关于UAA仪表板

UAA仪表板是predix.io提供的一个在线图形用户界面，用于配置和管理您的UAA实例。您可以使用UAA仪表板完成的任务包括：

* 查看与您的UAA实例相关的客户端、用户和身份提供者。
* 查看与您的服务实例相关的客户端。
* 创建和管理客户端。
* 创建和管理用户和用户组。
* 添加和管理用于SAML联合的身份提供者。
* 添加和管理密码策略。

查看、创建和管理客户端

您可以为认证具有相同UAA服务器实例的不同应用程序创建其他客户端。您可以根据您的应用程序的要求为每个客户端分配不同的访问许可。使用UAA仪表板中的客户端管理选项卡[查看、创建和管理您的OAuth客户端](https://www.predix.io/docs/?r=198842#uAyBrT9y-png4X1SX)。

查看、创建和管理用户和用户组

授权代码、隐式和资源所有者密码等授权类型的认证流程涉及到为应用程序提供其用户名和密码凭据的用户。您既可以在UAA本地创建用户也可以与外部身份提供者联合。使用UAA仪表板中的用户管理选项卡[在UAA本地创建用户](https://www.predix.io/docs/?r=198841#RulzoBew)并为您的应用程序管理这些用户。另外，您可以[创建用户组](https://www.predix.io/docs/?r=198840#ZHglCMC7)表示一名用户潜在拥有的权限。

添加和管理身份提供者

使用UAA仪表板中的身份提供者选项卡在UAA中管理您的身份提供者和服务提供者。关于与身份提供者联合的更多信息，请参见[使用UAA进行SAML联合](https://www.predix.io/docs/?r=198839#LigRvar4)。

添加和管理密码策略

使用UAA仪表板中的密码策略选项卡[通过密码策略在UAA中配置用户账号](https://www.predix.io/docs/?r=198838#SSqTSghl)，例如长度、接受或需要的字符类型、失效时间以及重置策略。

创建一个OAuth2客户端

创建其他OAuth2客户端以访问一个UAA实例的程序。

* 登录predix.io。

OAuth 2.0协议将一个客户端建立为一个应用程序，可以代表资源所有者并通过其授权执行保护资源请求。当您创建一个UAA实例时，会自动为您创建一个管理员客户端，能够配置您的UAA实例。管理员客户端的密码是您在创建UAA实例时指定的密码。您可以为认证具有相同UAA服务器实例的不同应用程序创建其他客户端。您可以根据您的应用程序的要求为每个客户端分配不同的访问许可。

1. 在控制台视图中，选择您的服务所处的空间。
2. 在服务实例页面中，选择您需要配置的UAA实例。
3. 选择配置服务实例选项。
4. 在UAA仪表板登录页面中，指定您的管理员客户端密码并点击登录。
5. 在UAA仪表板中，选择客户端管理选项卡。

客户端管理选项卡包括两部分，客户端和服务实例。服务实例部分显示您已经为您的服务创建的服务实例。

注：此处显示的服务实例是那些您使用您想要配置的UAA创建的实例。您使用其他UAA实例创建的服务实例不显示在此页面上。

1. 点击创建客户端按钮打开创建客户端窗体。
2. 在创建客户端窗体中指定以下值：

| **字段** | **描述** |
| --- | --- |
| 客户端ID | 为您正在创建的OAuth2客户端指定一个名称。 |
| 客户端密码 | 指定密码。您要记住该密码，这一点非常重要。如果丢失，该密码无法找回。 |
| 确认客户端密码 | 重新输入客户端密码。 |
| 授权的授权类型 | 从下面选择一个或多个授权类型：* + client\_credentials

当您使用客户端凭据授权类型时，UAA中的OAuth2端点接受客户端ID和客户端密码并提供访问令牌。* + implicit

当您使用隐式授权类型时，UAA直接为客户端发布访问令牌，无需认证该客户端。这减少了获取访问令牌所需往返行程的数量。 * + password

当您使用资源所有者密码凭据授权类型时，UAA中的OAuth2端点接受用户名和密码并提供访问令牌。* + authorization\_code

当您使用授权代码授权类型时，客户端将资源所有者定向至UAA，然后UAA反过来通过授权代码将资源所有者定向至客户端。 当您使用这种授权类型时，也会默认分配一个 refresh\_token 。关于不同授权类型的更多信息，请参见[RFC 6749](https://tools.ietf.org/html/rfc6749)。 |
| 范围 | 范围是与一个OAuth客户端相关的许可，用于用户通过应用程序访问资源。用户许可用于授权类型 authorization\_code, password 与 implicit。范围决定是否允许应用程序代表用户进行访问。默认为管理员客户端分配所有需要的范围。对于新的客户端，管理员可以根据客户端要求选择要添加的范围。关于可以分配的范围的完整列表，请参见[UAA授权的范围](https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#scopes-authorized-by-the-uaa)。 |
| 权限 | 权限是指当一个应用程序或API使用其自有凭据访问一个资源时与OAuth客户端相关的许可，不涉及用户。许可用于授权类型 client\_credentials。默认为管理员客户端分配所有需要的权限。对于新的客户端，管理员可以根据客户端要求选择要添加的权限。权限列表与范围列表相同。关于完整的列表，请参见[UAA授权的范围](https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#scopes-authorized-by-the-uaa)。 |
| 自动批准的范围 | 为客户端指定可以自动批准的范围，无需资源所有者的明确批准。 |
| 重定向URI | 指定一个重定向URI在登录后对客户端进行重定向。例如， http://example-app.com/welcome。当您开始将UAA用作服务提供者时，该URI用于您的外部身份提供者。 |
| 访问令牌有效性 | 指定访问令牌失效时间，单位毫秒。  |
| 刷新令牌有效性 | 指定刷新令牌失效时间，单位毫秒。 |
| 允许的提供者 | 如果有的话，指定外部身份提供者的名称。如果您正在使用外部身份提供者，同时将UAA用作服务提供者，则该字段为必填。 |

