

Article

[Guillaume Rongier](#) · Oct 19, 2022 13m de lecture

Construction d'un référentiel FHIR + le serveur d'autorisation OAuth2/serveur de ressources sur IRIS for Health - Partie 2

Bonjour, chers développeurs !

Dans cet article, nous allons nous concentrer sur OAuth2, un protocole qui est de plus en plus utilisé en combinaison avec FHIR pour effectuer des autorisations.

Dans cette partie 1, nous allons démarrer le conteneur Docker pour IRIS for Health et Apache, configurer la fonction de serveur d'autorisation OAuth2 sur IRIS for Health, y accéder depuis l'outil de développement REST Postman, et obtenir un jeton d'accès.

En outre, dans la deuxième partie et au-delà, nous ajouterons la fonctionnalité de référentiel FHIR à IRIS for Health, nous ajouterons la configuration du serveur de ressources OAuth2 et nous expliquerons comment exécuter des requêtes FHIR avec des jetons d'accès depuis Postman.

Plusieurs excellents articles ont déjà été publiés au sein de la communauté des développeurs pour expliquer la fonctionnalité OAuth2 des produits d'InterSystems ; cependant, je voudrais expliquer à nouveau comment configurer la dernière version.

[Mise en œuvre d'InterSystems IRIS Open Authorization Framework \(OAuth 2.0\) - partie 1](#)

Dans cet article, nous utiliserons la dernière version d'InterSystems IRIS for Health 2020.3 Preview Edition. Si vous envisagez de créer un environnement basé sur cet article, veuillez à utiliser cette version ou une version ultérieure du kit. Certaines fonctionnalités ne sont pas incluses dans les produits antérieurs à cette version.

Préparatifs préliminaires

La première étape consiste à effectuer des préparatifs préliminaires. Il y a beaucoup de choses à préparer pour construire un environnement sécurisé.

IRIS for Health 2020.3 Preview Edition est uniquement disponible en version conteneur Docker. ([InterSystems Docker Hub/IRIS for Health](#))

Pour effectuer la configuration d'OAuth2, vous devrez également effectuer la configuration du serveur web et de SSL. Dans cet article, nous utiliserons Apache.

Lors de la configuration SSL sur Apache, le certificat de configuration SSL doit correspondre au nom d'hôte du serveur. Veuillez tenir compte de ce point.

Obtenir des fichiers d'exemple à partir du dépôt GitHub d'intersystems-jp

Le fichier docker-compose.yml/Dockerfile et d'autres exemples de fichiers utilisés dans cette configuration sont disponibles dans le dépôt GitHub réservé à la communauté des développeurs InterSystems.

Tout d'abord, décompressez ce fichier dans votre environnement à l'aide de la commande suivante. (Vous pouvez également le faire à partir de la pièce jointe à cet article).

Ce fichier docker-compose.yml/Dockerfile et d'autres fichiers sont créés en se référant à l' [application iris-webgateway-example](#) published on OpenExchange.

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

Modification de la configuration en fonction du kit utilisé

Dans ce fichier docker-compose.yml, deux conteneurs sont configurés pour être démarrés : le conteneur IRIS for Health et le conteneur Apache (httpd) seront créés par la commande docker build.

Le fichier docker-compose.yml, disponible sur GitHub, fait référence à IRIS for Health Community Edition Preview Edition (2020.3.200.0).

L'édition communautaire peut être utilisée pour l'évaluation des produits InterSystems.

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

Si vous utilisez une version différente (version officielle ou version plus récente), veuillez modifier cette partie de la spécification.

Le conteneur Apache sera construit avec le contenu du Dockerfile, qui nécessite un kit [WebGateway](#) pour se connecter à IRIS depuis Apache.

Pour savoir comment obtenir ce kit, les partenaires d'InterSystems peuvent consulter le site de téléchargement du kit WRC ou contacter le centre de support WRC.

Pour toute autre question, veuillez nous contacter à [cet adresse](#).

Modifiez les parties suivantes du Dockerfile en fonction du produit que vous avez obtenu. Quel que soit le système d'exploitation de la machine hôte (Windows/Ubuntu/CentOS), la plate-forme sera lnxubuntux64 car le système d'exploitation du conteneur httpd de base est Debian.

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

Préparation d'un certificat SSL

À l'étape suivante, un certificat SSL est préparé. Lorsque l'on accède à l'autorisation OAuth2, le certificat SSL défini dans le serveur Web est vérifié pour voir s'il correspond à l'URL à laquelle on accède.

