



API REST

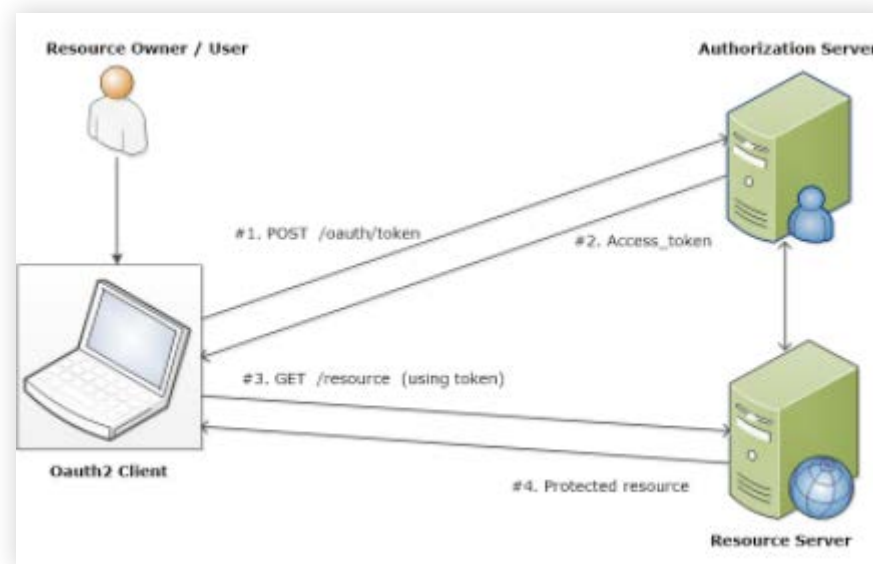
API REST EFACT-OSE
Versión 1.0

1. Introducción

LA API REST permite el envío de documentos a la plataforma OSE de Efact, la cual realizará las validaciones correspondientes de los documentos enviados. Posteriormente, la API podrá devolver el resultado de dicha validación, así como la constancia de recepción (CDR) generada en el Efact OSE

2. Autenticación

El API REST hace uso del protocolo OAuth2.0 para la autenticación de los clientes de los servicios. El objetivo de dicho protocolo es establecer una delegación de la autenticación a un servidor de autorización que entregará tokens que serán utilizados para acceder a los servicios.



Para acceder al servidor de autenticación debe hacer una petición POST a la siguiente dirección:

`https://ose-gw1.efact.pe:443/api-efact-ose/oauth/token`

En la cabecera del mensaje:

`Content-Type: application/x-www-form-urlencoded`
`Authorization: Basic Y2xpZW50OnNlY3JldA==`

Atributo	Descripción
Content-Type	Debe ir "application/x-www-form-urlencoded"
Authorization	Autenticación de la aplicación cliente. "Basic (client:secret en base 64)"

**En desarrollo, el header tendrá una autenticación básica para la aplicación cliente (user: client, password: secret)*

En el cuerpo del mensaje:

`username: 20338570041`
`password: 501ee725667`
`grant_type: password`

Atributo	Descripción
Username	RUC del usuario.
Password	Contraseña REST registrada en Efact.
grant_type	Este parámetro debe contener el valor password, ya que es ese tipo de token el que se está solicitando.

2.1. Respuesta: HTTP STATUS CODE: 200

```
{
  "access_token": "68ecdf44-6aaf-4128-a145-9f5745e02147",
  "token_type": "bearer",
  "refresh_token": "38bd8897-382e-411d-b256-3abaccdcb514",
  "expires_in": 43199,
  "scope": "read"
}
```

Campo	Descripción
access_token	El token de acceso proporcionado por Efact.
expires_in	Tiempo de vida del token de acceso (en segundos).
token_type	Tipo de token retornado. El valor será siempre: Bearer.

**Nota: el token solo tendrá una validez temporal, por lo cual, para cada consumo de los métodos deberá consumirse el servicio de autenticación y obtenerse un nuevo token.*

2.2. Excepciones

En caso de error en el servicio de autorización retornarán status HTTP mencionados a continuación.