使用UAAC创建一个OAuth2客户端

使用UAA命令行界面（UAAC）代替图形用户界面创建一个OAuth2客户端的可选程序。

您可以使用UAAC管理您的UAA实例。关于安装命令行界面的更多信息，请参见<https://github.com/cloudfoundry/cf-uaac>。

1. 将您的UAA实例指定为预定目标。

uaac target <uaa\_instance\_url>

<uaa\_instance\_url> 是您的可信发布者的URL，例如， https://11fa0273-9e2a-37e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io。您可以在将您的UAA实例与一个应用程序绑定之后从VCAP\_SERVICES环境变量中检索该URL。

1. 使用管理客户端登录。

uaac token client get admin

1. 在提示时指定管理客户端密码。
2. 使用以下命令创建OAuth2客户端：

uaac client add [client\_name]--authorities "uaa.resource"--scope "openid"--autoapprove "openid"--authorized\_grant\_types [authorization\_code|implicit|password|client\_credentials|refresh\_token]--redirect\_uri [redirect\_uri\_1, redirect\_uri\_2, ...]

关于UAA选项的更多信息，例如 scopes 与 authorized\_grant\_types，请参见<https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst>网页上的UAA文件。

更改管理客户端密码

1. 将您的UAA实例指定为预定目标。

uaac target <uaa\_instance\_url>

<uaa\_instance\_url> 是您的可信发布者的URL，例如， https://11fa0273-9e2a-37e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io。您可以在将您的UAA实例与一个应用程序绑定之后从VCAP\_SERVICES环境变量中检索该URL。

1. 使用管理客户端登录。

uaac token client get admin

1. 在提示时指定管理客户端密码。
2. 使用以下命令更改客户端密码。

uaac secret change

1. 在提示时输入管理客户端当前的密码。
2. 在提示时输入管理客户端的新密码。
3. 在提示时重新输入管理客户端的新密码。

注：

请不要使用 -s 选项指定管理客户端的密码。您的终端历史记录中保留了这些凭据的副本。

在一个UAA实例中创建用户组

* 登录predix.io。

在UAA中，一个用户组表示一名用户潜在拥有的权限。当一名用户重定向至UAA进行认证时，UAA在访问令牌中添加用户组信息。

1. 在控制台视图中，选择您的服务所处的空间。
2. 在服务实例页面中，选择您需要配置的UAA实例。
3. 选择配置服务实例选项。
4. 在UAA仪表板登录页面中，指定您的管理员客户端密码并点击登录。
5. 在UAA仪表板中，选择用户管理选项卡。

用户管理选项卡包括两部分，用户和用户组。用户组部分显示您在您的UAA实例中创建的用户组。

1. 点击添加用户组按钮打开新建用户组窗体。
2. 在新建用户组窗体中指定以下值：

| **字段** | **描述** |
| --- | --- |
| 显示名称 | 指定用户组的名称。 |
| 描述 | 指定用户组的描述。 |

在一个UAA实例中创建用户

在UAA本地创建用户用于认证的程序。

* 登录predix.io。

当您创建一个UAA实例时，会自动为您创建一个管理员客户端，能够配置您的UAA实例。默认为管理员客户端分配所需的所有权限和范围。

注：管理员客户端默认未分配能够更改用户密码的权限。如果您需要更新或更改用户密码，您必须为您的管理员客户端添加 uaa.admin 权限。更多信息，请参见[更新OAuth2客户端](https://www.predix.io/docs/?r=198832#cSznH26X-ao7RnIco)。

使用以下程序在UAA本地创建用户。

1. 在控制台视图中，选择您的服务所处的空间。
2. 在服务实例页面中，选择您需要配置的UAA实例。
3. 选择配置服务实例选项。
4. 在UAA仪表板登录页面中，指定您的管理员客户端密码并点击登录。
5. 在UAA仪表板中，选择用户管理选项卡。

用户管理选项卡包括两部分，用户和用户组。用户组部分显示您在您的UAA实例中创建的用户组。

1. 点击添加用户按钮打开新建用户窗体。
2. 在新建用户窗体中指定以下值：

| **字段** | **描述** |
| --- | --- |
| 用户姓名 | 指定用户姓名。 |
| 电子邮件 | 指定电子邮件地址。 |
| 密码 | 指定密码。 |
| 名 | 指定用户的名。 |
| 姓 | 指定用户的姓。 |
| 用户组 | 选择与用户相关的用户组 |

