記事
Toshihiko Minamoto · 2021年7月28日 7m read

InterSystems API Management で OAuth 2.0 を使用して API を保護する - パート 3

この 3 部構成の記事では、IAM を使って、以前に IRIS にデプロイされた認証されていないサービスに OAuth 2.0 標準に従ったセキュリティを追加する方法を説明します。

パート 1 では、サービスを保護するプロセス全体を理解しやすくするために、IRIS と IAM の基本的な定義と構成を示しながら OAuth 2.0 の背景を説明しました。

パート 2 では、着信リクエストに存在するアクセストークンを検証し、検証が成功した場合にはそのリクエストをバックエンドに転送するように IAM を構成する手順について詳しく説明しました。

連載の最後となるこのパートでは、IAM がアクセストークンを生成（承認サーバーとして機能します）してそれを検証するために必要な構成と、重要な最終考慮事項を説明します。

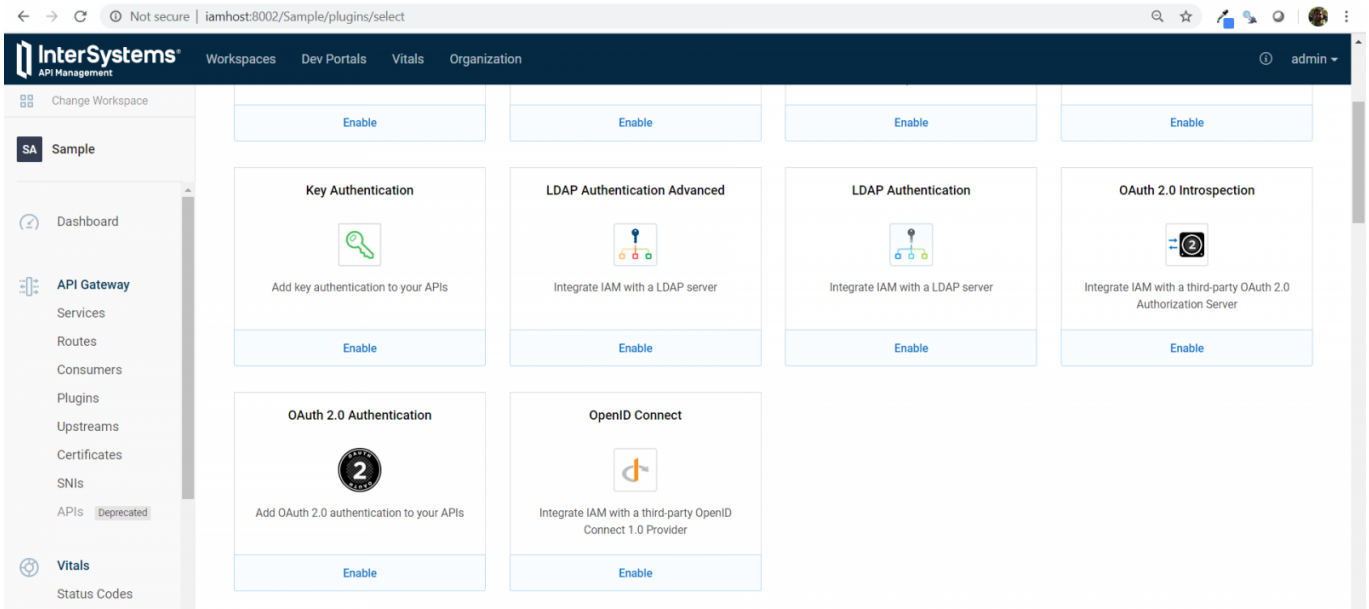
IAM をお試しになりたい方は、InterSystems 営業担当者にお問い合わせください。

シナリオ 2: 承認サーバーとアクセストークンのバリデーターとしての IAM

このシナリオでは、最初のシナリオとは異なり、「OAuth 2.0 Authentication」というプラグインを使用します。

このリソース所有者パスワード資格情報フローで IAM を承認サーバーとして使用するには、クライアントアプリケーションがユーザー名とパスワードを認証する必要があります。 IAM からアクセストークンを取得するリクエストは、認証が成功した場合にのみ行う必要があります。

プラグインを「SampleIRISService」に追加しましょう。 以下のスクリーンショットからわかるように、このプラグインを構成するために入力が必要なフィールドがいくつかあります。



まず、「SampleIRISService」の ID
を［serviceid］フィールドに貼り付けて、このプラグインをサービスに適用します。

［config.authheadername］フィールドには、承認トークンを運搬するヘッダー名を指定します。
ここでは、デフォルト値の「authorization」のままにします。

「OAuth 2.0 Authentication」プラグインは、認可コードグラント、クライアント資格情報、インプリシットグラント、またはリソース所有者パスワード資格情報グラントの OAuth 2.0 フローをサポートしています。 この記事ではリソース所有者パスワードを使用しているため、［config.enablepasswordgrant］チェックボックスをオンにします。

［config.provisionkey］フィールドには、プロビジョンキーとして使用される任意の文字列を入力します。
この値は、IAM にアクセストークンをリクエストするために使用されます。

