1.1. 角色

OAuth 定义了四种角色：

资源所有者（resource owner）

能够授权访问受保护资源的实体。

当资源所有者是个人时，也称为终端用户（end-user）。

资源服务器（resource server）

托管受保护资源的服务器，能够接受并使用访问令牌（access tokens）响应受保护资源的请求。

客户端（client）

代表资源所有者并以其授权权限请求受保护资源的应用程序。

术语“客户端”不特指任何具体的实现形式（例如，应用程序可运行于服务器、桌面设备或其他设备上）。

授权服务器（authorization server）

在成功验证资源所有者身份并获取授权后，向客户端颁发访问令牌的服务器。

授权服务器与资源服务器之间的交互不在本规范范围内。授权服务器可以与资源服务器是同一实体，也可以是独立实体。一个授权服务器可以颁发被多个资源服务器接受的访问令牌。

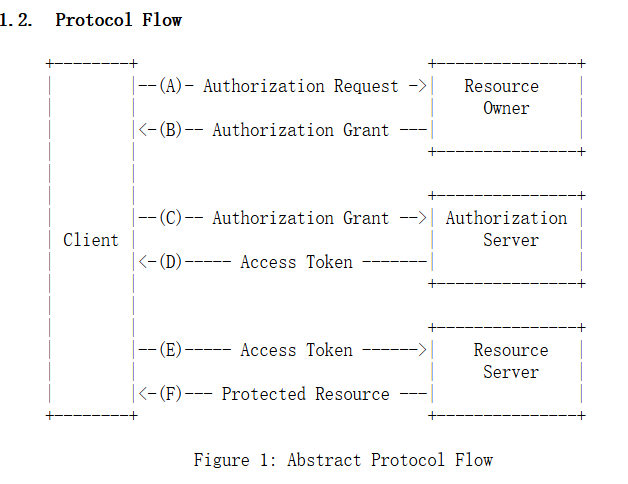


图1所示的OAuth 2.0抽象流程描述了四种角色之间的交互，包括以下步骤：

(A) 客户端向资源所有者请求授权。授权请求可以直接向资源所有者发起（如图所示），但更推荐通过授权服务器作为中介间接发起。

(B) 客户端获得授权许可（authorization grant），这是一个代表资源所有者授权的凭证，其形式采用本规范定义的四种许可类型之一，或使用扩展许可类型。具体授权许可类型取决于客户端请求授权的方式以及授权服务器支持的类型。

(C) 客户端通过向授权服务器认证并提交授权许可，请求访问令牌（access token）。

1. 授权服务器对客户端进行认证并验证授权许可的有效性，若验证通过，则颁发访问令牌。

(E) 客户端向资源服务器请求受保护资源，并通过提交访问令牌进行身份验证。

(F) 资源服务器验证访问令牌的有效性，若验证通过，则响应请求。

客户端从资源所有者处获取授权许可（如步骤(A)和(B)所示）的首选方式是通过授权服务器作为中介，该流程将在第4.1节的图3中详细说明。

### 授权许可

授权许可（authorization grant）是代表资源所有者授权（允许访问其受保护资源）的凭证，客户端使用该凭证获取访问令牌。本规范定义了四种许可类型——授权码（authorization code）、隐式（implicit）、资源所有者密码凭证（resource owner password credentials）和客户端凭证（client credentials）——同时支持通过扩展机制定义其他类型。

#### 1.3.1 授权码

授权码通过授权服务器作为客户端与资源所有者之间的中介来获取。客户端不直接向资源所有者请求授权，而是将资源所有者（通过其用户代理，如[RFC2616]所定义）重定向至授权服务器，授权服务器随后将资源所有者连同授权码一起重定向回客户端。

在将资源所有者重定向回客户端并附带授权码之前，授权服务器会对资源所有者进行身份验证并获取授权。由于资源所有者仅与授权服务器进行身份验证，其凭证永远不会与客户端共享。

授权码提供了若干重要的安全优势，例如能够验证客户端身份，以及将访问令牌直接传输给客户端，而无需经过资源所有者的用户代理，从而避免令牌可能泄露给包括资源所有者在内的其他方。

#### 1.3.2 隐式许可

隐式许可（implicit grant）是简化版的授权码流程，专为使用JavaScript等脚本语言实现的浏览器内客户端优化。在隐式流程中，客户端会直接获取访问令牌，而非先获取授权码。（作为资源所有者授权的结果）。该许可类型为隐式（implicit），因为不会颁发中间凭证（如授权码）（后续也不需要用它来获取访问令牌）。

在隐式许可流程中颁发访问令牌时，授权服务器不会对客户端进行身份验证。在某些情况下，可通过用于传递访问令牌给客户端的重定向URI来验证客户端身份。访问令牌可能会暴露给资源所有者或其他能够访问资源所有者用户代理的应用程序。

隐式许可提高了某些客户端（如浏览器内应用实现的客户端）的响应速度与效率，因为它减少了获取访问令牌所需的往返次数。但这一便利性需与隐式许可的安全风险（如第10.3节和10.16节所述）进行权衡，尤其是在授权码许可类型可用的情况下。

#### 1.3.3 资源所有者密码凭证

资源所有者密码凭证（即用户名和密码）可直接作为授权许可来获取访问令牌。该方式仅适用于资源所有者与客户端之间存在高度信任关系的情况（例如客户端是设备操作系统的一部分或高权限应用），且其他授权许可类型（如授权码）不可用时。

尽管此许可类型要求客户端直接获取资源所有者凭证，但这些凭证仅用于单次请求，并会兑换为访问令牌。通过将凭证换为长期有效的访问令牌或刷新令牌，该许可类型可避免客户端存储资源所有者凭证以供后续使用。

#### 1.3.4 客户端凭证

当授权范围仅限于客户端控制的受保护资源，或事先与授权服务器约定的受保护资源时，客户端凭证（或其他形式的客户端认证）可作为授权许可使用。客户端凭证通常用于以下场景：客户端代表自身（客户端即资源所有者）进行操作，或基于事先与授权服务器约定的授权请求访问受保护资源。

### 1.4 访问令牌

访问令牌（Access Token）是用于访问受保护资源的凭证。它是一个字符串，代表颁发给客户端的授权许可。该字符串通常对客户端不透明。令牌具有特定的访问范围和有效期，这些权限由资源所有者授予，并由资源服务器和授权服务器强制执行。

令牌可以是用于检索授权信息的标识符，也可以以可验证的方式（即由数据和签名组成的令牌字符串）直接包含授权信息。客户端在使用令牌时，可能需要额外的认证凭证（这已超出本规范的范围）。

访问令牌提供了一个抽象层，用资源服务器能够识别的单一令牌替代了不同的授权结构（如用户名和密码）。这种抽象使得访问令牌的权限可以比获取它们时使用的授权许可更加严格，同时也免除了资源服务器需要支持多种认证方法的负担。

根据资源服务器的安全需求，访问令牌可以具有不同的格式、结构和使用方法（例如加密属性）。访问令牌的属性及访问受保护资源的具体方法不在本规范范围内，而是由配套规范（如[RFC6750]）定义。

### 1.5 刷新令牌

刷新令牌（Refresh Token）是用于获取访问令牌的凭证。刷新令牌由授权服务器颁发给客户端，用于在当前访问令牌失效或过期时获取新的访问令牌，或者获取具有相同或更窄权限范围的额外访问令牌（访问令牌的生命周期可能较短，且权限可能少于资源所有者授予的范围）。是否颁发刷新令牌由授权服务器自行决定。如果授权服务器颁发了刷新令牌，它会在颁发访问令牌时一并提供（即图1中的步骤(D)）。

刷新令牌是一个字符串，代表资源所有者授予客户端的授权。该字符串通常对客户端不透明。令牌是一个标识符，用于检索授权信息。

授权信息。与访问令牌不同，刷新令牌（Refresh Token）仅用于与授权服务器的交互，而绝不会被发送至资源服务器。

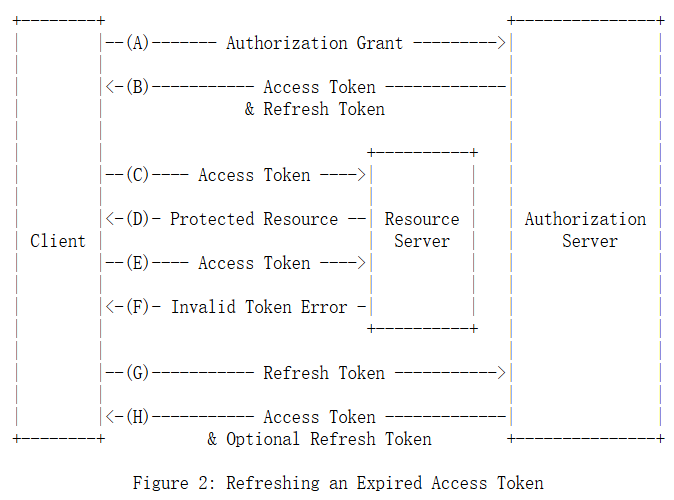


图2所示的流程包含以下步骤：

(A) 客户端通过向授权服务器进行身份验证并提交授权许可，请求获取访问令牌。

(B) 授权服务器对客户端进行身份验证并校验授权许可的有效性，若验证通过，则颁发访问令牌和刷新令牌。

(C) 客户端通过提交访问令牌，向资源服务器发起受保护资源请求。

(D) 资源服务器验证访问令牌的有效性，若验证通过，则响应请求。

(E) 步骤(C)和(D)会重复执行，直到访问令牌过期。若客户端已知访问令牌已过期，则直接跳转至步骤(G)；否则将继续发起新的受保护资源请求。

1. 由于访问令牌已失效，资源服务器返回令牌无效错误。

(G) 客户端通过向授权服务器进行身份验证并提交刷新令牌，请求获取新的访问令牌。客户端身份验证的具体要求取决于客户端类型及授权服务器的策略。

(H) 授权服务器对客户端进行身份验证并校验刷新令牌的有效性，若验证通过，则颁发新的访问令牌（以及可选的新的刷新令牌）。

步骤(C)、(D)、(E)和(F)不在本规范范围内，如第7节所述。

### 1.6 TLS版本

当本规范使用传输层安全协议(TLS)时，具体采用的TLS版本将随时间变化，主要基于广泛部署情况和已知安全漏洞。在撰写本文时，TLS 1.2版本[RFC5246]是最新版本，但部署基础非常有限，可能不易实现。TLS 1.0版本[RFC2246]是部署最广泛的版本，可提供最佳的互操作性。

实现方案也可以支持其他符合其安全要求的传输层安全机制。

### 1.7 HTTP重定向

本规范大量使用HTTP重定向技术，即客户端或授权服务器将资源所有者的用户代理引导至其他目标地址。虽然本规范示例中使用的是HTTP 302状态码，但允许使用用户代理支持的任何其他方法来实现重定向，这些方法被视为实现细节。

### 1.8 互操作性

OAuth 2.0提供了一个具有明确定义安全特性的丰富授权框架。然而，作为一个高度可扩展且包含许多可选组件的框架，仅凭本规范很可能会导致大量无法互通的实现。

此外，本规范对部分必要组件（如客户端注册、授权服务器能力、端点发现等）的定义不完全或完全未定义。若缺乏这些组件，客户端必须针对特定的授权服务器和资源服务器进行手动专门配置才能实现互通。

本框架在设计时明确预期，未来工作将定义必要的规范性配置文件和扩展，以实现完整的网络规模互操作性。

### 1.9 符号约定

本规范中的关键词"MUST"、"MUST NOT"、"REQUIRED"、"SHALL"、"SHALL NOT"、"SHOULD"、"SHOULD NOT"、"RECOMMENDED"、"MAY"和"OPTIONAL"应按照[RFC2119]中的描述进行解释。

本规范使用[RFC5234]的扩展巴科斯范式(ABNF)表示法。此外，URI-reference规则引用自"统一资源标识符(URI):通用语法"[RFC3986]。

某些安全相关术语应按照[RFC4949]中的定义理解。这些术语包括但不限于："attack"(攻击)、"authentication"(认证)、"authorization"(授权)、"certificate"(证书)、"confidentiality"(机密性)、"credential"(凭证)、"encryption"(加密)、"identity"(身份)、"sign"(签名)、"signature"(签名)、"trust"(信任)、"validate"(验证)和"verify"(校验)。

除非另有说明，所有协议参数名称和值都区分大小写。

### 2. 客户端注册

在启动协议之前，客户端需要在授权服务器注册。客户端与授权服务器之间的注册方式不在本规范范围内，但通常涉及终端用户通过HTML注册表单进行交互。

客户端注册不要求客户端与授权服务器直接交互。当授权服务器支持时，注册可以通过其他方式建立信任关系并获取所需的客户端属性（如重定向URI、客户端类型）。例如，可以通过自签发或第三方签发的断言完成注册，或由授权服务器通过可信通道执行客户端发现。

在注册客户端时，开发者必须：

1. 按照第2.1节的规定指定客户端类型；

2. 按照第3.1.2节的要求提供客户端重定向URI；

3. 包含授权服务器要求的其他任何信息（如应用名称、网站、描述、徽标图像、法律条款的接受声明）。

（注：翻译严格遵循RFC规范用语，"SHALL"译为强制性的"必须"。使用中文数字序号替代原文的"o"符号，更符合中文技术文档规范。括号内的示例列表保持完整，确保技术细节无遗漏。专业术语如"redirection URIs"统一译为"重定向URI"。）

### 2.1 客户端类型

OAuth根据客户端与授权服务器安全认证的能力（即维护客户端凭证机密性的能力），定义了两类客户端：

\*\*保密型(confidential)\*\*

能够维护其凭证机密性的客户端（例如运行在安全服务器上且对客户端凭证有访问限制的实现），或能通过其他方式实现安全客户端认证的客户端。

\*\*公开型(public)\*\*

无法维护其凭证机密性的客户端（例如运行在资源所有者所用设备上的客户端，如已安装的原生应用或基于网页浏览器的应用），且无法通过任何其他方式实现安全客户端认证的客户端。

