文章
Hao Ma · 一月 15, 2021 阅读大约需 分钟

IAM实践指南——OAuth 2.0的API保护(第三部分)

在这个由三个部分组成的系列文章中，介绍了如何在OAuth
2.0标准使用IAM简单地为IRIS中的未经验证的服务添加安全性

第一部分 介绍了一些OAuth
2.0背景知识，以及IRIS和IAM的一些初始定义和配置，以帮助读者理解确保服务安全的整个过程。

第二部分
详细讨论和演示了配置IAM所需的步骤——验证传入请求中的访问令牌，并在验证成功时将请求转发到后端。

本系列的最后一部分将讨论和演示IAM生成访问令牌（充当授权服务器）并对其进行验证时所需的配置，以及一些重要的最终考虑事项。

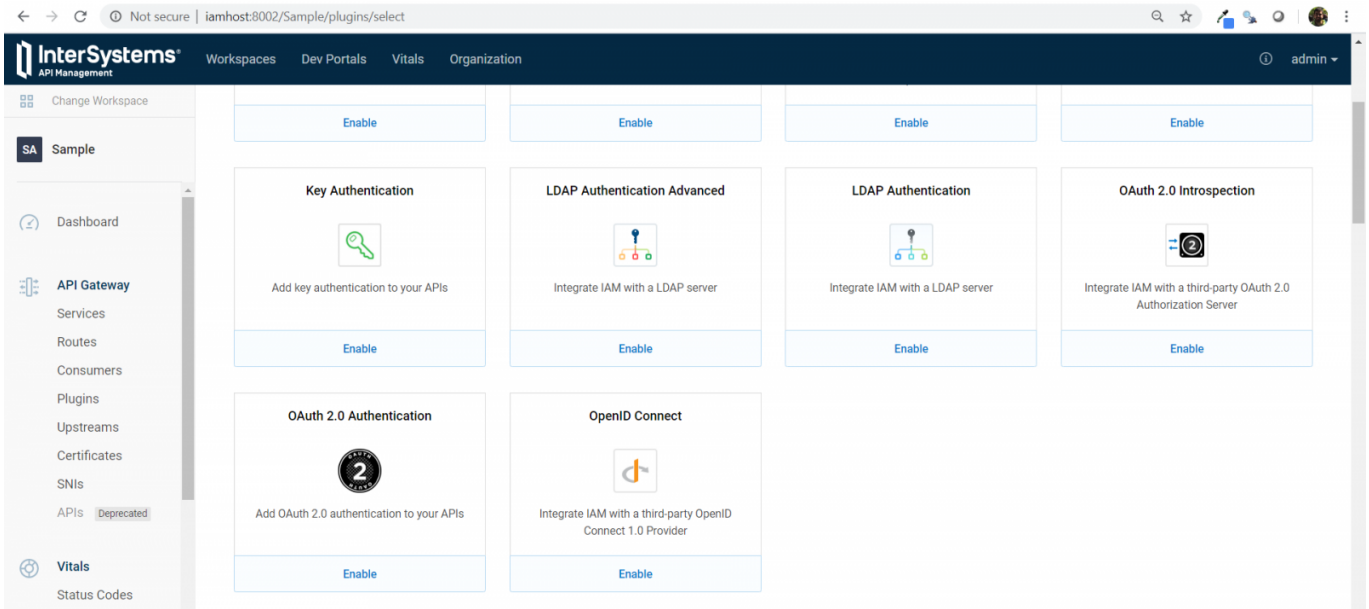
如果您想试用IAM，请联系InterSystems销售代表。

场景2：IAM作为授权服务器和访问令牌验证器

与上个场景不同的是，该场景中将使用一个名为“OAuth 2.0 Authentication”的插件。

如果要在资源所有者密码凭证流中将IAM作为授权服务器使用，客户端应用程序必须对用户名和密码进行身份验证。
只有在身份验证成功时，才能发出获取IAM访问令牌的请求。

