

Manual de API de Proveedores

FACe, Punto General de
Entrada de Facturas
Electrónicas de la
Administración General
del Estado



TLP: CLEAR

AUTOR	SGAD
ÁREA	Catálogos
PROYECTO	FACe, Punto General de Entrada de Facturas de la Administración General del Estado
NOMBRE DEL DOCUMENTO	Manual de API de Integradores

Control de Versiones del Documento

VERSIÓN	AUTOR	FECHA	DESCRIPCIÓN
2.0.0	Equipo FACe	19/11/2025	<ul style="list-style-type: none"> Cambio de plantilla Revisión y corrección de errores
1.0.0	Equipo FACe	03/1/2024	Versión inicial del documento

Índice

1	INTRODUCCIÓN	5
1.1	Objetivos	5
2	CONSIDERACIONES PREVIAS	6
2.1	Alta como integrador	6
2.2	Catálogo de APIs	6
2.3	Conceptos de autenticación	7
3	DESCRIPCIÓN DEL API	9
3.1	Grupo Facturas	9
3.1.1	Facturas	9
3.1.1.1	Remisión de facturas [POST]	9
3.1.1.2	Remisión de facturas a Repositorio [POST]	13
3.1.1.3	Búsqueda de una factura [GET]	17
3.1.1.4	Información del estado de las facturas [GET]	20
3.1.1.5	Solicitud de anulación de una factura [POST]	23
3.1.2	Cesión de crédito	25
3.1.2.1	Obtener facturas con cesión de crédito [GET]	25
3.1.2.2	Obtener fichero Facturae [GET]	29
3.1.2.3	Obtener anexos de la factura [GET]	31
3.1.2.4	Obtener factura europea [GET]	33
3.2	Grupo Información para Proveedores	35
3.2.1	Relaciones	35
3.2.1.1	Listado de Relaciones [GET]	35
3.2.1.2	Listado de Relaciones en formato CSV [GET]	37
3.2.2	Estadísticas [GET]	39
3.2.3	FAQS [GET]	40
3.3	Grupo Notificaciones	42
3.3.1	Gestión de notificaciones y noticias	42
3.3.1.1	Listado de notificaciones [GET]	42
3.3.1.2	Listado de noticias [GET]	44
3.3.1.3	Listado de noticias cortas [GET]	45
	ANEXO I: XML PARA REMITIR FACTURAS A REPOSITORIO	47
	ANEXO II: EJEMPLO DEL CÓDIGO EN JAVA PARA GENERAR EL TOKEN	48

RELACIÓN DE GRÁFICOS

Ilustración 1. Archivo CSV con las relaciones.....38

1 INTRODUCCIÓN

1.1 Objetivos

El presente documento tiene como objetivo servir de manual de uso de las APIs expuestas por el Punto General de Entrada de Facturas Electrónicas de la AGE, conocido como FACE.

En el apartado 2 Consideraciones previas se tratarán aspectos relacionados con el alta como integrador dentro del sistema FACE y de seguridad como es el modelo de autenticación que expone el API.

2 CONSIDERACIONES PREVIAS

2.1 Alta como integrador

Para poder utilizar los servicios web REST de FACE el usuario se tiene que dar de alta previamente en el **Portal de Integradores** cuyas URL son:

Servicios Estables	Producción
https://se-integradores-face.redsara.es/inicio	https://integradores.face.gob.es/inicio

En dicho portal podrá:

- Darse de alta como integrador
- Generar plataformas, tantas como se necesiten, y asociarles certificados (PEM)
- Gestionar autorizados.

Los certificados tendrán que ser los que se utilicen para firmar las peticiones SOAP para que FACE pueda verificar quién está llamando a los servicios web y si tiene permisos para ello.

2.2 Catálogo de APIs

Puede encontrar las APIs en las siguientes rutas:

- Servicios estables: <https://se-api-face.redsara.es/providers/doc>
- Producción: <https://api.face.gob.es/providers/doc>

En su versión V1 FACE expone la siguiente API:

- /v1/providers/

2.3 Conceptos de autenticación

Las invocaciones a la API están securizadas mediante tokens a través de un JWT firmado con JWS. El integrador deberá generar un token que tendrá que incluirse en la cabecera de las peticiones. El token no es más que la firma cifrada que permita al API identificar al usuario.

- **JWT (JSON Web Token):** El token en sí. Un JWT es un estándar abierto (RFC 7519, <https://datatracker.ietf.org/doc/html/rfc7519>) que define una forma compacta y segura de representar afirmaciones entre dos partes como un objeto JSON.
- **JWS (JSON Web Signature):** El mecanismo de firma. Es un estándar relacionado con JWT que define una forma de firmar digitalmente un mensaje JSON. La firma digital garantiza la integridad y autenticidad del mensaje.

El tiempo de validez del token será de 5 minutos. Una vez pasado este tiempo de validez, el servidor no permitirá más el acceso a recursos con dicho token y deberá generarse un nuevo token.