客户端类型的认定基于授权服务器对安全认证的定义及其可接受的客户端凭证暴露级别。授权服务器不应假设客户端类型。

一个客户端可由多个具有不同客户端类型和安全上下文的分布式组件组成（例如同时包含保密型服务器组件和公开型浏览器组件的分布式客户端）。若授权服务器不支持此类客户端或未提供相关注册指导，客户端应将每个组件注册为独立客户端。

### 客户端类型规范

本规范围绕以下客户端类型设计：

\*\*Web应用程序\*\*

Web应用是运行在Web服务器上的保密型客户端。资源所有者通过其设备上的用户代理(如浏览器)访问HTML用户界面。客户端凭证及颁发的访问令牌均存储在Web服务器上，资源所有者无法接触或获取。

\*\*基于用户代理的应用程序\*\*

此类应用属于公开型客户端，其代码从Web服务器下载并在资源所有者设备的用户代理(如浏览器)中执行。协议数据和凭证对资源所有者是可见且易获取的。由于运行在用户代理环境中，这类应用在请求授权时可无缝调用用户代理功能。

\*\*原生应用程序\*\*

原生应用是安装在资源所有者设备上的公开型客户端。协议数据和凭证对资源所有者是可访问的。假定应用内包含的所有客户端认证凭证都可能被提取。但动态颁发的凭证(如访问令牌或刷新令牌)可获得适当保护：至少能防范应用可能交互的恶意服务器，在某些平台上还能防范同一设备的其他应用。

### 2.2 客户端标识符

授权服务器为注册客户端颁发客户端标识符——代表客户端注册信息的唯一字符串。该标识符并非机密，会暴露给资源所有者，且禁止单独用于客户端认证。每个客户端标识符在授权服务器范围内具有唯一性。

本规范未定义客户端标识符的字符串长度。客户端应避免对标识符长度做任何假设。授权服务器应当在其文档中说明所颁发标识符的长度规格。

### 2.3 客户端认证

若客户端类型为保密型(confidential)，客户端与授权服务器需建立符合授权服务器安全要求的认证方式。授权服务器可接受任何满足其安全要求的客户端认证形式。

保密型客户端通常被颁发（或自行建立）一组用于授权服务器认证的凭证（如密码、公私钥对）。

授权服务器可为公开型(public)客户端建立认证方法，但不得依赖公开客户端的认证来识别客户端身份。

客户端在每次请求中不得使用超过一种认证方法。

#### 2.3.1 客户端密码

持有密码的客户端可采用[RFC2617]定义的HTTP基本认证方案进行认证。客户端标识符按附录B的"application/x-www-form-urlencoded"编码算法处理作为用户名，客户端密码以相同算法编码后作为密码。授权服务器必须支持通过HTTP基本认证方案对持有密码的客户端进行认证。或者，授权服务器可支持通过请求体传递以下参数：

\*\*client\_id\*\*

必需。客户端注册时获得的标识符（参见第2.2节）。

\*\*client\_secret\*\*

必需。客户端密钥。若密钥为空字符串，客户端可省略该参数。

### 2.3.1 客户端凭证传输规范（补充）

\*\*强烈不推荐\*\*在请求体中通过client\_id和client\_secret参数传递客户端凭证，该方式\*\*仅限\*\*于无法直接使用HTTP基本认证方案（或其他基于密码的HTTP认证方案）的客户端。这些参数必须\*\*严格通过请求体传输\*\*，\*\*严禁\*\*包含在URL中。

访问令牌刷新请求示例（第6节，换行仅用于排版）：

```http

POST /token HTTP/1.1

Host: server.example.com

Content-Type: application/x-www-form-urlencoded

grant\_type=refresh\_token&refresh\_token=tGzv3J0kF0XG5Qx2T1KWIA

&client\_id=s6BhdRkqt3&client\_secret=7Fjfp0ZBr1KtDRbnfVdmIw

授权服务器在使用密码认证发送请求时，\*\*必须\*\*按照第1.6节要求强制启用TLS协议。

由于此认证方法涉及密码凭证，授权服务器\*\*必须\*\*对相关接入端点实施防暴力破解攻击保护措施。

### 2.3.2 其他认证方法

授权服务器可支持符合其安全策略的任何HTTP认证方案。采用其他方法时，\*\*必须\*\*建立客户端标识符（注册记录）与认证方案的明确映射关系。

### 2.4 未注册客户端

本规范：

- 不排除未注册客户端的使用

- 但此类用法\*\*超出规范范围\*\*

- 需进行额外：

✓ 安全风险评估

✓ 互操作性影响分析

### 3. 协议端点 (什么是端点）

授权流程使用两类端点（HTTP资源）：

\*\*授权服务器端点\*\*

1. \*\*授权端点\*\*

客户端通过用户代理重定向从此获取资源所有者授权（authorization grant）

2. \*\*令牌端点\*\*

客户端将授权许可(grant) 兑换为访问令牌（通常需客户端认证）

\*\*客户端端点\*\*

3. \*\*重定向端点\*\*

授权服务器通过此端点（经资源所有者用户代理）向客户端返回授权凭证

注意：

- 非所有授权类型都需使用全部端点

- 扩展许可类型可定义额外端点

### 3.1 授权端点

\*\*核心功能\*\*

▸ 与资源所有者交互并获取授权许可

▸ 授权服务器\*\*必须\*\*先验证资源所有者身份

（认证方式如密码登录/会话Cookie等不在规范范围内）

\*\*端点配置\*\*

• 位置获取：通过服务文档提供（规范未定义获取方式）

• URI格式要求：

✓ 可包含`application/x-www-form-urlencoded`格式查询组件（附录B）

✓ 添加新参数时\*\*必须\*\*保留原始查询组件（[RFC3986]第3.4节）

✗ \*\*禁止\*\*包含片段组件

\*\*安全强制要求\*\*

1. \*\*必须启用TLS\*\*（第1.6节）

- 涉及用户认证

- 传输明文凭证（HTTP响应中）

2. \*\*必须支持HTTP GET方法\*\* [RFC2616]

3. 可选支持POST方法

参数处理规范

- \*\*空值参数\*\*：视为未发送（授权服务器必须忽略）

- \*\*未知参数\*\*：授权服务器必须忽略

- \*\*重复参数\*\*：请求/响应中\*\*禁止\*\*重复包含

### 3.1.1 响应类型 (response\_type)

\*\*功能\*\*：声明授权许可类型（授权端点专用参数）

\*\*强制要求\*\*：

✓ 必填参数

✓ 取值限定：

• `code` - 请求授权码（参见4.1.1节）

• `token` - 请求访问令牌（隐式许可，参见4.2.1节）

• 扩展值（需符合8.4节规范）

\*\*扩展类型特性\*\*：

▸ 支持空格分隔值列表（如 `a b`）

▸ 值顺序无关（`a b` ≡ `b a`）

▸ 复合类型语义由扩展规范定义

\*\*错误处理\*\*：

✗ 缺失参数 或 响应类型无效 → \*\*必须返回错误\*\*（遵循4.1.2.1节规范）

### 3.1.2 重定向端点

\*\*运行机制\*\*：

1. 授权服务器完成资源所有者交互后

2. 通过用户代理重定向至客户端预设端点

3. 端点来源：

• 客户端注册时提供

• 授权请求时声明

\*\*URI格式强制要求\*\*：

✓ \*\*绝对URI\*\*（符合[RFC3986]第4.3节）

✓ 可含`application/x-www-form-urlencoded`格式查询组件（附录B）

✓ 添加参数时\*\*必须保留\*\*原始查询组件（[RFC3986]第3.4节）

✗ \*\*禁止\*\*包含片段组件

### 3.1.2.1 端点请求机密性

\*\*TLS启用规则\*\*：

✓ \*\*应当启用TLS\*\*（第1.6节）当：

• 请求的响应类型为 `code` 或 `token`

• 重定向请求将在开放网络传输敏感凭证

✗ \*\*未强制要求原因\*\*：

→ 强制客户端部署TLS对开发者门槛过高（撰写本文时现状）

\*\*无TLS的处置要求\*\*：

⚠️ 授权服务器\*\*应当\*\*在重定向前向资源所有者提示不安全端点风险（例如在授权请求界面显示警告）

\*\*安全影响警告\*\*：

❗ 缺乏传输层安全将严重危害：

- 客户端安全

- 被授权访问的受保护资源

❗ 特别关键场景：

→ 当授权流程用于客户端委托终端用户认证时（如第三方登录服务）

### 3.1.2.2 注册要求

\*\*强制注册范围\*\*：

▸ \*\*必须\*\*注册重定向端口的客户端类型：

1. 公开型客户端（Public clients）

2. 采用隐式许可的保密型客户端（Confidential clients using implicit grant）

\*\*最佳实践建议\*\*：

• 授权服务器\*\*应当\*\*要求所有客户端在使用授权端点前注册重定向端口

• \*\*应当\*\*要求提供完整重定向URI

（允许客户端使用`state`参数实现请求级定制）

\*\*变通注册方案\*\*：

→ 若无法注册完整URI，\*\*应当\*\*要求注册以下组件：

✓ URI协议方案

✓ 授权机构

✓ 路径

（允许客户端仅在查询参数动态变化）

\*\*扩展灵活性\*\*：

• 可允许注册多个重定向端点

### 3.1.2.3 动态配置

\*\*适用场景\*\*：

▸ 注册多个重定向URI

▸ 仅注册部分URI组件

▸ 未注册任何重定向URI

\*\*强制要求\*\*：

✓ 客户端\*\*必须\*\*在授权请求中包含 `redirect\_uri` 参数

\*\*URI验证规则\*\*：

1. 授权服务器\*\*必须\*\*将接收值与至少一个已注册URI（或组件）比对（[RFC3986]第6节）

2. 完整URI注册场景 → \*\*必须\*\*采用简单字符串比对（[RFC3986]第6.2.1节）

### 3.1.2.4 无效端点处理

\*\*错误类型\*\*：

• 重定向URI缺失

• 格式无效

• 与注册值不匹配

\*\*处置要求\*\*：

⚠️ \*\*应当\*\*向资源所有者提示错误

✗ \*\*禁止\*\*自动重定向至无效URI

### 3.1.2.5 端点内容安全

\*\*风险警示\*\*：

❗ 重定向响应返回的HTML文档 → 内含脚本可\*\*完全访问URI凭证\*\*

\*\*客户端防护措施\*\*：

✓ \*\*不应\*\*包含第三方脚本（如分析工具/社交插件/广告网络）

✓ \*\*应当\*\*执行两步重定向：

1. 提取凭证

2. 跳转至新端点（不暴露凭证）

✓ \*\*必须\*\*优先执行自有脚本的场景：

→ 若含第三方脚本 → 自有脚本\*\*必须\*\*先执行凭证提取/清除

### 3.2 令牌端点

\*\*核心功能\*\*：

• 客户端提交授权许可或刷新令牌

• 换取访问令牌

\*\*例外说明\*\*：

✗ \*\*不用于\*\*隐式许可类型（因其直接发放访问令牌）

### 令牌端点规范

\*\*位置获取\*\*：

• 不在规范范围内（通常由服务文档提供）

\*\*URI格式要求\*\*：

✓ 可含 `application/x-www-form-urlencoded` 格式查询组件（附录B）

✓ 添加参数时\*\*必须保留\*\*原始组件（[RFC3986]第3.4节）

✗ \*\*禁止\*\*包含片段组件

\*\*安全强制要求\*\*：

1. \*\*必须启用TLS\*\*（第1.6节）

→ 涉及明文凭证传输（HTTP请求/响应中）

2. \*\*必须使用HTTP POST方法\*\*

\*\*参数处理规则\*\*：

▸ 空值参数：视为未发送

▸ 未知参数：授权服务器必须忽略

▸ 重复参数：\*\*严禁\*\*出现

### 3.2.1 客户端认证

\*\*适用对象\*\*：

✓ 保密型客户端

✓ 持有凭证的其他客户端

\*\*认证要求\*\*：

• \*\*必须\*\*按2.3节规范进行认证

\*\*核心作用\*\*： 你

1. \*\*令牌绑定强制实施\*\*

✓ 确保刷新令牌/授权码与颁发客户端绑定

✓ 关键应用场景：

- 授权码经不安全通道传输

- 重定向URI未完全注册

2. \*\*入侵恢复机制\*\*

✓ 通过禁用客户端或更新凭证遏制攻击

✓ 优势：

→ 单组凭证更新快于撤销全部刷新令牌

3. \*\*凭证轮换实践\*\*

✓ 符合定期凭证轮换的安全最佳实践

✓ 优势：

→ 单组客户端凭证轮换难度 << 刷新令牌全集轮换

### 令牌端点客户端标识

\*\*特殊场景处理\*\*：

✓ 客户端\*\*可（may）\*\*在请求中使用 `client\_id` 参数表明身份

\*\*强制要求\*\*（授权码许可类型）：

⚠️ 未认证客户端在令牌端点发起 `authorization\_code`、’grant\_type’ 请求时\*\*必须\*\*包含 `client\_id`

→ \*\*核心作用\*\*：防止接受其他客户端的授权码（抵御授权码替换攻击）

（注：此机制仅保护客户端安全，不增强受保护资源的安全性）

### 3.3 访问令牌范围

\*\*范围控制机制\*\*：

▸ \*\*请求端\*\*：客户端通过 `scope` 参数声明权限范围

▸ \*\*响应端\*\*：授权服务器通过 `scope` 参数返回实际授予范围

\*\*参数语法规范\*\*：

✓ 值格式：空格分隔的字符串列表

✓ 特性：

- 区分大小写

- 字符串顺序无关

- 每个字符串代表一个权限域

scope = scope-token \*( SP scope-token )  
scope-token = 1\*(%x21 / %x23-5B / %x5D-7E )

（注：允许可打印ASCII字符，排除空格/引号等控制字符）

\*\*授权服务器行为准则\*\*：

1. \*\*范围调整权限\*\*：

• 可基于策略或资源所有者指令\*\*调整请求范围\*\*

2. \*\*范围变化通知\*\*：

✓ 授予范围≠请求范围 → \*\*必须\*\*返回 `scope` 参数说明实际范围

3. \*\*缺省范围处理\*\*：

• 客户端未提供scope参数时\*\*必须\*\*：

✓ 使用预定义默认值处理

或

✗ 返回无效范围错误

• \*\*应当\*\*在文档中说明范围要求和默认值

### 4. 获取授权

\*\*授权流程本质\*\*：

客户端通过资源所有者的授权获得访问令牌，授权以\*\*授权许可\*\*形式体现

\*\*四大核心许可类型\*\*：

1. 授权码（authorization code）

2. 隐式许可（implicit）

3. 资源所有者密码凭证（resource owner password credentials）

4. 客户端凭证（client credentials）

\*\*扩展机制\*\*：

• 支持通过扩展机制定义新许可类型

### 4.1 授权码许可

\*\*核心用途\*\*：

• 获取访问令牌（access tokens）

• 获取刷新令牌（refresh tokens）

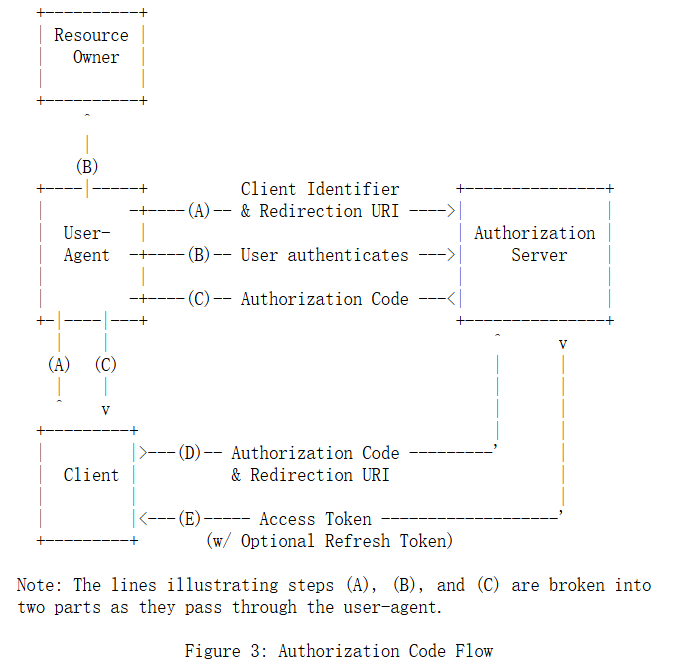
\*\*设计特性\*\*：

✓ \*\*专为保密型客户端（confidential clients）优化\*\*

✓ \*\*基于重定向的流程\*\* → 要求客户端：

1. 能与资源所有者的用户代理（通常为浏览器）交互

2. 能接收来自授权服务器的重定向请求



**### 图3所示流程步骤：**

\*\*(A) 客户端发起流程\*\*

✓ 引导资源所有者的用户代理访问授权端点

✓ 包含参数：

• `client\_id`（客户端标识符）

• 请求范围（scope）

• 本地状态（state）

• 重定向URI（授权后回调地址）

\*\*(B) 授权服务器执行\*\*

✓ 通过用户代理认证资源所有者身份

✓ 确认资源所有者批准/拒绝访问请求

\*\*(C) 批准访问时的响应\*\*

✓ 使用预设重定向URI回传用户代理

✓ 响应包含：

• 授权码（authorization code）

• 客户端提供的本地状态（state）

\*\*(D) 客户端令牌请求\*\*

✓ 向令牌端点提交前步获得的授权码

✓ \*\*强制操作\*\*：

1. 向授权服务器进行身份验证

2. 包含用于验证的重定向URI（需与步骤C一致）

\*\*(E) 授权服务器响应\*\*

✓ 身份验证客户端

✓ 校验授权码有效性

✓ \*\*关键验证\*\*：重定向URI ≡ 步骤(C)所用URI

✓ 验证通过则返回：

• 访问令牌（access token）

• 可选的刷新令牌（refresh token）

### 4.1.1 授权请求参数

\*\*请求URI构造规则\*\*：

▸ 按附录B格式在授权端点URI后添加查询参数

\*\*必需参数\*\*：

\*\*response\_type\*\*

值\*\*必须\*\*为 `code`

\*\*client\_id\*\*

客户端标识符（遵循第2.2节定义）

\*\*可选参数\*\*：

\*\*redirect\_uri\*\*

重定向URI（遵循第3.1.2节规范）

### 4.1.1 授权请求参数（续）

\*\*scope\*\*

可选。访问请求范围（遵循第3.3节规范）

\*\*state\*\*

推荐。客户端用于维持请求状态的随机值，授权服务器将在重定向时原样返回。

\*\*安全作用\*\*：

✓ 防范跨站请求伪造（CSRF）攻击（第10.12节）

### 客户端引导流程

▸ 通过HTTP重定向响应

▸ 或用户代理支持的其他方式

▸ 引导资源所有者访问构造的URI

\*\*TLS请求示例\*\*（换行仅用于排版）：

