Oauth工作组	J. Richer, Ed.
Internet-Draft	The MITRE Corporation
Intended status: Standards Track	J. Bradley
Expires: October 01, 2013	Ping Identity
	M.B. Jones
	Microsoft
	M. Machulak
	Newcastle University
	March 30, 2013

OAuth 2.0动态客户端注册道理

draft-ietf-oauth-dyn-reg-09

Abstract

该规范定义了OAuth 2.0客户端ATAN授权服务器的动态注册的终点和协议，以及动态注册的客户端管理其注册的方法。

本备忘录的状态

本互联网草案的提交完全符合 BCP 78 和 BCP 79 的规定。

Internet草稿是Internet工程工作组（IETF）的工作文件。请注意，其他组也可以将工作文档作为互联网组合分发。当前的Internet-Drafts列表在http://datatracker.ietf.org/drafts/current/。

互联网草稿是最长六个月有效的文件草案，并且可以随时更新，更换或封闭其他文件。使用Internet-Draftsas参考材料或将其引用为“正在进行的工作”是不合适的。

该互联网草稿将于2013年10月1日到期。

版权声明

目录

一、简介

在某些用例场景中，允许OAuth 2.0客户端从OAuth 2.0授权服务器获得授权，而无需双方进行交互。然而，为了使授权服务器准确，安全地阐明客户端寻求授权以访问最终用户资源的最终用户，需要自动且独特的客户注册。 OAuth 2.0授权FrameWork并未定义客户端与授权服务器之间的关系如何iSitialization或给定客户端如何分配唯一的客户端标识符。从历史上看，这使OAuth 2.0协议的带外挂钩。该草案为客户端托尔吉斯特本身提供了一种机制，该机制可用于动态提供客户端识别仪，并选择一个客户端的秘密。

作为注册过程的一部分，此规范还定义了客户端限制授权服务器的机制，其中一组元数据，例如在授权步骤中向用户表示的显示名称和图标。该草案还提供了一种机制，可以在初始注册措施之后阅读和更新此信息。

1.1。符号惯例

关键词“必须”，“必须”，“必需”，“应”，“不得”，“应”，“应该”，“不应该”，“推荐”，“可能”，“可能”和“可选”，如[RFC2119]中的描述。

除非另有说明，否则所有协议参数名称和值都是案例敏感的。

1.2. Terminology

This specification uses the terms "Access Token", "Refresh Token", "Authorization Code","Authorization Grant", "Authorization Server", "Authorization Endpoint", "Client", "Client Identifier","Client Secret", "Protected Resource", "Resource Owner", "Resource Server", and "Token Endpoint"defined by OAuth 2.0 [RFC6749].

该规范定义了以下其他术语：

该规范定义了以下其他术语：

- 客户端注册端点：OAUTH 2.0端点可以通过该端点请求NewRegistration。客户端获得此端点URL的平均值不超出规范。Client配置端点：OAUTH 2.0端点的
- OAUTH 2.0端点，特定客户端通过该端点Canmanage CanManage其注册信息，由授权服务器提供给客户端。此端点的thisUrl通过客户端信息响应中的授权服务器传达给客户端。注册访问令牌：由授权服务器通过客户端注册端点发出的OAuth
- 2.0 Bearer令牌，客户端使用该端点，该端口可通过客户端用来身份验证自身对其自身进行读取，更新，更新和Delete Operations。这个令牌与特定客户端关联。

2。客户元数据

客户通常在Theauthorization服务器上具有与其唯一客户端标识符相关的元数据。这些范围从面向人物的显示字符串（例如客户端名称）到影响协议安全性的项目，例如有效重定向URI的列表。

