記事
Shintaro Kaminaka · 2021年4月15日 7m read

IRIS for Health上でFHIRリポジトリ＋OAuth2認可サーバ/リソースサーバ構成を構築するパート3(OAuth2スコープ編)

開発者の皆さん、こんにちは。

以前の記事でIRIS for Health上でFHIRリポジトリを構築し、OAuth2認証を構成する方法をご紹介しました。
IRIS for Health
2020.4がリリースされ、FHIRリポジトリのアクセストークンをチェックする機能が追加になりました。

ドキュメントはAccess Token Scopesです。

この記事ではドキュメントの記載も抜粋しながらこの機能を紹介していきます。

Basic Processing

The access token that accompanies a request must include at least one patient clinical scope or user clinical scope, or else the request is rejected with an HTTP 403 error. If an access token contains both a patient clinical scope and a user clinical scope, the FHIR server enforces the patient clinical scope while ignoring the user clinical scope.

> (訳)リクエストに添付されるアクセストークンには、患者のクリニカル・スコープまたはユーザーのクリニカル・スコープが少なくとも1つ含まれていなければならず、そうでなければリクエストはHTTP 403エラーで拒否されます。アクセストークンに患者のクリニカル・スコープとユーザーのクリニカル・スコープの両方が含まれている場合、FHIRサーバーは患者のクリニカル・スコープを強制し、ユーザーのクリニカル・スコープを無視します。

この機能が追加になったことで、以前の記事でご紹介していたような、"scope1"のようなスコープ指定では403エラーとなり、FHIRリポジトリから応答を受け取ることはできなくなりました。(前回の構成が残っている方はぜひ2020.4にアップグレードして403エラーになることを試してみてください。)

この記事では、正しいスコープを指定して、FHIRリポジトリにアクセスする方法をご紹介します。
ベースとなるOA
uth2サーバ構成やクライアント構成に関しては以前と変わりませんので、以前の記事を参考にしてください。

SMART on FHIR : Scopes and Launch Context

今回のIRISの実装は、SMART on FHIRプロジェクトのスコープ定義をベースにしています。

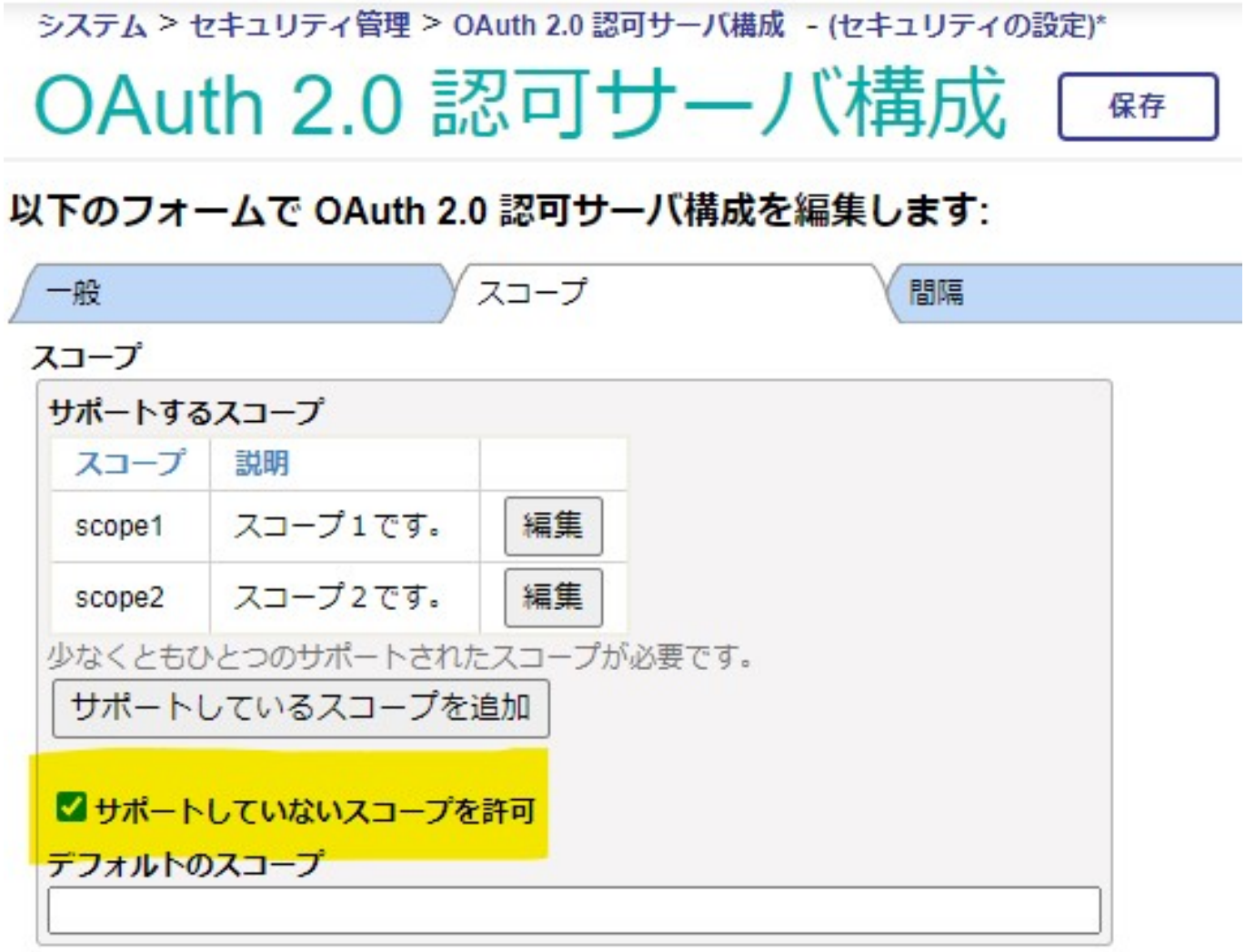
SMART App Launch: Scopes and Launch Context