HTTP STATUS CODE: 400

El servicio de autorización lo devolverá en caso los datos ingresados sean incorrectos.

```
{  
  "error": "invalid_grant",  
  "error_description": "Bad credentials"  
}
```

2.3. TEST MÉTODO - IMPLEMENTACIÓN EN JAVA

```
// Se establece el cliente POST para el servidor de autenticación
String resource = "https://ose-gwl.efact.pe:443/api-efact-ose/oauth/token";
CloseableHttpClient httpClient = HttpClients.createDefault();

HttpPost httpPost = new HttpPost(resource);

// Se codifica a base 64 las credenciales de la aplicación cliente
// Para las pruebas las credenciales son -> client:secret
String credentials =
Base64.getEncoder().encodeToString("client:secret".getBytes("UTF-8"));
httpPost.setHeader("Authorization", "Basic " + credentials);

//Se agregan los datos de autenticación para la plataforma Efact OSE
List<NameValuePair> params = new ArrayList<>();
params.add(new BasicNameValuePair("grant_type", "password"));
params.add(new BasicNameValuePair("username", "RUC_USER"));
params.add(new BasicNameValuePair("password", "*****"));
httpPost.setEntity(new UrlEncodedFormEntity(params));

//Se envía la petición y se recibe el json con el token
String json= null;
try {
    json = httpClient.execute(httpPost, new StringResponseHandler());

    // En caso de enviar datos correcto se obtiene el token de autenticación
    ObjectMapper mapper = new ObjectMapper();
    JsonNode rootNodeToker = mapper.readTree(json);
    accessToken = rootNodeToker.path("access_token").asText();
    System.out.println("TOKEN: " + accessToken);

} catch (ClientProtocolException e) {
    // En caso de error de autenticación retornará una excepción
    throw new Exception(e);
}
```

.....

Se implementa una clase estática para reconocer los códigos de error

```
private static class StringResponseHandler implements ResponseHandler<String> {  
    @Override  
    public String handleResponse (HttpResponse response) throws IOException {  
        int status = response.getStatusLine ().getStatusCode ();  
        if (status >= 200 && status < 300) {  
            HttpEntity entity = response.getEntity ();  
            return entity != null ? EntityUtils.toString (entity) : null;  
        } else {  
            throw new ClientProtocolException ("Unexpected response status: " + status);  
        }  
    }  
}
```

Nota: Para las implementaciones en JAVA se utilizan las siguientes dependencias:

```
<!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpclient -->  
<dependency>  
    <groupId>org.apache.httpcomponents</groupId>  
    <artifactId>httpclient</artifactId>  
    <version>4.5.3</version>  
</dependency>  
<!-- https://mvnrepository.com/artifact/org.apache.httpcomponents/httpmime -->  
<dependency>  
    <groupId>org.apache.httpcomponents</groupId>  
    <artifactId>httpmime</artifactId>  
    <version>4.3.1</version>  
</dependency>  
<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core -->  
<dependency>  
    <groupId>com.fasterxml.jackson.core</groupId>  
    <artifactId>jackson-core</artifactId>  
    <version>2.8.9</version>  
</dependency>  
<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->  
<dependency>  
    <groupId>com.fasterxml.jackson.core</groupId>  
    <artifactId>jackson-databind</artifactId>  
    <version>2.8.9</version>  
</dependency>
```


3. Métodos

3.1. Autenticación

Todos los métodos de la API REST deben contener, en la cabecera, el parámetro de autorización con el token obtenido para el usuario:

Cabecera:

Authorization: Bearer 9ca7a5ff-fd5e-4e93-b6d7-06e0e255639e

Atributo	Descripción
Authorization	"Bearer + TOKEN" autenticación con el token obtenido en la autenticación Oauth.

3.2. Método: **POST** <https://ose-gw1.efact.pe:443/api-efact-ose/v1/document>

Request:

Cabecera del mensaje:

content-type: multipart/form-data

Atributo	Descripción
content-type	Tipo de contenido de la petición, el valor será multipart/form-data para adjuntar archivos.

Cuerpo del mensaje:

file : @RUC-01-F99-0001.xml o @RUC-01-F99-0001.csv

Atributo	Descripción
File	El documento electrónico en formato XML o CSV.

3.2.1 Respuesta: HTTP STATUS CODE: 200

Cuerpo del mensaje:

```
{  
  "code": "0", ytw  
  "description": "9ca7a5ff-fd5e-4e93-b6d7-0123fd234fg"  
}
```

Atributo	Descripción
code	Código de error de la SUNAT.
description	En caso de error, retornará la descripción del error.