首先，将其添加到“SampleIRISService”中。正如下面截屏所示，需要填充一些不同的字段来配置此插件。



首先，将“SampleIRISService”的ID粘贴到“service_id”字段中，这样就可以在服务中启用该插件。

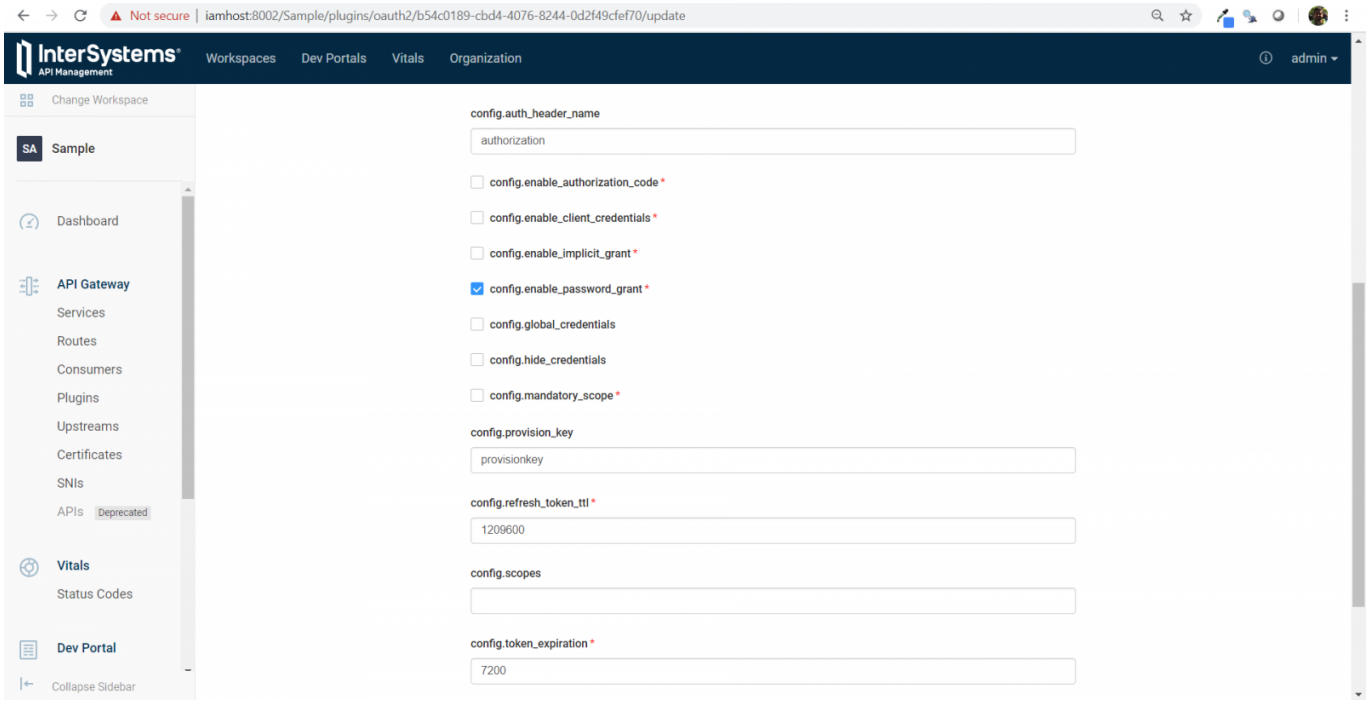
在“config.auth_header_name”字段中，需要指定携带授权令牌的头名称。本例中，我们保留默认值“authorization”。

“OAuth 2.0 Authentication”插件支持的OAuth 2.0流包括授权码授权(Authorization Code Grant)、客户端凭证(Client Credentials)、隐式授权(Implicit Grant)或资源所有者密码凭证授权(Resource Owner Password Credentials Grant)。我们在本文中使用的是“资源所有者密码凭证”流，故选中“config.enable_password_grant”。

