**OpenID Connect 1.0规范**

# 欢迎来到OpenID Connect

### OpenID Connect是什么？

OpenID Connect 1.0是基于OAuth 2.0协议的简单的身份层。它允许客户以验证最终用户的基础上由一个授权服务器，以及进行身份验证的身份，以获取有关最终??用户在一个可互操作和REST的方式基本个人资料信息。

OpenID Connect支持所有类型客户端，包括基于Web，移动和JavaScript的客户端，请求和接收有关身份验证的会话和最终用户的信息。该规范套件是可扩展的，允许参与者使用的可选功能，如加密身份数据，发现的OpenID提供商，和会话管理，当它是明智的。

见<http://openid.net/connect/faq/>一组答案的OpenID Connect常见问题的。

### OpenID Connect与OpenID 2.0有什么不同？

OpenID Connect 执行许多与支持OpenID 2.0相同的任务，但这样做的方式，就是API的友好，并通过本地和移动应用程序使用。OpenID Connect 定义了强劲的签名和加密可选的机制。而集成的OAuth 1.0A和支持OpenID 2.0的要求的延伸，在OpenID Connect，OAuth的2.0功能被集成到协议本身。

### 规范组成

OpenID Connect 1.0规范由六个文件：

* [核心](http://openid.net/specs/openid-connect-core-1_0.html) -定义OpenID Connect核心功能：身份验证建立在OAuth 2.0的顶部和使用声明沟通一下最终用户信息
* [发现](http://openid.net/specs/openid-connect-discovery-1_0.html) - （可选）定义客户端如何动态地发现的OpenID提供商的信息
* [动态注册](http://openid.net/specs/openid-connect-registration-1_0.html) - （可选）定义客户端如何动态地使用OpenID提供商注册
* [会话管理](http://openid.net/specs/openid-connect-session-1_0.html) - （可选）定义了如何管理OpenID Connect会话，包括退出功能
* [OAuth 2.0多响应类型](http://openid.net/specs/oauth-v2-multiple-response-types-1_0.html) -定义了几个具体的新的OAuth 2.0响应类型
* [OAuth 2.0表单提交响应模式](http://openid.net/specs/oauth-v2-form-post-response-mode-1_0.html) - （可选）定义了如何使用HTTP POST使用被自动提交用户代理HTML表单的值，返回的OAuth 2.0授权响应参数（包括 OpenID Connect认证响应参数）

两个实施者指南也可用来作为基本的基于Web的依赖方的实施者自足参考：

* [基本客户实施者指南](http://openid.net/specs/openid-connect-basic-1_0.html) -核心功能使用OAuth的码流一个基于网络的依赖方简单的子集
* [隐客户实施者指南](http://openid.net/specs/openid-connect-implicit-1_0.html) -核心功能使用OAuth的隐流一个基于网络的依赖方简单的子集

OpenID Connect规格，实施者指南，他们是建立在规格如下图所示。点击图中的框来查看规范。

### 参加工作组

来监视 OpenID Connect 1.0 规范进步的最简单的方法就是加入邮件列表的<http://lists.openid.net/mailman/listinfo/openid-specs-ab>。

请注意，虽然任何人都可以加入邮件列表作为一个只读的收件人，发布到邮件列表或有助于规范要求提交的知识产权协议。更多信息，请<http://openid.net/intellectual-property>。**确保指定的工作组为“OpenID AB /Connect”**，因为这个群体是一个合并工作组和两个名字**必须被指定**。

有关参赛详情，请参阅[OpenID Connect工作组页](http://openid.net/wg/connect/)。

### 互操作测试

[互操作测试的OpenID](http://osis.idcommons.net/wiki/OC5:OpenID_Connect_Interop_5) Connect正在进行。如果您有兴趣参与互操作的活动，加入[OpenID Connect互操作](http://groups.google.com/group/openid-connect-interop?hl=en)的邮件列表。

**OpenID Connect Core 1.0**

摘要

OpenID Connect1.0是对的OAuth2.0协议之上的简单的身份层。它使客户端验证最终用户的基础上由一个授权服务器，以及进行身份验证的身份，以获取有关最终用户在一个可互操作和REST的方式基本个人资料信息。