此规范的扩展和配置文件可能会扩展此列表，但必须至少接受此列表中的AllParameters。授权服务器必须忽略其不理解的客户端发送的任何其他参数。

[[编辑注：下表中的规范语言旨在在请求时适用于 *客户端 *。上面的段落是指服务器至少必须接受AllParameters，而不会在未知参数上出现错误，尤其是在下面列表中的错误时，extersensions需要明确调用，如果他们不打算使用这些基本参数之一来做某事，而不是忽略它们的存在。这是用于互操作性的参数的 *最低集合 *。]]]]

redirect_uris

受到推崇的。重定向URI的数组用于授权代码和隐式授予类型。授权服务器应要求对使用这些赠款类型的所有授予类型来防止令牌和凭据theftattacks进行有效重定向URI的注册。

client_name

受到推崇的。将向用户展示的客户的人类可读名称。如果符合人，则授权服务器可以向用户显示RAW CLIELT_ID值。该字段的值可以如人类可读的客户端MeterMetadata [HumanReadableClientMetadata]中进行国际化。

client_uri

受到推崇的。客户主页的URL。如果存在，服务器应以可单击的方式向最终用户显示此URL。该领域的价值可以是人类可读客户元数据中描述的国际化[HumanReadableClientMetadata]。

logo_uri

选修的。 URL引用客户端的徽标。如果存在，则服务器应在批准期间向最终用户显示此图像。该字段的价值可以在人类可读客户元数据中进行国际化asdecrized。

contacts

选修的。负责此客户的人的一系列电子邮件地址。授权服务器可以使最终用户可用于客户端的支持请求。授权服务器可以将这些电子邮件地址用作该客户端的管理页面的标识符。

tos_uri

选修的。指向客户的人类可读条款的URL。如果给出了theauthorization服务器，则应将此URL显示给最终用户。 Thisfield的价值可以按照人类可读客户元数据[HumanReadableClientMetadata]中所述进行国际化。

token_endpoint_auth_method

选修的。令牌端点的请求的身份验证类型。有效值为：

- none: this is a public client as defined in OAuth 2.0 and does not
- have a clientsecretclient_secret_post: the client uses the HTTP
- POST parameters defined inOAuth 2.0 section
- 2.3.1client_secret_basic: the client uses HTTP Basic defined in
- OAuth 2.0section 2.3.1client_secret_jwt: the client uses the JWT
- Assertion profile with asymmetric secret由
- serverprivate_key_jwt发行：客户端使用JWT断言配置文件和其自动密钥

其他身份验证方法可以通过扩展定义。如果未指定或省略，默认值为

client_secret_basic，用OAuth 2.0节的第2.3.1节中指定的HTTP基本身份验证方案。

scope

选修的。空间分离的范围值列表（如OAuth 2.0 2.0第3.3节[RFC6749]中所述），该客户端正在声明其在请求访问令牌时可能使用。 i，如果授权服务器可以注册默认范围范围的客户端。

grant_types

选修的。客户可能使用的OAuth 2.0赠款类型的数组。这些赠款类型的定义如下：

- authorization_code: The Authorization Code Grant described in
- OAuth 2.0Section 4.1.implicit: The Implicit Grant described in OAuth
- 2.0 Section 4.2.password: The Resource Owner Password Credentials
- Grant described inOAuth 2.0 Section 4.3client_credentials: The
- Client Credentials Grant described in OAuth 2.0Section
- 4.4refresh_token: The Refresh Token Grant described in OAuth 2.0
- Section 6.URN：IETF：参数：OAUTH：授予型：JWT-BEARER：
- OAuth JWT BEARER TOKEN PROFILES [OAUTH.JWT] .URN：IETF：
- IETF：ietf：params：oauth：oauth：oauth：grant-type：saml2-
- bearer：saml2-bearer：saml 2 bearer： [oauth.saml2]。

授权服务器可以允许在赠款类型扩展中定义的其他值TOAUTH 2.0。扩展过程在OAuth 2.0第2.5节中描述，此参数的值必须与该值的值相同

Grant_Type参数传递给扩展名中定义的令牌端点。

response_types

选修的。客户可能使用的OAuth 2.0响应类型的数组。这些响应型定义如下：

- 代码：OAUTH 2.0中描述的授权代码响应第4.1节中描述的词：OAuth 2.0中描述的隐
- 式响应4.2节

授权服务器可以允许在响应类型ExtensionSto OAuth 2.0中定义的其他值。扩展过程在OAuth 2.0第2.5节中描述，此参数的值必须与该参数的值相同