このサイトのQuick Startに記載されている以下のスコープをベースにこの記事の検証を進めてみたいと思います。
詳細についてはSMART on FHIRの上記解説サイトをご覧ください。

Scope	Grant
patient/*.read	現在の患者のあらゆるリソースを読むことができる許可
user/*.*	現在のユーザーがアクセスできるすべてのリソースの読み取りと書き込みの許可

OAuth 2.0 認可サーバ構成の変更

まず、OAuth2認可サーバ構成で、サポートしていないスコープが許可されるように構成します。
OAuth2認可サーバの「スコープ」タブで、「サポートしていないスコープを許可」にチェックを入れます。



スコープを指定してPatientリソースを登録する

それでまず、Patientリソースを登録してみましょう。
Create Interactionを行う場合には、**user clinical scope** が必要になります。以下はIRIS for Healthのドキュメントからの抜粋です。

Create Interaction

Requests to create a new Patient resource must include a user clinical scope that gives write permissions (**user/Patient.write** or **user/Patient.***). You cannot perform a **create** interaction for a Patient resource with a patient clinical scope; patient clinical scopes must include a patient context value, and the **create** interaction cannot include a resource id.

> (訳)新規にPatientリソースを作成する要求には、書き込み権限(user/Patient.writeまたはuser/Patient.*)を与えるユーザークリニカルスコープを含める必要があります。患者臨床スコープには患者コンテキスト値を含める必要があり、作成インタラクションにリソースIDを含めることはできません

先ほど、でてきたスコープ「user/*.*」を追加して、アクセストークンを取得し、Patientリソースを登録してみましょう。

以下の図はRESTツールPOSTMAN上で、スコープを指定している画面です。

任意のスコープを許可する設定が正しく動作していれば以下のような確認画面が表示されます。

「許可」をクリックして、アクセストークンを取得します。このアクセストークンを使ってPatientリソースを登録してみましょう。
まず登録するPatientリソースを用意してください。FHIR公式ページ
から取得しても良いですし、過去のコミュニティ記事から取得しても良いです。