该规范定义了核心的OpenID Connect功能：身份验证建立在OAuth的2.0之上，并利用声明沟通一下最终用户的信息。它也描述了使用OpenID连接的安全性和隐私的考虑。

目录

1 .介绍

    1.1 .要求记法和约定

    1.2 .术语

    1.3 .概观

2 . ID令牌

3 .认证

    3.1 .使用授权码流认证

        3.1.1 .授权码流步骤

        3.1.2 .授权端点

            3.1.2.1 .认证请求

            3.1.2.2 .验证请求验证

            3.1.2.3 .授权服务器验证最终用户

            3.1.2.4 .授权服务器获取最终用户许可/授权

            3.1.2.5 .成功的认证响应

            3.1.2.6 .验证错误响应

            3.1.2.7 .验证响应验证

        3.1.3 .令牌端点

            3.1.3.1 .令牌请求

            3.1.3.2 .令牌请求验证

            3.1.3.3 .成功的令牌响应

            3.1.3.4 .令牌错误响应

            3.1.3.5 .令牌响应验证

            3.1.3.6 . ID令牌

            3.1.3.7 . ID令牌验证

            3.1.3.8 .访问令牌验证

    3.2 .使用隐式认证流程

        3.2.1 .隐流步骤

        3.2.2 .授权端点

            3.2.2.1 .认证请求

            3.2.2.2 .验证请求验证

            3.2.2.3 .授权服务器验证最终用户

            3.2.2.4 .授权服务器获取最终用户许可/授权

            3.2.2.5 .成功的认证响应

            3.2.2.6 .验证错误响应

            3.2.2.7 .重定向的URI片段处理

            3.2.2.8 .验证响应验证

            3.2.2.9 .访问令牌验证

            3.2.2.10 . ID令牌

            3.2.2.11 . ID令牌验证

    3.3 .使用混合身份验证流程

        3.3.1 .混合流程步骤

        3.3.2 .授权端点

            3.3.2.1 .认证请求

            3.3.2.2 .验证请求验证

            3.3.2.3 .授权服务器验证最终用户

            3.3.2.4 .授权服务器获取最终用户许可/授权

            3.3.2.5 .成功的认证响应

            3.3.2.6 .验证错误响应

            3.3.2.7 .重定向的URI片段处理

            3.3.2.8 .验证响应验证

            3.3.2.9 .访问令牌验证

            3.3.2.10 .授权码验证

            3.3.2.11 . ID令牌

            3.3.2.12 . ID令牌验证

        3.3.3 .令牌端点

            3.3.3.1 .令牌请求

            3.3.3.2 .令牌请求验证

            3.3.3.3 .成功的令牌响应

            3.3.3.4 .令牌错误响应

            3.3.3.5 .令牌响应验证

            3.3.3.6 . ID令牌

            3.3.3.7 . ID令牌验证

            3.3.3.8 .访问令牌

            3.3.3.9 .访问令牌验证

4 .由第三方发起登录

5 .声明

    5.1 .标准声明

        5.1.1 .地址声明

        5.1.2 .其他声明

    5.2 .声明语言和脚本

    5.3 .使用者资讯端点

        5.3.1 .使用者资讯需求

        5.3.2 .成功的UserInfo响应

        5.3.3 .使用者资讯错误响应

        5.3.4 .使用者资讯响应验证

    5.4 .使用范围值，要求声明

    5.5 .使用“声明”请求参数要求声明

        5.5.1 .个人声明要求

            5.5.1.1 .要求“ ACR ”声明

        5.5.2 .语言和文字为个人声明

    5.6 .声明类型

        5.6.1 .普通债权

        5.6.2 .汇总和分布式声明

            5.6.2.1 .聚合权利要求的示例

            5.6.2.2 .分布式权利要求的示例

