ESQUEMA DE AUTENTICACIÓN DE SISTEMAS EXTERNOS PARA INTEROPERABILIDAD

IDENTIDAD
v.1.4.1

Identificador del Documento:	IDENTIDAD
Nombre del documento:	ESQUEMA DE AUTENTICACIÓN DE SISTEMAS EXTERNOS PARA INTEROPERABILIDAD
Estado del documento:	

Versión	Creación	Descripción Cambio	Autor/es
1.0	19/01/2018	Especificación interoperabilidad.	ERNESTO MEDINA, ANGELA SUAREZ, ARMANDO FRADE BELLO
1.1	05/03/2018	Se agregan ejemplos de las tramas de identidad.	ERNESTO MEDINA, ANGELA SUAREZ
1.2	04/04/2018	Soporte encriptación ClientSecret y Password en método login	ERNESTO MEDINA, ANGELA SUAREZ CARLOS DIAZ
1.3	15/04/2018	Alcance de seguridad detallado.	ERNESTO MEDINA
1.4	15/20/2020	Alineación con estándar de especificaciones y acuerdos	ERNESTO MEDINA, CLIMACO ALBERTO LLAMAS
1.4.1	04/03/2021	Actualización de página sugerida para cifrado AES-128	CLIMACO LLAMAS

Contenido

1. Introducción

1.1. Objetivo

Esta especificación de autenticación está orientada a la habilitación de la interoperabilidad con los sistemas de la DIAN. Describe los mecanismos y requerimientos para que un sistema externo (aplicación) pueda obtener un token de acceso para el consumo de servicios.

Los aspectos generales de seguridad tecnológica, estándares y políticas se encuentran en el anexo técnico "ESPECIFICACIONES TÉCNICAS DE INTEROPERABILIDAD" que se entrega con este documento.

1.2. Terminología

Para facilidad del entendimiento de este documento, se establece la siguiente terminología de uso común.

Término / Abreviatura	**Descripción**
TOKEN	También conocido como token de autenticación o token criptográfico es un elemento electrónico que se le da a un usuario autorizado de un servicio computarizado para facilitar el proceso de autenticación.
JWT	JSON Web Token es un estándar abierto (RFC-7519) basado en JSON para crear un token que sirva para enviar datos entre aplicaciones o servicios y garantizar que sean válidos y seguros. El caso más común de uso de los JWT es para manejar la autenticación en aplicaciones móviles o Web.
REST-Based Web Service	Representational State Transfer (REST por sus siglas en ingles), son una forma de proveer interoperabilidad entre sistemas. Los servicios que cumplen con los requerimientos REST permitir a sistemas acceder y manipular representaciones de Recursos Web usando una forma única y predefinada de operaciones sin estado.
JSON	JavaScript Object Notation, es un formato mínimo y legible para estructurar datos. Es utilizado para la transmisión de datos entre aplicaciones web como una alternativa al XML.
CLIENTE	Aplicación externa que quiere hacer interoperabilidad con los servicios de la entidad.
TOKEN ENDPOINT	Para el caso de la DIAN, el token endpoint es sinónimo del servicio de identidad de la organización, quien es el encargado de administrar el ciclo de vida del token.

2. Planteamiento general y consideraciones

El esquema de autenticación presenta tres escenarios/operaciones:

OBSERVACIONES

- La obtención de cualquier código de respuesta HTTP diferente a 200 OK en el consumo de servicios de autenticar, renovar o revocar se considera como error.

2.1. Autenticar.

Método HTTP: POST
URL Producción: https://api.dian.gov.co/identidad/sts/v1/tokens/login
URL Pruebas: https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/login

Dentro de este contexto, el sistema externo obtendrá un token del sistema de identidad de la DIAN. Aquí se solicita la siguiente información para poder acceder a un token:

Entradas:

Parámetro/Objeto	Tipo Dato	Longitud	Tipo Parámetro	Descripción
grant-type	String	15	Query	Atributo obligatorio que representa el tipo de operación, siempre será la palabra “password” literal. Ver Ejemplo.
client_id	String	50	Query	Representa la identificación de la aplicación que se conecta.
client_secret	String	50	Query	Representa la clave de aplicación que se dio en el registro de la aplicación. Este atributo debe ir encriptado usando AES-128 y aplicación de encoding al enviar la petición

tipoDocumento	String	3	Query	Identifica el tipo de documento conforme la especificación Swagger. Debe ser “US” para la mayoría de escenarios externos.
nroDocumento	String	15	Query	Identificación del usuario
nit	String	15	Query	Identificación de la empresa.
password	String	50	Query	Representa la clave del usuario de sistema habilitado. Este atributo debe ir encriptado usando AES-128 y aplicación de encoding al enviar la petición

Salidas:

Un objeto **DToken** en el body del response con los siguientes atributos:

Atributo	Tipo Dato	Longitud	Descripción
clientId	String	50	Identifica el ClientId al que se le genera el token.
accessToken	String	50	JSON Web Token entregado.
idToken	String	500	Identificador del token.
refreshToken	String	50	Token de renovación automática
tokenType	String	15	Siempre será “Bearer”
expireIn	Number	10	Identifica el tiempo de expiración del token.

