Artículo
Ricardo Paiva · 29 abr, 2021 Lectura de 16 min

Cómo crear un repositorio FHIR + Configuración del servidor de autorización/de recursos OAuth2 en IRIS for Health - Parte 1

¡Hola desarrolladores!

En este artículo, nos centraremos en OAuth2, un protocolo que se utiliza cada vez con más frecuencia en combinación con FHIR para realizar la autorización.

En esta primera parte, comenzaremos con el contenedor Docker para IRIS for Health y Apache, configuraremos la función del servidor de autorización OAuth2 en IRIS for Health, accederemos a él desde la herramienta de desarrollo Postman para REST y obtendremos un token de acceso. Además, en la Parte 2 y en las siguientes, añadiremos las funciones del repositorio FHIR a IRIS for Health, también agregaremos la configuración del servidor de recursos OAuth2, y explicaremos cómo ejecutar las solicitudes de FHIR utilizando los tokens de acceso desde Postman.

En la Comunidad de Desarrolladores ya se han publicado varios artículos excelentes donde se explican las funciones de OAuth2 en los productos de InterSystems; sin embargo, me gustaría explicar nuevamente cómo configurar la última versión. Cómo implementar la estructura de InterSystems IRIS Open Authorization (OAuth 2.0) - Parte 1

En este artículo, utilizaremos la última versión de InterSystems IRIS for Health 2020.3 (es una versión de prueba). Si quieres desarrollar un entorno basado en este artículo, asegúrate de utilizar esta versión o una posterior del kit. Algunas características no están incluidas en los productos anteriores a esta versión.

Preparativos preliminares

El primer paso es hacer unos preparativos preliminares. Son muchas las cosas que deben prepararse para crear un entorno seguro.

La versión de prueba de IRIS for Health 2020.3 solo está disponible en una versión para el contenedor Docker (InterSystems Docker Hub/IRIS for Health). Para efectuar la configuración de OAuth2, también deberás realizar la configuración del servidor web y de SSL. En este artículo, utilizaremos Apache. Al realizar la configuración de SSL en Apache, el certificado de configuración de SSL debe coincidir con el nombre del servidor. Ten en cuenta este punto.

Cómo obtener los archivos de muestra del repositorio intersystems-jp de GitHub

El archivo docker-compose.yml/Dockerfile y otros archivos de muestra utilizados en esta configuración están disponibles en el repositorio de GitHub reservado para la Comunidad de Desarrolladores de InterSystems. En primer lugar, descomprime este archivo en tu entorno utilizando el siguiente comando. (También puedes hacerlo desde el archivo adjunto que se encuentra en este artículo). Este archivo docker-compose.yml/Dockerfile y otros archivos se crean tomando como referencia la aplicación iris-webgateway-example publicada en OpenExchange.

```
git clone https://github.com/Intersystems-jp/IRIS4H-OAuth2-handson.git
```

Cómo cambiar la configuración para que coincida con el kit que se está utilizando

En este archivo docker-compose.yml, se configuran dos contenedores para iniciarse: el contenedor de IRIS for Health y el contenedor de Apache (httpd) se crearán por el comando docker build. El archivo docker-compose.yml, disponible en GitHub, hace referencia a la versión de prueba 2020.3.200.0 de la Community Edition de IRIS for Health. La Community Edition puede utilizarse para evaluar los productos de InterSystems.

```
iris:
    image: store/intersystems/irishealth-community:2020.3.0.200.0
```

Si utilizas una versión diferente (la versión oficial o una más reciente), cambia esta parte de la especificación.

El contenedor Apache se creará con el contenido del Dockerfile, que requiere un kit WebGateway para conectarse a IRIS desde Apache. Para saber cómo conseguir el kit, contacta con el Centro de Soporte Internacional (WRC) de Intersystems. Para consultas sobre el SO del servidor, puedes ponerte en contacto con nosotros en esta dirección.