    5.7 .权利要求稳定性和唯一性

6 .通过请求参数作为JWTs

    6.1 .按值传递一个请求对象

        6.1.1 .请使用“请求”请求参数

    6.2 .按引用传递对象请求

        6.2.1 . URL引用的请求对象

        6.2.2 .要求使用“ REQUEST\_URI ”请求参数

        6.2.3 .授权服务器取指请求对象

        6.2.4 . “ REQUEST\_URI ”的理由

    6.3 .智威汤逊验证基于请求

        6.3.1 .加密的请求对象

        6.3.2 .签署请求对象

        6.3.3 .请求参数组装和验证

7 .自发行OpenID提供者

    7.1 .自发行OpenID提供者发现

    7.2 .自发行OpenID提供者报名

        7.2.1 .与“注册”请求参数提供信息

    7.3 .自发行OpenID提供者请求

    7.4 .自发行OpenID提供者响应

    7.5 .自发行标识令牌验证

8 .主题标识符类型

    8.1 .成对标识算法

9 .客户端身份验证

10 .签名和加密

    10.1 .签名

        10.1.1 .不对称签名密钥轮换

    10.2 .加密

        10.2.1 .非对称加密密钥的旋转

11 .离线访问

12 .使用刷新令牌

    12.1 .刷新请求

    12.2 .成功刷新响应

    12.3 .刷新错误响应

13 .序列化

    13.1 .查询字符串序列化

    13.2 .序列化形式

    13.3 . JSON序列化

14 .字符串操作

15 .实施注意事项

    15.1 .强制性地对所有的OpenID提供商特点

    15.2 .强制实现动态的OpenID提供商特点

    15.3 .发现和注册

    15.4 .强制实现的功能为信任方

    15.5 .实现说明

        15.5.1 .授权代码实现说明

        15.5.2 .杜撰实现说明

        15.5.3 .重定向的URI片段操作实现说明

    15.6 .兼容性说明

        15.6.1 .准决赛IETF规范

        15.6.2 .谷歌“ ISS ”的价值

    15.7 .相关的规范和实施者指南

16 .安全注意事项

    16.1 .要求披露

    16.2 .服务器伪装

    16.3 .令牌制造/修改

    16.4 .访问令牌披露

    16.5 .服务器响应披露

    16.6 .服务器响应赖账

    16.7 .要求赖账

    16.8 .访问令牌重定向

    16.9 .令牌重用

    16.10 .窃听或泄漏授权码（中学验证器捕捉）

    16.11 .标记替换

    16.12 .时序攻击

    16.13 .其他加密相关的攻击

    16.14 .签名和加密订购

    16.15 .发行人标识符

    16.16 .隐流威胁

    16.17 . TLS的要求

    16.18 .访问令牌和刷新令牌的寿命

    16.19 .对称密钥熵

    16.20 .需要签名请求

    16.21 .需要加密的请求

17 .隐私考量

    17.1 .个人身份信息

    17.2 .数据访问监控

    17.3 .相关

    17.4 .离线访问

18 . IANA考虑

    18.1 . JSON网站令牌债权登记

        18.1.1 .注册表内容

    18.2 . OAuth的参数注册

        18.2.1 .注册表内容

    18.3 . OAuth的扩展错误登记

        18.3.1 .注册表内容

19 .参考文献

    19.1 .规范性引用文件

    19.2 .翔实的参考资料

附录A.授权示例

    A.1 .例如使用RESPONSE\_TYPE =代码

    A.2 .例如使用RESPONSE\_TYPE = id\_token

    A.3 .使用RESPONSE\_TYPE例如= id\_token令牌

    A.4 .例如使用RESPONSE\_TYPE =代码id\_token

    A.5.例如使用RESPONSE\_TYPE =令牌代码

    A.6 .使用示例RESPONSE\_TYPE =代码id\_token令牌

    A.7 . RSA密钥用于实例

附录B.致谢

附录C.声明

§作者地址

摘要

OpenID Connect1.0是对的OAuth2.0协议之上的简单的身份层。它使客户端验证最终用户的基础上由一个授权服务器，以及进行身份验证的身份，以获取有关最终用户在一个可互操作和REST的方式基本个人资料信息。

该规范定义了核心的OpenID Connect功能：身份验证建立在OAuth的2.0之上，并利用声明沟通一下最终用户的信息。它也描述了使用OpenID连接的安全性和隐私的考虑。