```http

GET /authorize?response\_type=code&client\_id=s6BhdRkqt3&state=xyz

&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb HTTP/1.1

Host: server.example.com

授权服务器处理流程

请求验证：

✓ 检查必需参数

✓ 验证参数有效性

认证与授权：

✓ 认证资源所有者身份

✓ 获取授权决策（直接询问或其他批准机制）

决策执行

▸ 通过HTTP重定向响应

▸ 或用户代理支持的其他方式

▸ 引导至客户端重定向URI

4.1.2 授权响应

资源所有者批准时 → 授权服务器：

生成授权码（authorization code）

按附录B格式添加到重定向URI查询组件

响应参数：

code

必需。授权服务器生成的授权码。

安全要求：

✓ 必须短期有效（降低泄露风险）

✓ 建议最长生命周期10分钟

✗ 客户端禁止重复使用

（注：关键技术要点：

state参数安全作用关联RFC 6819的CSRF防护规范

URI中的%3A%2F%2F等编码保留原文体现URL转义规则

授权码安全要求三重强调：时效性/最佳实践/一次性原则

决策获取机制明确两种方式：交互式询问或预设批准流程）

### 授权响应参数（续）

\*\*state\*\*

▸ 当客户端请求包含 `state` 参数时\*\*必需\*\*

▸ 值必须与客户端请求值\*\*完全一致\*\*

\*\*授权码安全约束\*\*：

✗ \*\*严禁重复使用\*\*

✓ 检测到重用 → 授权服务器\*\*必须\*\*拒绝请求

✓ \*\*应当\*\*撤销此前基于该授权码颁发的所有令牌

✓ 授权码绑定关系：

• 客户端标识符

• 重定向URI

\*\*响应示例\*\*：

```http

HTTP/1.1 302 Found

Location: <https://client.example.com/cb?code=Sp1x10BeZQQYbYS6WxsbIA&state=xyz>

客户端处理规则

▸ 必须忽略未知响应参数

▸ 禁止假设授权码长度（规范未定义）

▸ 授权服务器应当在文档中说明其授权码长度

4.1.2.1 错误响应

特定错误处理（满足以下条件时）：

• 重定向URI缺失/无效/不匹配

• 客户端标识符缺失/无效

处置要求：

⚠️ 应当向资源所有者提示错误

✗ 禁止自动重定向至无效URI

其他错误场景（如资源所有者拒绝访问）：

▸ 通过重定向URI返回错误参数（格式遵循附录B）

错误参数：

error

必需。ASCII错误代码，可选值：

invalid\_request：请求存在以下问题之一 →

✓ 缺少必需参数

✓ 包含无效参数值

✓ 参数重复出现

✓ 格式错误

（注：关键技术要点：

安全约束三级递进：

重用禁令（严禁）

即时处置（必须拒绝）

后续清理（应当撤销令牌）

错误类型invalid\_request的四项判定标准明确列出

HTTP 302重定向示例保留原始协议格式

参数绑定关系强调双重绑定（客户端ID+URI））

### 错误代码详解

\*\*error 参数值\*\*（必需，ASCII错误码）：

`unauthorized\_client`：

▸ 客户端无权使用此方法请求授权码

`access\_denied`：

▸ 资源所有者或授权服务器已拒绝请求

`unsupported\_response\_type`：

▸ 授权服务器不支持此方法获取授权码

`invalid\_scope`：

▸ 请求的范围无效/未知/格式错误

`server\_error`：

▸ 授权服务器遇到意外故障无法完成请求

（注：需此错误码因HTTP 500状态码无法通过重定向返回）

`temporarily\_unavailable`：

▸ 服务器过载或维护导致暂时无法处理

（注：需此错误码因HTTP 503状态码无法通过重定向返回）

### 参数格式强制要求

\*\*error 值字符集\*\*：

✗ \*\*禁止\*\*超出：%x20-21 / %x23-5B / %x5D-7E

（可打印ASCII字符，排除双引号/反斜杠等）

### 扩展错误参数

\*\*error\_description\*\*

可选。人类可读的ASCII文本，帮助开发者理解错误

\*\*字符集限制\*\*：

✗ \*\*禁止\*\*超出：%x20-21 / %x23-5B / %x5D-7E

\*\*error\_uri\*\*

可选。指向错误说明网页的URI

\*\*格式要求\*\*：

✓ \*\*必须\*\*符合URI引用语法

✗ \*\*禁止\*\*包含：%x21 / %x23-5B / %x5D-7E 外字符

（注：允许除空格/双引号/尖括号外可打印字符）

（注：关键技术要点：

1. 错误代码翻译保留英文原名+中文解释双轨制

2. 特殊说明强调错误码存在原因（HTTP状态码重定向限制）

3. 字符集范围用十六进制精确标注

4. URI语法要求关联RFC 3986标准

5. 错误描述文本定位为"开发者辅助"工具）

state

必需，如客户端授权请求中已包含 “state” 参数。必须为从客户端接收到的原始值。

例如，授权服务器通过发送以下 HTTP 响应重定向用户代理：

HTTP/1.1 302 Found

Location: https://client.example.com/cb?error=access\_denied&state=xyz

**4.1.3. 访问令牌请求 (Access Token Request)**

客户端通过使用 “application/x-www-form-urlencoded” 格式（按附录 B）发送以下参数（字符编码为 UTF-8）到令牌端点，以在 HTTP 请求实体正文中发起请求：

grant\_type

必需。值必须设置为 "authorization\_code"。

code

必需。从授权服务器接收到的授权码。

redirect\_uri

必需，如授权请求中已按 4.1.1 节所述包含 “redirect\_uri” 参数，则此处的值必须与其完全相同。

client\_id

必需，如客户端未按 3.2.1 节所述向授权服务器进行身份验证。

如客户端类型为机密客户端，或客户端被颁发了客户端凭据（或分配了其他身份验证要求），客户端必须按 3.2.1 节所述向授权服务器进行身份验证。

例如，客户端使用 TLS 发起以下 HTTP 请求（额外换行仅用于显示）：

POST /token HTTP/1.1

Host: server.example.com

Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Content-Type: application/x-www-form-urlencoded

grant\_type=authorization\_code&code=Sp1x10BeZQQYbYS6WxSbIA

&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb

**授权服务器必须：**

**o 对于机密客户端，或者对于任何被颁发了客户端凭据（或其他身份验证要求）的客户端，要求进行客户端认证；**

**o 如果包含了客户端身份验证信息，则对客户端进行身份验证；**

**o 确保授权码是颁发给经过认证的机密客户端的，或者如果客户端是公共客户端，则确保该码是颁发给请求中的“client\_id”的；**

**o 验证授权码是否有效；并且**

**o 如果在如 4.1.1 节所述的初始授权请求中包含了“redirect\_uri”参数，则确保此处“redirect\_uri”参数存在，并且如果包含了该参数，则确保其值与初始请求中的值完全相同。**

**4.1.4. 访问令牌响应 (Access Token Response**)

如果访问令牌请求有效且已获授权，授权服务器将颁发一个访问令牌以及可选的刷新令牌，具体如 5.1 节所述。如果请求的客户端身份验证失败或无效，授权服务器将返回错误响应，具体如 5.2 节所述。

一个成功响应的示例：

HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8

Cache-Control: no-store

Pragma: no-cache

{

"access\_token":"2YotnFZFEjr1zCsicMWpAA",

"token\_type":"example",

"expires\_in":3600,

"refresh\_token":"tGzv3J0kFOXG5Qx2T1KWIA",

"example\_parameter":"example\_value"

}

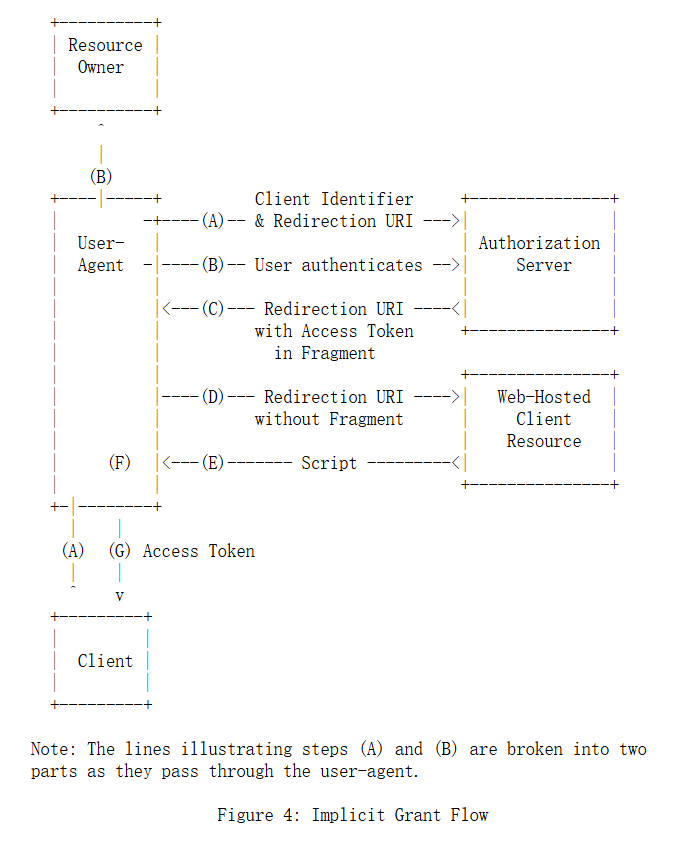
4.2. 隐式授权 (Implicit Grant)

隐式授权类型用于获取访问令牌（它不支持颁发刷新令牌），并针对已知运行于特定重定向URI的公共客户端进行了优化。这些客户端通常使用诸如JavaScript之类的脚本语言在浏览器中实现。

由于这是一个基于重定向的流程，客户端必须能够与资源所有者的用户代理（通常是网页浏览器）进行交互，并且能够接收来自授权服务器的传入请求（通过重定向）。

与授权码授权类型（其中客户端分别发出授权请求和访问令牌请求）不同，在隐式授权类型中，客户端作为授权请求的结果直接接收访问令牌。

隐式授权类型不包含客户端身份验证，而是依赖于资源所有者的在场和重定向URI的注册。由于访问令牌被编码在重定向URI中，它可能会暴露给资源所有者以及驻留在同一设备上的其他应用程序。



注：描绘步骤 (A) 和 (B) 的线条在穿过用户代理 (user-agent) 时断成两部分。

图 4：隐式授权流 (Implicit Grant Flow)

图 4 所描绘的流程包括以下步骤：

(A) 客户端通过将资源所有者的用户代理 (user-agent) 定向到授权端点 (authorization endpoint) 来发起流程。客户端包含其客户端标识符 (client identifier)、请求的作用域 (scope)、本地状态 (local state)、以及一个重定向 URI (redirection URI)，授权服务器将在访问被授予（或拒绝）后将用户代理发送回该 URI。

(B) 授权服务器对资源所有者（通过用户代理）进行身份验证 (authenticates)，并确认资源所有者是批准还是拒绝客户端的访问请求。

(C) 假设资源所有者批准访问，授权服务器使用之前提供的重定向 URI 将用户代理重定向回客户端。该重定向 URI 在 URI 片段 (URI fragment) 中包含了访问令牌 (access token)。

(D) 用户代理按照重定向指示，向托管在 Web 上的客户端资源 (web-hosted client resource) 发出请求（根据 [RFC2616]，此请求不包含片段 (fragment)）。用户代理在本地保留片段信息。

(E) 托管在 Web 上的客户端资源返回一个网页（通常是一个包含嵌入式脚本的 HTML 文档），该网页能够访问完整的重定向 URI（包括用户代理保留的片段），并能提取片段中包含的访问令牌（及其他参数）。

(F) 用户代理在本地执行托管在 Web 上的客户端资源提供的脚本，该脚本提取访问令牌。

(G) 用户代理将访问令牌传递给客户端。

有关使用隐式授权的背景信息，请参见第 1.3.2 节和第 9 节。关于使用隐式授权时的重要安全考量，请参见第 10.3 节和第 10.16 节。

4.2.1. 授权请求 (Authorization Request)

客户端通过向授权端点 URI 的查询组件 (query component) 添加以下参数（使用 "application/x-www-form-urlencoded" 格式，依据附录 B）来构建请求 URI：

response\_type

必需。值必须设为 "token"。

client\_id

必需。如第 2.2 节所述的客户端标识符。

redirect\_uri

可选 (OPTIONAL)。如第 3.1.2 节所述。

scope

可选 (OPTIONAL)。如第 3.3 节所述的访问请求的作用域 (scope)。

state

推荐 (RECOMMENDED)。一个由客户端使用的不透明值 (opaque value)，用于在请求和回调 (callback) 之间维护状态。当授权服务器将用户代理 (user-agent) 重定向回客户端时，会包含此值。该参数应该 (SHOULD) 用于防止跨站请求伪造 (cross-site request forgery)，如第 10.12 节所述。

客户端使用 HTTP 重定向响应 (HTTP redirection response)，或通过用户代理可用的其他方式，引导资源所有者 (resource owner) 访问构造好的 URI。

例如，客户端引导用户代理使用 TLS 发出以下 HTTP 请求（额外换行仅用于显示目的）：

授权服务器验证请求以确保所有必需参数都存在且有效。授权服务器必须 (MUST) 验证其将要重定向访问令牌的目标重定向 URI，与该客户端注册的重定向 URI（如第 3.1.2 节所述）相匹配。

如果请求有效，授权服务器则对资源所有者进行身份验证 (authenticates the resource owner) 并获得授权决策 (authorization decision)（通过询问资源所有者或通过其他方式建立许可）。

当决策确立后，授权服务器使用 HTTP 重定向响应，或通过用户代理可用的其他方式，引导用户代理前往提供的客户端重定向 URI。

4.2.2. 访问令牌响应 (Access Token Response)

如果资源所有者批准了访问请求，授权服务器将颁发一个访问令牌，并通过使用“application/x-www-form-urlencoded”格式（按照附录B）将以下参数添加到重定向URI的片段组件 (fragment component) 中，将其交付给客户端：

access\_token

必需 (REQUIRED)。授权服务器颁发的访问令牌。

token\_type

必需 (REQUIRED)。所颁发令牌的类型，如第7.1节所述。其值不区分大小写 (case insensitive)。

expires\_in

推荐 (RECOMMENDED)。访问令牌的生命周期（以秒为单位）。例如，值 "3600" 表示该访问令牌将在响应生成后一小时过期。如果省略，授权服务器应 (SHOULD) 通过其他方式提供过期时间或记录默认值。

scope

可选 (OPTIONAL)，如果与客户端请求的作用域相同；否则为必需 (REQUIRED)。如第 3.3 节所述的访问令牌的作用域。

state

必需 (REQUIRED)，如果客户端授权请求中曾包含 "state" 参数。必须为从客户端接收到的原始值 (exact value)。

授权服务器禁止 (MUST NOT) 颁发刷新令牌。

例如，授权服务器通过发送以下 HTTP 响应（额外的换行仅用于显示）来重定向用户代理：

HTTP/1.1 302 Found  
Location: <http://example.com/cb#access_token=2YotnFZFEjr1zCsicMWpAA&state=xyz&token_type=example&expires_in=3600>