使用UAAC在一个UAA实例中创建用户

使用UAAC代替图形用户界面在一个UAA实例中创建用户的可选程序。

对于访问您的UAA实例的应用程序，您可以创建具有所需范围的其他客户端和用户。

您可以使用UAA命令行界面（UAAC）创建用户并管理其范围。关于安装命令行界面的更多信息，请参见<https://github.com/cloudfoundry/cf-uaac>。

1. 创建一个新用户。

uaac user add <my-user> --emails <my\_user>-user@ge.com --password <my\_password>

1. 在您的UAA实例中创建用户组。

关于可用用户组的更多信息，请参见<https://github.com/cloudfoundry/uaa/blob/master/docs/UAA-APIs.rst#scopes-authorized-by-the-uaa>。

例如：

uaac group add scim.read uaac group add scim.write

1. 将新用户添加至所需的用户组。

例如：

uaac member add zones.<my\_uaa\_instance>.admin <my-user> uaac member add scim.read <my-user> uaac member add scim.write <my-user> uaac member add clients.write <my-user> uaac member add clients.read <my-user> uaac member add clients.admin <my-user> uaac member add clients.secret <my-user>

1. 验证创建的用户具有正确的范围。

uaac token owner get <my-oauth-client> <my-user> uaac token decode

创建一个密码策略

由UAA管理的用户账号数据库可用于基于密码的认证。您可以通过密码策略在UAA中配置用户账号，例如长度、接受或需要的字符类型、失效时间以及重置策略。

当您创建一个UAA实例时，会自动通过默认的密码策略创建一个类型为 uaa 的内部身份提供者。您可以在您的UAA实例中为用户创建新的密码策略。

1. 在控制台视图中，选择您的服务所处的空间。
2. 在服务实例页面中，选择您需要配置的UAA实例。
3. 选择配置服务实例选项。
4. 在UAA仪表板登录页面中，指定您的管理员客户端密码并点击登录。
5. 在UAA仪表板中，选择密码策略选项卡。
6. 在密码策略窗体中指定以下值：

| **字段** | **描述** |
| --- | --- |
| 最小长度 | 指定一个有效密码需要的最少字符数。 |
| 最大长度 | 指定一个有效密码需要的最多字符数。 |
| 需要大写字符 | 指定一个有效密码需要的最少大写字符数。 |
| 需要小写字符 | 指定一个有效密码需要的最少小写字符数。 |
| 需要特殊字符 | 指定一个有效密码需要的最少特殊字符数。 |
| 需要数字 | 指定一个有效密码需要的最少数字数。 |
| 密码过期月数 | 指定当前密码过期的月数。 |
| 锁定期秒数 | 指定失败的尝试次数超过设定的限值时账号被锁定的时间（单位：秒）。 |
| 锁定前的失败次数 | 指定账号锁定之前允许的失败次数。 |
| 失败计数 | 指定为了锁定账号锁定失败必须发生的秒数。 |

使用UAA进行SAML联合

UAA支持安全断言标记语言（SAML）协议，用于与身份提供者（IdP）或服务提供者（SP）进行通信。

IdP管理可能需要安全访问应用程序或服务的用户的账号。SP是从用户处接收请求的服务器，用于访问服务或应用程序。在一个典型的SAML流程中，当一名用户从SP处请求一项服务时，SP首先从IdP处请求并获取一个身份断言。IdP从SP处接收请求并根据用户账号信息生成一个身份断言。然后，SP决定是否根据IdP提供的断言执行服务。

您可以将您的UAA实例配置为用作IdP或SP或两者。以下场景决定了您的UAA需要的配置类型：

* 如果您使用UAA SCIM API或UAA仪表板在UAA本地管理用户账号，您无需为SAML联合配置UAA。
* 如果您在公司SSO等外部IdP上远程设定您的用户账号，您可以将UAA配置为重定向至外部IdP的SP。更多信息，请参见[使用脚本将UAA配置为一个服务提供者](https://www.predix.io/docs/?r=198829#aJ7mgIqn)。
* 如果您拥有提供SP功能的应用程序（例如，GitHub企业版或ServiceNow），您可以将UAA配置为IdP。更多信息，请参见[将UAA配置为一个身份提供者](https://www.predix.io/docs/?r=198828#N1i57sg)。
* 可以将UAA配置为SP和IdP两者。但是，这种配置仅用于测试环境。如果要将UAA配置为SP和IdP，您可以完成将UAA配置为SP和IdP两者的步骤。

将UAA配置为服务提供者用于外部身份提供者

* 从您的IdP管理员处获取您的身份提供者（IdP）元数据。
* 登录Predix.io并转到控制台视图。

如果您在公司SSO等IdP上远程设定您的用户账号，您可以将UAA配置为重定向至外部IdP的服务提供者（SP）。

1. 在Predix.io上的服务实例页面中，选择您需要配置的UAA实例。
2. 选择配置服务实例选项。
3. 在UAA仪表板登录页面中，指定您的管理员客户端密码并点击登录。
4. 在UAA仪表板中，点击身份管理选项卡。