官方文档：<http://openid.net/specs/openid-connect-core-1_0.html>

目录

1 .介绍

    1.1 .要求记法和约定

    1.2 .术语

    1.3 .概观

2 . ID令牌

3 .认证

    3.1 .使用授权码流认证

        3.1.1 .授权码流步骤

        3.1.2 .授权端点

            3.1.2.1 .认证请求

            3.1.2.2 .验证请求验证

            3.1.2.3 .授权服务器验证最终用户

            3.1.2.4 .授权服务器获取最终用户许可/授权

            3.1.2.5 .成功的认证响应

            3.1.2.6 .验证错误响应

            3.1.2.7 .验证响应验证

        3.1.3 .令牌端点

            3.1.3.1 .令牌请求

            3.1.3.2 .令牌请求验证

            3.1.3.3 .成功的令牌响应

            3.1.3.4 .令牌错误响应

            3.1.3.5 .令牌响应验证

            3.1.3.6 . ID令牌

            3.1.3.7 . ID令牌验证

            3.1.3.8 .访问令牌验证

    3.2 .使用隐式认证流程

        3.2.1 .隐流步骤

        3.2.2 .授权端点

            3.2.2.1 .认证请求

            3.2.2.2 .验证请求验证

            3.2.2.3 .授权服务器验证最终用户

            3.2.2.4 .授权服务器获取最终用户许可/授权

            3.2.2.5 .成功的认证响应

            3.2.2.6 .验证错误响应

            3.2.2.7 .重定向的URI片段处理

            3.2.2.8 .验证响应验证

            3.2.2.9 .访问令牌验证

            3.2.2.10 . ID令牌

            3.2.2.11 . ID令牌验证

    3.3 .使用混合身份验证流程

        3.3.1 .混合流程步骤

        3.3.2 .授权端点

            3.3.2.1 .认证请求

            3.3.2.2 .验证请求验证

            3.3.2.3 .授权服务器验证最终用户

            3.3.2.4 .授权服务器获取最终用户许可/授权

            3.3.2.5 .成功的认证响应

            3.3.2.6 .验证错误响应

            3.3.2.7 .重定向的URI片段处理

            3.3.2.8 .验证响应验证

            3.3.2.9 .访问令牌验证

            3.3.2.10 .授权码验证

            3.3.2.11 . ID令牌

            3.3.2.12 . ID令牌验证

        3.3.3 .令牌端点

            3.3.3.1 .令牌请求

            3.3.3.2 .令牌请求验证

            3.3.3.3 .成功的令牌响应

            3.3.3.4 .令牌错误响应

            3.3.3.5 .令牌响应验证

            3.3.3.6 . ID令牌

            3.3.3.7 . ID令牌验证

            3.3.3.8 .访问令牌

            3.3.3.9 .访问令牌验证

4 .由第三方发起登录

5 .声明

    5.1 .标准声明

        5.1.1 .地址声明

        5.1.2 .其他声明

    5.2 .声明语言和脚本

    5.3 .使用者资讯端点

        5.3.1 .使用者资讯需求

        5.3.2 .成功的UserInfo响应

        5.3.3 .使用者资讯错误响应

        5.3.4 .使用者资讯响应验证

    5.4 .使用范围值，要求声明

    5.5 .使用“声明”请求参数要求声明

        5.5.1 .个人声明要求

            5.5.1.1 .要求“ ACR ”声明

        5.5.2 .语言和文字为个人声明

    5.6 .声明类型

        5.6.1 .普通债权

        5.6.2 .汇总和分布式声明

            5.6.2.1 .聚合权利要求的示例

            5.6.2.2 .分布式权利要求的示例

    5.7 .权利要求稳定性和唯一性

6 .通过请求参数作为JWTs

    6.1 .按值传递一个请求对象

        6.1.1 .请使用“请求”请求参数

    6.2 .按引用传递对象请求

        6.2.1 . URL引用的请求对象

        6.2.2 .要求使用“ REQUEST\_URI ”请求参数

        6.2.3 .授权服务器取指请求对象

        6.2.4 . “ REQUEST\_URI ”的理由