响应_type参数传递给扩展名中定义的授权端点。

policy_uri

选修的。客户端向最终用户提供的URL位置，以阅读如何使用profile数据。授权服务器应将此URL显示给给出的最终用户。该领域的价值可以如人类读取元素元数据[HumanReadableClientMetadata]中所述进行国际化。

jwks_uri

选修的。客户端的JSON Web密钥集[JWK]文档的URL，用于签名列表，例如使用Private_key_key_jwt AssertionClient凭据到令牌端点的请求。这些键也可以用于需要签名Orencyption的更高级别协议。

2.1。赠款类型与响应类型之间的关系

上面描述的授予_ types和wenders_types值是部分正交的，因为他们将参数转移到了OAuth协议中的不同端点。但是，它们与客户端可用的授予_TYPES会影响客户使用的响应_型，反之亦然。例如，包含Authorization_code的Grant_Types值表示包含代码的响应_types值，因为这两个值都定义为OAuth 2.0 2.0授权代码授予的一部分。因此，支持这些字段的服务器应采取步骤，以确保客户端不能将自己注册到不一致的状态。

下表列出了两个字段之间的相关性。

grant_types值包括：response_types值包括：	**response_types value includes:**
authorization_code	code
implicit	token
password	(none)
client_credentials	(none)
refresh_token	(none)
urn：ietf：params：oauth：授予型：jwt-bearer（无）	

本文档的扩展和配置文件向grant_types orresponse_types参数引入新值，必须记录参数类型之间的所有对应关系。

2.2。人类可读客户元数据

人类可读的客户端元数据值和客户元数据值可以用多种语言和脚本表示人类可读值。例如，诸如client_name，tos_uri，polition_uri，logo_uri和client_uri之类的值可能在某些客户端注册中具有特定于LocaLe的特定值。

要指定语言和脚本，BCP47 [RFC5646]语言标签添加到客户端MeterMetadata成员名称中，由＃字符界定。由于JSON成员名称对病例敏感，因此建议使用索赔名称中使用的语言标签值使用thecharacter案例拼写，并在IANA语言子标签注册表中注册它们[iana.language]。特别是，通常用小写字符拼写语言名称，区域名称用大写字符拼写，并且语言用混合的casecharacters拼写。但是，由于BCP47语言标签值是案例不敏感的，因此实现应该解释以情况不敏感的方式提供的语言标签值。在BCP47中，根据元数据成员名称中使用的语言标签值仅应在必要时尽其所能。例如，在许多情况下，使用FR可能就足够了，而是FR-CA或FR-FR。

例如，客户端可以用英语代表其名称为“ client_name＃en”：“我的客户端”及其名称为“ client_name＃ja-jpan-jp”：“” SamereGistration请求中。授权服务器可以在授权步骤中向Theresource所有者显示任何或全部这些名称，选择基于SystemConfiguration，用户偏好或其他因素显示的名称。

如果在没有语言标签的情况下发送任何人类可读字段，则使用它的各方不得对字符串值的语言，字符集或脚本进行任何误解，并且在用户界面中呈现的任何地方都将使用字符串valuemust。为了促进互操作性，请申请客户和服务器在没有任何语言标签的情况下使用可读字段，而不会对任何语言特定的字段进行任何操作，建议任何没有语言标签的人类可读字段都包含适合在各种系统上显示的值的人类可读字段。

实施者的注释：许多JSON库使将JSON对象的成员引用为库的本机编程环境中的对象构造的成员。
as members of an Object construct in the native programming environment of the library.

但是，尽管＃字符是JSON对象的成员名称内部的有效字符，但它不是在许多编程环境中在对象成员名称中使用的有效字符。因此，实现将需要为这些索赔使用替代访问表单。 forinstance，在JavaScript中，如果可以使用JavaScript语法J [“ client_name＃en-en-us”]。

3。客户注册端点

客户端注册端点是本文档中定义的OAUTH 2.0端点，它具有设计为允许客户端在授权服务器上注册的端点。客户端登记点必须接受HTTP发布消息，其中包含使用应用程序/JSON格式的实体中编码的请求参数。客户端注册端点必须受Atransport-Layer安全机制保护，并且服务器必须支持TLS 1.2 RFC 5246 [RFC5246]和/或TLS 1.0 [RFC2246]，可以支持其他运输层机制。使用TLS时，客户端必须按照RFC 6125 [RFC6125]执行TLS/SSL Server证书。