\*\*开发者需要注意：\*\* 有些用户代理 (user-agents) 不支持在 HTTP "Location" 响应头字段中包含片段组件 (fragment component)。对于此类客户端，将需要（MUST）使用 3xx 重定向响应以外的方法来重定向客户端——例如，返回一个包含链接到该重定向URI的 '继续'(continue) 按钮的 HTML 页面。

客户端必须 (MUST) 忽略无法识别的响应参数。访问令牌字符串的大小在本规范中未作定义。客户端应避免对值的大小做任何假设。授权服务器应该 (SHOULD) 记录其颁发的任何值的大小。

4.2.2.1. 错误响应 (Error Response)

如果请求失败是由于重定向 URI (redirection URI) 缺失、无效或不匹配，或者由于客户端标识符缺失或无效，授权服务器应 (SHOULD) 告知资源所有者错误情况，并且禁止 (MUST NOT) 自动将用户代理 (user-agent) 重定向至无效的重定向 URI。

如果资源所有者拒绝了访问请求，或者请求因重定向 URI 缺失或无效以外的原因失败，授权服务器则通过使用 "application/x-www-form-urlencoded" 格式（依据附录 B），将以下参数添加到重定向 URI 的片段组件 (fragment component) 中来通知客户端：

error

必需 (REQUIRED)。一个单一的 ASCII [USASCII] 错误码，取值于以下列表：

invalid\_request

请求缺失必要参数、包含无效参数值、重复包含某个参数，或其他形式的结构错误。

unauthorized\_client

客户端未被授权使用此方法请求访问令牌。

access\_denied

资源所有者或授权服务器拒绝了该请求。

unsupported\_response\_type

授权服务器不支持使用此方法获取访问令牌。

invalid\_scope

请求的作用域无效、未知或格式错误。

server\_error

授权服务器遇到意外情况，导致其无法完成请求。（需要此错误代码是因为无法通过 HTTP 重定向向客户端返回 500 Internal Server Error HTTP 状态码。）

temporarily\_unavailable

授权服务器由于临时过载或维护，当前无法处理请求。（需要此错误代码是因为无法通过 HTTP 重定向向客户端返回 503 Service Unavailable HTTP 状态码。）

"error" 参数的值必须不包含 (MUST NOT include) 除 %x20-21 / %x23-5B / %x5D-7E 字符集以外的任何字符。

error\_description

可选 (OPTIONAL)。提供额外信息的、人类可读的 ASCII [USASCII] 文本，用于帮助客户端开发者理解所发生的错误。

"error\_description" 参数的值必须不包含 (MUST NOT include) 除 %x20-21 / %x23-5B / %x5D-7E 字符集以外的任何字符。

error\_uri

可选 (OPTIONAL)。标识包含错误信息的人类可读网页的 URI，用于向客户端开发者提供关于错误的额外信息。

"error\_uri" 参数的值必须符合 (MUST conform to) URI-reference 语法，因此必须不包含 (MUST NOT include) 除 %x21 / %x23-5B / %x5D-7E 字符集以外的任何字符。

state

必需 (REQUIRED)，如果客户端授权请求中曾包含 "state" 参数。必须为从客户端接收到的原始值 (exact value)。

例如，授权服务器通过发送以下 HTTP 响应来重定向用户代理：

HTTP/1.1 302 Found

Location: https://client.example.com/cb#error=access\_denied&state=xyz

4.3. 资源所有者密码凭据授权 (Resource Owner Password Credentials Grant)

资源所有者密码凭据授权类型适用于资源所有者与客户端之间存在信任关系的情况，例如设备操作系统或高权限应用程序等。授权服务器在启用此授权类型时需特别审慎，仅应在其他授权流程均不适用时允许此类型。

此授权类型适用于能够获取资源所有者凭据（通常通过交互式表单获取用户名和密码）的客户端。它也用于将使用直接认证方案（如 HTTP Basic 或 Digest 认证）的现有客户端迁移至 OAuth，通过将存储的凭据转换为访问令牌来实现。

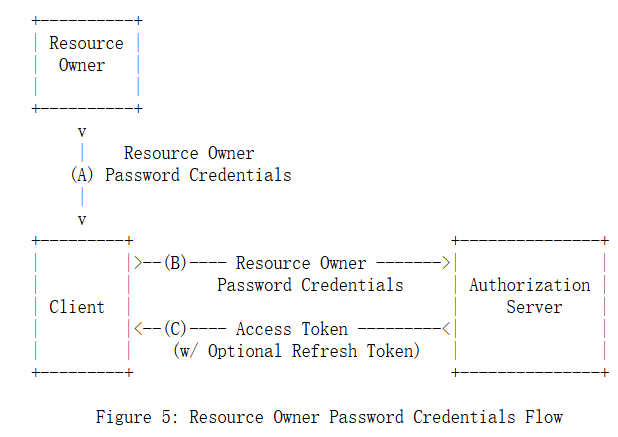


图 5：资源所有者密码凭据授权流 (Resource Owner Password Credentials Flow)

图 5 描绘的流程包括以下步骤：

(A) 资源所有者向客户端提供其用户名和密码。

(B) 客户端通过向授权服务器的令牌端点发出请求来请求访问令牌，该请求中包含了从资源所有者处接收到的凭据。在发出请求时，客户端需向授权服务器进行身份验证。

(C) 授权服务器对客户端进行身份验证，并验证资源所有者凭据的有效性；如验证成功，则颁发访问令牌（可选包含刷新令牌）。

4.3.1. 授权请求与响应 (Authorization Request and Response)

客户端获取资源所有者凭据的方法超出本规范的范畴。客户端必须 (MUST) 在获取访问令牌后立即丢弃 (discard) 这些凭据。

4.3.2. 访问令牌请求 (Access Token Request)

客户端通过使用 "application/x-www-form-urlencoded" 格式（依据附录 B），以 UTF-8 字符编码将以下参数添加到 HTTP 请求的实体主体 (entity-body) 中，向令牌端点发出请求：

grant\_type

必需 (REQUIRED)。值必须 (MUST) 设为 "password"。

username

必需 (REQUIRED)。资源所有者的用户名 (resource owner username)。

password

必需 (REQUIRED)。资源所有者的密码 (resource owner password)。

scope

可选 (OPTIONAL)。如第 3.3 节所述的访问请求的作用域 (scope)。

如果客户端类型为机密客户端 (confidential)，或者该客户端被颁发了客户端凭据（或被分配了其他身份验证要求），客户端则必须 (MUST) 按第 3.2.1 节所述向授权服务器进行身份验证。

例如，客户端使用传输层安全 (transport-layer security) 发出以下 HTTP 请求（额外换行仅用于显示目的）：

POST /token HTTP/1.1  
Host: server.example.com  
Authorization: Basic czZCaGRSa3FOMzpnWDFmQmFOM2JW  
Content-Type: application/x-www-form-urlencoded

grant\_type=password&username=johndoe&password=A3ddj3w

授权服务器必须 (MUST)：

o 对于机密客户端 (confidential clients)，或者对于任何被颁发了客户端凭据（或具备其他身份验证要求）的客户端，要求进行客户端身份验证；

o 如果包含了客户端身份验证信息，则对客户端进行身份验证；并且

o 使用其现有的密码验证算法 (existing password validation algorithm) 来验证资源所有者密码凭据 (resource owner password credentials)。

由于此访问令牌请求涉及使用资源所有者的密码，授权服务器必须 (MUST) 保护该端点 (endpoint) 免受暴力攻击 (brute force attacks)（例如，采用速率限制 (rate-limitation) 或生成警报）。

4.3.3. 访问令牌响应 (Access Token Response)

如果访问令牌请求有效且已获授权 (valid and authorized)，授权服务器将颁发一个访问令牌以及可选的刷新令牌，具体如第 5.1 节所述。如果请求的客户端身份验证失败或无效，授权服务器将返回错误响应，具体如第 5.2 节所述。

一个成功响应的示例：

HTTP/1.1 200 OK

Content-Type: application/json;charset=UTF-8

Cache-Control: no-store

Pragma: no-cache

{

"access\_token":"2YotnFZFEjr1zCsicMWpAA",

"token\_type":"example",

"expires\_in":3600,

"refresh\_token":"tGzv3J0kF0XG5Qx2TlKWIA",

"example\_parameter":"example\_value"

}

4.4. 客户端凭据授权 (Client Credentials Grant)

当客户端请求访问其控制下的受保护资源，或者请求访问之前已与授权服务器安排好的、属于另一资源所有者的受保护资源时（具体安排方式超出本规范范围），客户端可以仅使用其客户端凭据（或其他支持的认证方式）来请求访问令牌。

客户端凭据授权类型 (client credentials grant type) 必须仅由 (MUST only be used by) 机密客户端 (confidential clients) 使用。

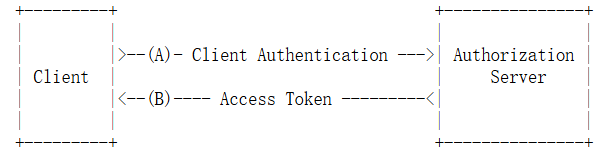


图 6：客户端凭据授权流 (Client Credentials Grant Flow)

图 6 描绘的流程包括以下步骤：

(A) 客户端向授权服务器进行身份验证，并向令牌端点 (token endpoint) 请求访问令牌。

(B) 授权服务器对客户端进行身份验证，如验证有效，则颁发访问令牌。

4.4.1. 授权请求与响应 (Authorization Request and Response)

由于客户端身份验证 (client authentication) 本身即被用作授权授予 (authorization grant)，因此不需要额外的授权请求。

4.4.2. 访问令牌请求 (Access Token Request)

客户端通过使用 "application/x-www-form-urlencoded" 格式（依据附录 B），以 UTF-8 字符编码将以下参数添加到 HTTP 请求的实体主体 (entity-body) 中，向令牌端点 (token endpoint) 发出请求：

grant\_type

必需 (REQUIRED)。值必须 (MUST) 设为 "client\_credentials"。

scope

可选 (OPTIONAL)。如第 3.3 节所述的访问请求的作用域 (scope)。

客户端必须 (MUST) 按照第 3.2.1 节所述，向授权服务器进行身份验证。

例如，客户端使用传输层安全 (transport-layer security) 发出以下 HTTP 请求（额外换行仅用于显示目的）：

POST /token HTTP/1.1  
Host: server.example.com  
Authorization: Basic czZCaGRSa3FOMzpnWDFmQmFOM2JW  
Content-Type: application/x-www-form-urlencoded

grant\_type=client\_credentials

授权服务器必须 (MUST) 对客户端进行身份验证。

4.4.3. 访问令牌响应 (Access Token Response)

如果访问令牌请求有效且已获授权 (valid and authorized)，授权服务器将按照第 5.1 节所述颁发访问令牌。响应中应不包含 (SHOULD NOT be included) 刷新令牌。如果请求的客户端身份验证失败或无效，授权服务器将按照第 5.2 节所述返回错误响应。

一个成功响应的示例：

HTTP/1.1 200 OK  
Content-Type: application/json;charset=UTF-8  
Cache-Control: no-store  
Pragma: no-cache

{  
"access\_token":"2YotnFZFEjr1zCsicMWpAA",  
"token\_type":"example",  
"expires\_in":3600,  
"example\_parameter":"example\_value"  
}

4.5. 扩展授权 (Extension Grants)

客户端通过以下方式使用扩展授权类型：

\* 在令牌端点的 "grant\_type" 参数中指定一个由授权服务器定义的绝对 URI (absolute URI) 作为其值；并且

\* 添加所需的任何额外参数 (adding any additional parameters necessary)。

例如，如[OAuth-SAML2]所定义，要使用安全断言标记语言 (Security Assertion Markup Language, SAML) 2.0 断言授权类型请求访问令牌，客户端可使用 TLS 发出以下 HTTP 请求（额外换行仅用于显示目的）：

POST /token HTTP/1.1  
Host: server.example.com  
Content-Type: application/x-www-form-urlencoded

grant\_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Asaml2-bearer&assertion=PEFzc2VydGlvbiBJc3N1ZUluc3RhbnQ9IjIwMTEtMDU[...]此处省略...]aG5TdGF0ZW1lbnQ-PC9Bc3NlcnRpb24-

如果访问令牌请求有效且已获授权 (valid and authorized)，授权服务器将按照第 5.1 节所述颁发访问令牌及可选的刷新令牌。如果请求的客户端身份验证失败或无效，授权服务器将按照第 5.2 节所述返回错误响应。

5. 颁发访问令牌 (Issuing an Access Token)

如果访问令牌请求有效且已获授权 (valid and authorized)，授权服务器将按照第 5.1 节所述颁发访问令牌及可选的刷新令牌。如果请求的客户端身份验证失败或无效，授权服务器将按照第 5.2 节所述返回错误响应。

5.1. 成功响应 (Successful Response)

授权服务器颁发访问令牌及可选的刷新令牌，并通过向 HTTP 响应（状态码为 200 (OK)）的实体主体 (entity-body) 中添加以下参数来构建响应：

access\_token

必需 (REQUIRED)。授权服务器颁发的访问令牌 (access token)。

token\_type

必需 (REQUIRED)。所颁发令牌的类型，如第 7.1 节所述。其值不区分大小写 (case insensitive)。

expires\_in

推荐 (RECOMMENDED)。访问令牌的生命周期（以秒为单位）。例如，值 "3600" 表示该访问令牌将在响应生成后一小时过期。如果省略，授权服务器应 (SHOULD) 通过其他方式提供过期时间或记录默认值。

refresh\_token