3.2.2 TEST MÉTODO - IMPLEMENTACIÓN EN JAVA

```
String nombreDocumento="20551093035-01-FG99-01304.xml";

// Se establece el cliente POST para el servidor de autenticación
HttpPost post = new HttpPost("https://ose-gwl.efact.pe:443/api-efact-ose/v1/document");

// Se agrega un Header de autorización con el token recibido por el servidor de autenticación
post.setHeader("Authorization", "Bearer " + accessToken);

// Se adjuntarán los documentos electrónicos en el body de la petición
MultipartEntityBuilder builder = MultipartEntityBuilder.create();
builder.setMode(HttpMultipartMode.BROWSER_COMPATIBLE);

byte[] xml = Files.readAllBytes(Paths.get("tmp/"+nombreDocumento));

builder.addBinaryBody
    ("file", xml, ContentType.create("text/xml"), nombreDocumento);
HttpEntity entity = builder.build();
post.setEntity(entity);

// Se envía el documento electrónico a la plataforma Efact OSE
HttpResponse response = httpClient.execute(post);

// Se valida el código de estado de la petición
int status = response.getStatusLine().getStatusCode();
System.out.println("STATUS CODE: " + status );
//Se valida que sea el status correcto
if (status == 200) {
    HttpEntity entity = response.getEntity();

    //Se obtiene el resultado del response
    String jsonResponse = entity != null ? EntityUtils.toString(entity) : null;

    String ticket=null;
    // Se transforma el Json y se obtiene el ticket
    ObjectMapper mapper = new ObjectMapper();
    JsonNode rootNodeResponse = mapper.readTree(jsonResponse);
    ticket = rootNodeResponse.path("description").asText();
    System.out.println("TICKET: " + ticket);

}
}
```

3.3. MÉTODO: GET <https://ose-gw1.efact.pe:443/api-efact-ose/v1/document>

Atributo	Descripción
ticket	Ticket obtenido al enviar el documento por el método POST api/v1/documents/

Response:

HTTP STATUS CODE: 200 OK - Descarga el CDR en formato xml.

3.3.1 EXCEPCIONES

HTTP STATUS CODE: 202

Se obtendrá en caso que el documento aún se encuentre en proceso de validacion.

HTTP STATUS CODE: 404

Se obtendrá en caso que el ticket buscado sea incorrecto.

```
{  
  "code": "0127",  
  "description": "El ticket no existe."  
}
```

HTTP STATUS CODE: 412

Se obtendrá en caso de que exista una validación incorrecta en el XML.
Cuerpo del mensaje:

```
{
  "code": "2334",
  "description": "El documento electrónico ingresado ha sido alterado."
}
```

Atributo	Descripción
code	Ticket obtenido al enviar el documento por el método POST api/v1/documents/
description	Descripcion del error en la validacion.

3.3.2 TEST MÉTODO - IMPLEMENTACIÓN EN JAVA

```
// Se establece el cliente GET para el servidor de autenticación
// El ticket se obtuvo en el método de envío de documentos
HttpGet httpget = new HttpGet("https://ose-gw1.efact.pe:443/api-efact-ose/v1/cdr/" +
ticket);

// Se agrega un Header de autorización con el token recibido por el servidor de autenticación
httpget.setHeader("Authorization", "Bearer " + accessToken);

// Se envía la petición a la plataforma Efact OSE
HttpResponse response = httpClient.execute(httpget);

// Se valida el código de estado de la petición
int status = response.getStatusLine().getStatusCode();
System.out.println("STATUS CODE: " + status );

// Si el código de estado es 200 (OK) entonces la respuesta contiene el CDR en formato XML
if (status == 200) {
  File myFile = new File("CDR-document.xml");
  HttpEntity entity = response.getEntity();
  if (entity != null) {
    try (FileOutputStream outstream = new FileOutputStream(myFile)) {
      entity.writeTo(outstream);
    }
  }
}
```

3.4. MÉTODO: GET <https://ose-gw1.efact.pe:443/api-efact-ose/v1/xml/{ticket}>

Atributo	Descripción
ticket	Ticket obtenido al enviar el documento por el método POST api/v1/documents/

Response:

HTTP STATUS CODE: 200 OK - Descarga el XML en formato xml.

3.3.1 EXCEPCIONES

HTTP STATUS CODE: 202

Se obtendrá en caso que el documento aún se encuentre en proceso de validación.