客户注册端点可以接受AnoAuth 2.0 [RFC6749]访问令牌的形式的初始授权凭据，以便将注册限制在仅前的授权工作人员中。注册人获得此访问令牌的方法通常是带外的，并且不超出此规范的范围。

为了支持开放注册并促进更广泛的互操作性，客户端登记处应允许在没有身份验证的情况下进行初始注册请求。这些请求可能会受到限制或以其他方式限制，以防止对客户登记处的拒绝服务攻击。

为了促进注册客户更新其信息，客户端注册端口ississuse请求访问令牌供客户端在以后的连接中安全地识别自己的端口端端[AccessEndPoint]。因此，对这些操作的OAUTH 2.0载体令牌[RFC6750]的客户端配置端点要接受请求，无论是Ornot Ornot Ornot the Intial Registration Call是否需要对某些形式进行身份验证。

客户端注册端点必须忽略其不了解的所有参数。

3.1。客户注册请求

该操作将新客户端注册到授权服务器。授权服务器分配了该客户端一个唯一的客户端标识符，可选为客户端秘密分配，并将请求中给出的temetadata与已发行的客户端标识符相关联。该请求包括客户元数据[客户端 - 米达塔]中描述的任何参数，客户希望在注册过程中指定自己的forit。授权服务器可以为客户元数据中的任何项目提供默认值。

客户端将HTTP帖子发送到客户端注册端点，其中包含“应用程序/JSON”的内容类型。 HTTP Entity有效负载是JSON [RFC4627]文档，该文档由JSOnObject和所有参数作为该JSON对象的顶级成员组成。

例如，客户可以将以下注册请求发送到客户端注册点：

以下是非规范示例请求（仅用于显示目的的行包装）：

```
POST /register HTTP/1.1
Content-Type: application/json
Accept: application/json
Host: server.example.com

{ “ redirect_uris”：[ “ https：//client.example.org/callback”， “
https：//client.example.org/callback2”] “ client_name”： “我的 ，
example client”， “我的example client”， “
token_endpoint_authpoint_auth_auth_method_method” dolphin “，”
logo_uri “：” https：//client.example.org/logo.png “，” jwk_uri “：”
https：//client.example.org/my_rsa_rsa_rsa_public_key.jwk “

 "jwk_uri":"https://client.example.org/my_rsa_public_key.jwk"
```

```
}
```

3.2。客户注册响应

成功注册后，授权服务器会为theclient生成新的客户端标识符。该客户端标识符必须在服务器上是唯一的，并且不得由任何其他client使用。服务器以HTTP 201创建的代码和类型应用程序/JSONWITH内容的响应，客户信息响应[客户端INFO-RESPONSE]中描述的内容。

在不成功的注册后，授权服务器响应一个错误，如所述包含的注册错误[Client-Registration-Error]。

4。客户端配置端点

客户端配置端点是OAuth 2.0受保护的端点，由这些端点提供，以便特定客户端能够查看和更新其注册信息。客户必须在此端点的所有呼叫中以OAuth 2.0携带者令牌[RFC6750]的所有调用中的注册访问令牌。

该端点上的操作通过使用不同的HTTP方法[RFC2616]切换。

4.1。形成客户端配置端点URL

授权服务器必须在客户信息响应[客户端Info-response]中为客户端提供完全合格的URL。授权服务器不得期望客户端必须自行构建或发现此URL。客户端必须使用服务器给出的URL，并且不得必须构造此url fromll promonement部分。

根据部署特征，客户端配置端点URL可能会采用任何表格。建议通过使用ASERVER构造的URL字符串来形成此端点URL，该URL字符串结合了客户端注册端点的URL和该客户端的theissed Client_id，后者是路径参数或查询参数。前提，可以为客户ID S6BHDRKQT3的客户端提供一个客户端配置端点https://server.example.com/register.com/register/s6bhdrkqt3（path parameter）或ofhttps：//server.example.example.com/registe.com/register.climper_id = s6bdrkqu quameter parameter。在这两种情况下，客户只需遵循所给出的URL即可。