可选 (OPTIONAL)。刷新令牌 (refresh token)，可用于按照第 6 节所述，使用相同的授权授予 (authorization grant) 来获取新的访问令牌 (access tokens)。

scope

可选 (OPTIONAL)，如果与客户端请求的作用域相同；否则为必需 (REQUIRED)。如第 3.3 节所述的访问令牌的作用域 (scope of the access token)。

这些参数被包含在 HTTP 响应的实体主体 (entity-body) 中，使用 [RFC4627] 定义的 "application/json" 媒体类型。参数通过将每个参数添加到最高结构级别（顶级）序列化为 JavaScript 对象表示法 (JavaScript Object Notation, JSON) 结构。参数名称 (Parameter names) 和字符串值 (string values) 作为 JSON 字符串 (JSON strings) 包含。数值 (Numerical values) 作为 JSON 数字 (JSON numbers) 包含。参数的顺序无关紧要且可以变化。

授权服务器必须在 (MUST include) 任何包含令牌 (tokens)、凭据 (credentials) 或其他敏感信息 (sensitive information) 的响应中，包含值为 "no-store" 的 HTTP "Cache-Control" 响应头字段 [RFC2616]，以及值为 "no-cache" 的 "Pragma" 响应头字段 [RFC2616]。

例如：

HTTP/1.1 200 OK  
Content-Type: application/json;charset=UTF-8  
Cache-Control: no-store  
Pragma: no-cache

{  
"access\_token":"2YotnFZFEjr1zCsicMWpAA",  
"token\_type":"example",  
"expires\_in":3600,  
"refresh\_token":"tGzv3J0kFOXG5Qx2TlKWIA",  
"example\_parameter":"example\_value"  
}

客户端必须 (MUST) 忽略响应中无法识别的值名称 (unrecognized value names)。从授权服务器接收到的令牌 (tokens) 和其他值的大小未被定义 (left undefined)。客户端应避免 (should avoid) 对值的大小做出假设。授权服务器应该 (SHOULD) 记录 (document) 其颁发的任何值的大小。

5.2. 错误响应 (Error Response)

授权服务器将使用 HTTP 400（错误请求）状态码进行响应（除非另有规定），并在响应中包含以下参数：

error

必需 (REQUIRED)。一个来自下列清单的 ASCII [USASCII] 错误码：

invalid\_request

请求缺失必要参数、包含了不支持的参数值（不包括授权类型参数）、重复包含某个参数、包含了多种凭据、使用了超过一种认证客户端机制 (mechanism for authenticating the client)，或者存在其他形式的结构错误。

invalid\_client

客户端身份验证失败（例如，未知客户端、未包含客户端身份验证信息，或使用了不受支持的身份验证方法）。授权服务器\*可\*(MAY) 返回 HTTP 401（未授权）状态码来表明其支持哪些 HTTP 认证方案 (HTTP authentication schemes)。如果客户端试图通过“Authorization”请求头字段进行身份验证，则授权服务器\*必须\*(MUST) 以 HTTP 401（未授权）状态码响应，并在“WWW-Authenticate”响应头字段中包含与客户端使用的认证方案相匹配的信息。

invalid\_grant

提供的授权授予（例如，授权码、资源所有者凭据）或刷新令牌 (refresh token) 无效、已过期、已被撤销、与授权请求中使用的重定向 URI 不匹配，或是颁发给了其他客户端。

unauthorized\_client

经过身份验证的客户端无权使用此授权授予类型。

unsupported\_grant\_type

该授权授予类型不受授权服务器支持。

invalid\_scope

请求的作用域（scope）无效、未知、格式错误，或超出了资源所有者授予的范围。

"error" 参数的值必须不包含 (MUST NOT include) 除 %x20-21 / %x23-5B / %x5D-7E 字符集以外的任何字符。

error\_description

可选 (OPTIONAL)。提供额外信息的、人类可读的 ASCII [USASCII] 文本，用于帮助客户端开发者理解所发生的错误。

"error\_description" 参数的值必须不包含 (MUST NOT include) 除 %x20-21 / %x23-5B / %x5D-7E 字符集以外的任何字符。

error\_uri

可选 (OPTIONAL)。标识包含错误信息的人类可读网页的 URI，用于向客户端开发者提供关于错误的额外信息。

"error\_uri" 参数的值必须符合 (MUST conform to) URI-reference 语法，因此必须不包含 (MUST NOT include) 除 %x21 / %x23-5B / %x5D-7E 字符集以外的任何字符。

这些参数被包含在 HTTP 响应的实体主体 (entity-body) 中，使用 [RFC4627] 定义的 "application/json" 媒体类型。参数通过将每个参数添加到最高结构级别（顶级）序列化为 JavaScript 对象表示法 (JSON) 结构。参数名称 (Parameter names) 和字符串值 (string values) 作为 JSON 字符串 (JSON strings) 包含。数值 (Numerical values) 作为 JSON 数字 (JSON numbers) 包含。参数的顺序无关紧要且可以变化。

例如：

HTTP/1.1 400 Bad Request

Content-Type: application/json;charset=UTF-8

Cache-Control: no-store

Pragma: no-cache

{

"error":"invalid\_scope",

"error\_description":"The requested scope 'profile email address' exceeds the granted scope 'profile'."

}

6. 刷新访问令牌 (Refreshing an Access Token)

如果授权服务器已向客户端颁发了刷新令牌 (refresh token)，客户端可通过在 HTTP 请求的实体主体 (entity-body) 中使用 “application/x-www-form-urlencoded” 格式（依据附录 B），并以 UTF-8 字符编码添加以下参数，向令牌端点 (token endpoint) 发出刷新请求 (refresh request)：

grant\_type

必需 (REQUIRED)。值必须 (MUST) 设为 "refresh\_token"。

refresh\_token

必需 (REQUIRED)。颁发给客户端的刷新令牌。

scope

可选 (OPTIONAL)。如第 3.3 节所述的访问请求的作用域。\*\*请求的作用域必须不包含 (MUST NOT include) 任何原始授权范围之外的作用域；如果省略此参数，则视其等于资源所有者原始授权的范围 (scope originally granted by the resource owner)。\*\*

由于刷新令牌通常是用于请求额外访问令牌的长时有效凭据 (long-lasting credentials)，因此该刷新令牌被绑定 (bound) 于颁发给它的客户端。如果客户端类型为机密客户端 (confidential)，或该客户端被颁发了客户端凭据（或被分配了其他身份验证要求），则客户端必须 (MUST) 按照第 3.2.1 节所述向授权服务器进行身份验证。

例如，客户端使用传输层安全 (transport-layer security) 发出以下 HTTP 请求（额外换行仅用于显示目的）：

POST /token HTTP/1.1  
Host: server.example.com  
Authorization: Basic czZCaGRSa3FOMzpnWDFmQmFOM2JW  
Content-Type: application/x-www-form-urlencoded

grant\_type=refresh\_token&refresh\_token=tGzv3J0kFOXG5Qx2TlKWIA

授权服务器必须 (MUST)：

o 对于机密客户端 (confidential clients)，或者对于任何被颁发了客户端凭据（或具备其他身份验证要求）的客户端，要求进行客户端身份验证；

o 如果包含了客户端身份验证信息，则对客户端进行身份验证，并确保该刷新令牌是颁发给经过身份验证的客户端；并且

o 验证刷新令牌 (validate the refresh token)。

如果验证有效且已获授权 (valid and authorized)，授权服务器将按照第 5.1 节所述颁发访问令牌。如果请求的验证失败或无效，授权服务器将按照第 5.2 节所述返回错误响应。

授权服务器\*可\*(MAY) 颁发一个新的刷新令牌；在此情况下，客户端必须 (MUST) 丢弃旧的刷新令牌并用新的刷新令牌替换它。在向客户端颁发新的刷新令牌后，授权服务器\*可\*(MAY) 撤销旧的刷新令牌。如果颁发了新的刷新令牌，则该刷新令牌的作用域必须 (MUST) 与客户端在请求中包含的刷新令牌的作用域完全相同。

7. 访问受保护资源 (Accessing Protected Resources)

客户端通过向资源服务器 (resource server) 提供访问令牌 (access token) 来访问受保护资源 (protected resources)。资源服务器必须 (MUST) 验证该访问令牌，并确保其未过期且其作用域覆盖所请求的资源。资源服务器用于验证访问令牌的方法（以及任何错误响应）超出本规范的范畴，但通常涉及资源服务器与授权服务器之间的交互或协调。

客户端利用访问令牌向资源服务器进行身份验证的方法，取决于授权服务器颁发的访问令牌的类型。通常，它涉及使用定义了所用访问令牌类型（如 [RFC6750]）规范的认证方案 (authentication scheme)，配合 HTTP "Authorization" 请求头字段 [RFC2617]。

7.1. 访问令牌类型 (Access Token Types)

访问令牌类型 (access token type) 向客户端提供成功使用访问令牌发出受保护资源请求所需的信息（以及类型特定的属性）。如果客户端不理解该令牌类型，则其禁止使用 (MUST NOT use) 该访问令牌。

例如，定义于 [RFC6750] 中的 "持有者" 令牌类型 ("bearer" token type)，仅需将访问令牌字符串包含在请求中即可使用：

GET /resource/1 HTTP/1.1  
Host: example.com  
Authorization: Bearer mF\_9.B5f-4.1JqM

而定义于 [OAuth-HTTP-MAC] 中的 "MAC" 令牌类型 ("mac" token type)，通过颁发一个消息认证码 (Message Authentication Code, MAC) 密钥与访问令牌一起使用，该密钥用于对 HTTP 请求的某些组件进行签名：

GET /resource/1 HTTP/1.1  
Host: example.com  
Authorization: MAC id="h480djs93hd8",nonce="274312:dj83hs9s",mac="kDZvddkndxvhGRXZhvuDjEWhGe="

以上示例仅供参考。开发者在使用前应咨询 (advised to consult) [RFC6750] 和 [OAuth-HTTP-MAC] 规范。

每个访问令牌类型定义都规定了与 "access\_token" 响应参数一同发送给客户端的附加属性（如有）。它还定义了在发出受保护资源请求时用于包含访问令牌的 HTTP 认证方法 (HTTP authentication method)。

7.2. 错误响应 (Error Response)

如果资源访问请求失败，资源服务器应该 (SHOULD) 将错误告知客户端。虽然此类错误响应的具体细节超出本规范的范畴，但本文档在第 11.4 节建立了一个公共注册表 (common registry)，用于在 OAuth 令牌认证方案 (OAuth token authentication schemes) 之间共享错误值 (error values)。

主要为 OAuth 令牌认证设计的新的认证方案应该 (SHOULD) 定义一种向客户端提供错误状态码的机制，其中允许的错误值应在本规范建立的错误注册表中注册。

此类方案\*可\*(MAY) 将有效错误码集合限定在已注册值的一个子集内。如果错误码是通过一个具名参数 (named parameter) 返回的，该参数名称\*应该\*(SHOULD) 为 "error"。

其他能够用于OAuth令牌认证，但并非主要为此目的设计的方案，\*可\*(MAY) 以同样方式将其错误值绑定到该注册表。

新的认证方案\*可\*(MAY) 选择也指定使用 "error\_description" 和 "error\_uri" 参数，以与此规范中的使用方式并行地返回错误信息。

8. 扩展性 (Extensibility)

8.1. 定义访问令牌类型 (Defining Access Token Types)

访问令牌类型可通过以下两种方式之一定义：在访问令牌类型注册表 (Access Token Types registry) 中注册（遵循第 11.1 节的程序），或使用一个唯一的绝对 URI (unique absolute URI) 作为其名称。

使用 URI 名称的类型\*应该\*(SHOULD) 仅限于不普遍适用、且特定于其所用资源服务器实现细节的供应商特定实现。

所有其他类型\*必须\*(MUST) 注册。类型名称\*必须\*(MUST) 符合类型名称的 ABNF。如果该类型定义包含一种新的 HTTP 认证方案 (HTTP authentication scheme)，则该类型名称\*应该\*(SHOULD) 与 HTTP 认证方案名称相同（如 [RFC2617] 所定义）。令牌类型名称 "example" 保留用于示例。

type-name = 1\*name-char *(类型名称 = 1个或多个名称字符)*  
name-char = "-" / "." / "\_" / DIGIT / ALPHA *(名称字符 = "-" / "." / "\_" / 数字 / 字母)*

*8.2. 定义新端点参数 (Defining New Endpoint Parameters)*

*在授权端点或令牌端点中使用的新请求或响应参数，需遵循第 11.2 节的程序在 OAuth 参数注册表 (OAuth Parameters registry) 中定义和注册。*

*参数名称\*必须\*(MUST) 符合参数名称的 ABNF，并且参数值的语法\*必须\*(MUST) 是明确定义的 (well-defined)（例如，使用 ABNF，或引用现有参数的语法）。*