Un JWT firmado con JWS consta de tres partes codificadas en base 64 y separadas por puntos:

- **Header:** Un JSON que contiene información sobre el tipo de token (JWT), el algoritmo de firma utilizado y otros metadatos. En nuestro caso es un JSON con el tipo de token, su codificación (RSA Signature with SHA-256 (RS256) o HMAC with SHA-256 (HS256)) y una clave xc5 cuyo valor es un array con el que llamaremos PEM limpio, y que es el PEM, o parte publica, del certificado con el que se va a firmar en base 64, en una línea, sin espacios ni saltos de línea, ni los tags de -----BEGIN/END CERTIFICATE-----. Todo ello, codificado a base 64:

```
{  
  "typ": "JWT",  
  "alg": "RS256",  
  "x5c": [ "(PEM limpio )" ]  
}
```

- **Payload:** Un JSON con los claims o atributos, que son pares clave-valor que definen al token y contienen información sobre la entidad a la que pertenece el token. Todo ello, codificado a base 64:

```
{
  "username": "(Hash o huella digital SHA1 del PEM limpio)"
  "iat": "(fecha de creación del token en formato de tiempo
UNIX)",
  "exp": "(fecha de expiración del token en formato de
tiempo UNIX.)"
}
```

- Signature: Un valor calculado sobre el header y el payload. Se concatenan las representaciones base64 del header y payload, separadas por un punto. Se calcula el hash de la cadena concatenada utilizando el algoritmo SHA-256. Se utiliza la clave privada RSA para firmar el hash. Se codifica en base64 la firma resultante y el resultado es la signature.

Tiene más información, una herramienta de creación de token JWT, decodificadores y generadores de JWT, conversor de tiempo unix y de hash en las páginas:

- <https://jwt.io/>
- <https://jwt.one/https://tribestream.io/tools/jwt/>
- <https://www.jstoolset.com/jwt>
- https://time.is/es/Unix_time_converter
- <https://www.convertstring.com/es/Hash/SHA1>

Para mayor privacidad en la generación del JWT tienen bibliotecas de lenguajes de programación como:

- Node.js: jsonwebtoken o Passport.js
- PHP: Firebase PHP Admin SDK, Lcobucci JWT, Sodium PHP, RobRichards/jwt
- Python: PyJWT
- Java: Nimbus JOSE+JWT jjwt
- .NET: System.IdentityModel.Tokens.Jwt

En el [Anexo I](#) se muestra un ejemplo del código en java para generar el token.

3 DESCRIPCIÓN DEL API

3.1 Grupo Facturas

Los recursos ubicados bajo la ruta */invoices* están relacionados con la gestión de las facturas.

3.1.1 Facturas

Las facturas podrán ser consultadas por los proveedores y evolucionadas por los organismos adheridos a FACe.

3.1.1.1 Remisión de facturas [POST]

Realiza el proceso de remisión y registro de una factura.

- API Invoice

/v1/invoices

- Parámetros de body
 - filename (string, obligatorio): nombre asignado a la factura.
 - content (string, obligatorio): XML de la factura codificada en Base64.
 - email (string, obligatorio): dirección de email asociada a la remisión de factura.
 - attachments
 - filename (string, opcional): nombre del anexo.
 - content (string, opcional): PDF o HTML del anexo codificado en Base64.


```
"alias": "string",
},
"repository": true,
"hasRcfFace": true,
"status": "string",
"statusComment": "string",
"statusHistory": [
  {
    "code": "string",
    "name": "string"
  }
],
"cancellationRequestStatus": "string",
"cancellationRequestStatusCode": "string",
"cancellationRequestComment": "string",
"serial": "string",
"number": "string",
"serialNumber": "string",
"amount": 0,
"currency": "string",
"expeditionDate": "2025-03-
24T11:54:57.027Z",
"email": "string",
"registryCode": "string",
"registryDate": "2025-03-24T11:54:57.027Z",
"seller": {
  "identifier": "string",
  "fullName": "string"
},
"buyer": {
  "identifier": "string",
  "fullName": "string"
},
"signer": {
  "identifier": "string",
  "fullName": "string"
},
"third": {
  "identifier": "string",
  "fullName": "string"
},
"factoringAssignment": {
  "identifier": "string",
  "fullName": "string"
},
"facturae": {
  "uuid": "string",
  "filename": "string",
```

```
    "mime": "string",
    "hash": "string",
    "isDeleted": true
  },
  "europeanInvoice": {
    "uuid": "string",
    "filename": "string",
    "mime": "string",
    "hash": "string",
    "isDeleted": true
  },
  "deliverer": {
    "identifier": "string",
    "fullName": "string"
  },
  "attachments": [
    {
      "uuid": "string",
      "filename": "string",
      "mime": "string",
      "hash": "string",
      "isDeleted": true
    }
  ],
  "factoringAssignmentDocuments": [
    {
      "isDeleted": true,
      "uuid": "string",
      "filename": "string",
      "mime": "string",
      "hash": "string"
    }
  ],
  "type": "string"
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 500 (application/json)

- Body

```
{  
  "code": 500,  
  "message": "Internal Server Error"  
}
```

3.1.1.2 Remisión de facturas a Repositorio [POST]

Realiza el proceso de remisión y registro de una factura, pero a un organismo que no está adherido a FACe, por lo que se requiere firma.



Nota

El XML necesario para poder remitir una factura a Repositorio está definido en el [Anexo I](#).

- API Invoices

```
/v1/invoices/repository
```

- Parámetros de body
 - filename (string, obligatorio): nombre asignado a la factura.
 - content (string, obligatorio): XML de la factura codificada en Base64.
 - email (string, obligatorio): dirección de email asociada a la remisión de factura.
 - attachments
 - filename (string, opcional): nombre del anexo.
 - content (string, opcional): PDF o HTML del anexo codificado en Base64.
 - repositorySignature (string, obligatorio): XML firmado.

- Petición (text/plain)