ACLARACIONES

- La duración del token puede variar en el tiempo, producto de cambios en las políticas internas de la entidad. Por lo tanto, siempre se deberá verificar la vigencia de este al momento de la obtención para confirmar la fecha de expiración.
- En un ambiente de producción, donde existirán varios “servidores” clientes de Identidad, cada uno deberá autenticarse de forma independiente. Esto significa que cada “servidor” cliente deberá gestionar su token, así como renovarlo o revocarlo de forma autónoma.
- Solo deberían generarse token que van a ser utilizados. Cuando el token no se requiera más para su uso, deberá revocarse.
- No existen limitaciones para la cantidad de tokens generados por un cliente durante un tiempo particular. Sin embargo, esto no evita que las políticas y reglas en el uso de las APIs detecten los malos usos y generen restricciones al cliente.

2.1.1. Cifrado AES-128

A pesar de tener la opción de cifrado como obligatoria en ambiente producción, será posible en pruebas ejecutar la autenticación sin cifrado. Para tal efecto, solo se requiere informar a la DIAN para habilitar/deshabilitar el cifrado en la prueba. En producción, este proceso se podrá realizar por auto-gestión.

Para el cifrado de los atributos se debe realizar el siguiente procedimiento:

1. Se debe concatenar el contenido a encriptar con la fecha del sistema. (Debe estar sincronizado con la hora colombiana al minuto usando estándar ISO 8601).
   Ejemplo: [ClientSecret]-[2018-08-13T09:30:47] y [Password]-[2018-08-13T09:30:47]
2. Se aplica el algoritmo de encriptación AES-128 CBC al contenido utilizando el EncryptionKey (alfanumérico de 16 caracteres) dado por la DIAN al registrar la aplicación del Banco.
   Ejemplo del EncryptionKey: “68BD795F133F0132”
3. La DIAN descifrará el contenido, comparando que la fecha y hora descifrada no difiera en más o menos 3 minutos de la hora actual. Una vez verificado la hora exitosamente, realizará el proceso de autenticación normal usando el “clientId” y “password” obtenidas del descifrado para retornar un token válido.

El proceso de encriptación con los atributos involucrados puede ser generado en la siguiente página (sugerida) para obtener datos cifrados y realizar pruebas en un cliente rest:

https://paiza.io/projects/3HVUVns3UNsVOpTfHxE7Cg

Código implementado en el cual se registra el client_secret, password y encription key, el cual al ejecutarse entrega esos datos cifrados aplicando el algoritmo AES-128

2.1.2. Ejemplo detallado del procedimiento para el Client Secret

Se tiene un client secret inicial así:

CLIENT SECRET Cadena Inicial:

trttwNZ0OMkHS7CC1_nFx6wlKnca

Definición Fecha:

2018-03-18T19:09:10

Formación cadena:

[trttwNZ0OMkHS7CC1_nFx6wlKnca]-[2018-03-18T19:09:10]

Características para el Cifrado de la cadena:
AES
CBC
128 Key Size

Ejemplo Encryption Key:
6A28CE819A9E001A

Definición Vector Ej:

```
byte[] iv = new byte[256 / 16];
IvParameterSpec ivspec = new IvParameterSpec(iv);
ivspec = new IvParameterSpec(new byte[16]);
```

Cifrado Aplicado

oTr1sm/Yk5lkJVkkRIj6QASsKukRahpvltJoXLes0q2XE9TqmgzZaq8jOF493SP7ZzOCLabdRtIAtW9ZpYf+AQ==

Cifrado Aplicado en formato codificado para URL

oTr1sm%2FYk5lkJVkkRIj6QASsKukRahpvltJoXLes0q2XE9TqmgzZaq8jOF493SP7ZzOCLabdRtIAtW9ZpYf%2BAQ%3D%3D

2.1.3. Ejemplo detallado del procedimiento para el Password

Se tiene un password inicial así:

PASSWORD Cadena Inicial:

Prueba2006

Definición Fecha:

2018-03-18T19:09:10

Formación cadena:

[Prueba2006]-[2018-03-18T19:09:10]

Características para el Cifrado de la cadena:
AES
CBC
128 Key Size

Ejemplo Encryption Key:
6A28CE819A9E001A

Definición Vector Ej:
```
byte[] iv = new byte[256 / 16];
IvParameterSpec ivspec = new IvParameterSpec(iv);
ivspec = new IvParameterSpec(new byte[16]);
```