在“config.provision_key”字段中输入要用作配置密钥的字符串。此值用来向IAM请求访问令牌。

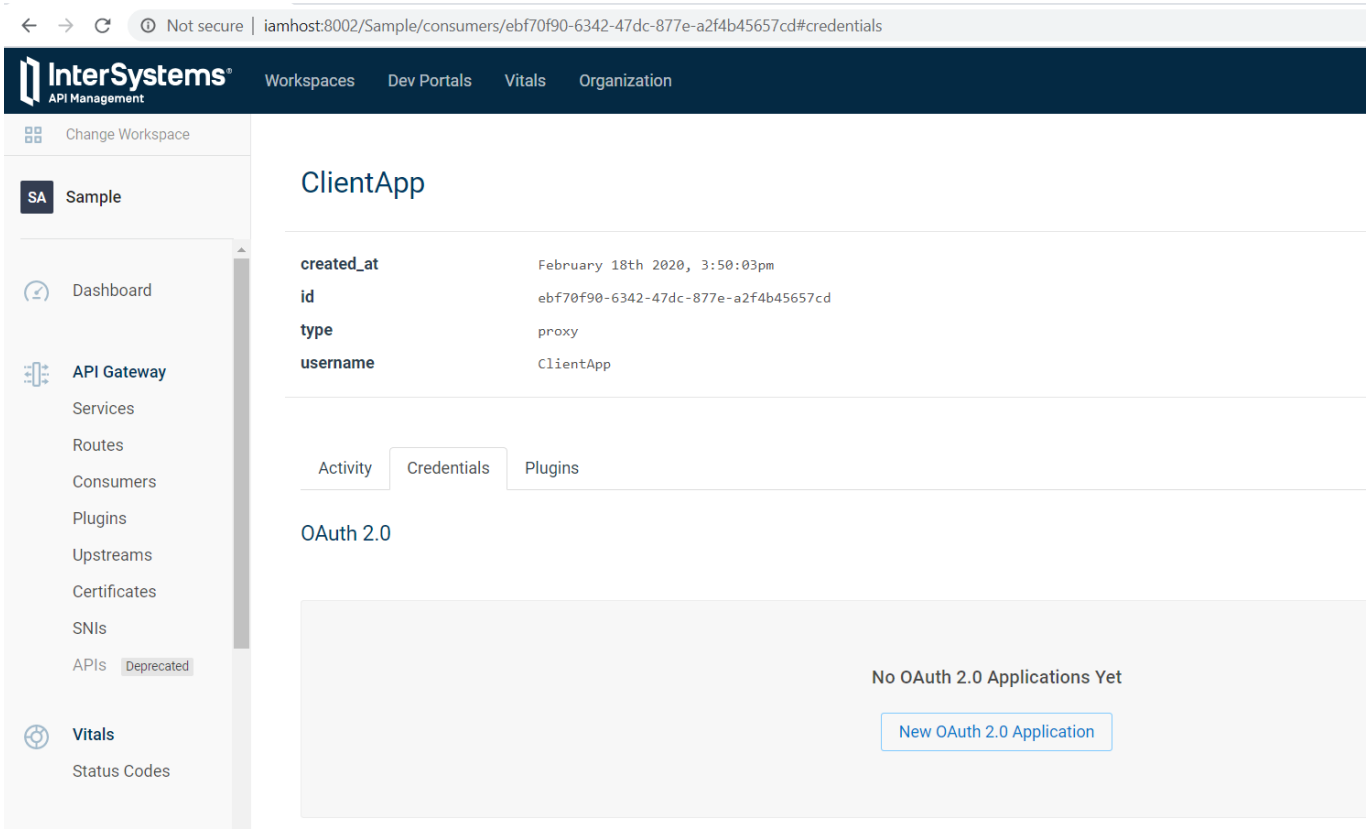
本例中，我们保留了所有其他字段的默认值。可以在 此处 查看插件文档中每个字段的完整引用。