```
{
  "filename": "factura",
  "content":
  "PD94bWwgdMvYc2lrbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz
  48ZmU6RmFjdHVyYWUgeG1sbnM6ZmU9Imh0dHA6Ly93d3cuZmFjd
  HVyYWUuZ29iLmVzL2ZvcmlhdG8vVmVyc2lrbmVzL0ZhY3R1cmFl
  djNfMl8yLnhtbCIgeG1sbnM6ZHM9Imh0dHATxTY2h1bWFWZXJza
  W9uPjMuMi4yPC9TY2h1bWFWZXJzaW9uFtb3VudD4zNi4zNjwVG
  90YWxBbW91bnQ+",
  "email": "a@a.com",
  --"repositorySignature":
}
```

- Response 201 (application/json)

- Body

```
{
  "oc": {
    "code": "string",
    "name": "string",
    "alias": "string"
  },
  "og": {
    "code": "string",
    "name": "string",
    "alias": "string"
  },
  "ut": {
    "code": "string",
    "name": "string",
    "alias": "string"
  },
  "repository": true,
  "hasRcfFace": true,
  "status": "string",
  "statusComment": "string",
  "statusHistory": [
    {
      "code": "string",
      "name": "string"
    }
  ],
  "cancellationRequestStatus": "string",
  "cancellationRequestStatusCode": "string",
  "cancellationRequestComment": "string",
  "serial": "string",
  "number": "string",
}
```

```
"serialNumber": "string",
"amount": 0,
"currency": "string",
"expeditionDate": "2025-03-
24T12:36:15.258Z",
"email": "string",
"registryCode": "string",
"registryDate": "2025-03-24T12:36:15.258Z",
"seller": {
  "identifier": "string",
  "fullName": "string"
},
"buyer": {
  "identifier": "string",
  "fullName": "string"
},
"signer": {
  "identifier": "string",
  "fullName": "string"
},
"third": {
  "identifier": "string",
  "fullName": "string"
},
"factoringAssignment": {
  "identifier": "string",
  "fullName": "string"
},
"facturae": {
  "uuid": "string",
  "filename": "string",
  "mime": "string",
  "hash": "string",
  "isDeleted": true
},
"europeanInvoice": {
  "uuid": "string",
  "filename": "string",
  "mime": "string",
  "hash": "string",
  "isDeleted": true
},
"deliverer": {
  "identifier": "string",
  "fullName": "string"
},
"attachments": [
  {
```

```
    "uuid": "string",
    "filename": "string",
    "mime": "string",
    "hash": "string",
    "isDeleted": true
  }
],
"factoringAssignmentDocuments": [
  {
    "isDeleted": true,
    "uuid": "string",
    "filename": "string",
    "mime": "string",
    "hash": "string"
  }
],
"type": "string"
}
```

- Response 400 (application/json)

- Body

```
{
  "content": [
    "string"
  ],
  "filename": [
    "string"
  ],
  "attachments": [
    "string"
  ],
  "email": [
    "string"
  ],
  "repositorySignature": [
    "string"
  ],
  "errors": [
    "string"
  ],
  "code": "400",
  "message": "Bad Request"
}
```


- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.1.1.3 Búsqueda de una factura [GET]

La búsqueda de facturas por número de registro devuelve toda la información detallada de dicha factura.

- API Invoices

```
/v1/invoices/{registryCode}
```

- Parámetros de path

- registryCode (string, obligatorio). Número de registro de la factura.

- Petición (text/plain)

```
{
  "registry": "REGAGE24e0000009xxx",
}
```

- Response 200 (application/json)

- Body

```
{
  "oc": {
    "code": "string",
    "name": "string",
    "alias": "string"
  },
  "og": {
    "code": "string",
    "name": "string",
  }
}
```

```
"alias": "string",
},
"ut": {
  "code": "string",
  "name": "string",
  "alias": "string"
},
"repository": true,
"hasRcfFace": true,
"status": "string",
"statusComment": "string",
"statusHistory": [
  {
    "code": "string",
    "name": "string"
  }
],
"cancellationRequestStatus": "string",
"cancellationRequestStatusCode": "string",
"cancellationRequestComment": "string",
"serial": "string",
"number": "string",
"serialNumber": "string",
"amount": 0,
"currency": "string",
"expeditionDate": "2025-03-24T14:19:43.947Z",
"email": "string",
"registryCode": "string",
"registryDate": "2025-03-24T14:19:43.947Z",
"seller": {
  "identifier": "string",
  "fullName": "string"
},
"buyer": {
  "identifier": "string",
  "fullName": "string"
},
"signer": {
  "identifier": "string",
  "fullName": "string"
},
"third": {
  "identifier": "string",
  "fullName": "string"
},
"factoringAssignment": {
  "identifier": "string",
  "fullName": "string"
},
"facturae": {
  "uuid": "string",
  "filename": "string",
```

```
    "mime": "string",
    "hash": "string",
    "isDeleted": true
  },
  "europeanInvoice": {
    "uuid": "string",
    "filename": "string",
    "mime": "string",
    "hash": "string",
    "isDeleted": true
  },
  "deliverer": {
    "identifier": "string",
    "fullName": "string"
  },
  "attachments": [
    {
      "uuid": "string",
      "filename": "string",
      "mime": "string",
      "hash": "string",
      "isDeleted": true
    }
  ],
  "factoringAssignmentDocuments": [
    {
      "isDeleted": true,
      "uuid": "string",
      "filename": "string",
      "mime": "string",
      "hash": "string"
    }
  ],
  "type": "string"
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 403 (application/json)

- Body

```
{
  "code": "403",
  "message": "Forbidden"
}
```

- Response 404 (application/json)