Modifica las siguientes partes del Dockerfile según el producto que hayas obtenido. Independientemente del SO del servidor (Windows/Ubuntu/CentOS), la plataforma será lnxubuntux64 porque el SO base del contenedor httpd es Debian.

```
ARG version=2020.3.0.200.0
ARG platform=lnxubuntux64
ADD WebGateway-${version}-${platform}.tar.gz /tmp/
```

Cómo preparar un certificado SSL

En el siguiente paso, se prepara un certificado SSL. Cuando se accede a la autorización OAuth2, debe comprobarse si el certificado SSL establecido en el servidor web coincide con la URL a la que se tiene acceso. No es necesario utilizar un certificado oficial, es posible utilizar OpenSSL, etc. Introduce el nombre del servidor en el campo "Common Name" al crear el certificado.

Además, como el certificado que acabas de crear se cargará automáticamente durante el arranque, debes cambiar el archivo por uno que no requiera una contraseña. Consulta el siguiente comando.

```
$ openssl rsa -in cert.key.org -out cert.key
```

Coloca los archivos CRT y KEY que se crearon, en el mismo directorio con el Dockerfile, con los nombres de archivo **server.crt** / **server.key** respectivamente.

Además de utilizarlo con el servidor web Apache, necesitarás un certificado SSL para la configuración de OAuth2. Estos no requieren que introduzcas un nombre de servidor, etc., pero debes crear tres conjuntos (En las configuraciones posteriores, aparecen como auth.cer/auth.key, client.cer/client.key, resserver.cer/resserver.key).

Cómo crear e iniciar un contenedor docker

¡Ya estás listo! Además de los cuatro archivos que descargaste, ahora tienes un kit de instalación Web Gateway y dos certificados SSL en tu directorio. Ten cuidado con los permisos de acceso y ejecución de cada archivo (por ejemplo, yo añadí el permiso de ejecución para webgateway-entrypoint.sh).

```
docker-compose build
docker-compose up -d
```

Una vez iniciado, utiliza el comando docker ps para verificar que los dos contenedores están en ejecución.

```
Apache Container name?<directoryname>_web
IRIS for Health container name?store/intersystems/irishealth-
community:2020.3.0.200.0?or other name depend on kit)
```

Ahora, intenta acceder al portal de administración mediante alguna de estas tres formas. Si el tercer método funciona, ¡tu configuración SSL a través del servidor web Apache es un éxito!

http://[hostname]**:52773**/csp/sys/UtilHome.csp ：A esta URL se accede a través del servidor Apache privado en el contenedor IRIS. Este no pasa por el Apache que se configuró.

http://[hostname]/csp/sys/UtilHome.csp ：Esta URL accede al portal de administración a través del Apache que se configuró.