Il n'est pas nécessaire d'utiliser un certificat officiel ; il est possible d'utiliser OpenSSL, etc. Saisissez le nom d'hôte dans le champ "Nom commun" lors de la création du certificat.

De plus, comme le certificat que vous avez créé sera chargé automatiquement au moment du lancement, vous devez modifier le fichier pour qu'il ne nécessite pas de mot de passe. Veuillez vous référer à la commande suivante.

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

Placez les fichiers CRT et KEY créés dans le même répertoire que le Dockerfile, avec les noms de fichiers server.crt / server.key respectivement.

En plus de l'utiliser avec le serveur web Apache, vous aurez besoin d'un certificat SSL pour la configuration d'OAuth2. Il n'est pas nécessaire d'entrer un nom d'hôte, etc., mais vous devez créer trois ensembles. (Dans les configurations suivantes, ils apparaissent sous la forme auth.cer/auth.key , client.cer/client.key , resserver.cer/resserver.key)

Construction de docker et démarrage d'un conteneur docker

Maintenant, vous êtes enfin prêt ! En plus des quatre fichiers que vous avez téléchargés, vous avez maintenant un

ensemble d'installation de la passerelle Web et deux certificats SSL dans votre répertoire. Faites attention aux autorisations d'accès et d'exécution de chaque fichier. (Par exemple, j'ai ajouté la permission d'exécution à `webgateway-entrypoint.sh`).

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

Une fois lancé, utilisez la commande `docker ps` pour vérifier que les deux conteneurs fonctionnent.

```
Nom du conteneur Apache?<directoryname>_web
Nom du conteneur IRIS for Health?store/intersystems/irishealth-
community:2020.3.0.200.0?ou autre nom en fonction de l'ensemble)
```

Essayez maintenant d'accéder au portail de gestion selon les trois méthodes suivantes. Si la troisième méthode fonctionne, votre configuration SSL via le serveur web Apache est un succès !

`http://[hostname]:52773/csp/sys/UtilHome.csp` : L'accès à cette URL se fait par le biais de Private Apache dans le conteneur IRIS. Elle ne passe pas par l'Apache configuré.

`http://[hostname]/csp/sys/UtilHome.csp` : Cette URL permet d'accéder au portail de gestion via l'Apache configuré.

`https://[hostname]/csp/sys/UtilHome.csp` : Cette URL permet d'accéder au portail de gestion en utilisant une connexion SSL via Apache, que vous avez configurée.

Création d'une configuration SSL

Maintenant que IRIS for Health est opérationnel et que nous avons accès au portail de gestion, créons la configuration SSL pour les derniers préparatifs.

Allez sur le Portail de gestion -> Administration du système -> Sécurité -> Configuration SSL/TLS et créez trois configurations SSL en utilisant les trois paires de clés de certificat que vous avez préparées.

Vous pouvez choisir le nom que vous voulez, mais dans cet article, nous utiliserons `SSL4AUTH/SSL4CLIENT/SSL4RESSERVER`, conformément aux articles précédents sur OAuth2.

System > Security Management > SSL/TLS Configurations > New SSL/TLS Configuration - (security settings)*

New SSL/TLS Configuration

Use the form below to create a new SSL/TLS configuration:

Configuration Name
Required.

Description

Enabled

Type Client Server

Server certificate verification None Require

File containing trusted Certificate Authority certificate(s)

This client's credentials

Note: Only necessary if this client will be asked to authenticate itself to servers.

File containing this client's certificate

File containing associated private key

Private key type RSA DSA

Private key password

Private key password (confirm)

Cryptographic settings

Minimum Protocol Version	<input type="text" value="TLSv1.2"/>
Maximum Protocol Version	<input type="text" value="TLSv1.3"/>
Enabled cipherlist (TLSv1.2 and below)	<input type="text" value="ALL:!aNULL:!eNULL:!EXP:!SSLv2"/>
Enabled ciphersuites (TLSv1.3)	<input type="text" value="TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES"/>

*À propos du partage de répertoire entre les hôtes et les conteneurs

La spécification des volumes suivants dans le fichier docker-compose indique l'emplacement actuel du répertoire hôte = /ISC dans le conteneur.

Veuillez utiliser ce répertoire lorsque vous spécifiez le fichier de certificat dans les paramètres ci-dessus, etc.

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

Ce répertoire contiendra non seulement des fichiers mais aussi des fichiers de base de données IRIS et des fichiers de configuration.

Consultez le document [Persistant %SYS pour le stockage des données d'instance persistantes](#) pour plus d'information.