param-name = 1\*name-char *(参数名称 = 1个或多个名称字符)*  
name-char = "-" / "." / "\_" / DIGIT / ALPHA *(名称字符 = "-" / "." / "\_" / 数字 / 字母)*

*未注册的供应商特定参数扩展 (unregistered vendor-specific parameter extensions)，如果其不具备普遍适用性，并且特定于其所使用的授权服务器的实现细节，则\*应该\*(SHOULD) 使用一个不太可能与其他注册值冲突的供应商特定前缀 (vendor-specific prefix)（例如，以 'companyname\_' 开头）。*

*8.3. 定义新的授权许可类型 (Defining New Authorization Grant Types)*

*可以通过分配一个唯一的绝对 URI (unique absolute URI) 以用于 "grant\_type" 参数来定义新的授权许可类型 (authorization grant types)。如果该扩展许可类型 (extension grant type) 需要额外的令牌端点参数，这些参数\*必须\*(MUST) 在 OAuth 参数注册表 (OAuth Parameters registry) 中注册，遵循第 11.2 节所述的程序。*

*8.4. 定义新的授权端点响应类型 (Defining New Authorization Endpoint Response Types)*

*用于授权端点 (authorization endpoint) 的新响应类型需遵循第 11.3 节的程序，在授权端点响应类型注册表 (Authorization Endpoint Response Types registry) 中定义和注册。响应类型名称\*必须\*(MUST) 符合响应类型 (response-type) 的 ABNF 语法。*

response-type = response-name \*( SP response-name ) *(响应类型 = 响应名称 ( 空格 响应名称 ))  
response-name = 1*response-char *(响应名称 = 1个或多个 响应字符)*  
response-char = "\_" / DIGIT / ALPHA *(响应字符 = "\_" / 数字 / 字母)*

*如果一个响应类型包含一个或多个空格字符 (%x20)，它将被视为一个以空格分隔的值列表 (space-delimited list of values) 进行比较，其中值的顺序无关紧要。只能注册一种值的顺序 (only one order of values can be registered)，该注册涵盖相同值集合的所有其他排列 (covers all other arrangements of the same set of values)。*

*例如，响应类型 "token code" 在本规范中\*未定义\*(left undefined)。但是，某个扩展可以定义并注册 "token code" 响应类型。一旦注册，相同的组合就不能再注册为 "code token"，但这两个值都可以用来表示同一响应类型 (denote the same response type)。*

*8.5. 定义额外的错误代码 (Defining Additional Error Codes)*

*在协议扩展（即访问令牌类型、扩展参数或扩展许可类型）需要额外的错误代码 (additional error codes) 用于授权码许可错误响应（第 4.1.2.1 节）、隐式许可错误响应（第 4.2.2.1 节）、令牌错误响应（第 5.2 节）或资源访问错误响应（第 7.2 节）的情况下，\*可以\*(MAY) 定义此类错误代码。*

*扩展错误代码（Extension error codes） \*\*必须\*\*注册（遵循第 11.4 节的程序），如果与之关联的扩展属于以下任一情况：已注册的访问令牌类型 (registered access token type)、已注册的端点参数 (registered endpoint parameter)，或已注册的扩展许可类型 (extension grant type)。与未注册扩展一起使用的错误代码 \*\*可以\*\*注册。*

*错误代码 \*\*必须\*\*符合错误的 ABNF (Augmented Backus–Naur Form，扩展巴科斯范式) 语法，且\*\*应尽可能加上标识性名称前缀\*\*。例如，用于标识传递到扩展参数 "example" 的无效值的错误，\*\*应\*\*命名为 "example\_invalid"。*

error = 1\*error-char  
error-char = %x20-21 / %x23-5B / %x5D-7E

\*(错误 = 1个或多个错误字符

错误字符 = 编码为十六进制20-21 / 23-5B / 5D-7E的字符)\*

9. 原生应用程序 (Native Applications)

原生应用程序是安装在资源所有者所使用的设备上并执行的客户端（例如，桌面应用程序、原生移动应用程序）。原生应用程序在安全性、平台功能和整体最终用户体验方面需要特殊考量。

授权端点要求客户端与资源所有者的用户代理进行交互。原生应用程序可以调用外部用户代理 (external user-agent) 或在应用程序内部嵌入用户代理 (embed a user-agent)。例如：

o \*\*外部用户代理 (External user-agent)\*\* - 原生应用程序可通过以下方式捕获来自授权服务器的响应：使用向操作系统注册了处理程序机制的 URI 方案（以唤起客户端作为处理程序）、手动复制粘贴凭据、运行本地 Web 服务器、安装用户代理扩展程序，或者提供一个指向客户端控制下的服务器托管资源的重定向 URI（该资源随后将响应提供给原生应用程序）。

o \*\*嵌入用户代理 (Embedded user-agent)\*\* - 原生应用程序通过与嵌入式用户代理直接通信来获取响应，其方式包括：监控资源加载过程中触发的状态变化，或访问用户代理的 Cookie 存储。

在选择外部用户代理还是嵌入用户代理时，开发者应考虑以下方面：

o 外部用户代理可提高完成率，因为资源所有者可能已拥有授权服务器的活跃会话，从而无需重新认证。它提供了熟悉的最终用户体验和功能。

资源所有者也可能依赖用户代理特性或扩展程序来协助身份验证（例如，密码管理器、双因素设备读取器）。

o \*\*嵌入式用户代理（Embedded user-agent）\*\* 可提供更好的可用性，因为它消除了切换上下文和打开新窗口的需要。

o \*\*嵌入式用户代理（Embedded user-agent）\*\* 带来了安全挑战，因为资源所有者是在一个未经验证且缺乏大多数外部用户代理所具有视觉保护机制的窗口中完成身份验证的。嵌入式用户代理会引导最终用户去信任身份不明的身份验证请求（使得钓鱼攻击更容易实施）。

在选择隐式许可类型（implicit grant type）和授权码许可类型（authorization code grant type）时，应考虑以下因素：

o \*\*使用授权码许可类型的原生应用程序，应当不使用客户端凭据（SHOULD do so without using client credentials）\*\*，因为原生应用无法保密客户端凭据。

o \*\*使用隐式许可流程时，不会返回刷新令牌\*\*，这要求在访问令牌过期后必须重复授权流程。

10. 安全考虑 (Security Considerations)

作为一个灵活且可扩展的框架，OAuth 的安全考量取决于许多因素。以下各节为实施者提供安全指南，重点关注第 2.1 节描述的三种客户端类型：Web 应用程序、基于用户代理的应用程序和原生应用程序。

[OAuth-THREATMODEL] 提供了全面的 OAuth 安全模型与分析，以及协议设计的背景知识。

10.1. 客户端认证 (Client Authentication)

授权服务器与 Web 应用程序客户端建立客户端凭据，用于客户端身份验证。\*\*鼓励授权服务器考虑比客户端密码更强的客户端身份验证方式\*\*。Web 应用程序客户端\*\*必须确保（MUST ensure）\*\* 客户端密码及其他客户端凭据的机密性。

授权服务器\*\*禁止\*\*(MUST NOT) 向原生应用程序或基于用户代理的应用程序客户端颁发客户端密码 (client passwords) 或其他客户端凭据，\*\*用于客户端身份验证的目的\*\*。授权服务器\*\*可\*\*(MAY) 向在特定设备上安装的特定原生应用程序客户端实例颁发客户端密码或其他凭据。

当无法进行客户端身份验证时，授权服务器\*\*应当\*\*(SHOULD) 采用其他方式来验证客户端的身份——例如，要求注册客户端重定向 URI (redirection URI)，或协调资源所有者确认身份。在请求资源所有者授权时，有效的重定向 URI 不足以验证客户端的身份，但\*\*可\*\*用于在获取资源所有者授权后防止将凭据发送给伪造的客户端。

授权服务器必须考虑与未经认证的客户端交互的安全风险，并采取措施限制向此类客户端颁发的其他凭据（例如，刷新令牌）的潜在暴露。

## 10.2. 客户端冒充 (Client Impersonation)

如果被冒充的客户端未能或无法保密其客户端凭据，恶意客户端就可能冒充该客户端并获取对受保护资源的访问权限。

授权服务器\*\*必须\*\*(MUST) 在可能的情况下对客户端进行身份验证。如果授权服务器因客户端的性质而无法对其进行身份验证，则\*\*必须\*\*(MUST) 要求注册用于接收授权响应的任何重定向 URI，并且\*\*应当\*\*(SHOULD) 利用其他措施保护资源所有者免受此类潜在恶意客户端的侵害。例如，授权服务器可以协调资源所有者协助识别客户端及其来源。

授权服务器\*\*应当\*\*(SHOULD) 强制执行显式的资源所有者身份验证，并向资源所有者提供有关客户端、请求的授权范围以及有效期的信息。资源所有者有责任在当前客户端的上下文中审查这些信息，并批准或拒绝该请求。

授权服务器\*\*应当禁止\*\*(SHOULD NOT) 在不进行客户端身份验证或依赖其他措施（以确保重复请求来自原始客户端而非冒充者）的情况下，自动处理重复的授权请求（未经资源所有者的主动交互）。

10.3. 访问令牌 (Access Tokens)

访问令牌凭据 (access token credentials)（以及任何保密的访问令牌属性）\*\*必须\*\*(MUST) 在传输和存储过程中保持\*\*保密性\*\*，并且\*\*只能\*\*(only) 在授权服务器、该访问令牌有效的资源服务器以及被颁发该访问令牌的客户端之间共享。访问令牌凭据\*\*必须\*\*(MUST) 仅使用第 1.6 节所述（带 RFC2818 定义的服务器认证）的 TLS 进行传输。

当使用隐式许可类型 (implicit grant type) 时，访问令牌在 URI 片段 (URI fragment) 中传输，这可能将其暴露给未授权方 (unauthorized parties)。

授权服务器\*\*必须\*\*(MUST) 确保访问令牌无法被未授权方生成、修改或猜解 (guessed) 以产生有效的访问令牌。

客户端\*\*应当\*\*(SHOULD) 请求具有必要最小作用域 (minimal scope) 的访问令牌。在选择如何兑现所请求的作用域时，授权服务器\*\*应当\*\*(SHOULD) 将客户端的身份 (client identity) 纳入考量，并且\*\*可\*\*(MAY) 颁发权限小于请求的访问令牌。

本规范\*\*并未提供\*\*(does not provide any methods) 任何方法，供资源服务器确保其从给定客户端收到的访问令牌是由授权服务器颁发给该客户端的。

10.4. 刷新令牌 (Refresh Tokens)

授权服务器\*\*可\*\*(MAY) 向 Web 应用程序客户端和原生应用程序客户端颁发刷新令牌。

刷新令牌\*\*必须\*\*(MUST) 在传输和存储过程中保持\*\*保密性\*\*，并且\*\*只能\*\*(only) 在授权服务器和向其颁发刷新令牌的客户端之间共享。授权服务器\*\*必须\*\*(MUST) 维护刷新令牌与其所颁发客户端之间的绑定关系 (binding)。刷新令牌\*\*必须\*\*(MUST) 仅使用第 1.6 节所述（带 RFC2818 定义的服务器认证）的 TLS 进行传输。

只要可以认证客户端的身份，授权服务器\*\*必须\*\*(MUST) 在每次使用时验证刷新令牌与客户端身份之间的绑定。当无法进行客户端认证时，授权服务器\*\*应当\*\*(SHOULD) 部署其他手段来检测刷新令牌滥用 (abuse)。

例如，授权服务器可以采用\*\*刷新令牌轮换 (refresh token rotation)\*\* 机制：在每次访问令牌刷新响应中颁发一个新的刷新令牌，同时\*\*使之前的刷新令牌失效\*\*。

### **10.4. 刷新令牌 *(Refresh Tokens)* (续)**

​\*\*...但授权服务器保留该绑定关系。​****​ 如果刷新令牌被泄露（compromised）并被攻击者和合法客户端同时使用，其中一方将提交一个****已失效的****刷新令牌，这将通知授权服务器发生了安全泄露（breach）。  
授权服务器****必须****​(MUST) 确保刷新令牌****无法****被未授权方（unauthorized parties）​****生成、修改或猜解\*\*​（guessed）出有效的刷新令牌。

### **10.5. 授权码 *(Authorization Codes)***

授权码的传输****应该****​(SHOULD) 通过安全通道进行。如果重定向URI（redirection URI）标识的是网络资源，客户端****应该****​(SHOULD) 要求在其重定向URI上使用TLS。由于授权码是通过用户代理重定向传输的，因此它们****有可能****通过用户代理的历史记录（history）和HTTP Referrer标头（HTTP referrer headers）被泄露。  
授权码以****明文持有者凭证****​（plaintext bearer credentials）的方式运作，用于****验证****在授权服务器授予授权的资源所有者，与返回到客户端以完成流程的资源所有者是****同一实体****。因此，​****如果客户端依赖授权码进行自身的资源所有者身份验证****，那么客户端的重定向端点（redirection endpoint）​****必须****​(MUST) 要求使用TLS。  
授权码****必须****​(MUST) ​****生命周期短****​（short lived）且****仅限单次使用****​（single-use）。如果授权服务器观察到****多次尝试****使用同一个授权码来交换访问令牌，授权服务器****应该****​(SHOULD) 尝试****撤销****​（revoke）所有已基于该遭泄露的授权码颁发的访问令牌。  
​****如果可以对客户端进行身份验证****，授权服务器****必须****​(MUST) 对该客户端进行身份验证，并****确保****该授权码是****颁发给同一个客户端的****。

10.6. 授权码重定向URI操纵 (Authorization Code Redirection URI Manipulation) (续)