下面是插件配置的最终效果：

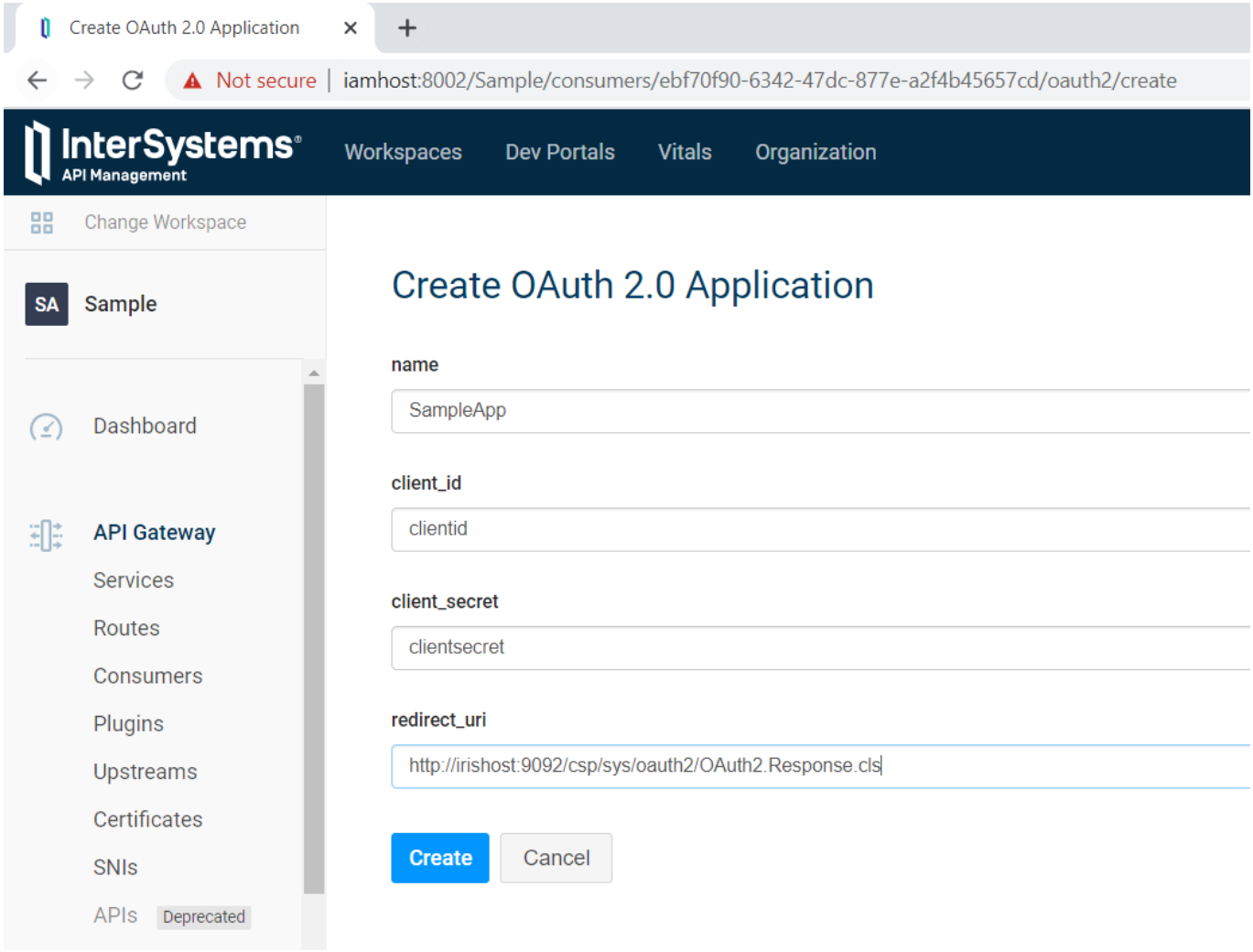


创建插件后，需为“ClientApp”客户端创建凭证。

为此，打开左侧菜单上的“Consumers”，然后单击“ClientApp”。接下来，点击“Credentials”标签，然后点击“New OAuth 2.0 Application”按钮。



在此个页面的“name”字段输入名称，以标识应用程序，在“client_id”和“client_secret”字段分别定义客户端ID和客户端密钥，最后在应用程序中输入URL，在“redirect_uri”字段上授权后，用户将被发送到该URL。然后，单击“Create”。



现在，可以发送请求了。

要发出第一个请求是获取IAM访问令牌。“OAuth 2.0 Authentication”插件自动创建一个端点，并将“/oauth2/token”路径附加到已经创建的路由上。
注意：必须使用HTTPS协议和IAM的代理端口（默认端口为8443）监听TLS/SSL请求。这是OAuth 2.0规范要求。

因此在本例中，要向URL发出一个POST请求：

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

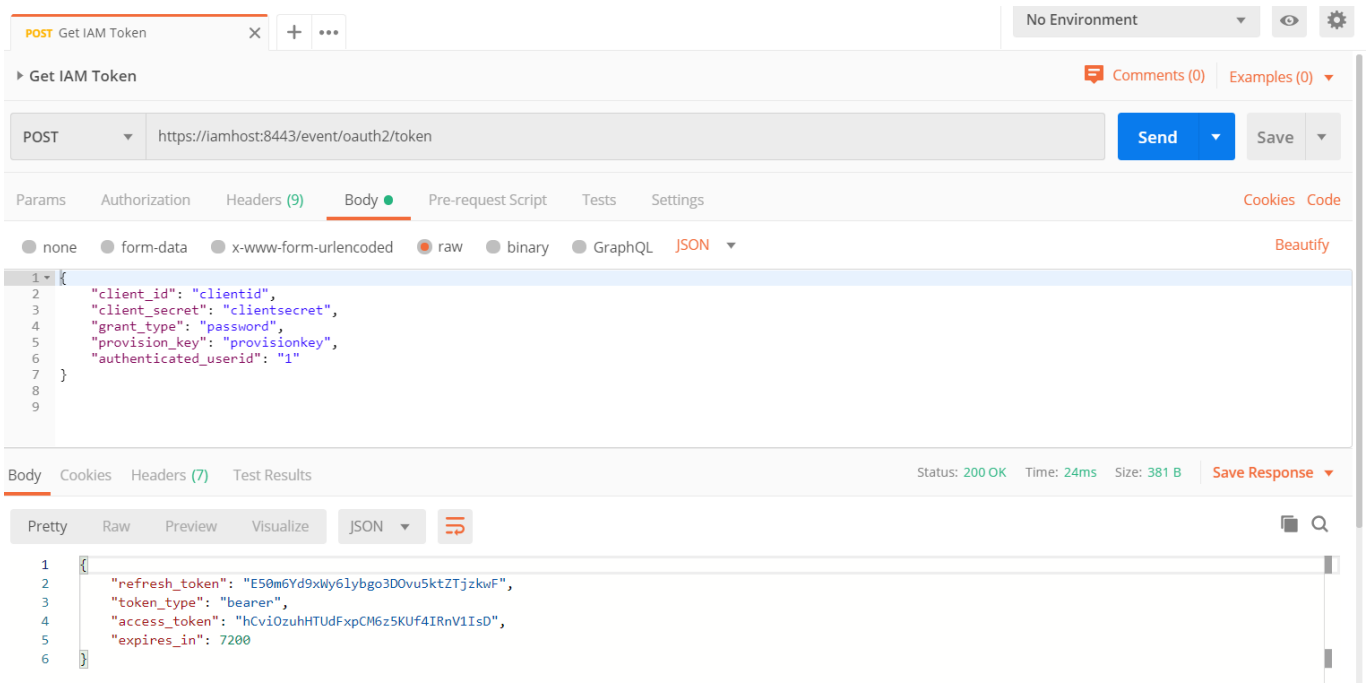
请求主体应包括以JSON：

```
{
  "client_id": "clientid",
  "client_secret": "clientsecret",
  "grant_type": "password",
  "provision_key": "provisionkey",
  "authenticated_userid": "1"
}
```