这些常见的模式可以帮助服务器更轻松地确定限定范围的客户，这些端口必须与发出注册访问权限的客户端相匹配。如果需要，服务器可以简单地将客户端注册端点URL作为客户端配置端点URL和基于注册访问令牌提供的身份验证上下文更改行为。

4.2。客户阅读请求

为了读取授权服务器上客户端的当前配置，客户端将HTTP获取请求到客户端配置端点，并使用其registration访问令牌进行身份验证。

以下是非规范示例请求（仅用于显示目的的行包装）：

```
GET /register/s6BhdRkqt3 HTTP/1.1
Accept: application/json
Host: server.example.com
Authorization: Bearer reg-23410913-abewfq.123483
```

成功阅读了当前活动客户端的信息后，具有HTTP 200的授权serverResponds，具有内容类型的应用程序/JSON，以及在客户端信息响应[Client-Info-response]中的有效载荷。

如果该服务器上不存在客户端，则服务器必须返回禁止的HTTP 403。

如果客户端没有读取其记录的权限，则服务器必须返回HTTP 403forbidden。
Forbidden.

4.3。客户端更新请求

该操作在授权服务器上使用新的元数据更新了先前注册的客户端。该请求通过发给客户的注册访问令牌来验证。

客户端将HTTP发送给客户端配置端点，并使用内容类型的Application/JSON发送。 HTTP Entity有效负载是JSON [RFC4627]文档，该文档由JSONOBJECT和所有参数作为该JSON对象的顶级成员组成。

该请求必须包括客户元数据[客户端 - 米达塔]中描述的所有字段，作为从上一个寄存器，读取或更新操作中返回的客户端。 The Client MUST NOT include theregistration_access_token, registration_client_uri, expires_at, or issued_at fieldsdescribed in Client Information Response [client-info-response].

此请求中客户端元数据字段的有效值必须替换与该客户端相关的值的值。省略的字段必须由服务器视为空值或空值。

客户端必须在请求中包含其客户端_id字段，并且必须与其递送的客户端标识符相同。如果客户端在请求中包含client_secret字段，则该字段的值必须与该客户端的当前发布的客户端秘密匹配。客户必须用自己选择的价值覆盖其现有客户秘密。

对于所有元数据字段，授权服务器可以用suebableFeault值替换任何无效的值，并且必须将任何此类字段返回响应中的客户端。

例如，客户端可以将以下请求发送到客户注册端点，以在上面的示例中使用新信息：

以下是非规范示例请求（仅用于显示目的的行包装）：

```
PUT /register/s6BhdRkqt3 HTTP/1.1
Accept: application/json
Host: server.example.com
Authorization: Bearer reg-23410913-abewfq.123483

{
 "client_id":"s6BhdRkqt3",
 "client_secret": "cf136dc3c1fc93f31185e5885805d",
 "redirect_uris":["https://client.example.org/callback",
    "https://client.example.org/alt"],
 "scope": "read write dolphin",
 "grant_types": ["authorization_code", "refresh_token"]
 "token_endpoint_auth_method": "client_secret_basic",
 "jwk_uri": "https://client.example.org/my_rsa_public_key.jwk"
 "client_name":"My New Example",
 "logo_uri":"https://client.example.org/newlogo.png"
}
```

成功更新后，授权服务器使用HTTP 200 OK消息响应，使用Concontent Type应用程序/JSON和client信息响应中所述的有效载荷[客户端INFO-RESPONDE]。授权服务器可以在其响应中包含一个新的客户端秘密和/或登记访问令牌。如果是这样，客户必须立即丢弃其以前的秘密和/或注册访问令牌。

如果该服务器上不存在客户端，则服务器必须返回禁止的HTTP 403。

如果不允许客户端更新其记录，则服务器必须使用http 403forbiddend响应。

如果客户端试图设置无效的元数据字段，并且授权服务器未设置Adefault值，则授权服务器以客户client RegistrationError响应[Client-Registration-Error]中所述的错误响应。