身份管理选项卡显示了一个您配置的IdP的列表，所选的UAA实例配置为SP。

1. 如果要添加一个新的IdP，可以点击添加身份提供者按钮。
2. 在身份提供者窗体中，指定以下信息并按下提交：

|  |  |
| --- | --- |
| **字段名称** | **描述** |
| 名称 | 指定您的IdP的名称。 |
| 描述 | 为您的IdP指定一个简短描述。 |
| 类型 | 选择IdP的类型。  |
| 有效 | 选择此选项将IdP设定为有效。如果您为一个UAA实例定义了多个IdP，UAA仅与有效的IdP进行交互。如果您的一个UAA实例有多个有效IdP，您必须确保通过相应信息对IdP的相关客户端进行更新。尽管可以更新一个客户端与多个IdP进行交互，但是作为一项最佳实践，您可以为与UAA交互的每个应用程序定义一个新客户端。 |
| 电子邮件 | 指定在您的IdP中定义的与在UAA中定义的 email 属性相对应的属性。 |
| 名 | 指定在您的IdP中定义的与在UAA中定义的 Given Name 属性相对应的属性。 |
| 姓 | 指定在您的IdP中定义的与在UAA中定义的 Family Name 属性相对应的属性。 |
| 元数据 | 指定IdP元数据并将UAA SP元数据下载至该字段。您必须从您的IdP管理员处获取您的IdP元数据。您必须将UAA SP元数据提供给您的IdP管理员。 |

1. IdP在外部身份提供者列表中显示。
2. 如果要为访问UAA的IdP设置客户端，可以点击您的IdP的姓名旁边的查看/编辑选项。
3. 为您的IdP指定一个客户端。

您既可以选择一个现有的客户端也可以创建一个新客户端。作为开发的最佳实践，您可以为每个应用程序添加一个新客户端。

如果要创建一个新客户端，可以点击创建新客户端按钮。默认情况下，创建客户端窗体填充用于授权代码授权类型。您也可以将其更新为使用隐式或密码授权类型。对于客户端凭据授权类型，您必须首先创建一个用户。

关于创建一个客户端的更多消息，请参见[创建一个OAuth2客户端](https://www.predix.io/docs/?r=198827#uAyBrT9y-png4X1SX)。

使用脚本将UAA配置为一个服务提供者

使用UAAC和配置脚本代替图形用户界面将UAA配置为服务提供者的可选程序。

* 从[GitHub](https://github.com/PredixDev/uaa-scripts.git)下载以下脚本。
	+ create-saml-idp.sh
	+ create-client-for-idp.sh

您可以将UAA配置为重定向至公司SSO等外部身份提供者（IdP）的服务提供者（SP）。

您可以使用UAA命令行界面（UAAC）管理您的UAA实例。关于安装命令行界面的更多信息，请参见<https://github.com/cloudfoundry/cf-uaac>。

1. 将您的UAA实例指定为预定目标。

uaac target <uaa\_instance\_url>

<uaa\_instance\_url> 是您的可信发布者的URL，例如， https://11fa0273-9e2a-37e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io。您可以在将您的UAA实例与一个应用程序绑定之后从VCAP\_SERVICES环境变量中检索该URL。

1. 使用管理客户端登录。

uaac token client get admin

1. 在提示时指定管理客户端密码。
2. 从UAA导出SAML服务提供者元数据。将该元数据提供给您的IdP管理员，以便将其导入IdP。

uaa-sp-metadata.xml 文件是您的UAA实例生成的服务提供者SAML元数据。可用于以下端点：

<uaa\_instance\_url>/saml/metadata/alias/<uaa\_instance\_guid>.cloudfoundry-saml-login

例如，

https://13fa0384-9e2a-48e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io/saml/metadata/alias/13fa0384-9e2a-48e2-9d06-2c95a1f4f5ea.cloudfoundry-saml-login

1. 您的IdP管理员将配置其IdP、导入SAML服务提供者元数据并生成身份提供者元数据。从您的IdP管理员处获取该身份提供者元数据。您需要将该元数据导入UAA。
2. 通过运行一个XML解析器验证SAML身份提供者元数据。确保包含一个有效的XML标头，例如：

<?xml version="1.0" encoding="UTF-8"?>

如果您的文件不包含所需的标头，将其添加到您的元数据文件的顶部。

1. 使用 create-saml-idp.sh 脚本为特定UAA实例创建SAML身份提供者。脚本使用特定的SSO元数据文件。

./create-saml-idp.sh -n <my-idp-name> -m <idp-metadata>.xml -a -c <mapping-config-file> -g <groups-config-file>

* + -n 为您的身份提供者指定一个名称。
	+ -m指定您的IdP管理员提供的SSO元数据。例如， idp-metadata.xml
	+ -a 指定应该在UAA中创建的一个影子用户。默认情况下不创建影子用户。当您首次设置身份提供者时需要使用 -a 选项。
	+ -c 是一个指定配置映射文件的可选参数。配置映射文件包含将SAML断言属性转换为JWT令牌的属性映射。

配置映射文件的示例如下：

{"email":"mail","given\_name":"first name","family\_name":"last name"}

* + -g 是一个指定用户组配置文件的可选参数。用户组配置文件包含需要从外部IdP映射的所有用户组。

用户组配置文件的示例如下：

["group1","group2","group3","group4","group5"]

例如，如果要使用GE SSO，您可以指定：

./create-saml-idp.sh -n gesso-staging-idp -m gesso-staging-idp-metadata.xml -a

注：如果需要，您可以在为您的IdP配置UAA之后更新IdP配置。更多信息，请参见[更新IdP配置](https://www.predix.io/docs/?r=198824)。