Configuration de OAuth2 dans IRIS for Health

Il est maintenant temps d'entrer dans les détails de l'accès à IRIS for Health en utilisant OAuth2 !

Configuration du serveur d'autorisation OAuth2

Tout d'abord, configurons le serveur d'autorisation OAuth2 !

Allez dans Portail de gestion Administration du système Sécurité OAuth 2.0 Serveur.

Suivez les instructions ci-dessous pour configurer les paramètres.

Paramètres dans l'onglet "Général"

Point de terminaison de l'émetteur : Nom d'hôte

Point de terminaison de l'émetteur : Prefix

Saisissez le nom d'hôte réel.

Vous pouvez saisir la valeur de votre choix, mais ici nous l'avons fixée à "authserver".

Types de subventions pris en charge

Dans cet article, nous n'utiliserons que le "Code d'autorisation", mais si vous souhaitez tester d'autres

Paramètres dans l'onglet "Général"

"Types de subventions", veuillez ajouter une coche.

Ajoutez également une coche à "Autorisation JWT"

Configuration SSL/TLS

Spécifiez la configuration SSL que vous venez d'ajouter.

Dans l'onglet "Scopes", cliquez sur "Ajouter un Scope supporté" pour les ajouter.

Plus tard, l'écran de connexion du code d'autorisation affichera la "description" que vous avez écrite ici.

Ne modifiez pas l'onglet "Intervalles" par rapport à la valeur par défaut.

Dans l'onglet "Paramètres JWT", sélectionnez "RS512" comme algorithme de signature.

Dans le dernier onglet "Personnalisation", changez la spécification "Génération de classe de jeton" en %OAuth2.Server.JWT.

Une fois que vous avez saisi les informations, cliquez sur le bouton "Enregistrer" pour sauvegarder la configuration.

Maintenant que vous avez la configuration nécessaire pour qu'IRIS for Health fonctionne comme un serveur d'autorisation OAuth2, vous êtes prêt à l'essayer ! Essayons d'y accéder à partir de Postman et voyons si nous pouvons obtenir un jeton d'accès !

Cependant, avant de faire cela, nous devons effectuer deux autres configurations.

Ajout d'une description du client

Tout d'abord, ajoutez les informations de Postman auxquelles vous souhaitez accéder en tant que client OAuth2. L'enregistrement du client OAuth2 peut être ajouté par le biais d'un enregistrement dynamique ou d'autres méthodes.

Cliquez sur "Description du client" sur la page de configuration du serveur pour continuer.

Cliquez sur "Créer une description du client" pour ajouter une entrée.

Suivez les instructions ci-dessous pour créer une souscription du client.

Paramètres dans l'onglet "Général"

Nom

Entrez un nom de votre choix. Dans ce cas, nous avons choisi "postman".

Type du Client

Sélectionnez "Confidentiel"

Redirection d'URLs

Cliquez sur le bouton "Add URL" pour ajouter une URL de redirection pour Postman.

<https://www.getpostman.com/oauth2/callback> comme URL de redirection pour Postman.

Types de subventions pris en charge

Spécifiez le même "Code d'autorisation" (Authorization Code) qui a été configuré dans les paramètres du serveur d'autorisation OAuth2. (Par défaut) Ajoutez un contrôle si vous souhaitez également tester d'autres types de subventions. Cependant, les paramètres doivent être les mêmes que la configuration du serveur d'autorisation. Cochez également la case "Autorisation JWT". Précisez ici

Authenticated Signing Algorithm

Vérifiez "JWT authorization" sous Supported grant Types (Types de subventions pris en charge) pour pouvoir le sélectionner. Sélectionnez "RS512".

Une fois que vous avez saisi les informations, cliquez sur le bouton "Enregistrer" pour sauvegarder la description du client.

Cliquez sur l'onglet "Références du client" pour voir l'ID du client et la clé privée du client pour cette entrée. Vous aurez besoin de cet ID et de cette clé privée lorsque vous effectuerez des tests à partir de POSTMAN.

Ajout d'une application Web

Un autre paramètre important doit être ajouté avant d'y accéder à partir de POSTMAN.

L'écran de configuration du serveur d'autorisation OAuth2 a déterminé que le point de terminaison pour cette configuration est `https://<hostname>/authserver/oauth2`.

Pour que l'accès à ce point de terminaison soit traité correctement par IRIS, nous devons ajouter une application Web pour cette route URL.

Allez dans Administration système > Sécurité > Applications > Applications Web, et cliquez sur "Créer une nouvelle application Web".

Un modèle d'application web OAuth2 est fourni, il faut donc d'abord sélectionner "/oauth2" dans " Copier à partir de ".