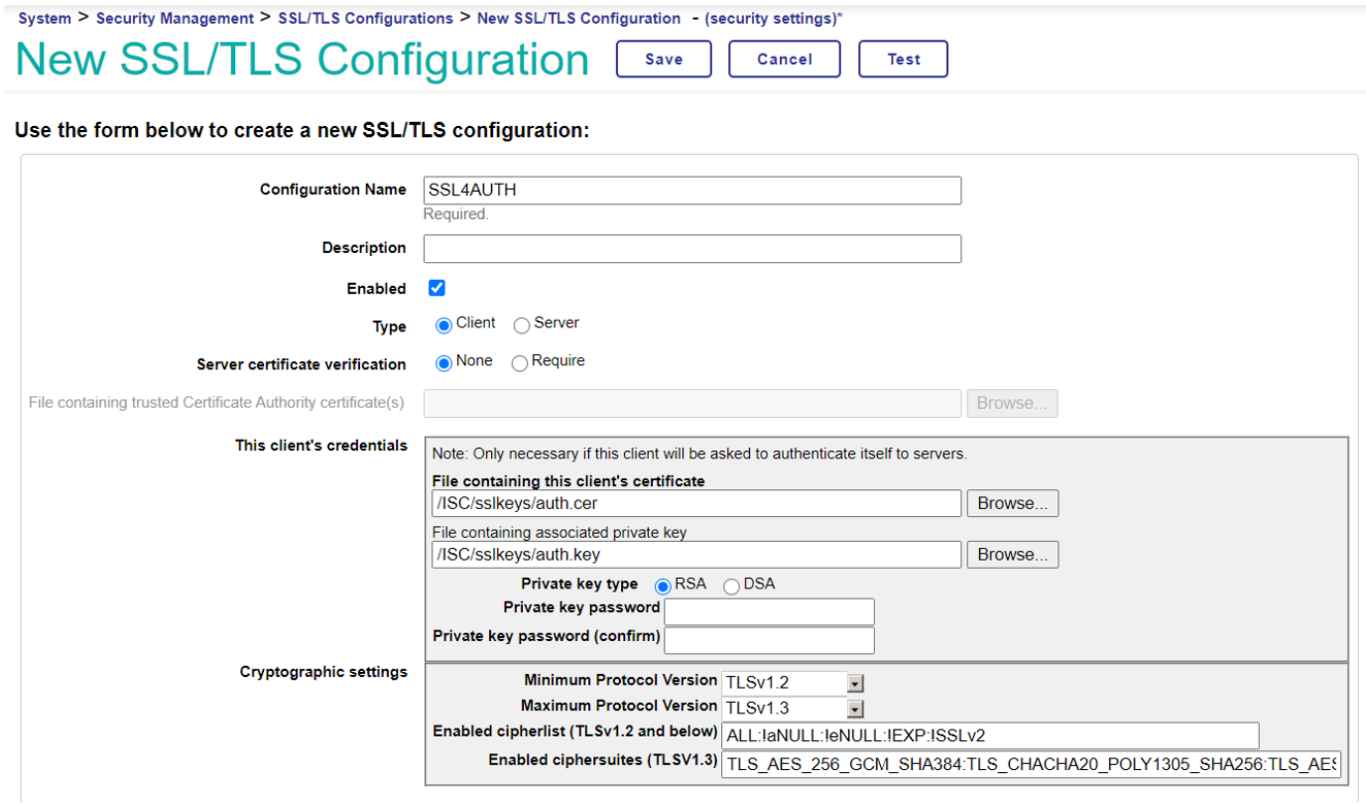
http**s**://[hostname]/csp/sys/UtilHome.csp ：Esta URL accede al Portal de administración utilizando una conexión SSL a través del Apache que se configuró.

Cómo crear una configuración SSL

Ahora que IRIS for Health está listo y en funcionamiento, y tenemos acceso al Portal de administración, vamos a crear la configuración SSL para los últimos preparativos.

Ve a Management Portal -> System Administration -> Security -> SSL/TLS Configuration y crea tres configuraciones SSL utilizando los tres pares de certificado/llave que preparaste.

Puedes seleccionar el nombre que quieras, pero en este artículo utilizaremos SSL4AUTH/SSL4CLIENT/SSL4RESSERVER, siguiendo los artículos anteriores relacionados con OAuth2.



*Acerca de compartir directorios entre servidores y contenedores

En la siguiente especificación de volúmenes que se encuentra en el archivo docker-compose se muestra la ubicación actual del directorio host = /ISC en el contenedor. Utiliza este directorio cuando especifiques el archivo del certificado en la configuración anterior, etc.

```
volumes:
  - .:/ISC
```

Este directorio no solo contendrá archivos, sino también archivos de la base de datos IRIS y archivos de la configuración. Consulta el documento "Persistencia de %SYS para almacenar datos de instancias persistentes" para obtener más información.

Cómo configurar OAuth2 en IRIS for Health

¡Ahora es el momento de entrar en los detalles del acceso a IRIS for Health utilizando OAuth2!

Configuración del servidor de autorización OAuth2

Primero, vamos a configurar el servidor de autorización OAuth2. Ve a Management Portal → System Administration → Security → OAuth 2.0 → Server.