- Body

```
{
  "code": "404",
  "message": "Not Found"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.1.1.4 Información del estado de las facturas [GET]

Se puede obtener el listado de los estados en los que pueda estar una factura, así como a solicitud de anulación y la cesión de crédito.

- API Invoices

```
/v1/invoices-status-list
```

- No tiene parámetros de entrada.

- Response 200 (application/json)

- Body

```
{
  "eInvoiceStatus": [
    {
      "code": "1200",
      "name": "Registrada",
      "description": "La factura ha sido registrada en el registro electrónico"
    }
  ]
}
```

```

    },
    {
      "code": "1300",
      "name": "Registrada en RCF",
      "description": "La factura ha sido registrada
en RCF"
    },
    {
      "code": "2400",
      "name": "En Trámite",
      "description": "Contabilizada la obligación
reconocida"
    },
    {
      "code": "2500",
      "name": "Pagada",
      "description": "Factura pagada"
    },
    {
      "code": "2600",
      "name": "Rechazada",
      "description": "La Unidad rechaza la factura"
    },
    {
      "code": "3100",
      "name": "Anulada",
      "description": "La Unidad aprueba la
propuesta de anulación"
    }
  ],
  "cancellationRequestStatus": [
    {
      "code": "4100",
      "name": "No solicitada",
      "description": "Solicitud de anulación no
solicitada"
    },
    {
      "code": "4200",
      "name": "Solicitada",
      "description": "Solicitud de anulación
solicitada"
    },
    {
      "code": "4300",
      "name": "Aceptada",
      "description": "Solicitud de anulación
aceptada"
    },
    {
      "code": "4400",
      "name": "Rechazada",

```

```
    "description": "Solicitud de anulación rechazada"
  },
  ],
  "factoringAssignmentStatus": [
    {
      "code": "6100",
      "name": "Pendiente",
      "description": "Pendiente"
    },
    {
      "code": "6200",
      "name": "Aceptada",
      "description": "Aceptada"
    },
    {
      "code": "6300",
      "name": "Rechazada",
      "description": "Rechazada"
    },
    {
      "code": "6400",
      "name": "No Procesable",
      "description": "No Procesable"
    }
  ]
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 404 (application/json)

- Body

```
{
  "code": "404",
  "message": "Not Found"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.1.1.5 Solicitud de anulación de una factura [POST]

Los proveedores tienen la capacidad de solicitar la anulación de una factura cuando consideran que hay algún dato erróneo.

- API Invoices

```
v1/invoices/{registryCode}/cancellation-requests
```

- Parámetros de path

- registryCode (string, obligatorio). Número de registro que identifica unívocamente a la factura.

- Parámetros de body

- comment (string, obligatorio). Comentario que sustente la solicitud de anulación.

- Petición (text/plain)

```
{
  "comment": "necesito que se anule esta factura",
}
```

- Response 200 (application/json)

- Body

```
{
  "registryCode": " REGAGE24e0000000XXXXX",
  "registryDate": "2024-01-04T15:14:22.000+01:00",
  "uuid": "6596c172b375f",
  "requestComment": "Necesito que se anule esta factura",
  "processed": false,
  "requestedAt": "2024-01-04T15:32:18.735+01:00"
}
```

- Response 400 (application/json)

- Body

```
{
  "comment": [
    "string"
  ],
  "errors": [
    "string"
  ],
  "code": "400",
  "message": "Bad Request"
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 404 (application/json)

- Body

```
{
  "code": "404",
  "message": "Not Found"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```


3.1.2 Cesión de crédito

Los cesionarios podrán obtener información sobre las facturas en las que intervienen. Los recursos sobre la cesión de crédito estarán ubicados en la URL /factoring-assignment.

3.1.2.1 Obtener facturas con cesión de crédito [GET]

Obtiene un listado de facturas en las que el usuario es cesionario de la cesión de crédito.

- API factoring-assignment

/v1/factoring-assignment/invoices

- Parámetros de query
 - page (string, obligatorio): número de páginas en las que obtener el resultado., por defecto aparece 1.
 - limit (string, obligatorio): número de ítems por página, por defecto aparecen 1000
 - registryDateFrom (string, opcional): fecha de registro desde, con formato dd/mm/yyyy
 - registryDateTo (string, opcional): fecha de registro hasta. , con formato dd/mm/yyyy
 - seller (string, opcional): Identificador del emisor de la factura.
 - orderBy (string, obligatorio): campo de ordenación de los resultados, por defecto es el número de registro
 - order (string, obligatorio): orden ascendente o descendente de los datos, por defecto es descendente.

- Petición (text/plain)

```
{
  "page": "1",
  "limit": "1000",
  "registryDateFrom": "01/01/2019",
  "registryDateTo": "31/12/2019",
  "seller": "12345678Z",
  "orderBy": " REGAGE24e0000000XXXXX",
  "order": "DESC"
}
```

- Response 200 (application/json)

- Body

```
{
  "items": [
    {
      "serial": "string",
      "number": "string",
      "serialNumber": "string",
      "seller": {
        "identifier": "string",
        "fullName": "string"
      },
      "buyer": {
        "identifier": "string",
        "fullName": "string"
      },
      "hasRcfFace": true,
      "oc": {
        "code": "string",
        "name": "string",
        "alias": "string"
      },
      "og": {
        "code": "string",
        "name": "string",
        "alias": "string"
      },
      "ut": {
        "code": "string",
        "name": "string",
        "alias": "string"
      },
      "status": "string",
      "statusHistory": [
        {
          "code": "string",
          "name": "string",
          "comment": "string",
          "createdAt": "2025-03-24T14:28:59.234Z"
        }
      ]
    }
  ]
}
```