4.4。客户端删除请求

[[编辑注：此功能的实用性和性质仍在积极的讨论中。这是服务器可以选择实现的一组功能集，否则给对任何尝试的端子进行405响应，如果不能支持它。]]]]

为了在授权服务器上剥夺本身，客户端将http deleterequest放在客户端配置端点上。该请求由发给客户的注册表令牌进行了身份验证。

以下是非规范示例请求（仅用于显示目的的行包装）：

```
DELETE /register/s6BhdRkqt3 HTTP/1.1
Accept: application/json
Host: server.example.com
Authorization: Bearer reg-23410913-abewfq.123483
```

成功的删除操作将为此客户端提供client_id，client_secret和registration_access_token的无效，从而阻止client_id在授权服务器的授权端点令牌端点上使用。授权服务器应立即宣布与此客户端关联的所有现有授权赠款和当前活跃的令牌。

如果客户端已成功剥夺，则授权服务器使用HTTP204响应没有内容消息。

如果没有此类客户端，则服务器以HTTP 403禁止响应。

如果不允许客户端删除自身，则服务器以HTTP 403的响应禁止响应。

如果服务器不支持删除方法，则它以不支持的HTTP 405响应。

以下是一个非规范的示例响应：

```
HTTP/1.1 204 No Content
Cache-Control: no-store
Pragma: no-cache
```

5。回答

响应客户端对客户端注册端点的某些请求或本规范中所述的端配置端点，授权服务器发送了遵守响应主体。

5.1。客户信息响应

如果客户端是保密信息，则响应包含客户端标识符以及客户端的秘密。该响应还包含对客户端配置端点特定客户端的完全资格的URL，客户端可以用来获取和更新有关自身的信息。theresponse还包含一个注册访问令牌，客户将在客户端配置端点上执行SSUBSESTESSESTER操作。

client_id

必需的。唯一的客户端标识符当前不得对任何其他注册现象有效。

client_secret

选修的。客户秘密。如果发行，则必须对每个客户端_ID唯一。机密客户端使用的这个值为对令牌端点进行身份验证，如OAuth2.0第2.3.1节中所述。

expires_at

如果发出client_secret，则需要。从UTC中提出的1970-01-01-01T0：0：0 Z的秒数，client_secret将过期或0，如果不过期。有关日期/时间，尤其是UTC，请参见RFC 3339 [RFC3339]。

issued_at

选修的。在发出客户端标识符时指定时间戳。时间戳记必须是一个积极的整数。自1970年2月1日以来，该值以几秒钟的数量表示。

registration_access_token

必需的。客户使用的访问令牌可以在客户端限制端点上执行操作。

registration_client_uri

必需的。该客户端的客户端配置端点的完全合格的URL。与客户端configurationEndpoint通信时，theclient必须使用此URL。

此外，授权服务器必须返回有关该客户端的所有注册元数据[客户端 - 米达塔]，包括授权服务器本身提供的任何字段。授权服务器可以拒绝或替换在此期间提交或更新请求的客户所请求的元数据值，并用合适的值代替它们。

响应是一个应用程序/JSON文档，其中所有参数是AJSON对象的顶级成员[RFC4627]。

以下是一个非规范的示例响应：

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
 "registration_access_token": "reg-23410913-abewfq.123483",
 "registration_client_uri":
       "https://server.example.com/register/s6BhdRkqt3",
 "client_id":"s6BhdRkqt3",
 "client_secret": "cf136dc3c1fc93f31185e5885805d",
 "expires_at":2893276800
 "redirect_uris":["https://client.example.org/callback",
    "https://client.example.org/callback2"]
 "scope": "read write dolphin",
 "grant_types": ["authorization_code", "refresh_token"]
 "token_endpoint_auth_method": "client_secret_basic",
 "logo_uri": "https://client.example.org/logo.png",
 "jwk_uri": "https://client.example.org/my_rsa_public_key.jwk"
}
```

5.2. Client Registration Error Response

当发生OAuth 2.0错误条件时，例如呈现无效registrationAccess令牌的客户端时，授权服务器返回了TheoAuth 2.0规范第5.2节中定义的错误响应。

当发生注册错误条件时，授权服务器将返回由Content类型应用程序/JSON组成的HTTP 400 STATUS代码，该代码由JSON对象[RFC4627]组成，描述了响应主体中的ERROR。

JSON对象包含两个成员：

error

错误代码，一个ASCII字符串。

error_description

调试错误的人类可读文本描述。

该规范定义了以下错误代码：

invalid_redirect_uri

一个或多个redirect_uris的值无效。

invalid_client_metadata

客户端元数据[client-metadata]字段之一的值无效，服务器已对此请求进行了哈西。请注意，授权服务器可以选择将有效值替换为客户端元数据的任何请求参数。

invalid_client_id

client_id的值无效。

以下是错误响应的非规范示例（具有显示目的的线包）：

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{“error”: “invalid_redirect_uri”, “error_description”: “此服务器不允许使用http://sketchy.example.comis的重定向URI。”}
```