ここでは、ほかのすべてのフィールドはデフォルト値のままにしました。
各フィールドの完全なリファレンスは、こちらからアクセスできるプラグインのドキュメントをご覧ください。

最終的に、プラグイン構成は次のようになります。

プラグインが作成されたら、「ClientApp」コンシューマーへの資格情報を作成する必要があります。

作成するには、左メニューの［コンシューマー］に移動して［ClientApp］をクリックします。
次に［資格情報］タブをクリックして［新しい OAuth 2.0 アプリケーション］ボタンをクリックします。

次のページでは、［名前］フィールドにアプリケーションを識別するための任意の名前を入力し、［clientid］フィールドと［clientsecret］にクライアント ID とクライアントシークレットを定義し、［redirecturi］フィールドに承認後にユーザーが送信されるアプリケーションの URL を入力します。
そして、［作成］をクリックします。

これで、リクエストを送信する準備が整いました。

最初に行うのは、IAM からアクセストークンを取得するためのリクエストです。 「OAuth 2.0 Authentication」プラグインは自動的に、作成済みのルートに「/oauth2/token」パスを追加して、エンドポイントを作成します。
注意: HTTPS プロトコルと、TLS/SSL リクエストをリスンする IAM のプロキシポート（デフォルトポートは 8443）を使用していることを確認してください。 これは OAuth 2.0 の要件です。

したがって、この場合、次の URL に POST リクエストを行う必要があります。

https://iamhost:8443/event/oauth2/token

リクエスト本文に、次の JSON を含める必要があります。

```
 {
 "clientid": "clientid",
 "clientsecret": "clientsecret",
 "granttype": "password",
 "provisionkey": "provisionkey",
 "authenticateduserid": "1"
 }
```

ご覧のとおり、この JSON には「OAuth 2.0 Authentication」プラグイン作成中に定義した値（「granttype」や「provisionkey」など）とコンシューマーの資格情報の作成中に定義した値（「clientid」や「clientsecret」など）が含まれています。

提供されたユーザー名とパスワードが正しく認証された場合には、クライアントアプリケーションによって「authenticateduserid」パラメーターも追加される必要があります。
この値は、認証されたユーザーを一意に識別するために使用されます。

リクエストとそれに対応するレスポンスは次のようになります。

これで、上記のレスポンスの「accesstoken」値を次の URL への GET リクエストの「ベアラートークン」として含めて、イベントデータを取得するリクエストを行えるようになりました。

https://iamhost:8443/event/1

アクセストークンが期限切れになった場合は、アクセストークンを取得するために使用したのと同じエンドポイントに、わずかに異なる本文を使って POST リクエストを送信し、期限切れのアクセストークンとともに受け取ったリフレッシュトークンを使用して、新しいアクセストークンを生成することができます。

```
 {
 "clientid": "clientid",
 "clientsecret": "clientsecret",
 "granttype": "refreshtoken",
 "refreshtoken": "E50m6Yd9xWy6lybgo3DOvu5ktZTjzkwF"
 }
```

リクエストとそれに対応するレスポンスは次のようになります。

「OAuth 2.0
Authentication」プラグインには、アクセストークンを表示して無効にするという興味深い機能があります。

トークンを一覧表示するには、次に示す IAM の管理 API のエンドポイントに GET リクエストを送信します。

https://iamhost:8444/**{workspacename}/oauth2tokens**

上記の {workspacename} は使用される IAM ワークスペースの名前です。 RBAC
を有効している場合に備え、IAM の管理 API を呼び出すために必要な資格情報を入力してください。

「credentialid」は ClientApp コンシューマー内に作成した OAuth アプリケーションの ID（この場合は SampleApp）で、「serviceid」はこのプラグインが適用される「SampleIRISService」の ID
であることに注意してください。

トークンを無効にするには、次のエンドポイントに DELETE リクエストを送信します。

https://iamhost:8444/Sample/oauth2tokens/**{tokenid}**

上記の {tokenid} は無効にされるトークンの ID です。

無効化されたトークンを使おうとした場合、この無効なトークンをベアラートークンとして含む GET リクエストを 次の URL
に送信すると、トークンが無効であるか期限切れであるというメッセージが表示されます。

https://iamhost:8443/event/1

最終的な考慮事項

この記事では、IRIS にデプロイされている認証されていないサービスに対し、IAM で OAuth 2.0 認証を追加する方法を示しました。 サービスそのものは、IRIS で認証されないままとなることに注意してください。 したがって、IRIS サービスのエンドポイントを IAM レイヤーを介さずに直接呼び出すと、情報は認証なしで表示されます。
そのため、ネットワークレベルでセキュリティルールを設定し、不要なリクエストが IAM レイヤーをバイパスしないようにすることが重要です。

IAM の詳細についてはこちらをご覧ください。

IAM をお試しになりたい方は、InterSystems 営業担当者にお問い合わせください。

#API #OAuth2 #REST API #セキュリティ #InterSystems IRIS

ソースURL:https://jp.community.intersystems.com/post/intersystems-api-management-%E3%81%A7-oauth-20-%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%A6-api-%E3%82%92%E4%BF%9D%E8%AD%B7%E3%81%99%E3%82%8B-%E3%83%91%E3%83%BC%E3%83%88-3