POSTMANを使う場合、Bodyタブに登録したいリソースを貼り付けます。

前回記事同様、アクセストークン＋ベーシック認証を行う必要があるので、取得したアクセストークンを
accesstoke=eyXXX としてパラメータに追加し、Authorizationタブではベーシック認証を選択してください。

メソッドが**POST**
であること、リクエストURLが正しくPatientを指定していることなども確認出来たら、登録を実行してみてください。Status 201 Created が返ってきたら登録は成功です。

登録が成功したら、続いて**GET**も試してみてください。
このスコープは読み取りの権限もありますから取得できるはずです。

リソースが取得できたら、今登録したPatientリソースのリソースIDも確認してください。以下の例では **1** になります。

スコープを指定してPatientリソースを取得する

では、Patientリソースが登録できたので、今度は異なるスコープを指定して先ほどのPatientリソースを取得してみましょう。

患者に紐づく情報を取得する場合、**patient clinical scope** が必要になります。

Patient Clinical Scope / Patient Context Value

If an access token includes a patient clinical scope, it must also include a patient context value (also known as “ launch context ”) that is the id of a Patient resource. This patient context value provides access to the specified Patient and its related resources.

> (訳)アクセストークンに患者の臨床範囲が含まれる場合、Patient リソースの ID である患者コンテキスト値（「起動コンテキスト」とも呼ばれる）も含まれなければならない。この患者コンテキスト値は、指定された患者及びその関連リソースへのアクセスを提供する。

今回はスコープに「patient/*.read」を追加します。もう一点どの患者情報にアクセスできるのか判断するために、Patientリソースのリソース**ID**情報を含むPatient Context Valueをスコープとして渡します。具体的には、「launch/patient/1」(リソースID＝１の場合)もスコープとして指定します。

POSTMANでは以下のように指定します。スコープの間は半角スペースで区切ります。

アクセストークンを取得できたら、先ほどと同じ手順でリクエストを投げてみましょう。
ただし、今回取得したアクセストークンで取得できるFHIRリソースはリソースID＝１のPatientリソースに紐づく情報だけであることにご注意ください。ということは、取得するためにRESTのパスは **Patient/1** となります。

例：https://<server>/csp/healthshare/fhirserver/fhir/r4/Patient/**1**?accesstoken=eyXXX

無事に先ほどのPatientリソースが取得できたらアクセス成功です！
次は、このトークンが本当にリソースID=1だけに制限されているかも確認してみましょう。他のリソースIDを指定するか、Patientリソース全体を取得するようなリクエストを投げてみてください（ただし存在しないPatientのリソースIDを指定するとそのリソースは存在しませんという別のエラーになります。）

例：https://<server>/csp/healthshare/fhirserver/fhir/r4/Patient/**2**?accesstoken=eyXXX
例：https://<server>/csp/healthshare/fhirserver/fhir/r4/Patient?accesstoken=eyXXX

Status 403エラーになりましたか？エラーが返れば確認は成功です。

まとめ

どのようなFHIRリソースをFHIRリポジトリに格納しているのか？また、どのようなアプリケーションを構築して、どのようにユーザにアクセス制限を設定したいのか？等により、利用すべきスコープの使い方は異なってきます。

安全なFHIRアプリケーションを構築するために、この新しいアクセストークンスコープ機能（とあとこの記事が）活用されると幸いです。

#FHIR #OAuth2 #セキュリティ #InterSystems IRIS for Health

ソースURL:https://jp.community.intersystems.com/post/iris-health%E4%B8%8A%E3%81%A7fhir%E3%83%AA%E3%83%9D%E3%82%B8%E3%83%88%E3%83%AA%EF%BC%8Boauth2%E8%AA%8D%E5%8F%AF%E3%82%B5%E3%83%BC%E3%83%90%E3%83%AA%E3%82%BD%E3%83%BC%E3%82%B9%E3%82%B5%E3%83%BC%E3%83%90%E6%A7%8B%E6%88%90%E3%82%92%E6%A7%8B%E7%AF%89%E3%81%99%E3%82%8B%E3%83%91%E3%83%BC%E3%83%883oauth2%E3%82%B9%E3%82%B3%E3%83%BC%E3%83%97%E7%B7%A8