1. 验证您指定的身份提供者存在。

uaac curl /identity-providers

1. 使用 create-client-for-idp.sh 脚本创建一个客户端，将指定的身份提供者（IdP）作为客户端的允许提供者。每个OAuth客户端可以指定一个或多个负责认证用户的身份提供者。默认的身份提供者是 uaa，这就意味着使用UAA登录页面对用户进行认证。使用该脚本指定一个用于用户登录的不同身份提供者，例如一个SAML身份提供者。但是，不允许您指定多个身份提供者。

./create-client-for-idp.sh -c <client\_id> -s <client-secret> -p <my-sso-idp> -r <redirect\_uri>

* + -c 指定客户端Id。
	+ -s 指定客户端密码。
	+ -p 为您的身份提供者指定一个名称。
	+ -r 指定您的应用程序URI。

例如，

./create-client-for-idp.sh -c apm\_app -s changeme -p gesso-staging-idp -i  -r Predix-HelloWorld-WebApp-myapp.run.aws-usw02-pr.ice.predix.io

1. 验证您的客户端存在而且为您的IdP设置了 allowed providers 属性。

uaac client get <client\_id>

如果要测试您的设置，可以使用以下URL：

<uaa\_instance\_url>/oauth/authorize?client\_id=<my\_app>&response\_type=code&redirect\_uri=<redirect\_uri>

注：如果您正在使用授权代码进行授权，可以使用该URL进行测试。

* <client\_id> 是第6步设定的客户端的名称
* redirect\_uri 是应用程序URL。例如， https://security-predix-seed.grc-apps.svc.ice.ge.com。

请求被重定向至您的IdP登录页，您可以在该页面输入您为您的IdP设定的用户凭据。如果登录成功，您被重定向回到 redirect\_uri。

注：火狐浏览器可以使用 SAML Tracer 等浏览器插件工具验证SAML流程。

更新IdP配置

* 从[GitHub](https://github.com/PredixDev/uaa-scripts.git)下载以下脚本。
	+ update-saml-idp.sh

您可以更新UAA配置，以反映对您的IdP设置进行的更改。

1. 将您的UAA实例指定为预定目标。

uaac target <uaa\_instance\_url>

<uaa\_instance\_url> 是您的可信发布者的URL，例如， https://11fa0273-9e2a-37e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io。您可以在将您的UAA实例与一个应用程序绑定之后从VCAP\_SERVICES环境变量中检索该URL。

1. 使用管理客户端登录。

uaac token client get admin

1. 在提示时指定管理客户端密码。
2. 使用 update-saml-idp.sh 脚本为特定UAA实例更新SAML身份提供者。脚本使用特定的SSO元数据文件。

./update-saml-idp.sh -n <my-idp-name> -m <idp-metadata>.xml -d <idp-id> -a -c <mapping-config-file> -g <groups-config-file>

* + -n 指定您在创建IdP时所指定的身份提供者的名称，它符合 /identity-providers 载荷的名称属性。
	+ -d 指定IdP的Id。IdP Id是由UAA自动生成的id，符合 id 属性，用于 /identity-providers 载荷。
	+ 其他参数的描述，请参见[使用脚本将UAA配置为一个服务提供者](https://www.predix.io/docs/?r=198822#aJ7mgIqn)。

将UAA配置为一个身份提供者

* 从[GitHub](https://github.com/PredixDev/uaa-scripts.git)下载以下脚本。
	+ create-saml-sp.sh

您可以将UAA配置为SAML身份提供者，以便与其他服务提供者集成。

1. 将您的UAA实例指定为预定目标。

uaac target <uaa\_instance\_url>

<uaa\_instance\_url> 是您的可信发布者的URL，例如， https://11fa0273-9e2a-37e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io。您可以在将您的UAA实例与一个应用程序绑定之后从VCAP\_SERVICES环境变量中检索该URL。

1. 使用管理客户端登录。

uaac token client get admin

1. 在提示时指定管理客户端密码。
2. 从UAA导出身份提供者（IdP）元数据。将该元数据提供给您的服务提供者（SP）管理员，以便将其导入SP应用程序。

uaa\_idp\_metadata.xml 文件是您的UAA实例生成的IdP元数据。可用于以下端点：

<uaa\_instance\_url>/saml/idp/metadata

例如，

https://13fa0384-9e2a-48e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io/saml/idp/metadata

1. 将UAA IdP元数据导入您的SP应用程序。

这一步取决于您使用的服务提供者的类型。更多信息，请参见您的服务提供者的文件。

1. 为UAA IdP设定一个用户。

uaac target <UAA\_IDP\_INSTANCE\_URL> uaac token client get admin client secret: <admin client secret> uaac user add <user-name> --given\_name <first-name> --family\_name <last-name> --emails <email> -p <password> uaac group add zones.uaa.admin uaac member add zones.uaa.admin <user-name>

1. 为UAA IdP添加服务提供者配置。
	1. 获取一个访问令牌。您获取访问令牌所用的客户端必须拥有 sps.write 范围。

您被重定向至UAA IdP登录页面。输入在第3步中为用户设定的凭据。

关于获取访问令牌的更多信息，请参见[使用UAA获取OAuth2访问令牌](https://www.predix.io/docs/?r=198820#inJm68Sl)。