如上所示，该JSON包含了在创建“OAuth 2.0 Authentication”插件时定义的值（如“grant_type”和“provision_key”），以及在创建客户端凭证时定义的值（如“client_id”和“client_secret”）。

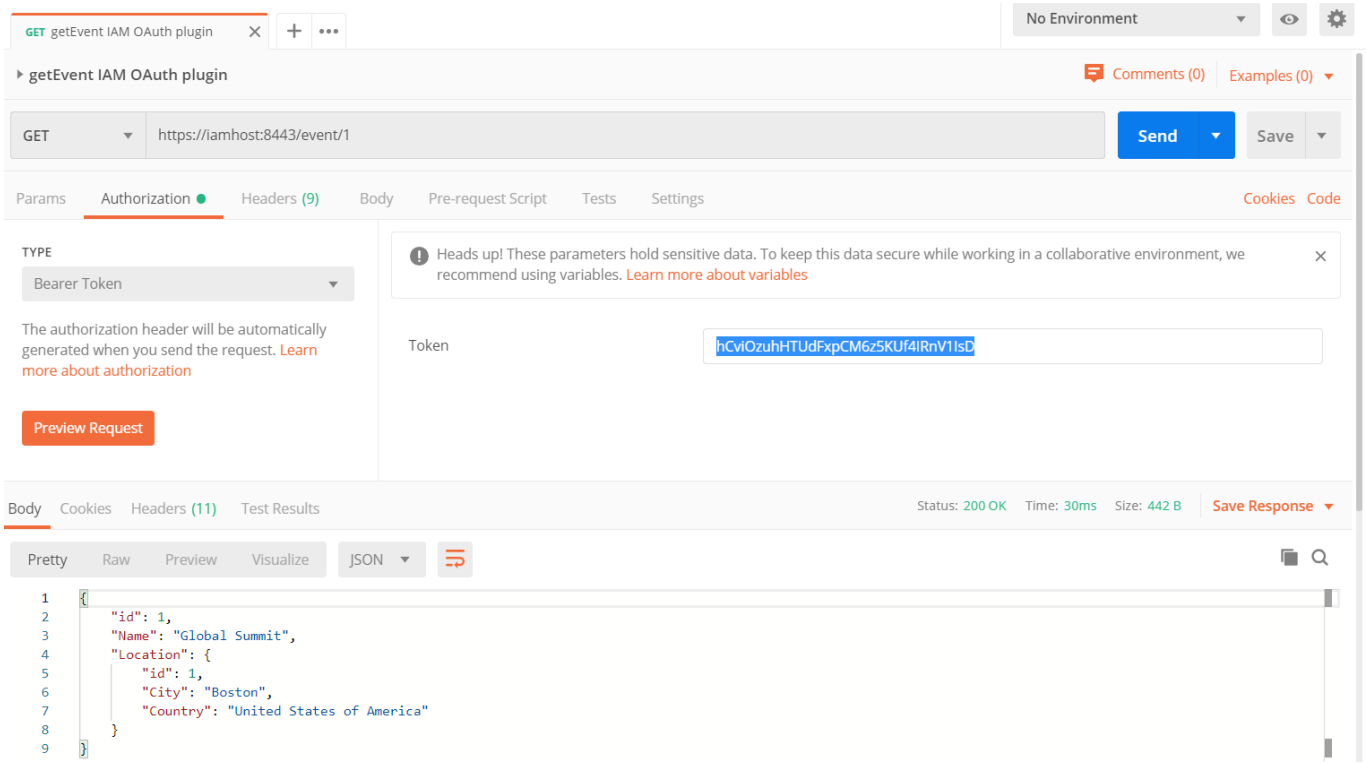
当提供的用户名和密码成功通过身份验证后，客户端应用程序还应该添加“authenticated_userid”参数值作为已通过身份验证的用户的唯一标志。