6. IANA Considerations

该文档不提出IANA的要求。

7. 安全考虑

由于对客户端注册端点的请求会导致清除textcredentials的传输（在HTTP请求和响应中），因此服务器在将请求发送到注册端点时必须需要使用运输层安全机制。该服务器必须支持TLS 1.2 RFC 5246 [RFC5246]和/或TLS 1.0 [RFC2246]，并可能支持满足其安全要求的其他Transport-Layer机制。使用TLS时，根据RFC 6125 [RFC6125]，客户端必须使用TLS/SSL服务器证书检查。

由于此端点是OAuth 2.0受保护的资源，因此注册端点的请求应限制故障的速率，以防止注册访问令牌被限制在重复访问尝试时被限制。

授权服务器必须将所有客户端元数据视为自我掌握。流氓客户可能会为合法客户提供姓名和徽标，并试图模仿它。授权服务器需要采取措施来减轻这种网络钓鱼风险，因为徽标可能会使用户混淆他们登录合法客户。例如，授权服务器可以警告徽标的域/站点与重定向URI的域/站点不符。在所有情况下，授权服务器还可以向最终用户提供有关不受信任客户端的警告消息，尤其是如果此类客户端已动态注册并且以前没有被任何用户信任授权服务器。

在授权服务器支持开放客户端注册的情况下，必须对将向用户显示的客户提供的任何URL谨慎（例如logo_uri和policy_uri）。 Rogue客户端可以指定注册请求，并在policy_uri中使用参考toa驱动器下载。授权服务器应检查thelogo_uri和polition_uri是否具有与Redredect_uris数组中定义的主机相同的主机。

虽然客户秘密可以到期，但注册访问令牌不应在客户积极注册时过期。如果该令牌要到期，则可以将客户留在一个可以自行更新并必须重新注册的情况下。由于注册访问令牌Arelong-Term凭据，并且由于注册访问令牌是一个携带者令牌，并且用作客户端配置端点的疗法验证，因此必须受到OAuth 2.0 Bearer中描述的客户端的保护[RFC6750]。

如果客户端从服务器中删除，则任何未出色的注册访问令牌

客户必须同时无效。否则，这可能会导致不一致的情况下，客户可能会向客户端配置端点提出请求，在该端点会成功，但由于客户端不再有效，因此该操作会失败。

8。规范参考

[RFC2119] Bradner，S。，“在RFC中使用以指示要求级别的关键词”，BCP 14，RFC 2119，1997年3月。[RFC2246] Dierks，T。和C. Allen，TLS协议1.0“1.0”，RFC 2246，RFC 2246，1999年1月9日。Frystyk，H.，Masinter，L.，Leach，P。和T.Berners-Lee，“超文本转移协议-HTTP/1.1”，RFC 2616，1999年6月。[RFC3339] Klyne，G。和C. Newman，G。和C. Newman和C. Newman，”，互联网上的日期和时间：时间段：timestamps”，timestamps“，rfc3339，2002年7月22. RFC52。 Rescorla, "The Transport Layer Security (TLS) Protocol Version1.2", RFC 5246, August 2008. [RFC5646]Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646,September 2009.[RFC6125]Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public在运输层安全性（TLS）中使用X.509（PKIX）的关键基础结构”，RFC6125，2011年3月。[RFC6749] Hardt，D。，“OAuth 2.0授权框架”，RFC 6749，2012年10月10日，2012年10月。Tokenusage”，RFC 6750，2012年10月。[RFC4627] Crockford，D。，“JavaScript对象符号（JSON）的应用/JSON媒体类型”，RFC 4627，2006年7月。[JWK] Jones，M.B.莫蒂莫尔（Mortimore 2012年。[iana.language]互联网分配的数字权威（IANA），“语言子标签注册表”，2005年。