* 1. 使用 create-saml-sp.sh 脚本添加SP配置。

./create-saml-sp.sh -n <uaa-sp-name> -m uaa-sp-metadata.xml -i

1. 检查SP配置是否成功添加。

uaac curl /saml/service-providers

使用UAA获取OAuth2访问令牌

UAA提供一个OAuth2定义的端点用于获取访问令牌。关于令牌端点的详细信息，请参见[UAA文件](https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#oauth2-token-endpoint-post-oauth-token)。

您可以为以下授权类型获取访问令牌：

* 授权代码
* 客户端凭据
* 隐式
* 资源所有者密码凭据
* JSON网络令牌（JWT）不记名令牌

关于授权类型的详细信息，请参见[Oauth2 rfc-6749](http://www.rfc-base.org/)。

使用授权代码授权获取令牌

当您使用授权代码授权类型时，客户端将资源所有者定向至UAA，然后UAA反过来通过授权代码将资源所有者定向至客户端。UAA中的OAuth2端点接受授权代码以提供一个访问令牌。

当您使用授权代码授权类型时，在服务器侧对令牌进行存储和管理。因此，令牌中的任何技术信息（例如范围名称）不会泄露给用户，令牌不易于被误用。另外，通过授权代码授权类型，终止用户会话更简单，无需等待令牌的使用期限到期。授权代码的安全性依赖于稳健的会话管理方案，包括使用HTTP安全标头、HTTP Cookie安全标志、充足的会话使用期限和安全会话终止（登出机制）进行HTTP协议强化等最佳实践。

关于授权代码授权类型的详细信息，请参见[RFC 6749](https://tools.ietf.org/html/rfc6749#page-24)。

下图显示了使用授权获取访问令牌的高级流程。



1. 客户端请求访问一个资源。请求包括客户端标识符、请求的范围、本机状态和一个重定向URI。
2. UAA（通过用户代理）认证资源所有者并确定资源所有者授权还是拒绝客户端的访问请求。
3. 如果资源所有者授权访问，UAA使用上一步提供的重定向URI将授权代码发送回客户端。
4. 客户端通过包括在上一步中接收的授权代码从UAA令牌端点请求一个访问令牌。
5. UAA认证客户端、验证授权代码并确保接收的重定向URI与重定向客户端所用的URI匹配。如果客户端被认证，UAA将发送回一个访问令牌。

关于生成授权代码和访问令牌所用的UAA端点的详细信息，请参见<https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#authorization-code-grant>。

使用隐式授权获取令牌

当您使用隐式授权类型时，UAA直接为客户端发布访问令牌，无需认证该客户端。这减少了获取访问令牌所需往返行程的数量。但是，通过这种授权类型，访问令牌在URI片段中传输，可能会将其暴露给未授权方。另外，如果公共客户端使用隐式流程，UAA无法确定将一个访问令牌发布给哪个客户端。

使用隐式授权的优点是可以改进某些客户端的响应性和效率（例如，作为浏览器的内部应用程序执行的客户端）。但是，您必须将该优点与使用隐式授权的安全隐患进行权衡。

关于隐式授权类型的详细信息，请参见[RFC 6749](https://tools.ietf.org/html/rfc6749#page-24)。

下图显示了使用隐式授权类型获取访问令牌的概要流程。



1. 客户端将一个请求发送至UAA中的授权端点。请求包括客户端标识符、请求的范围、本机状态和一个重定向URI。UAA需要使用重定向URI发送访问令牌。
2. UAA认证资源所有者并确定资源所有者授权还是拒绝客户端的访问请求。
3. 如果资源所有者授权访问，UAA使用在第1步中指定的重定向URI将用户代理重定向回客户端。重定向URI包括URI碎片中的访问令牌。
4. 用户代理向网络托管客户端资源发出一个请求。用户代理不将URI碎片信息传递给客户端资源。
5. 网络托管客户端资源返回一个网页，该网页可以访问包括用户代理保留的URI碎片的重定向URI，而且可以提取URI碎片中包含的访问令牌。
6. 用户代理提取访问令牌。通常提取的内容包括执行网络托管客户端资源本地提供的脚本。
7. 用户代理将访问令牌传递给客户端。

关于通过隐式授权类型生成访问令牌所用的UAA端点的详细信息，请参见<https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#implicit-grant-with-credentials-post-oauth-authorize>。

使用资源所有者密码凭据授权获取令牌

当您使用资源所有者密码凭据授权类型时，UAA中的OAuth2端点接受用户名和密码并提供访问令牌。

您只能在资源所有者和客户端之间具有较高的信任度（例如，客户端是设备操作系统的一部分或者是一个具有较高权限的应用程序），而且授权代码等其他授权类型不可用时使用这种授权类型。

当您使用这种授权类型时，客户端无需存储资源所有者凭据以供将来使用。客户端简单地与长期访问令牌或刷新令牌交换凭据。

关于资源所有者密码凭据授权类型的详细信息，请参见[RFC 6749](https://tools.ietf.org/html/rfc6749#page-37)。

下图显示了使用资源所有者密码凭据授权获取访问令牌的概要流程。



1. 资源所有者向客户端提供其用户名和密码。
2. 客户端将凭据包含在从UAA令牌端点获取访问令牌的请求当中。
3. UAA认证客户端并验证资源所有者凭据，如果有效，发布一个访问令牌。