Sigue estas instrucciones para ajustar la configuración.

Configuración en la pestaña "General"	
Issuer endpoint: Host name	Escribe el nombre actual del servidor
Issuer endpoint: Prefix	Puedes escribir cualquier valor que quieras, pero aquí lo establecimos como "authserver".
Supported grant types	En este artículo, solo utilizaremos "Authorization Code", pero si quieres probar otros "Grant Types", añade una marca de verificación. Además, añade una verificación a la "JWT authorization"
SSL/TLS configuration	Especifica la configuración SSL que acabas de añadir.

En la pestaña "Scopes", haz clic en "Add Supported Scope" para añadirlos. Más adelante, la pantalla de inicio de sesión del Código de Autorización mostrará la "Descripción" que escribiste aquí.

No hagas cambios de la configuración predeterminada en la pestaña "Intervals". En la pestaña "JWT Settings", seleccionaremos "RS512" como el algoritmo de firma.

En la última pestaña de "Customization", cambia la especificación Generate Token Class a %OAuth2.Server.JWT.

Una vez que hayas introducido la información, haz clic en el botón "Save" para guardar la configuración.

Ahora que tienes la configuración necesaria para que IRIS for Health se ejecute como un servidor de autorización de OAuth2, ¡estás listo para probarlo! ¡Intentaremos acceder desde Postman y veremos si podemos obtener un token de acceso!

Sin embargo, antes de hacer eso, necesitamos configurar dos cosas más.

Añadir la descripción del cliente

En primer lugar, añade la información de Postman para acceder como un cliente de OAuth2. El registro de clientes de OAuth2 se puede añadir por medio del registro dinámico u otros métodos.

Haz clic en "Client Description" en la página de configuración del servidor para continuar.

Haz clic en "Create Client Description" para añadir una entrada.

Sigue estas instrucciones para crear una suscripción de cliente.

Configuración en la pestaña “ General ”	
Name	Escribe el nombre que quieras. En este caso, hemos elegido “ postman ” .
Client Type	Elige “ Confidential ”
Redirect URLs	Haz clic en el botón “ Add URL ” para añadir una URL de redirección para Postman. https://www.getpostman.com/oauth2/callback como la URL de redirección para Postman.
Supported grant types	Especifica el mismo “ Authorization Code ” (Código de autorización) configurado en la configuración del servidor de autorización de OAuth2. (Predeterminado) Añade una marca si quieres probar también otros tipos de permisos. Sin embargo, la configuración debe ser la misma que la del servidor de autorización. Además, marca la casilla “ JWT authorization ” . Especifícalo aquí
Authenticated Signing Algorithm	Marca la opción "Autorización JWT" en Supported grant Types para que se pueda seleccionar. Selecciona “ RS512 ” .

Una vez que hayas introducido la información, haz clic en el botón “ Save ” para guardar la descripción del cliente.

Haz clic en la pestaña “ Client Credentials ” para ver el client ID y la clave privada del cliente para esta entrada. Necesitarás este ID y la clave privada cuando realices las pruebas desde POSTMAN.

Añadir una aplicación web

Es necesario agregar una configuración más antes de acceder desde POSTMAN. La pantalla de configuración del servidor de autorización OAuth2 ha determinado que el endpoint para esta configuración es https://<hostname>/authserver/oauth2. Para que el acceso a este endpoint sea administrado correctamente por IRIS, necesitamos añadir una aplicación web para esta ruta de URL.

Ve a System Administration→Security→Applications→Web Applications, y haz clic en “ Create a new web application ” .

Se proporciona una plantilla de aplicación web OAuth2, así que primero, selecciona “ /oauth2 ” desde “ Copy from ” .

Configuración de “ Edit Web Application ”	
Copy From	“ /oauth2 ” : Siempre selecciona este primero desde el menú desplegable.
Name	/authserver/oauth2
Enable	Marca el botón de la opción “ REST ” .

Después de introducir cada valor, guárdalo.