攻击者可以在合法的客户端处创建账户并发起授权流程。

当攻击者的用户代理被发送到授权服务器去授予访问权限时，攻击者抓取合法客户端提供的授权URI，并将其中的客户端重定向URI替换为由攻击者控制的URI。攻击者随后诱骗受害者点击这个被篡改的链接，以授权访问合法客户端。

受害者到达授权服务器后，会看到一个代表合法可信客户端的正常有效请求，并批准该请求。随后受害者被重定向到攻击者控制的端点，​同时携带授权码。攻击者通过使用客户端提供的原始重定向URI，将授权码发送给客户端来完成授权流程。客户端随后用该授权码交换访问令牌，并将其关联到攻击者的客户端账户——该账户现在即可访问受害者授权的受保护资源。

防御措施：

为防范此类攻击，授权服务器必须​(MUST)：

确保用于获取授权码的重定向URI，与兑换访问令牌时提供的重定向URI完全一致​

​要求​(MUST) 公共客户端注册其重定向URI，并应当要求​(SHOULD require) 机密客户端注册重定向URI

如果请求中提供了重定向URI，则必须​(MUST) 验证其与注册值是否匹配

10.7. 资源所有者密码凭证 (Resource Owner Password Credentials)

资源所有者密码凭证授权类型通常仅用于遗留或迁移场景。它通过不在客户端存储用户名密码来降低整体风险，但无法消除向客户端暴露高权限凭证的需求。

此授权类型比其他类型风险更高，原因在于：

违背了协议旨在避免的密码反模式​（客户端可能滥用密码，或意外泄露给攻击者）

资源所有者失去对授权流程的控制​（提交凭证即结束参与），导致客户端可能获取超出资源所有者意图的访问范围

使用建议：

授权服务器与客户端应当​(SHOULD)：

最小化此授权类型的使用，优先选择其他授权类型

​严格限制通过此类型颁发的访问令牌作用域和生命周期

考虑访问令牌的权限范围，避免授予超出必要的高权限

### **10.8. 请求保密性 *(Request Confidentiality)***

访问令牌 (access tokens)、刷新令牌 (refresh tokens)、资源所有者密码 (resource owner passwords) 及客户端凭据 (client credentials) ​****必须禁止 (MUST NOT)​****​ 以明文传输。授权码 (authorization codes) ​****应当避免 (SHOULD NOT)​****​ 以明文传输。  
"state" 和 "scope" 参数****应当避免 (SHOULD NOT)​****​ 包含明文的敏感客户端或资源所有者信息，因为它们可能通过非安全通道传输或不安全地存储。

### **10.9. 确保端点真实性 *(Ensuring Endpoint Authenticity)***

为防止中间人攻击 (man-in-the-middle attacks)，授权服务器****必须要求 (MUST require)​****​：

* 向授权端点 (authorization endpoint) 和令牌端点 (token endpoint) 发送的任何请求，均采用定义了服务器认证的 TLS（遵循 [RFC2818]）。  
  客户端****必须 (MUST)​****​ 按照 [RFC6125] 的定义验证授权服务器的 TLS 证书，并****严格遵循****该规范中服务器身份认证的要求。

### **10.10. 凭证猜测攻击防护 *(Credentials-Guessing Attacks)***

授权服务器****必须 (MUST)​****​ 防止攻击者猜出访问令牌、授权码、刷新令牌、资源所有者密码及客户端凭据。  
攻击者猜出生成的令牌（以及其他****非供最终用户处理****的凭证）的概率****必须 (MUST)​****​ 小于或等于 2^(-128)，并****应当尽量 (SHOULD)​****​ 小于或等于 2^(-160)。  
对于****供最终用户使用****的凭据 (credentials intended for end-user usage)，授权服务器****必须 (MUST)​****​ 采用其他防护手段。

### **10.11. 防范钓鱼攻击 *(Phishing Attacks)***

本协议及类似协议的广泛部署可能导致最终用户逐渐习惯被重定向到要求输入密码的网站的做法。如果最终用户在输入凭据前未仔细验证这些网站的真实性，攻击者将可能利用此行为窃取资源所有者的密码。  
​****服务提供商应当 (SHOULD):​****​

1. 尝试教育最终用户认识钓鱼攻击的风险。
2. 提供易于最终用户验证其网站真实性的机制。  
   ​****客户端开发者应当 (SHOULD):​****​
3. 考虑其与用户代理交互方式（例如，外部的、嵌入式的）的安全影响。
4. 考虑最终用户验证授权服务器真实性的能力。

### **10.8. 请求保密性 *(Request Confidentiality)***

访问令牌 (access tokens)、刷新令牌 (refresh tokens)、资源所有者密码 (resource owner passwords) 及客户端凭据 (client credentials) ​****必须禁止 (MUST NOT)​****​ 以明文传输。授权码 (authorization codes) ​****应当避免 (SHOULD NOT)​****​ 以明文传输。  
"state" 和 "scope" 参数****应当避免 (SHOULD NOT)​****​ 包含明文的敏感客户端或资源所有者信息，因为它们可能通过非安全通道传输或不安全地存储。

### **10.9. 确保端点真实性 *(Ensuring Endpoint Authenticity)***

为防止中间人攻击 (man-in-the-middle attacks)，授权服务器****必须要求 (MUST require)​****​：

* 向授权端点 (authorization endpoint) 和令牌端点 (token endpoint) 发送的任何请求，均采用定义了服务器认证的 TLS（遵循 [RFC2818]）。  
  客户端****必须 (MUST)​****​ 按照 [RFC6125] 的定义验证授权服务器的 TLS 证书，并****严格遵循****该规范中服务器身份认证的要求。

### **10.10. 凭证猜测攻击防护 *(Credentials-Guessing Attacks)***

授权服务器****必须 (MUST)​****​ 防止攻击者猜出访问令牌、授权码、刷新令牌、资源所有者密码及客户端凭据。  
攻击者猜出生成的令牌（以及其他****非供最终用户处理****的凭证）的概率****必须 (MUST)​****​ 小于或等于 2^(-128)，并****应当尽量 (SHOULD)​****​ 小于或等于 2^(-160)。  
对于****供最终用户使用****的凭据 (credentials intended for end-user usage)，授权服务器****必须 (MUST)​****​ 采用其他防护手段。

### **10.11. 防范钓鱼攻击 *(Phishing Attacks)***

本协议及类似协议的广泛部署可能导致最终用户逐渐习惯被重定向到要求输入密码的网站的做法。如果最终用户在输入凭据前未仔细验证这些网站的真实性，攻击者将可能利用此行为窃取资源所有者的密码。  
​****服务提供商应当 (SHOULD):​****​

1. 尝试教育最终用户认识钓鱼攻击的风险。
2. 提供易于最终用户验证其网站真实性的机制。  
   ​****客户端开发者应当 (SHOULD):​****​
3. 考虑其与用户代理交互方式（例如，外部的、嵌入式的）的安全影响。
4. 考虑最终用户验证授权服务器真实性的能力。

### **10.13. 点击劫持 *(Clickjacking)***

在点击劫持攻击中，攻击者****注册一个合法客户端****，随后****构建一个恶意网站****。该恶意网站在一组****精心布置的虚拟按钮****之上，​****透明地加载****授权服务器的授权端点网页（置于透明 iframe 中）。这些虚拟按钮的位置被设计成正好位于授权页面上关键按钮（例如“授权”按钮）的下方。当最终用户点击这个****误导性的可见按钮****时，实际上****点击了授权页面上一个不可见的按钮****​（例如“Authorize”按钮）。这使攻击者能够****在最终用户不知情的情况下，诱骗资源所有者授予其客户端访问权限****。

#### **防范措施：**

* ​****原生应用程序应当 (SHOULD)：​****​ 在请求最终用户授权时，​****使用外部浏览器****而非在应用程序内部嵌入浏览器。
* ​****授权服务器应当：​****​
  + 对于较新的浏览器，可使用****​（非标准）"x-frame-options" 响应头****来避免 iframe 嵌入。此响应头可设两个值：
    - "deny": ​****阻止任何框架（iframe）嵌入****。
    - "sameorigin": ​****仅允许同源站点嵌入****。
  + 对于旧版浏览器，可采用 ​****JavaScript 反 iframe 嵌入技术****​（frame-busting techniques），但其在所有浏览器中的****有效性可能无法保证****。

### **10.14. 代码注入和输入验证 *(Code Injection and Input Validation)***

当应用程序****未净化（unsanitized）​****​ 地使用输入或其他外部变量，并导致****应用程序逻辑被修改****时，就会发生代码注入攻击。这可能使攻击者能够：

* 获取应用程序设备或其数据的访问权限
* 导致****拒绝服务 (denial of service)​****​
* 引入广泛的****恶意副作用****​

#### **核心要求：**

​****授权服务器和客户端必须 (MUST)：​****​

1. ​****净化 (Sanitize)​****​ 任何接收到的值。
2. ​****尽可能进行验证 (Validate when possible)​****。  
   ​****特别需要关注 "state" 和 "redirect\_uri" 参数值的安全性。****

### **10.15. 开放重定向器 *(Open Redirectors)***

授权服务器、授权端点以及客户端重定向端点****可能因配置不当而成为开放重定向器****。开放重定向器是一种端点，其使用某个参数（例如 redirect\_uri）来自动将用户代理重定向到由参数值指定的位置，​****而不对该值进行任何验证****。

#### **风险与利用：**

* ​****钓鱼攻击 (Phishing attacks)：​****​ 攻击者利用用户对熟悉且受信任网站域名（URI authority 部分）的信任，诱使最终用户访问恶意网站。
* ​****部分注册风险：​****​ 如果授权服务器****仅允许客户端注册重定向 URI 的一部分****​（例如只注册域名），攻击者便可利用目标域上****由他人运营或已遭入侵的开放重定向器****来完成攻击链条。

### **10.16. 隐式流程中滥用访问令牌冒充资源所有者 *(Misuse of Access Token to Impersonate Resource Owner in Implicit Flow)***

对于使用隐式流程的****公共客户端****，本规范****并未提供****任何方法让客户端确定访问令牌是颁发给****哪个客户端****的。

资源所有者可能****自愿****通过向攻击者的恶意客户端授予访问令牌，从而****委托****其访问受保护资源。这可能是由于****钓鱼攻击****或其他****欺骗手段****所致。攻击者也可能通过其他****机制窃取****令牌。随后，攻击者可以通过向一个****合法的公共客户端****提供该访问令牌，尝试****冒充****资源所有者。

在隐式流程 (response\_type=token) 中，攻击者可以轻易****调换****来自授权服务器的响应中的令牌，用先前颁发给攻击者自己的访问令牌****替换****掉真正的访问令牌。

与****本地应用程序****进行通信的服务器，如果依赖在****后端通信通道 (back channel)​****​ 中传递的访问令牌来识别****客户端用户****，同样可能因攻击者创建一个能注入****任意被盗访问令牌****的****恶意/受控应用程序****而遭到入侵。

任何做出****​“只有资源所有者才能向其提供针对该资源的有效访问令牌”​****这一假设的公共客户端，都****易受此类攻击****。

此类攻击可能将****合法客户端****上有关资源所有者的****信息暴露给攻击者（恶意客户端）​****。这也将允许攻击者在合法客户端上以****与最初授权访问令牌或授权码的资源所有者相同的权限****执行操作。

向客户端****认证资源所有者****超出了本规范的范围。​****任何****使用授权流程作为****向客户端进行委托式最终用户认证****​（例如，第三方登录服务）的规范，​****禁止****在不采用额外安全机制的情况下使用隐式流程；这些机制应使客户端能够判断访问令牌****是否是为供其使用而颁发的****​（例如，对访问令牌****添加受众 (audience) 限制****）。

以下是严格按照您提供的 IANA Considerations 技术文档内容的精准中文翻译，严格遵循 RFC 技术规范术语和流程描述：

11. IANA 考虑事项 (IANA Considerations)

11.1. OAuth 访问令牌类型注册表 (OAuth Access Token Types Registry)

本规范建立了 ​OAuth 访问令牌类型注册表 (OAuth Access Token Types registry)​。

访问令牌类型的注册遵循 ​​“需通过规范审核 ([RFC5226])”​​ 的要求。在为期两周的 oauth-ext-review@ietf.org 邮件列表评审期之后，注册需在 ​一名或多名指定专家 (Designated Experts)​​ 的建议下进行。​但是，为允许在规范发布前分配值，一旦指定专家确信相关规范将被发布，即可批准注册。

注册请求必须 (must)​​ 发送至 oauth-ext-review@ietf.org 邮件列表进行评审和评议，邮件主题应恰当明确​（例如：“Request for access token type: example”）。

在评审期内，指定专家将批准或拒绝注册请求，并将决定通知评审邮件列表和 ​IANA。拒绝决定应包含 (should include)​​ 解释说明，并在适用时给出使请求成功的修改建议。

​IANA 必须 (must)​​ 仅接受来自指定专家的注册表更新，并应当 (should)​​ 将所有注册请求引导至前述评审邮件列表。

11.1.1. 注册模板 (Registration Template)

​类型名称 (Type name):​​

所请求的名称（例如："example"）。

​附加令牌端点响应参数 (Additional Token Endpoint Response Parameters):​​

与 access\_token 参数一同返回的额外响应参数。新参数必须 (MUST)​​ 按照第 11.2 节所述，在 ​OAuth 参数注册表 (OAuth Parameters registry)​​ 中单独注册。

​HTTP 认证方案 (HTTP Authentication Scheme(s)):​​