P 14, 999. 9. FC rsion 6, - Using RFC 12. oken ion uth

附录A.致谢

作者感谢OAuth工作组，用户管理的访问工作组和Theopenid Connect工作组参与者对本文档的投入。特别是，遵守个人在审查和对本文档的各种版本的贡献中发挥了作用：Amanda Anganes，Tim Bray，Domenico Catalano，George Fletcher，Torstenlodderstedt，Eve Maler，Thomas Hardjono，Nat Sakimura和Christian Scholz。

附录B。文件历史记录

[[在出版作为RFC之前，RFC编辑器将其删除]]

-09

- 添加了客户元数据值的国际化方法

-08

- JWK_URI，JWK_ENCRYPTIC_URI，X509_uri和X509_ENCRYPTICT_URI折叠成一个JWKS_URIPARAMETRERENED
- grant_type to Grant_types，因为它是多元化的价值形式的“Oauth 2.0”
- oauth 2.0的多元化值，以示例为“Oauth 2.0”。参数和解释性文本与其使用和关系togrant_types

-07

- Changed registration_access_url to registration_client_uriFixed
- missing text in 5.1Added Pragma: no-cache to examplesChanged "no
- such client" error to 403Renamed Client Registration Access Endpoint
- to Client Configuration EndpointChanged all the parameter names
- containing "_url" to instead use "_uri"Updated example text for forming
- Client Configuration Endpoint URL

-06

- 删除的Secret_Rotation作为客户启动的动作，包括删除客户的秘
- 密终点和参数。更改为单个值
- regumtration_access_url.collapsed创建/更新/读取客户信息响
- 应的结构。在主体中指定JSON。编辑器的注释以删除有关其包含
- 的操作。编辑器的注释docration_access_url关于替代语法建议。
-
-

-05

- changed redirect_uri and contact to lists instead of space
- delimited stringsremoved operation parameteradded _links
- structuremade client update management more RESTfulsplit
- endpoint into three partschanged input to JSON from form-
- encodedadded READ and DELETE operationsremoved
- Requirements sectionchanged token_endpoint_auth_type back to
- token_endpoint_auth_method to match OIDCwho更改以匹配我们

-04

- removed default_acr, too undefined in the general OAuth2 caseremoved
- default_max_auth_age, since there's no mechanism for supplying a non-
- defaultmax_auth_age in OAuth2clarified signing and encryption URLschanged
- token_endpoint_auth_method to token_endpoint_auth_type to match OIDC

-03

- 添加了范围和grant_type索取各种错别字并更改了措辞以获得更好的
- clarityEndpoint现在返回client_update上的全套客户端信息操作允许在元数据上
- 进行三个操作：留下现有值，clearexisting value，替换现有值，用新值替换现有值

-02

- 重组的贡献者和参考文献对RFCreoranized模型/协议部分
- 进行重新组织，以验证“客户端寄存器”，而不是“客户端
- 签名”，而不是指定client_id必须在所有后续请求中匹配的
- client_id，尤其是在列表中匹配的，尤其是在列表中匹配
-

-01

- 合并的UMA和OpenID连接注册中的单个文档已更改为
- 形式参数输入，以基于Pultemper的登记

-00

- 导入原始UMA草稿规范

作者地址

Justin Richer (editor)
The MITRE Corporation
EMail: jricher@mitre.org

John Bradley
Ping Identity
EMail: ve7jtb@ve7jtb.com

Michael B. Jones
Microsoft
EMail: mbj@microsoft.com

Maciej Machulak
Newcastle University
EMail: m.p.machulak@ncl.ac.uk
URI: http://ncl.ac.uk/