Cómo probar OAuth2 desde POSTMAN

Lo vamos a probar desde POSTMAN. Las pruebas también se pueden ejecutar desde otras herramientas o desde el propio programa. La explicación detallada de POSTMAN va más allá del alcance de este artículo, pero cabe señalar que la verificación del certificado SSL debe cambiar a OFF en la configuración de POSTMAN.

Después de crear una nueva solicitud en POSTMAN, selecciona “ OAuth 2.0 ” en la pestaña TYPE of Authorization y haz clic en “ Get New Access Token ” .

En la siguiente pantalla, introduce los valores según lo siguiente.

Configuración「GET NEW ACCESS TOKEN」	
Token Name	Escriba el nombre que quieras.
Grant Type	Selecciona “ Authorization Code ” .

Configuración「GET NEW ACCESS TOKEN」	
Callback URL	https://www.getpostman.com/oauth2/callback
Auth URL	"https://<hostname>/authserver/oauth2/authorize" Introduce el valor del endpoint +/"authorize". Al añadir "?uilocales=ja", puedes mostrar la pantalla de inicio de sesión en japonés.
Auth Token URL	https:///authserver/oauth2/token. Introduce el valor del endpoint +/"token".
Client ID	Introduce el client ID que aparece en la pestaña de Client Credentials después de registrarte para la descripción del cliente.
Client Secret	Introduce la clave privada del cliente, que se muestra en la pestaña Client Credentials después de registrar la descripción del cliente.
Scope	Introduce el scope registrado en la configuración del servidor de autorización, por ejemplo, “ scope1 ” . También puedes especificar varios scopes separados por espacios.
State	Introduce el parámetro State, que se utiliza para las medidas contra el CSRF. No se utiliza explícitamente, pero no se puede dejar en blanco, por lo que se introduce una cadena arbitraria.

Después de introducir los parámetros y hacer clic en el botón “ Request Token ” , verás la pantalla de inicio de sesión:

Intenta iniciar sesión con la información del usuario (por ejemplo, SYSTEM) con acceso al Portal de administración.

En la siguiente pantalla después del inicio de sesión, puedes decidir conceder permisos a esta aplicación. Después de hacer clic en “ Allow ” , si el token de acceso aparece en la siguiente pantalla, como se muestra a continuación, ¡la prueba de adquisición del token de acceso se realizó con éxito!

Cómo probar OpenID Connect

IRIS for Health puede llevar a cabo el procesamiento de autorización de OAuth2, así como el procesamiento de autenticación compatible con OpenID Connect. Consulta este documento para obtener más información.

En esta configuración, OpenID Connect está habilitado, ¡así que vamos a probar si también podemos obtener el token de ID de OpenID Connect!

Es fácil de implementar. En la pantalla GET NEW ACCESS TOKEN, añade “ openid ” al scope y realiza una solicitud.

OpenID Connect también se mostrará en la página de solicitud de autorización. Después de que hayas iniciado sesión y hayas dado tus permisos, asegúrate de obtener también un token de identificación (idtoken) cuando veas la siguiente pantalla. (Es posible que necesites desplazarse hacia abajo por la pantalla.)

¿Pudiste obtener el token de acceso y el idtoken?

Aunque hay algunos preparativos, como los certificados que requieren un poco de tiempo y esfuerzo, podríamos construir un servidor de autorización OAuth2 muy sencillo utilizando IRIS for Health.

En el siguiente artículo de esta serie, finalmente os mostraré cómo crear un repositorio FHIR y registrarlo como un servidor de recursos OAuth2. Y también cómo acceder con REST al repositorio FHIR por medio de un token de acceso OAuth2 desde POSTMAN.

#FHIR #OAuth2 #InterSystems IRIS for Health

URL de fuente:https://es.community.intersystems.com/post/c%C3%B3mo-crear-un-repositorio-fhir-configuraci%C3%B3n-del-servidor-de-autorizaci%C3%B3nde-recursos-oauth2-en