关于生成访问令牌所用的UAA端点的详细信息，请参见<https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#password-grant-with-client-and-user-credentials-post-oauth-token>。

使用客户端凭据授权获取令牌

当您使用客户端凭据授权类型时，UAA中的OAuth2端点接受客户端ID和客户端密码并提供访问令牌。

当授权范围限于在客户端控制下受保护的资源，或者之前通过UAA授权的受保护资源时，可以使用客户端凭据。客户端凭据授权通常用于客户端自己执行（客户端也是资源所有者）或者根据之前的UAA授权请求访问受保护的资源时。

关于授权代码授权类型的详细信息，请参见[RFC 6749](https://tools.ietf.org/html/rfc6749#page-40)。

下图显示了使用客户端凭据授权获取访问令牌的高级流程。



1. 客户端从令牌端点请求一个访问令牌。
2. UAA认证客户端，如果有效，发布一个访问令牌。

关于通过客户端凭据授权生成访问令牌所用的UAA端点的详细信息，请参见<https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#trusted-authentication-from-login-server>。

使用JWT不记名令牌授权获取令牌

UAA支持[RFC 7523](https://tools.ietf.org/html/rfc7523)规范，该规范将JSON网络令牌（JWT）不记名令牌的使用定义为请求一个OAuth 2.0访问令牌以及用于客户端认证的方法。JWT不记名令牌流程支持RSA SHA256算法，该算法将一个上传的证书用作登录密码。UAA使用一个适用于JWT的数字签名认证授权的设备。

JWT不记名令牌授权可用于云上运行的设备与服务（以及应用程序）之间的安全通信，所述设备和服务使用OAuth2进行保护。设备是基于或不基于网络的设备、客户端、机器或第三方应用程序使用的一个更广泛的术语。

简要的JWT不记名令牌流程如下：

1. 应用程序开发人员获取一个X509证书并通过其所在组织的证书管理系统对证书进行注册。该证书符合其应用程序的私有密钥。
2. 应用程序开发人员创建一个符合RFC 7523中所述字段的JWT令牌，并通过证书的私有密钥对其进行登记。
3. 客户端请求UAA使用RFC 7523规范中所述的JWT不记名授权，以获取一个访问令牌。
4. UAA根据RFC 7523中的描述验证设备的JWT令牌，并使用分配给客户端的公共密钥验证令牌签名。
5. 如果JWT有效，UAA发布一个访问令牌。

注：在该流程中，不发布刷新令牌。另外，不需要将客户端密码传递至check\_token端点。

使用JWT不记名令牌流程

如果要使用JWT令牌授权流程，客户端/设备创建一个JWT令牌并通过以下参数向UAA令牌端点发出一个请求：

* 端点： /oauth/token
* HTTP请求参数
	+ grant\_type=urn:ietf:params:oauth:grant-type:jwt-bearer
	+ assertion==<jwt-token>

例如，

uaac curl /oauth/token -X POST \-H "content-type:application/x-www-form-urlencoded; charset=utf-8" -H "accept:application/json;charset=utf-8" \ --data 'grant\_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer&assertion=<asdfjhasdfhasdjfkh ---truncatedfor brevity>'

使用UAA获取ID令牌

OpenID Connect 1.0是OAuth 2.0协议之上的一个身份层。它允许客户端根据授权服务器执行的认证验证最终用户的身份，并获取关于最终用户的基本简介信息。

如果要使用OpenID Connect执行，客户端需要包括授权请求中的openid范围值。OAuth 2.0授权服务器返回与被称为ID令牌的JSON网络令牌（JWT）中执行的认证相关的信息。ID令牌包含关于授权服务器执行的最终用户认证的声明，而且可以选择包含其他请求的声明。关于ID令牌的更多信息，请参见[OpenID Connect Core 1.0规范](http://openid.net/specs/openid-connect-core-1_0.html#IDToken)。

支持OpenID Connect的认证服务器也称为OpenID提供者（OP）。使用OpenID Connect的OAuth 2.0客户端也称为依赖方（RP）。

您可以为以下授权类型获取ID令牌：

* 授权代码（response\_type=code）
* 隐式（response\_type=id\_token token 或 response\_type=id\_token）
* 混合（使用[OAuth 2.0多响应类型编码实践](http://openid.net/specs/openid-connect-core-1_0.html#OAuth.Responses) [OAuth.Responses]中定义的其他响应类型值）

获取ID令牌的基本流程与获取访问令牌的相同。下图显示了基本的ID令牌流程。



1. 客户端（依赖方）向UAA（OpenID提供者）发送一个请求。
2. UAA认证最终用户并获取授权。
3. UAA返回一个ID令牌和一个访问令牌。
4. 客户端通过访问令牌向UAA中的用户信息端点发送一个请求。
5. UAA返回关于最终用户的声明。

使用UAA进行令牌验证

UAA使用基于令牌的认证，当客户端应用程序代表Cloud Foundry用户帐号时，UAA发布令牌供其使用。基于令牌的认证方法应用于访问只能通过安全的令牌做验证的应用程序。‘