```
    }  
  ],  
  "cancellationRequestStatus": "string",  
  "cancellationRequestHistory": [  
    {  
      "requestComment": "string",  
      "requestResponse": "string",  
      "requestedAt": "2025-03-  
24T14:28:59.234Z",  
      "processedAt": "2025-03-  
24T14:28:59.234Z",  
      "processed": true,  
      "accepted": true  
    }  
  ],  
  "registryCode": "string",  
  "registryDate": "2025-03-24T14:28:59.234Z",  
  "amount": 0,  
  "currency": "string",  
  "third": {  
    "identifier": "string",  
    "fullName": "string"  
  },  
  "deliverer": {  
    "identifier": "string",  
    "fullName": "string"  
  },  
  "factoringAssignmentDocuments": [  
    {  
      "isDeleted": true,  
      "uuid": "string",  
      "filename": "string",  
      "mime": "string",  
      "hash": "string"  
    }  
  ],  
  "europeanInvoice": {  
    "uuid": "string",  
    "filename": "string",  
    "mime": "string",  
    "hash": "string",  
    "isDeleted": true  
  },  
  "facturae": {  
    "uuid": "string",  
    "filename": "string",  
    "mime": "string",  
    "hash": "string",  
    "isDeleted": true  
  },  
  "attachments": [  
    {
```

```
        "uuid": "string",
        "filename": "string",
        "mime": "string",
        "hash": "string",
        "isDeleted": true
    }
],
"factoringAssignmentStatus": "string",
"factoringAssignmentComment": "string",
"hasFactoringAssignment": true,
"factoringAssignment": {
    "identifier": "string",
    "fullName": "string"
},
"signer": {
    "identifier": "string",
    "fullName": "string"
}
}
},
"count": 0,
"total": 0,
"limit": 0,
"page": 0
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```


- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 403 (application/json)

- Body

```
{
  "code": "403",
  "message": "Forbidden"
}
```

- Response 409 (application/json)

- Body

```
{
  "errors": [
    "string"
  ],
  "code": "400",
  "message": "Bad Request"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.1.2.3 Obtener anexos de la factura [GET]

Obtiene los anexos de la factura en la que participa como cesionario.

- API factoring-assignment

```
/v1/factoring-  
assignment/invoices/{registryCode}/attachments/{fi  
le}
```

- Parámetros de query
 - registryCode (string, obligatorio): número de registro de la factura.
 - file (string, obligatorio): nombre del anexo

- Petición (text/plain)

```
{  
  "registryCode": " REGAGE24e0000000XXXXX",  
  "file": "6596c2d3a78c3"  
}
```

- Response 200 (application/json)

- Body

```
{  
  "uuid": "6596c2d3a78c3",  
  "filename": "anexo_cesion",  
  "content":  
  "PD94bWwgdMvYc2lvcj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48ZmU  
6RmFjdHVyYWUgeG1sbnM6ZmU9Imh0dHA6Ly93d3cuZmFjdHVyYWUuZ29  
iLmVzL2ZvcmlhdG8vVmVyc2lvcmlhdG8vZmVzL0ZlY3R1cmFldjNfM18yLnhtbCI  
geG1sbnM6ZHM9Imh0dHA6Ly93d3cuZmFjdHVyYWUuZ29iLmVzL2ZvcmlhdG8vVmVyc2lvcmlhdG8vZmVzL0ZlY3R1cmFldjNfM18yLnhtbCI  
pZyMiPgo8RmlsZUhlYWRlcj4KCTxTY2h1bWFWZXJzaW9uPjMuMi4yPC9  
TY2h1bWFWZXJzaW9uPjMuMi4yPC9TY2h1bWFWZXJzaW9uPjMuMi4yPC9  
CQk8QmF0Y2h1bWFWZXJzaW9uPjMuMi4yPC9TY2h1bWFWZXJzaW9uPjMuMi4yPC9  
CQk8QmF0Y2h1bWFWZXJzaW9uPjMuMi4yPC9TY2h1bWFWZXJzaW9uPjMuMi4yPC9  
bnRpdj48RmlsZUhlYWRlcj4KCTxTY2h1bWFWZXJzaW9uPjMuMi4yPC9TY2h1bWFWZXJzaW9uPjMuMi4yPC9  
PgoJCTxUb3RhbEludm9pY2VzQW1vdW50PjxUb3RhbEFtb3VudD4xMzg0  
LjU4PC9Ub3RhbEFtb3VudD48L1RvdGFsSW52b2ljZXNBbW91bnQ+Cgk  
PFRvdGFsT3V0c3RhbMRpbmdBbW91bnQ+PFRvdGFsQW1vdW50PjEzODQu  
NTg8L1RvdGFsQW1vdW50PjwvVG90YWwPdXRzdGFuZGluZ0Ftb3VudD4K  
CQk8VG90YWwFeGVjdXRhYm91bW50PjxUb3RhbEFtb3VudD4xMzg0  
LjU4PC9Ub3RhbEFtb3VudD48L1RvdGFsRXhlY3V0YVJzZUJ0Ftb3VudD4K  
CQk8SW52b2ljZUN1cnJlbnM5Q29kZT5FVVI8L0ludm9pY2V0dD48RmlsZUhlYWRlcj4KCTxTY2h1bWFWZXJzaW9uPjMuMi4yPC9  
eUNvZGU+Cgk8L0ludm9pY2V0dD48RmlsZUhlYWRlcj4KCTxTY2h1bWFWZXJzaW9uPjMuMi4yPC9TY2h1bWFWZXJzaW9uPjMuMi4yPC9  
PgoJCTxB",  
  "mime": " application/pdf",  
  "hash": " 0b61ba393c70eb9473058f1be68925b8f8c3363"  
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 403 (application/json)

