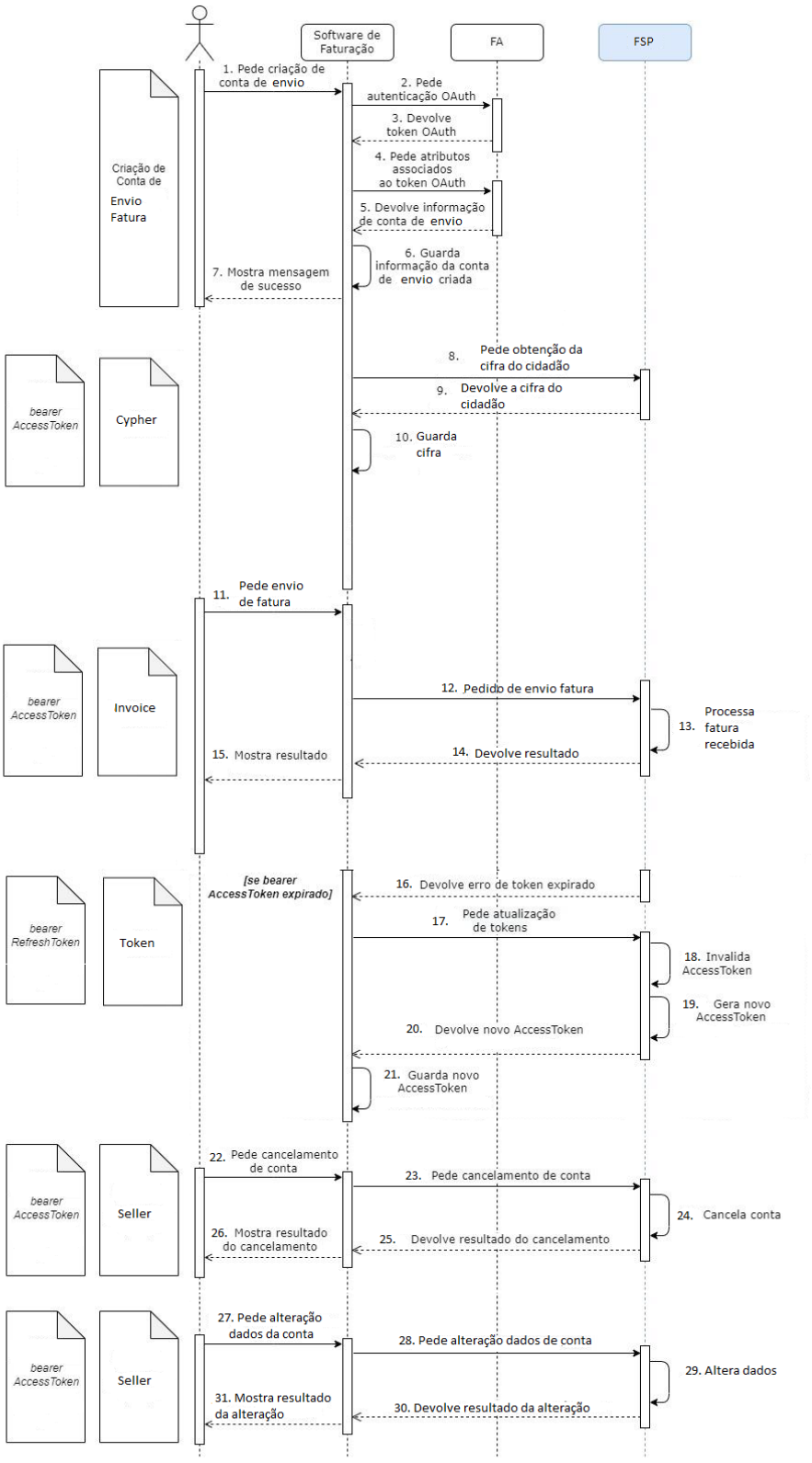
Depois do Comerciante estar registado no FSP, o processo de envio/reenvio de faturas mantêm-se uma vez que o SW de faturação está a enviar ao FSP o token do comerciante.

4.6.3 Aviso de certificado a expirar

De forma a garantir que todos os certificados registados sejam válidos, o FSP tem um mecanismo autónomo que controla a validade dos certificados. Isto é, **sempre que um certificado esteja a 31 dias ou menos de expirar**, o FSP irá enviar um Email, que servirá de notificação, ao email que está associado ao registo do SW de faturação que a AMA realizou.

Após o certificado ter expirado, a rotina de envio de email de notificação deixará de ser executada.

4.7 Fluxo Típico


Software de Faturação
FA
FSP
1. Pede criação de conta de envio
2. Pede autenticação OAuth
3. Devolve token OAuth
4. Pede atributos associados ao token OAuth
5. Devolve informação de conta de envio
6. Guarda informação da conta de envio criada
7. Mostra mensagem de sucesso
Criação de Conta de Envio Fatura
8. Pede obtenção da cifra do cidadão
9. Devolve a cifra do cidadão
10. Guarda cifra
bearer AccessToken
Cypher
11. Pede envio de fatura
12. Pedido de envio fatura
13. Processa fatura recebida
14. Devolve resultado
15. Mostra resultado
bearer AccessToken
Invoice
[se bearer AccessToken expirado]
16. Devolve erro de token expirado
17. Pede atualização de tokens
18. Invalida AccessToken
19. Gera novo AccessToken
20. Devolve novo AccessToken
21. Guarda novo AccessToken
bearer RefreshToken
Token
22. Pede cancelamento de conta
23. Pede cancelamento de conta
24. Cancela conta
25. Devolve resultado do cancelamento
26. Mostra resultado do cancelamento
bearer AccessToken
Seller
27. Pede alteração dados da conta
28. Pede alteração dados de conta
29. Altera dados
30. Devolve resultado da alteração
31. Mostra resultado da alteração
bearer AccessToken
Seller

5 Geração de UUID

Um Universally Unique IDentifier (UUID) é um número de 16 octetos (128 bits).

Na sua forma canónica, um UUID é representado por 32 dígitos em formato hexadecimal, exibidos em cinco grupos separados por hífens, na forma 8-4-4-4-12 para um total de 36 caracteres (32 caracteres alfanuméricos e 4 hífens, utilizando exclusivamente letras minúsculas).

Por exemplo:

- 123e4567-e89b-12d3-a456-426655440000 – Corresponde a um UUID.
- 123E4567-E89B-12D3-A456-426655440000 – Não corresponde a um UUID.

A geração deve seguir a norma da especificação (disponível em https://www.ietf.org/rfc/rfc4122.txt).

6 Especificação dos Serviços

Os ficheiros que contêm as especificações do serviço encontram-se disponíveis no repositório da Fatura Sem Papel em https://github.com/amagovpt/doc-FSP. Estes documentos estão formatados segundo a especificação da OpenAPI (https://swagger.io/specification) e podem ser lidos por qualquer ferramenta de leitura de especificações OpenAPI (e.g. https://editor.swagger.io).

A comunicação entre o Software de Faturação e o FSP deve ser feita através do protocolo HTTPS com autenticação Bearer Token.

6.1 Ambientes

Os métodos que constam na especificação estão publicados nos ambientes que constam da Tabela 1.

Ambiente	URL
Pré-Produção	https://providerapi.ppr-faturas.digital.gov.pt/
Produção	https://providerapi.faturas.digital.gov.pt/

Tabela 1 - Ambientes

7 Processo de Certificação

Em atualização.

8 Guidelines de Integração

As guidelines de integração encontram-se disponíveis no repositório da Fatura Sem Papel em https://github.com/amagovpt/doc-FSP