HTTP STATUS CODE: 404

Se obtendrá en caso que el ticket buscado sea incorrecto.

```
{  
  "code": "0127",  
  "description": "El ticket no existe."  
}
```

HTTP STATUS CODE: 412

Se obtendrá en caso de que exista una validación incorrecta en el XML.
Cuerpo del mensaje:

```
{
  "code": "2334",
  "description": "El documento electrónico ingresado ha sido alterado."
}
```

Atributo	Descripción
code	Ticket obtenido al enviar el documento por el método POST api/v1/documents/
description	Descripcion del error en la validacion.

3.4.2 TEST MÉTODO - IMPLEMENTACIÓN EN JAVA

```
// Se establece el cliente GET para el servidor de autenticación
// El ticket se obtuvo en el método de envío de documentos
HttpGet httpget = new HttpGet("https://ose-gw1.efact.pe:443/api-efact-ose/v1/xml/" +
ticket);

// Se agrega un Header de autorización con el token recibido por el servidor de autenticación
httpget.setHeader("Authorization", "Bearer " + accessToken);

// Se envía la petición a la plataforma Efact OSE
HttpResponse response = httpClient.execute(httpget);

// Se valida el código de estado de la petición
int status = response.getStatusLine().getStatusCode();
System.out.println("STATUS CODE: " + status );

// Si el código de estado es 200 (OK) entonces la respuesta contiene el CDR en formato XML
if (status == 200) {
  File myFile = new File("XML-document.xml");
  HttpEntity entity = response.getEntity();
  if (entity != null) {
    try (FileOutputStream outstream = new FileOutputStream(myFile)) {
      entity.writeTo(outstream);
    }
  }
}
```

3.5. MÉTODO: GET <https://ose-gw1.efact.pe:443/api-efact-ose/v1/pdf/{ticket}>

Atributo	Descripción
ticket	Ticket obtenido al enviar el documento por el método POST api/v1/documents/

Response:

HTTP STATUS CODE: 200 OK - Descarga el PDF en formato pdf.

3.5.1 EXCEPCIONES

HTTP STATUS CODE: 202

Se obtendrá en caso que el documento aún se encuentre en proceso de validación.

HTTP STATUS CODE: 404

Se obtendrá en caso que el ticket buscado sea incorrecto.

```
{  
  "code": "0127",  
  "description": "El ticket no existe."  
}
```


HTTP STATUS CODE: 412

Se obtendrá en caso de que exista una validación incorrecta en el XML.
Cuerpo del mensaje:

```
{
  "code": "2334",
  "description": "El documento electrónico ingresado ha sido alterado."
}
```

Atributo	Descripción
code	Ticket obtenido al enviar el documento por el método POST api/v1/documents/
description	Descripcion del error en la validacion.

3.5.2 TEST MÉTODO - IMPLEMENTACIÓN EN JAVA

```
// Se establece el cliente GET para el servidor de autenticación
// El ticket se obtuvo en el método de envío de documentos
HttpGet httpget = new HttpGet("https://ose-gw1.efact.pe:443/api-efact-ose/v1/pdf/" +
ticket);

// Se agrega un Header de autorización con el token recibido por el servidor de autenticación
httpget.setHeader("Authorization", "Bearer " + accessToken);

// Se envía la petición a la plataforma Efact OSE
HttpResponse response = httpClient.execute(httpget);

// Se valida el código de estado de la petición
int status = response.getStatusLine().getStatusCode();
System.out.println("STATUS CODE: " + status );

// Si el código de estado es 200 (OK) entonces la respuesta contiene el CDR en formato XML
if (status == 200) {
  File myFile = new File("pdf-document.pdf");
  HttpEntity entity = response.getEntity();
  if (entity != null) {
    try (FileOutputStream outstream = new FileOutputStream(myFile)) {
      entity.writeTo(outstream);
    }
  }
}
```

Acuerdo de Confidencialidad

El material adjunto en este documento es propiedad de Efact S.A.C, el mismo que contiene información confidencial, privilegiada y es para uso exclusivo de la persona o entidad a la que expresamente ha sido enviado.

Si no es usted el destinatario indicado, queda notificado de que cualquier revisión, difusión, copia o uso sin autorización de dicha información está prohibido.

efact

www.efact.pe