- Body

```
{
  "code": "403",
  "message": "Forbidden"
}
```

- Response 409 (application/json)

- Body

```
{
  "errors": [
    "string"
  ],
  "code": "400",
  "message": "Bad Request"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```


- Response 401 (application/json)

- Body

```
{  
  "code": "401",  
  "message": "Unauthorized"  
}
```

- Response 403 (application/json)

- Body

```
{  
  "code": "403",  
  "message": "Forbidden"  
}
```

- Response 409 (application/json)

- Body

```
{  
  "errors": [  
    "string"  
  ],  
  "code": "400",  
  "message": "Bad Request"  
}
```

- Response 500 (application/json)

- Body

```
{  
  "code": 500,  
  "message": "Internal Server Error"  
}
```

3.2 Grupo Información para Proveedores

En este apartado se desglosarán todos los recursos de información de relaciones y estadísticas que tienen los usuarios proveedores a su disposición.

3.2.1 Relaciones

Los recursos ubicados bajo la ruta */relations* contienen la información de las relaciones de organismos activas.

3.2.1.1 Listado de Relaciones [GET]

Devuelve la información de las relaciones entre centros administrativos.

- API relations

```
/v1/relations
```

- Parámetros de query
 - page (string, obligatorio): número de páginas en las que obtener el resultado. Por defecto aparecerá 1
 - limit (string, obligatorio): número de ítems por página, por defecto aparecerán 1000
 - fulltext (string, opcional): filtro por administración o rol de la unidad.
 - adminLevel (string, opcional): nivel de administración.
 - oc (string, opcional): código de unidad DIR con rol Oficina Contable.
 - og (string, opcional): código de unidad DIR con rol Órgano Gestor.
 - ut (string, opcional): código de unidad DIR con rol Unidad Tramitadora.
- Petición (text/plain)

```
{  
  "page": "1",  
  "limit": "1000",  
  "oc": "P00000000"  
}
```

- Response 200 (application/json)

- Body

```
{
  "items": [
    {
      "hash":
"a983c2d1a327a2abfb366e650cca3df75e7cfef0",
      "oc": {
        "code": "P00000015",
        "name": "UNIDAD DIR PRUEBAS 15",
        "alias": "Prof. Reilly Orn"
      },
      "og": {
        "identifier": "1111111H",
        "code": "P00000015",
        "name": "UNIDAD DIR PRUEBAS 15",
        "alias": "Alexandrea Ward Sr."
      },
      "ut": {
        "code": "P00000015",
        "name": "UNIDAD DIR PRUEBAS 15",
        "alias": "Zechariah Johns"
      },
      "active": true,
      "administration": {
        "code": "P00000005",
        "name": "UNIDAD DIR PRUEBAS 5"
      }
    },
    {
      "hash":
"5b72cbb98cbd85caa81297ffdf0fcc6e113f0bc1",
      "oc": {
        "code": "P00000009",
        "name": "UNIDAD DIR PRUEBAS 9",
        "alias": "nuevo alias"
      },
      "og": {
        "identifier": "1111111H",
        "code": "P00000009",
        "name": "UNIDAD DIR PRUEBAS 9",
        "alias": "Miss Arvilla Kilback"
      },
      "ut": {
        "code": "P00000009",
        "name": "UNIDAD DIR PRUEBAS 9",
        "alias": "Dameon Gleichner III"
      },
      "active": true,
      "administration": {
        "code": "P00000009",

```

```
{
  "name": "UNIDAD DIR PRUEBAS 9"
}
],
"count": 1000,
"total": 1004,
"limit": 1000,
"page": 1
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.2.1.2 Listado de Relaciones en formato CSV [GET]

Devuelve la información de las relaciones entre centros administrativos en formato CSV.

- API relations

```
/v1/relations.csv
```

- Parámetros de query

- fulltext (string, opcional): filtro por administración o rol de la unidad.
- adminlevel (string, opcional): nivel de administración.
- oc (string, opcional): unidad que con rol Oficina Contable.
- og (string, opcional): unidad con rol Órgano Gestor.
- ut (string, opcional): unidad con rol Unidad Tramitadora.

- Response 200 (application/json): devuelve un archivo CSV con el listado de relaciones.

Code

Details

200

Response body

[Download file](#)

Response headers

```
access-control-allow-headers: x-requested-with,Content-Type,origin,authorization,accept,client-security-token,X-HTTP-Method-Override,apikey
access-control-allow-methods: GET,POST,PATCH,PUT,DELETE,OPTIONS
cache-control: max-age=0,no-cache,no-store
connection: Keep-Alive
content-disposition: attachment; filename="relaciones.csv"
content-type: text/csv; charset=ISO-8859-1
date: Thu,04 Jan 2024 15:31:24 GMT
expires: 0
keep-alive: timeout=15,max=500
pragma: no-cache
referrer-policy: origin-when-cross-origin,strict-origin-when-cross-origin
server: Apache
strict-transport-security: max-age=63072000; includeSubdomains; preload
transfer-encoding: chunked
vary: User-Agent
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1;mode=block
```