该请求及其响应如下：



现在可以请求从上面的响应中获取事件数据(包括"access_token"值)，并作为对URL的GET请求中的"bearner token"

https://iamhost:8443/event/1

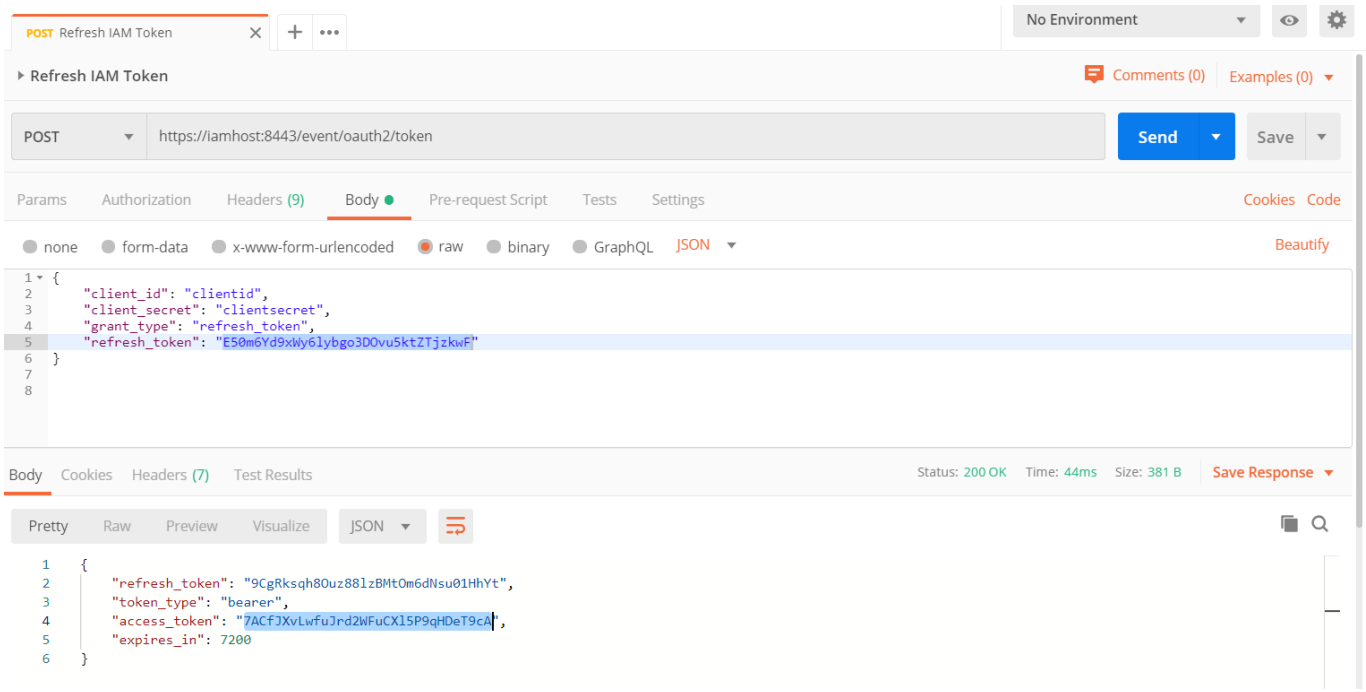


如果访问令牌过期，可以使用收到的刷新令牌和过期的访问令牌一起生成一个新的访问令牌，方法是向用于获取访问令牌的相同端点发出POST请求，但主体有不同：

```
{
  "client_id": "clientid",
  "client_secret": "clientsecret",
  "grant_type": "refresh_token",
```

```
  "refresh_token": "E50m6Yd9xWy6lybgo3DOvu5ktZTjzkwF"
}
```