Cifrado Aplicado

x0H+zSi8QK6Hr3SvlEdPr121Ck/XCL0YYzzxvVxl5ln2ez9v+zKwfBDD7fClauRG

Cifrado Aplicado en formato codificado para URL

x0H%2BzSi8QK6Hr3SvlEdPr121Ck%2FXCL0YYzzxvVxl5ln2ez9v%2BzKwfBDD7fClauRG

2.1.4. Ejemplo de tramas

REQUEST

POST

https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/login?grant_type=password&client_id=wAyXOuElL_w01O8MyUhDLK_Z_Xsa&client_secret=oTr1sm/Yk5lkJVkkRIj6QASsKukRahpvltJoXLes0q2XE9TqmgzZaq8jOF493SP7ZzOCLabdRtIAtW9ZpYf+AQ==&tipoDocumento=US&nroDocumento=800130643&nit=800130643&password=oTr1sm/Yk5lkJVkkRIj6QASsKukRahpvltJoXLes0q2XE9TqmgzZaq8jOF493SP7ZzOCLabdRtIAtW9ZpYf+AQ==

Host: 10.255.5.103:9091

Connection: keep-alive

Content-Length: 0

Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36

Content-Type: application/json

Accept: */*

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9

RESPONSE

HTTP/1.1 200 OK

X-Powered-By: Undertow/1

Server: WildFly/10

Server: Restlet-Framework/2.3.7

Accept-Ranges: bytes

Date: Thu, 08 Mar 2018 15:53:53 GMT

Connection: keep-alive

Access-Control-Allow-Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept

Access-Control-Allow-Credentials: true

Content-Type: application/json;charset=UTF-8

Content-Length: 579

{
"accessToken": "18800b30-fa7d-3df4-a4a4-2c2736c42a7c",
"idToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE1MjAwMjk0NjA
sInN1YiI6IjkwMDAwMDAwMDYwMDA0LTAtODkwOTAzOTM4IiwiYXVkIjpbIlBfeDhKakFZV

```
npFTGdyb0lxdE02N0QydjFRUWEiXSwiaXNzIjoiaHR0cHM6XC9cL211aXNjYS5kaWFuLmd
vdi5jbyIsImp0aSI6IjUxZWZiMDJiLWNjZDUtNDZlMi1iNjlmLWM5ZDljZTU1YjNmZSIsI
mlhdCI6MTUyMDAyNTg2MH0.6kRI310oQMSBBb9ZSMaqV-GcdNzKwqz5QJnu9Jmbldg",
tokenType: "Bearer",
expireIn: 1101,
refreshToken: "e161c94b-789e-3c4f-8903-310867440658",
clientId: "P_x8JjAYVzELgroIqtM67D2v1QQa"
}
```


2.2. Refrescar Token.

Método HTTP: POST
URL Producción: https://api.dian.gov.co/identidad/sts/v1/tokens/refresh
URL Pruebas: https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/refresh

Dentro de este contexto, se busca obtener un nuevo token cuando el token entregado se encuentra próximo a vencer o vencido. Sólo se requiere el envío del refresh token en el Authorization

Entradas:

En el Header de la solicitud deben ir:

Parámetro	Tipo Parámetro	Longitud	Descripción
Authorization	Header	500	Debe ser el refreshToken recibido al momento de la autenticación.
ClientId	Header	50	Se refiere al ClientId para el cual se generó el refreshToken.

Salidas:

Un objeto **DToken** en el body del response con los siguientes atributos:

Atributo	Tipo Dato	Longitud	Descripción
clientId	String	50	Identifica el ClientId al que se le generá el token.
accessToken	String	50	JSON Web Token entregado.
idToken	String	500	Identificador del token.

refreshToken	String	50	Token de renovación automática
tokenType	String	15	Siempre será “Bearer”
expireIn	Number	10	Identifica el tiempo de expiración del token.

Ejemplo de uso:

REQUEST

POST https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/refresh HTTP/1.1

Host: 10.255.5.103:9091

Connection: keep-alive

Content-Length: 0

Authorization: Bearer 64261f2a-f61a-343f-b730-581e11b93b38

ClientId: wAyXOuElL_w01O8MyUhDLK_Z_Xsa

Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36

Content-Type: application/json

Accept: */*

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9

RESPONSE

HTTP/1.1 200 OK

X-Powered-By: Undertow/1

Server: WildFly/10

Server: Restlet-Framework/2.3.7

Accept-Ranges: bytes

Date: Thu, 08 Mar 2018 16:07:06 GMT

Connection: keep-alive

Access-Control-Allow-Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept

Access-Control-Allow-Credentials: true

Content-Type: application/json;charset=UTF-8

Content-Length: 579


```
{
      accessToken: "0e9a0104-b26c-345b-8a04-96348671e1f5",
      idToken: "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI5MDAwMDAwMDA
      2MDAwMy0wLTgwMDEzMDY0MyIsImF1ZCI6WyJ3QXlYT3VFbExfdzAxTzhNeVVoRExLX1pfW
      HNhIl0sImlzcyI6Imh0dHBzOlwvXC9tdWlzY2EuZGlhbi5nb3YuY28iLCJleHAiOjE1MjA
      1Mjg4MjYsImlhdCI6MTUyMDUyNTIyNiwicm9sIjoiaHR0cHM6XC9cL2dvb2dsZS5jb20uY
      28iLCJqdGkiOiJmMTE3NGNlYy1lNjA0LTRkOTMtOTg4Mi00MTIxZDgxMTk0ZTkifQ._FyX
      kGUOYrr4LtCwmWNVjtW9mT0TJezubkykx3PgWos",
      tokenType: "Bearer",
      expireIn: 3600,
      refreshToken: "50be2bea-032d-31ac-963b-d46893d26a3b",
      clientId: " wAyXOuElL_w01O8MyUhDLK_Z_Xsa "
}
```


2.3. Revocar Token.

Método HTTP: POST
URL Producción : https://api.dian.gov.co/identidad/sts/v1/tokens/revoke
URL Pruebas: https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/revoke

Entradas:

En el Header de la solicitud deben ir:

Parámetro	Tipo Parámetro	Longitud	Descripción
Authorization	Header	500	Debe ser el accessToken recibido al momento de la autenticación.
ClientId	Header	50	Se refiere al ClientId para el cual se generó el refreshToken.

Salidas:

Un objeto **DToken** en el body del response con los siguientes atributos:

Atributo	Tipo Dato	Longitud	Descripción
clientId	String	50	Identifica el ClientId al que se le generá el token.
accessToken	String	50	JSON Web Token entregado.
idToken	String	500	Identificador del token.
refreshToken	String	50	Token de renovación automática
tokenType	String	15	Siempre será “Bearer”
expireIn	Number	10	Identifica el tiempo de expiración del token.

Ejemplo de uso:

REQUEST

POST https://apipruebasexternas.dian.gov.co/identidad/sts/v1/tokens/revoke HTTP/1.1

Host: 10.255.5.103:9091

Connection: keep-alive

Content-Length: 0

Authorization: Bearer

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiI5MDAwMDAwMDA2MDAw
My0wLTgwMDEzMDY0MyIsImF1ZCI6WyJ3QXlYT3VFbExfdzAxTzhNeVVoRExLX1
pfWHNhIl0sImlzcyI6Imh0dHBzOlwvXC9tdWlzY2EuZGlhbi5nb3YuY28iLCJleHAiOjE1

MjA1MjkxNjYsImlhdCI6MTUyMDUyNTU2Niwicm9sIjoiaHR0cHM6XC9cL2dvb2dsZS5jb20uY28iLCJqdGkiOiI5ODRjZjg0Zi02ZGIzLTRmMmYtYjQ2Mi01YTc3OTE0MGI2ZTcifQ.UFjN9tJ7JmTFVF6w3l_jvHQ9Q-YbScC1qh_Z-qU_wPU

ClientId: wAyXOuElL_w01O8MyUhDLK_Z_Xsa

Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.84 Safari/537.36

Accept: */*

Accept-Encoding: gzip, deflate

Accept-Language: en-US,en;q=0.9

RESPONSE

HTTP/1.1 200 OK

X-Powered-By: Undertow/1

Server: WildFly/10

Server: Restlet-Framework/2.3.7

Accept-Ranges: bytes

Date: Thu, 08 Mar 2018 16:13:05 GMT

Connection: keep-alive

Access-Control-Allow-Origin: chrome-extension://aejoelaoggembcahagimdiliamlcdmfm

Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept

Access-Control-Allow-Credentials: true

Content-Type: application/json;charset=UTF-8

Content-Length: 21

{"message":"success"}

2.4. Confirmación de la recepción

El servicio que recibe la petición envía una respuesta HTTP al cliente. Los posibles códigos de respuesta HTTP generalmente usados en los servicios de la DIAN son:

- 200 – Petición recibida exitosamente. Pendiente de procesamiento.
- 204 – Para cuando no hay contenido (específicamente para peticiones tipo GET)
- 400 – Petición invalida (ErrorResponse - especificación técnica Swagger - es el objeto estándar para especificar la causa del rechazo y deberá ser entregada en el cuerpo de la respuesta).
- 401 – Credencial de autenticación inválida.
- 429 – Demasiadas peticiones. Por favor reintente más tarde.
- 500 – Error interno de servidor.
- 503 – Servicio no disponible. Reintente tarde.

3. Referencias

Los siguientes documentos están referenciados en este documento:

- Este documento referencia la especificación Swagger de autenticación de identidad: API Identidad-EXTERNOS-Swagger20.json