生成一个访问令牌之后，访问您的应用程序的客户端向您的应用程序呈上不记名令牌进行认证。您的应用程序使用UAA验证该令牌。令牌可以通过以下其中一种方式进行验证：

* 使用UAA远程验证令牌

UAA提供一个端点（/check\_token）验证来自资源服务器的访问令牌。关于端点的详细信息，请参见[UAA文件](https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#oauth2-token-validation-service-post-check-token)。

* 本地验证令牌

Predix平台提供一个基于Java的FastTokenService库在本地验证令牌。FastTokenService执行以下任务：

* + 下载并缓存UAA身份区公共密钥并使用该密钥验证令牌签名。
	+ 验证令牌由一个有效的发布者发布。
	+ 验证令牌未过期。
	+ 验证令牌未发布用于未来日期。

关于FastTokenService库的更多信息，请参见[设置快速令牌验证](https://www.predix.io/docs/?r=198805#YzQ01I3v)。

您使用的令牌验证类型取决于您的应用程序的验证需求。例如，如果您需要使用令牌撤销，那么您必须使用远程令牌验证。但是，如果您只需要验证令牌未过期，您可以在本地验证令牌并尽量减少对UAA的联网请求。远程验证提供完整的验证支持，而本地验证更快而且网络负担更小。

下表显示了远程和本地验证可以使用的不同场景：

|  |  |  |
| --- | --- | --- |
| **验证需求** | **远程验证** | **本地验证** |
| 令牌是否被篡改？ | 是 | 是 |
| 令牌是否过期？ | 是 | 是 |
| 令牌是否发布用于未来日期？ | 是 | 是 |
| 令牌是否已被撤销？ | 是 | 否 |
| 用户是否仍然可用而且激活？ | 是 | 否 |
| 令牌是否由UAA发布？ | 是 | 是 |
| 发布应用程序是否仍然可用？ | 是 | 否 |

设置快速令牌验证

在一个典型的安全场景中，当您的应用程序使用UAA时，访问您的应用程序的客户端呈上其用于认证的UAA令牌。您的应用程序将该令牌重定向至UAA做验证。如果您的应用程序需要频繁验证这些令牌，会影响应用程序的性能。如果要减少这种风险，您可以使用Predix FastTokenService 库在启动和执行验证时提取UAA令牌签名密钥。 FastTokenService 库在整个maven artifactory中可用。

如果要使用 FastTokenService 库，

1. 可以将 uaa-token-lib依赖添加到您的应用程序POM文件。

<dependency>    <groupId>com.ge.predix</groupId>    <artifactId>uaa-token-lib</artifactId>    <version>3.1.1</version>    <type>pom</type></dependency>

1. 更新您的Spring配置：

<bean id="fasttokenServices"class="com.ge.predix.uaa.token.lib.FastTokenServices"><property name="storeClaims" value="true" /><property name="trustedIssuers"><list><value>${issuerId}</value></list></property></bean>

在本例中，如果 storeClaims 被设置为 true，它包括从UAA/check\_token端点接收的所有声明，用作字符串请求参数。

1. 更新您的Spring配置，以包括对 fastTokenService的引用。

 <oauth:resource-server id="oauth2ServiceFilter"token-services-ref="fasttokenServices" />

注：

Spring安全配置参见 oauth2ServiceFilter 如下所示：

<http pattern="/" request-matcher="ant"xmlns="http://www.springframework.org/schema/security"disable-url-rewriting="true" use-expressions="true"entry-point-ref="preAuthenticationEntryPoint" create-session="stateless"><intercept-url pattern="/about" access="isFullyAuthenticated()" /><anonymous enabled="false" /><custom-filter ref="oauth2ServiceFilter" position="PREAUTH\_FILTER" /></http>

1. 在您的代码中获取已验证的zoneId。

ZoneOAuth2Authentication zoneAuth = (ZoneOAuth2Authentication)SecurityContextHolder.getContext().getAuthentication();String zoneId = zoneAuth.getZoneId();

配置UAA以设置令牌失效时间

您可以配置您的UAA客户端，以设置令牌的过期时间，调节令牌过期的过程。

根据不同的授权类型，令牌过期时间会被指定为一个默认值。如果需要，您可以在一个设定范围内更新令牌过期。您既可以使用UAA API设定过期时间，也可以使用UAAC命令行工具。

关于使用UAA API的更多信息，请参见[UAA Documentation](https://github.com/GESoftware-CF/uaa/blob/master/docs/UAA-APIs.rst#client-registration-administration-apis)。

以下程序使用UAAC设定令牌过期时间。

1. 将您的UAA实例指定为预定目标。

uaac target <uaa\_instance\_url>

<uaa\_instance\_url> 是您的可信发布者的URL，例如， https://11fa0273-9e2a-37e2-9d06-2c95a1f4f5ea.predix-uaa.run.aws-usw02-pr.ice.predix.io。您可以在将您的UAA实例与一个应用程序绑定之后从VCAP\_SERVICES环境变量中检索该URL。

1. 使用管理员客户账户登录。

uaac token client get admin

1. 在提示时指定管理客户端密码。
2. 使用以下命令为令牌过期时间更新客户端。

uaac client update [client\_name] --access\_token\_validity <seconds>