该请求及其响应如下：

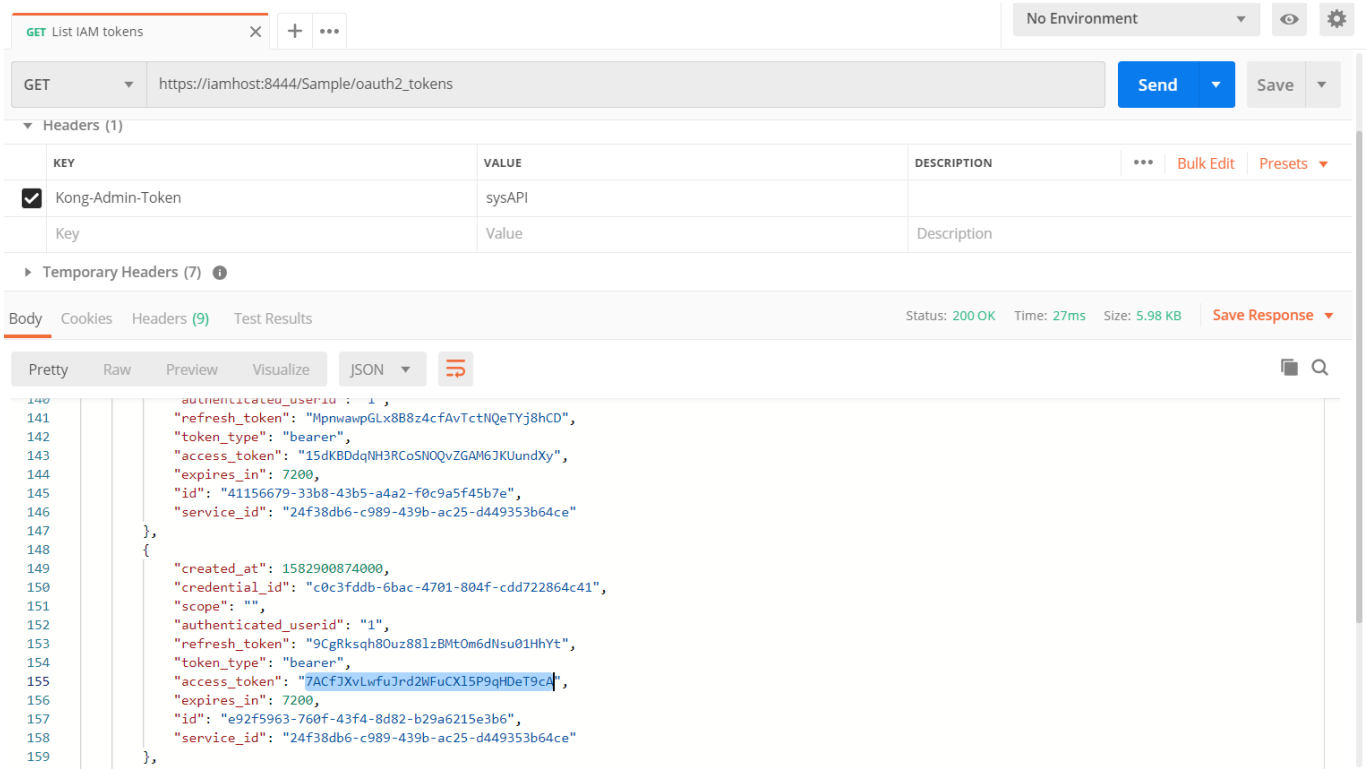


“OAuth 2.0 Authentication”插件的一个有趣好的特性能够查询和禁用访问令牌。

若要查询令牌列表，向以下IAM的Admin API终端发送GET请求即可：

https://iamhost:8444/**{workspace_name}/oauth2_tokens**

其中{workspace_name}是IAM工作区的名称。如果启用了RBAC，则需输入必要的凭证才能调用IAM Admin API。

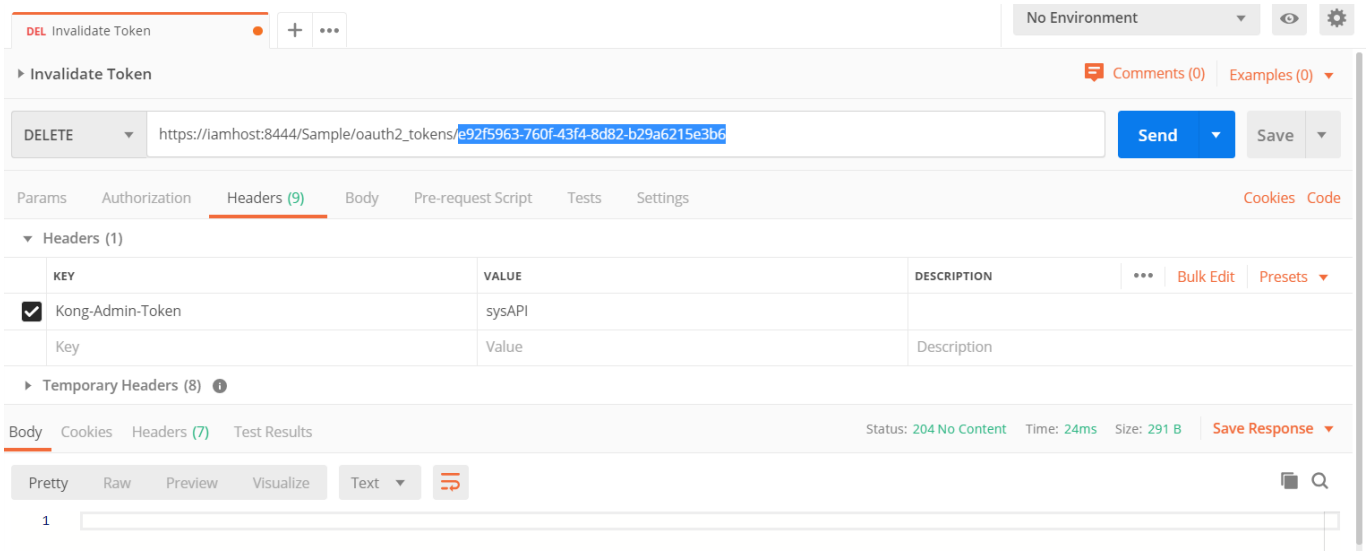


注意，“credential_id”是在ClientApp客户端内部创建的OAuth应用程序（本例中名为SampleApp）的id，“service_id”是应用此插件的“SampleIRISService”的id。