Paramètres "Editer l'application Web"

Copie à partir de

“ /oauth2 ” : Sélectionnez toujours celui-ci en premier dans la liste déroulante.

Nom

/authserver/oauth2

Activation

Cochez la case d'option "REST".

Après avoir saisi chaque valeur, enregistrez-la.

Test d'OAuth2 à partir de POSTMAN

Testons-le à partir de POSTMAN.

Les tests peuvent également être effectués à partir d'autres outils ou du programme lui-même.

L'explication détaillée de POSTMAN dépasse le cadre de cet article, mais un point à noter est que la vérification du certificat SSL doit être changée en OFF dans les paramètres de POSTMAN.

Après avoir créé une nouvelle demande dans POSTMAN, sélectionnez "OAuth 2.0" dans l'onglet TYPE d'autorisation et cliquez sur "Obtenir un nouveau jeton d'accès".

Dans l'écran suivant, saisissez les valeurs selon les indications suivantes.

Paramètres 「 GET NEW ACCESS TOKEN 」

Nom du jeton

Entrez un nom de votre choix.

Type de subvention

Choisissez "Code d'autorisation".

Callback URL

<https://www.getpostman.com/oauth2/callback>

Auth URL

`https://<hostname>/authserver/oauth2/authorize`

Saisissez la valeur du point de terminaison `+/authorize`.

En ajoutant `?ui_locales=ja`, vous pouvez afficher l'écran de connexion en japonais.

Auth Token URL

`https://authserver/oauth2/token`. Saisissez la valeur du point de terminaison `+/token`.

Client ID

Saisissez l'ID du client affiché dans l'onglet Références du client après l'enregistrement de la description du client.

Clé Secrète du client

Saisissez la clé privée du client, affichée dans l'onglet Références du client après l'enregistrement de la description du client.

Champ

Entrez le champ d'application enregistré dans la configuration du serveur d'autorisation, par exemple "scope1". Vous pouvez également spécifier plusieurs champs séparés par des espaces.

État

Entrez le paramètre d'état "State", qui est utilisé pour les contre-mesures contre CSRF. Il n'est pas explicitement utilisé mais ne peut pas être laissé vide, donc nous entrons une chaîne arbitraire.

Après avoir entré les paramètres et cliqué sur le bouton " Demande de jeton ", vous voyez l'écran de connexion

comme indiqué ci-dessous.

Essayez de vous connecter avec les informations de l'utilisateur (par exemple, SYSTEM) ayant accès au portail de gestion.

Sur l'écran suivant après la connexion, vous pouvez décider d'accorder des permissions à cette application. Après avoir cliqué sur " Autoriser ", si le jeton d'accès s'affiche sur l'écran suivant, comme indiqué ci-dessous, le test d'acquisition du jeton d'accès est réussi !

Test d'OpenID Connect

IRIS for Health peut effectuer un traitement d'autorisation OAuth2 ainsi qu'un traitement d'authentification conforme à OpenID Connect.

Pour plus de détails consultez [ce document](#).

Dans cette configuration, OpenID Connect est activé, alors testons si nous pouvons également obtenir le jeton d'identification OpenID Connect !

C'est facile à mettre en œuvre. Dans l'écran GET NEW ACCESS TOKEN, ajoutez "openid" à son champ d'application et faites une demande.

OpenID Connect sera également affiché sur la page de demande d'autorisation. Après avoir ouvert une session et donné vos autorisations, assurez-vous que vous obtenez également un jeton d'identification (idtoken) lorsque vous voyez l'écran suivant. (Vous devrez peut-être faire défiler l'écran).

Avez-vous réussi à obtenir le jeton d'accès et l'idtoken ?

Bien que certains préparatifs, tels que les certificats, nécessitent un peu de temps et d'efforts, nous pourrions construire un serveur d'autorisation OAuth2 avec une telle simplicité en utilisant IRIS for Health, une plateforme de base de données.

Dans la prochaine partie de cette série, je vous montrerai enfin comment construire un référentiel FHIR, enregistrer le référentiel FHIR en tant que serveur de ressources OAuth2 et vous montrer comment accéder par REST au référentiel FHIR en utilisant un jeton d'accès OAuth2 depuis POSTMAN.

[#FHIR #OAuth2 #InterSystems IRIS for Health](#)

URL de la source:<https://fr.community.intersystems.com/post/construction-dun-r%C3%A9f%C3%A9rentiel-fhir-le-serveur-dautorisation-oauth2serveur-de-ressources-sur-0>