Article
Guillaume Rongier · Juin 22, 2022 4m de lecture

Sécurisation de vos API avec OAuth 2.0 dans le cadre de la gestion des API d'InterSystems - Partie 2

Dans cette série d'articles en trois parties, il est montré comment vous pouvez utiliser IAM pour ajouter simplement de la sécurité, selon les normes OAuth 2.0, à un service précédemment non authentifié déployé dans IRIS.

Dans la première partie, nous avons fourni des informations sur OAuth 2.0 ainsi que des définitions et des configurations initiales d'IRIS et d'IAM afin de faciliter la compréhension de l'ensemble du processus de sécurisation de vos services.

Cette partie va maintenant discuter et montrer en détail les étapes nécessaires pour configurer IAM pour valider le jeton d'accès présent dans la demande entrante et transmettre la demande au backend si la validation réussit.

La dernière partie de cette série abordera et démontrera les configurations nécessaires pour que IAM génère un jeton d'accès (agissant comme un serveur d'autorisation) et le valide, ainsi que quelques considérations finales importantes.

Si vous voulez essayer IAM, veuillez contacter votre représentant commercial d'InterSystems.

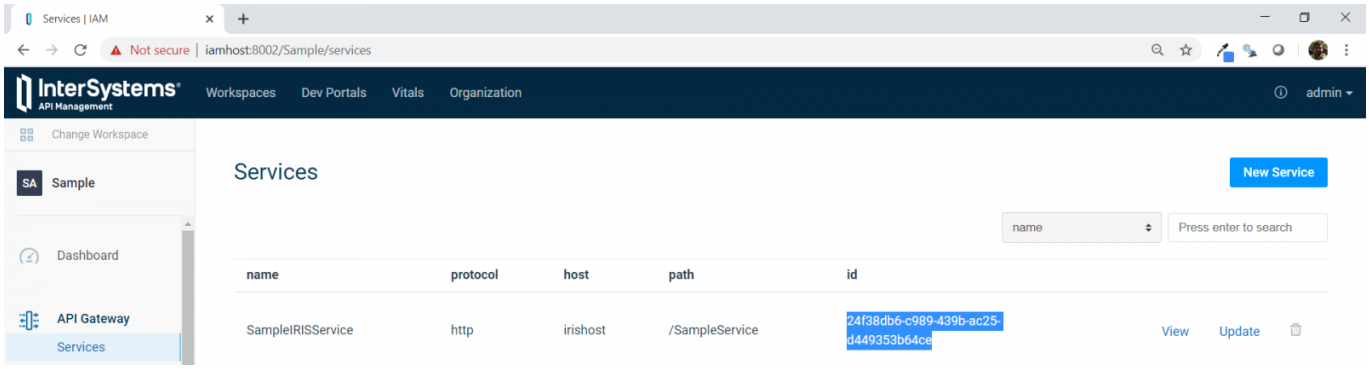
Scénario 1 : IAM comme validateur de jetons d'accès

Dans ce scénario, il sera utilisé un serveur d'autorisation externe qui génère un jeton d'accès au format JWT (JSON Web Token). Ce JWT est signé à l'aide de l'algorithme RS256 et d'une clé privée. Afin de vérifier la signature du JWT, l'autre partie (dans ce cas IAM) doit avoir la clé publique, fournie par le serveur d'autorisation.

Ce JWT généré par le serveur d'autorisation externe comprend également, dans son corps, une déclaration appelée "exp" contenant l'horodatage de la date d'expiration de ce jeton, et une autre déclaration appelée "iss" contenant l'adresse du serveur d'autorisation.

Par conséquent, IAM doit vérifier la signature du JWT avec la clé publique des serveurs d'autorisation et l'horodatage d'expiration contenu dans la déclaration "exp" à l'intérieur du JWT avant de transmettre la demande à IRIS.

Afin de configurer cela dans IAM, commençons par ajouter un plugin appelé "JWT" à notre "SampleIRISService" dans IAM. Pour ce faire, allez sur la page Services dans IAM et copiez l'identifiant du "SampleIRISService", nous allons l'utiliser plus tard.



Ensuite, allez dans Plugins, cliquez sur le bouton "New Plugin", localisez le plugin "JWT" et cliquez sur Enable.

Dans la page suivante, copiez l'identifiant "SampleIRISService" dans le champ "serviceid" et cochez la case "exp" dans le paramètre "config.claimstoverify".

Notez que la valeur du paramètre "config.keyclaimname" est "iss". Nous allons l'utiliser plus tard.

Ensuite, appuyez sur le bouton "Create" (créer).

Cela fait, allez dans la section "Consumers" dans le menu de gauche et cliquez sur notre "ClientApp" précédemment créée. Allez dans l'onglet "Credentials" (identifiants) et cliquez sur le bouton " New JWT Credential " (nouveau Identifiants JWT).

Dans la page suivante, sélectionnez l'algorithme utilisé pour signer le JWT (ici RS256) et copiez la clé publique dans le champ "rsapublickey" (il s'agit de la clé publique qui vous a été fournie par le serveur d'autorisation au format PEM).

Dans le champ "key", vous devez insérer le contenu du revendication JWT que vous avez entré dans le champ "config.keyclaimname" lors de l'ajout du plugin JWT. Donc, dans ce cas, je dois insérer le contenu de la revendication iss de mon JWT, qui, dans mon cas, est l'adresse du serveur d'autorisation.

Après cela, cliquez sur le bouton "Créer".
Hint: À des fins de débogage, il existe un outil en ligne permettant de décoder le JWT afin que vous puissiez vérifier les revendications et leurs valeurs et vérifier les signatures en insérant la clé publique. Voici le lien de cet outil en ligne : https://jwt.io/#debugger

Maintenant, avec l'ajout du plugin JWT, il n'est plus possible d'envoyer la requête sans authentification. Comme vous pouvez le voir ci-dessous, une simple demande GET, sans authentification, renvoie à l'URL

http://iamhost:8000/event/1

un message non autorisé avec le code de statut "401 Unauthorized".

Afin d'obtenir les résultats d'IRIS, nous devons ajouter le JWT à la requête.

Par conséquent, nous devons d'abord demander le JWT au serveur d'autorisation. Le serveur d'autorisation personnalisé que nous utilisons ici renvoie un JWT si une demande POST est faite avec quelques paires clé-valeur dans le corps, y compris des informations sur l'utilisateur et le client, à l'URL suivante :

https://authorizationserver:5001/auth

Voici à quoi ressemblent cette requête et sa réponse :

Ensuite, vous pouvez ajouter le JWT obtenu à partir de la réponse ci-dessous dans le header d'autorisation en tant que jeton de porteur Bearer Token et envoyer une requête GET à la même URL utilisée précédemment :

http://iamhost:8000/event/1

Vous pouvez également l'ajouter en tant que paramètre de chaîne de recherche, la clé de la chaîne de recherche étant la valeur spécifiée dans le champ "config.uriparamnames" lors de l'ajout du plugin JWT qui, dans ce cas, est "jwt" :

Enfin, il y a aussi la possibilité d'inclure JWT dans la requête en tant que cookie, si un nom est saisi dans le champ "config.cookienames".

Passez à la troisième et dernière partie de cette série pour comprendre les configurations nécessaires pour générer un jeton d'accès IAM et le valider, ainsi que quelques considérations finales importantes.

#API #OAuth2 #REST API #Sécurité #InterSystems IRIS

URL de la source:https://fr.community.intersystems.com/post/s%C3%A9curisation-de-vos-api-avec-oauth-20-dans-le-cadre-de-la-gestion-des-api-dintersystems-0