要想禁用令牌，可以向以下端点发送删除请求

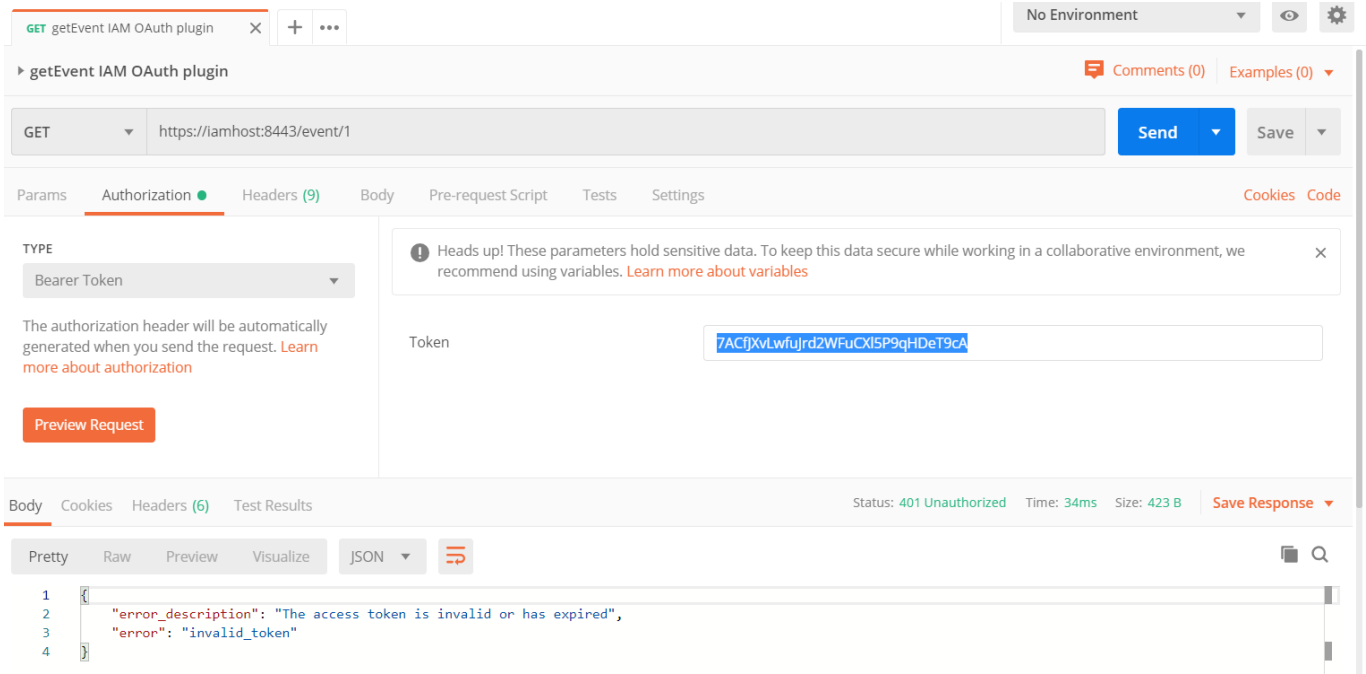
https://iamhost:8444/Sample/oauth2_tokens/**{token_id}**

其中{token_id}是要禁用的令牌id。



如果尝试使用无效令牌，将包含该无效令牌的GET请求作为Bearer Token 发送到URL，则会收到一条消息，提示令牌无效或已过期：

https://iamhost:8443/event/1



最后要考虑的因素

本文演示了如何将IAM中的OAuth 2.0身份验证添加到IRIS中未经身份验证的服务中。需牢记的是，IRIS中的服务本身仍是未经身份验证的。因此，如果有人绕过IAM层直接调用IRIS服务端点，则可以在不经过任何身份验证的情况

查看信息。出于该原因，在网络级别设置安全规则以防止不必要的请求绕过IAM层是很重要的。

可以在 此处了解更多有关IAM的信息。

如果您想试用IAM，请联系InterSystems销售代表。

#API #OAuth2 #REST API #安全 #InterSystems IRIS

源 URL: https://cn.community.intersystems.com/post/iam%E5%AE%9E%E8%B7%B5%E6%8C%87%E5%8D%97%E2%80%94%E2%80%94oauth-20%E4%B8%8B%E7%9A%84api%E4%BF%9D%E5%8D%AB%E6%88%98%EF%BC%88%E7%AC%AC%E4%B8%89%E9%83%A8%E5%88%86%EF%BC%89