    6.3 .智威汤逊验证基于请求

        6.3.1 .加密的请求对象

        6.3.2 .签署请求对象

        6.3.3 .请求参数组装和验证

7 .自发行OpenID提供者

    7.1 .自发行OpenID提供者发现

    7.2 .自发行OpenID提供者报名

        7.2.1 .与“注册”请求参数提供信息

    7.3 .自发行OpenID提供者请求

    7.4 .自发行OpenID提供者响应

    7.5 .自发行标识令牌验证

8 .主题标识符类型

    8.1 .成对标识算法

9 .客户端身份验证

10 .签名和加密

    10.1 .签名

        10.1.1 .不对称签名密钥轮换

    10.2 .加密

        10.2.1 .非对称加密密钥的旋转

11 .离线访问

12 .使用刷新令牌

    12.1 .刷新请求

    12.2 .成功刷新响应

    12.3 .刷新错误响应

13 .序列化

    13.1 .查询字符串序列化

    13.2 .序列化形式

    13.3 . JSON序列化

14 .字符串操作

15 .实施注意事项

    15.1 .强制性地对所有的OpenID提供商特点

    15.2 .强制实现动态的OpenID提供商特点

    15.3 .发现和注册

    15.4 .强制实现的功能为信任方

    15.5 .实现说明

        15.5.1 .授权代码实现说明

        15.5.2 .杜撰实现说明

        15.5.3 .重定向的URI片段操作实现说明

    15.6 .兼容性说明

        15.6.1 .准决赛IETF规范

        15.6.2 .谷歌“ ISS ”的价值

    15.7 .相关的规范和实施者指南

16 .安全注意事项

    16.1 .要求披露

    16.2 .服务器伪装

    16.3 .令牌制造/修改

    16.4 .访问令牌披露

    16.5 .服务器响应披露

    16.6 .服务器响应赖账

    16.7 .要求赖账

    16.8 .访问令牌重定向

    16.9 .令牌重用

    16.10 .窃听或泄漏授权码（中学验证器捕捉）

    16.11 .标记替换

    16.12 .时序攻击

    16.13 .其他加密相关的攻击

    16.14 .签名和加密订购

    16.15 .发行人标识符

    16.16 .隐流威胁

    16.17 . TLS的要求

    16.18 .访问令牌和刷新令牌的寿命

    16.19 .对称密钥熵

    16.20 .需要签名请求

    16.21 .需要加密的请求

17 .隐私考量

    17.1 .个人身份信息

    17.2 .数据访问监控

    17.3 .相关

    17.4 .离线访问

18 . IANA考虑

    18.1 . JSON网站令牌债权登记

        18.1.1 .注册表内容

    18.2 . OAuth的参数注册

        18.2.1 .注册表内容

    18.3 . OAuth的扩展错误登记

        18.3.1 .注册表内容

19 .参考文献

    19.1 .规范性引用文件

    19.2 .翔实的参考资料

附录A.授权示例

    A.1 .例如使用RESPONSE\_TYPE =代码

    A.2 .例如使用RESPONSE\_TYPE = id\_token

    A.3 .使用RESPONSE\_TYPE例如= id\_token令牌

    A.4 .例如使用RESPONSE\_TYPE =代码id\_token

    A.5.例如使用RESPONSE\_TYPE =令牌代码

    A.6 .使用示例RESPONSE\_TYPE =代码id\_token令牌

    A.7 . RSA密钥用于实例

附录B.致谢

附录C.声明

§作者地址

1 .介绍

OpenID的连接1.0是对的OAuth 2.0 [ RFC6749 ]协议之上一个简单的身份层.它使客户端验证最终用户的基础上由一个授权服务器，以及进行身份验证的身份，以获取有关最终用户在一个可互操作和REST的方式基本个人资料信息.

OpenID的核心连接1.0规范定义了核心的OpenID Connect功能：身份验证建立在OAuth的2.0之上，并利用索赔沟通一下最终用户的信息.它也描述了使用OpenID连接的安全性和隐私的考虑.

作为背景，在2.0的OAuth授权框架[ RFC6749 ]和OAuth 2.0承载令牌用法[ RFC6750 ]规范提供了第三方应用程序，以获取和使用有限的访问HTTP资源的总体框架.他们定义的机制来获取和使用访问令牌来访问资源，但没有定义标准的方法来提供身份信息.值得注意的是，没有分析的OAuth 2.0 ，它不能提供有关最终用户的认证信息.我们希望读者熟悉这些规范.

OpenID的连接实现身份验证的扩展的OAuth 2.0授权过程.使用这个扩展的要求由客户通过在授权请求中OpenID的范围值.有关执行的认证信息中返回一个JSON的Web令牌（ JWT ） [智威汤逊]称为一个ID令牌（见第2节） .实施的OpenID的OAuth连接2.0认证服务器也被称为的OpenID提供商（OPS ） .使用OpenID的OAuth连接2.0客户端也被称为依赖方（ RPS） .