Ilustración 1. Archivo CSV con las relaciones

```
oficinaContableNombre;oficinaContableCodigoDir3;organo
GestorNombre;organoGestorCodigoDir3;unidadTramitadoraN
ombre;unidadTramitadoraCodigoDir3;activo
"UNIDAD DIR PRUEBAS 0";P00000000;"UNIDAD DIR PRUEBAS
10";P00000010;"UNIDAD DIR PRUEBAS 20";P00000020;1
"UNIDAD DIR PRUEBAS 0";P00000000;"UNIDAD DIR PRUEBAS
10";P00000010;"UNIDAD DIR PRUEBAS 21";P00000021;1
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.2.2 Estadísticas [GET]

Devuelve la información estadística de las facturas en las que participa el integrador.

- API statistics

```
/v1/statistics
```

- Response 200 (application/json)

- Body

```
{
  "years": [
    {
      "year": 0,
      "invoices": 0,
      "registered": 0,
      "registeredInRcf": 0,
      "verifiedInRcf": 0,
      "receivedInDestination": 0,
      "conformed": 0,
      "paymentProposed": 0,
      "paid": 0,
      "rejected": 0,
      "cancelled": 0,
      "rejectedFace": 0
    }
  ]
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.2.3 FAQs [GET]

Devuelve la información de las preguntas más frecuentes sobre el portal de proveedores.

- API statistics

/v1/faqs

- No tiene parámetros de entrada.
- Response 200 (application/json)
 - Body

```
{
  {
    "id": 1,
    "block": "FACE",
    "title": "¿Qué es FACE?",
    "description": "FACE es el Punto General de Entrada de Facturas de la Administración General del Estado. Permite la remisión de facturas en formato electrónico a aquellos organismos de las administraciones que acepten la recepción de facturas en formato electrónico y que estén dadas de alta en el sistema. Además, ya puede presentar a través de FACE facturas a las administraciones que carezcan de punto general de entrada de facturas. \n Para ello, simplemente deberá incluir como oficina contable, como órgano gestor y como unidad tramitadora el código DIR3 de esa administración.",
    "createdAt": "09/11/2023",
    "active": true,
    "sites": [
      "public_providers",
      "private_providers"
    ]
  },
  {
    "id": 2,
    "block": "FACE",
    "title": "¿Cuál es la URL de acceso?",
    "description": "La URL del portal de proveedores para la presentación de facturas es: http://face.gob.es.",
    "createdAt": "09/11/2023",
```



```
"active": true,  
"sites": [  
  "public_providers",  
  "private_providers",  
  "public_organisms",  
  "private_organisms",  
  "public_integrators",  
  "private_integrators"  
]  
}  
]
```

- Response 401 (application/json)

- Body

```
{  
  "code": "401",  
  "message": "Unauthorized"  
}
```

- Response 500 (application/json)

- Body

```
{  
  "code": 500,  
  "message": "Internal Server Error"
```

3.3 Grupo Notificaciones

Los recursos ubicados bajo la ruta */notifications* y */news* están relacionados con la gestión de las notificaciones del sistema a sus usuarios y las noticias de interés de estos.

3.3.1 Gestión de notificaciones y noticias

3.3.1.1 Listado de notificaciones [GET]

Permite obtener el listado de todas las notificaciones que el sistema tiene para el usuario logado.

- API notifications

```
/v1/notifications
```

- Parámetros de query string
 - site (string, opcional). Puede ser public o private.
- Petición (text/plain)

```
{  
  "site": "public"  
}
```

- Response 200 (application/json)
 - Body

```
{
  "items": [
    {
      "id": 1,
      "title": "notification sed-et-atque-sed-in-  
dolorum-minus-ipso",
      "description": "description soluta-ducimus-  
temporibus-sint-magni-ut-nesciunt-animi",
      "notificationsButton": {
        "text": "esse-qui-sapiente-quidem-  
architecto-aut-necessitatibus-alias",
        "link": "https://rogahn.com/eum-soluta-  
atque-reiciendis-qui-consectetur-vel-facilis-  
illo.html"
      },
      "dateFrom": "2023-11-09T13:00:00+01:00",
      "dateTo": "2024-01-05T13:00:00+01:00",
      "createdAt": "09/11/2023",
      "active": true,
      "fixed": false,
      "notificationLevel": "danger",
      "sites": [
        "public_integrators",
        "public_organisms",
        "public_providers"
      ],
      "position": 2
    }
  ]
}
```

- Response 401 (application/json)

- Body

```
{
  "code": "401",
  "message": "Unauthorized"
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.3.1.2 Listado de noticias [GET]

Devuelve la información de las noticias publicadas en el portal de proveedores.

- API statistics

```
/v1/news
```

- Parámetros de query

- site (string, obligatorio): puede ser *private* o *public*.

- Petición (text/plain)

```
{  
  "site": "public"  
}
```

- Response 200 (application/json)

- Body

```
{  
  "items": [  
    {  
      "id": 0,  
      "title": "string",  
      "description": "string",  
      "dateFrom": "2025-03-24T15:42:57.238Z",  
      "dateTo": "2025-03-24T15:42:57.238Z",  
      "createdAt": "string",  
      "active": true,  
      "fixed": true,  
      "sites": [  
        "string"  
      ],  
      "position": 0  
    }  
  ]  
}
```

- Response 401 (application/json)

- Body

```
{  
  "code": "401",  
  "message": "Unauthorized"  
}
```

- Response 500 (application/json)

- Body

```
{
  "code": 500,
  "message": "Internal Server Error"
}
```

3.3.1.3 Listado de noticias cortas [GET]

Devuelve la información de las *slides* publicadas en el portal de proveedores.

- API statistics