​HTTP 认证方案名称​（如有），用于在使用此类访问令牌进行受保护资源请求 (protected resource requests)​​ 时的身份验证。

​变更控制者 (Change controller):​​

对于标准轨道 RFC，注明“IETF”。对于其他情况，提供责任方的名称。也可包含其他详细信息（如邮政地址、电子邮件地址、主页 URI）。

### **11.2. OAuth 参数注册表 *(OAuth Parameters Registry)***

本规范建立了 ​****OAuth 参数注册表 (OAuth Parameters registry)​****。

在授权端点请求、授权端点响应、令牌端点请求或令牌端点响应中包含的****附加参数****，需遵循 ​****​“需通过规范审核 ([RFC5226])”​****​ 的要求进行注册。注册将在为期两周的 oauth-ext-review@ietf.org 邮件列表评审期之后，并在****一名或多名指定专家 (Designated Expert(s))​****​ 的建议下进行。​****但是，为允许在规范发布前分配值****，一旦指定专家确信相关规范****将被发布****，即可批准注册。

注册请求****必须 (must)​****​ 发送至 oauth-ext-review@ietf.org 邮件列表进行评审和评议，邮件主题应****恰当明确****​（例如："Request for parameter: example"）。

在评审期内，指定专家将****批准或拒绝****该注册请求，并将决定通知评审邮件列表和 ​****IANA****。拒绝决定****应包含 (should include)​****​ 解释说明，并在适用时****给出使请求成功的修改建议****。

​****IANA 必须 (must)​****​ 仅接受来自指定专家的注册表更新，并****应当 (should)​****​ 将所有注册请求引导至前述评审邮件列表。

#### **11.2.1. 注册模板 *(Registration Template)***

​****参数名称 (Parameter name):​****​  
所请求的名称（例如："example"）。

​****参数使用位置 (Parameter usage location):​****​  
可使用该参数的****位置****。可能的位置包括：authorization request（授权请求）、authorization response（授权响应）、token request（令牌请求）或 token response（令牌响应）。

​****变更控制者 (Change controller):​****​  
对于****标准轨道 RFC (Standards Track RFC)​****，注明 "IETF"。对于其他情况，提供****责任方的名称****。也可包含其他详细信息（如邮政地址、电子邮件地址、主页 URI）。

​****规范文档 (Specification document(s)):​****​  
对该参数进行规范的****文档参考****，​****最好 (preferably)​****​ 包含一个可用于检索文档副本的 ​****URI****。也可以****包括 (may also be included)​****​ 相关章节的指示，但****非必须 (not required)​****。

### **11.2.2. 初始注册表内容 *(Initial Registry Contents)***

OAuth 参数注册表的初始内容如下：

* ​****参数名称 (Parameter name):​****​ client\_id
  + ​****参数使用位置 (Parameter usage location):​****​ 授权请求 (authorization request), 令牌请求 (token request)
  + ​****变更控制者 (Change controller):​****​ IETF
  + ​****规范文档 (Specification document(s)):​****​ RFC 6749
* ​****参数名称 (Parameter name):​****​ client\_secret
  + ​****参数使用位置 (Parameter usage location):​****​ 令牌请求 (token request)
  + ​****变更控制者 (Change controller):​****​ IETF
  + ​****规范文档 (Specification document(s)):​****​ RFC 6749
* ​****参数名称 (Parameter name):​****​ response\_type
  + ​****参数使用位置 (Parameter usage location):​****​ 授权请求 (authorization request)
  + ​****变更控制者 (Change controller):​****​ IETF
  + ​****规范文档 (Specification document(s)):​****​ RFC 6749
* ​****参数名称 (Parameter name):​****​ redirect\_uri
  + ​****参数使用位置 (Parameter usage location):​****​ 授权请求 (authorization request), 令牌请求 (token request)
  + ​****变更控制者 (Change controller):​****​ IETF
  + ​****规范文档 (Specification document(s)):​****​ RFC 6749
* ​****参数名称 (Parameter name):​****​ scope
  + ​****参数使用位置 (Parameter usage location):​****​ 授权请求 (authorization request), 授权响应 (authorization response), 令牌请求 (token request), 令牌响应 (token response)
  + ​****变更控制者 (Change controller):​****​ IETF
  + ​****规范文档 (Specification document(s)):​****​ RFC 6749
* ​****参数名称 (Parameter name):​****​ state
  + ​****参数使用位置 (Parameter usage location):​****​ 授权请求 (authorization request), 授权响应 (authorization response)
  + ​****变更控制者 (Change controller):​****​ IETF
  + ​****规范文档 (Specification document(s)):​****​ RFC 6749
* ​****参数名称 (Parameter name):​****​ code
  + ​****参数使用位置 (Parameter usage location):​****​ 授权响应 (authorization response), 令牌请求 (token request)
  + ​****变更控制者 (Change controller):​****​ IETF
  + ​****规范文档 (Specification document(s)):​****​ RFC 6749

​****规范文档说明 (Specification document(s) description):​****​

应引用指定该参数的文档，最好包含一个可用于检索文档副本的 URI。也可以包含相关章节的指示，但非必须。

### **11.3. OAuth 授权端点响应类型注册表 *(OAuth Authorization Endpoint Response Types Registry)***

本规范建立了 ​****OAuth 授权端点响应类型注册表 (OAuth Authorization Endpoint Response Types registry)​****。

用于授权端点的****附加响应类型 (additional response types)​****​ 需遵循 ​****​“需通过规范审核 ([RFC5226])”​****​ 的要求进行注册。在为期两周的 oauth-ext-review@ietf.org 邮件列表评审期之后，并在****一名或多名指定专家 (Designated Expert(s))​****​ 的建议下进行注册。​****但是，为允许在规范发布前分配值****，一旦指定专家确信相关规范****将被发布****，即可批准注册。

注册请求****必须 (must)​****​ 发送至 oauth-ext-review@ietf.org 邮件列表进行评审和评议，邮件主题应****恰当明确****​（例如："Request for response type: example"）。

在评审期内，指定专家将****批准或拒绝****该注册请求，并将决定通知评审邮件列表和 ​****IANA****。拒绝决定****应包含 (should include)​****​ 解释说明，并在适用时****给出使请求成功的修改建议****。

​****IANA 必须 (must)​****​ 仅接受来自指定专家的注册表更新，并****应当 (should)​****​ 将所有注册请求引导至前述评审邮件列表。

#### **11.3.1. 注册模板 *(Registration Template)***

​****响应类型名称 (Response type name):​****​  
所请求的名称（例如："example"）。

​****变更控制者 (Change controller):​****​  
对于****标准轨道 RFC (Standards Track RFC)​****，注明 "IETF"。对于其他情况，提供****责任方的名称****。也可包含其他详细信息（如邮政地址、电子邮件地址、主页 URI）。

​****规范文档 (Specification document(s)):​****​  
对该响应类型进行规范的****文档参考****，​****最好 (preferably)​****​ 包含一个可用于检索文档副本的 ​****URI****。也可以****包括 (may also be included)​****​ 相关章节的指示，但****非必须 (not required)​****。

### **11.3.2. 初始注册表内容 *(Initial Registry Contents)***

OAuth 授权端点响应类型注册表的初始内容如下：  
• ​****响应类型名称 (Response type name):​****​ code  
• ​****变更控制者 (Change controller):​****​ IETF  
• ​****规范文档 (Specification document(s)):​****​ RFC 6749  
• ​****响应类型名称 (Response type name):​****​ token  
• ​****变更控制者 (Change controller):​****​ IETF  
• ​****规范文档 (Specification document(s)):​****​ RFC 6749

### **11.4. OAuth 扩展错误注册表 *(OAuth Extensions Error Registry)***

本规范建立了 ​****OAuth 扩展错误注册表 (OAuth Extensions Error registry)​****。

与其他协议扩展（即扩展许可类型、访问令牌类型或扩展参数）​****共同使用****的附加错误代码，需遵循 ​****​“需通过规范审核 ([RFC5226])”​****​ 的要求进行注册。在为期两周的 oauth-ext-review@ietf.org 邮件列表评审期之后，并在****一名或多名指定专家 (Designated Expert(s))​****​ 的建议下进行注册。​****但是，为允许在规范发布前分配值****，一旦指定专家确信相关规范****将被发布****，即可批准注册。

注册请求****必须 (must)​****​ 发送至 oauth-ext-review@ietf.org 邮件列表进行评审和评议，邮件主题应****恰当明确****​（例如："Request for error code: example"）。

在评审期内，指定专家将****批准或拒绝****该注册请求，并将决定通知评审邮件列表和 ​****IANA****。拒绝决定****应包含 (should include)​****​ 解释说明，并在适用时****给出使请求成功的修改建议****。

​****IANA 必须 (must)​****​ 仅接受来自指定专家的注册表更新，并****应当 (should)​****​ 将所有注册请求引导至前述评审邮件列表。

### **11.4.1. 注册模板 *(Registration Template)***

​****错误名称 (Error name):​****​  
所请求的名称（例如："example"）。错误名称的值****必须禁止 (MUST NOT include)​****​ 包含除 %x20-21 / %x23-5B / %x5D-7E 字符集以外的任何字符。

​****错误使用位置 (Error usage location):​****​  
该错误可被使用的****位置****。可能的位置包括：

* + 授权码许可错误响应 (authorization code grant error response)（第 4.1.2.1 节）
  + 隐式许可错误响应 (implicit grant error response)（第 4.2.2.1 节）
  + 令牌错误响应 (token error response)（第 5.2 节）
  + 资源访问错误响应 (resource access error response)（第 7.2 节）

​****相关协议扩展 (Related protocol extension):​****​  
该错误代码所****关联使用****的****扩展许可类型 (extension grant type)​****、****访问令牌类型 (access token type)​****​ 或****扩展参数 (extension parameter)​****​ 的名称。

​****变更控制者 (Change controller):​****​  
对于****标准轨道 RFC (Standards Track RFC)​****，注明 "IETF"。对于其他情况，提供****责任方的名称****。也可包含其他详细信息（如邮政地址、电子邮件地址、主页 URI）。

​****规范文档 (Specification document(s)):​****​  
对该错误代码进行规范的****文档参考****，​****最好 (preferably)​****​ 包含一个可用于检索文档副本的 ​****URI****。也可以****包括 (may also be included)​****​ 相关章节的指示，但****非必须 (not required)​****。