```
/v1/slides
```

- Parámetros de query

- site (string, obligatorio): puede ser *private* o *public*.

- Petición (text/plain)

```
{
  "site": "public"
}
```

- Response 200 (application/json)

- Body

```
{
  "items": [
    {
      "id": 0,
      "image": "string",
      "title": "string",
      "description": "string",
      "notificationsButton": {
        "text": "string",
        "link": "string"
      },
      "dateFrom": "2025-03-24T15:47:40.020Z",
      "dateTo": "2025-03-24T15:47:40.020Z",
      "active": true,
      "fixed": true,
      "sites": [
        "string"
      ],
      "position": 0
    }
  ]
}
```

- Response 401 (application/json)

- Body

```
{  
  "code": "401",  
  "message": "Unauthorized"  
}
```

- Response 500 (application/json)

- Body

```
{  
  "code": 500,  
  "message": "Internal Server Error"  
}
```

ANEXO I: XML PARA REMITIR FACTURAS A REPOSITORIO

El XML para poder remitir facturas a Repositorio llevará la siguiente estructura:

```
<?xml version='1.0' encoding='utf-8'?>
<repository>
  <responsibility-conditions>
    <conditions>
      <![CDATA[El firmante y emisor de la factura, declara
expresamente bajo su responsabilidad: Que la Administración Pública
destinataria de la factura objeto de esta emisión no dispone de Punto
general de entrada de facturas electrónicas ni se ha adherido al de otra
Administración. Se declara expresamente que se ha buscado expresamente en
el <a href="https://face.gob.es/es/directorio">directorio del Punto
General de Entradas de Facturas electrónicas - FACE</a> así como en el <a
href="https://www.facturae.gob.es/que-desea/Paginas/buscador-
administraciones-publicas.aspx">directorio de puntos generales de entrada
de facturas</a> y que, por tanto, ejerce su derecho a presentar su
factura en el Punto general de entrada de facturas electrónicas de la
Administración General del Estado, quien depositará automáticamente la
factura en un repositorio donde la Administración competente será
responsable de su acceso, y de la gestión y tramitación de la factura de
acuerdo con lo establecido en el artículo 6 apartado 7 de la Ley 25/2013,
de 27 de diciembre, de impulso de la factura electrónica y creación del
registro contable de facturas en el Sector Público.]]></conditions>
      <invoice>9c635de6021a30dc91eba698b527c75e7adeb125</invoice>
      <attachments/>
    </responsibility-conditions>
  </repository>
```

La etiqueta <invoice> es un hash sha-1 del contenido de la factura.

ANEXO II: EJEMPLO DEL CÓDIGO EN JAVA PARA GENERAR EL TOKEN

El siguiente anexo muestra un ejemplo en java de generación del token.

```
import java.io.File;
import java.io.FileInputStream;
import java.security.KeyStore;
import java.security.PrivateKey;
import java.security.Signature;
import java.util.Arrays;
import java.util.Base64;
import java.util.Enumeraion;

def project = context.testCase.testSuite.project

this.getClass().classLoader.addURL(new
File(context.expand(project.resourceRoot) +
"/libs/google-http-client-jackson2-
1.28.0.jar").toURL());
this.getClass().classLoader.addURL(new
File(context.expand(project.resourceRoot) +
"/libs/google-http-client-1.28.0.jar").toURL());

File key_pem = new
File(context.expand(project.resourceRoot) + "Ruta a
la clave publica del certificado");

String pem =
org.apache.commons.io.IOUtils.toString(new
FileInputStream(key_pem));

Signature sig =
Signature.getInstance("SHA256WithRSA");

KeyStore ks =
java.security.KeyStore.getInstance("PKCS12");
File key = new
File(context.expand(project.resourceRoot) + "Ruta
a la clave privada del certificado");
ks.load(new
FileInputStream(key),"alias_certificado".toCharArray());

Enumeration aliases = ks.aliases();
String keyAlias = "";
while (aliases.hasMoreElements()) {
    keyAlias = (String) aliases.nextElement();
    System.out.println(keyAlias);
}
//sig.initSign((PrivateKey)ks.getKey(keyAlias,
"alias_certificado".toCharArray()));
```



```
PrivateKey privateKey ;
com.google.api.client.json.webtoken.JsonWebSignature.Header header = new
com.google.api.client.json.webtoken.JsonWebSignature.Header();
header.setAlgorithm("RS256");
//header.setX509Certificates(Arrays.asList(new
String[]
{Base64.getEncoder().encodeToString((ks.getCertificate(keyAlias).getEncoded()))});
def cert_enc =
Base64.getEncoder().encodeToString(ks.getCertificate(keyAlias).getEncoded());
def res = [];
res.add(cert_enc);
List certs = Arrays.asList(res);
header.setX509Certificates(certs);
com.google.api.client.json.webtoken.JsonWebToken.Payload payload = new
com.google.api.client.json.webtoken.JsonWebToken.Payload();
long now = System.currentTimeMillis() / 1000L;
long exp = now + 7200;
payload.set("exp", exp);
payload.set("iat", now);
com.google.api.client.json.JsonFactory JSON_FACTORY =
com.google.api.client.json.jackson2.JacksonFactory.getDefaultInstance();
def token =
com.google.api.client.json.webtoken.JsonWebSignature.signUsingRsaSha256((PrivateKey)ks.getKey(keyAlias),
"alias_certificado".toCharArray(),
JSON_FACTORY, header, payload);
context.testCase.setPropertyValue('token', new
String(token));
log.info(token)
```