本规范假定依赖方已经获得了有关OpenID提供者的配置信息，包括其授权端点和令牌端点的位置.通常是通过发现获得这个信息，如在OpenID的连接愉1.0 [ OpenID.Discovery ]中描述的，或者可以通过其他机制获得.

同样地，本规范假定依赖方已经获得足够的凭据，并提供使用OpenID提供所需的信息.这是通过动态注册正常完成，如在OpenID的连接动态客户注册1.0 [ OpenID.Registration ]中描述的，或者可以通过其他机制获得.

 TOC

1.1 .要求记法和约定

中的关键字“必须”，“必须不”，“要求” ， “应” ， “不应” ， “应该” ， “不应该” ，“建议” ， “或许”，本文件中“可选”在RFC 2119 [ RFC2119 ]中解释.

在这份文件中的. txt文本，值被引用，以表明他们是按字面解释.当使用在协议消息这些值，报价必须不能被用作值的一部分.在本文档的HTML版本，可以从字面上理解的值是通过使用这个固定宽度字体显示.

JSON的网络签名（ JWS ） [ JWS ]和JSON的Web加密（ JWE ） [ JWE ]数据结构在本规范的所有用途使用的JWS契约序列化或JWE契约序列化;的JWS JSON序列化和JWE JSON序列不使用.

1.2 .术语

本规范使用术语“访问令牌” ， “授权码” ， “授权端点” ， “授权格兰特” ， “授权服务器” ， “客户” ， “客户身份验证” ， “客户身份识别” ， “客户端秘密” ， “格兰特类型“ ， ”保护资源“ ， ”重定向URI “ ， ”刷新令牌“ ， ”资源所有者“ ， ”资源服务器“ ， ”反应型“和”令牌端点“通过的OAuth 2.0 [ RFC6749 ]中定义，术语”声明名称“ ， ”索赔的价值“ ， ” JSON的Web令牌（ JWT ） “和”嵌套智威汤逊用户代理“由RFC 2616 [ RFC2616 ]定义的， ”由JSON的Web令牌（ JWT ） [智威汤逊] ，术语定义“术语“应答模式”通过OAuth的2.0多重响应类型编码实践[ OAuth.Responses ]中定义.

该规范还定义了以下条款：

认证

使用过程中达到足够的信心，实体及提交身份之间的绑定.

认证请求

利用OpenID的连接定义的扩展参数和范围要求的最终用户的授权服务器，这是一个OpenID提供商连接进行身份验证，在客户端，这是一个连接的OpenID依赖方的OAuth 2.0授权请求.

鉴定环境

信息，依赖方可以要求它使一种权利决定就认证响应之前.这样的环境可以包括，但不限于，使用的实际验证方法或保证诸如ISO / IEC 29115 [ ISO29115 ]实体认证保证水平的水平.

验证上下文类

置的，被认为是等同于彼此在特定上下文验证方法或程序.

验证上下文类参考

标识符的验证上下文类.

授权码流

的OAuth 2.0流程中的授权码从授权端点返回的所有令牌从令牌端点返回.

授权申请

由[ RFC6749 ]中定义的OAuth 2.0授权请求.

要求

资料片断言有关的实体.

声明类型

语法，用于表示一个索赔的价值.该规范定义了标准，聚集和分布的声明类型.

声明提供

服务器可以返回有关一个实体索赔.

凭据

数据表示为要使用的身份或其他资源的权利的证据.

终端用户

人类的参与者.

实体

一些具有独立和独特的存在，并且可以在一个范围内加以确定.最终用户是一个实体的一个例子.

了基本要求

由客户指定为必要的，以确保由最终用户所请求的特定任务的平滑授权权利要求的经验.

混合流程

的OAuth 2.0流程中的授权码从授权端点返回，一些令牌从授权端点返回的，有些则是从令牌端点返回.

ID令牌

JSON的Web令牌（ JWT ） [智威汤逊] ，其中包含身份验证事件的索赔.它可能包含其他诉讼请求.

识别码

值，它唯一的特点实体在特定的上下文中.

特性

设置与一个实体的属性.

隐流

的OAuth 2.0流程中的所有令牌从授权端点回来，无论是令牌端点，也不是一个授权码被使用.

发行者

实体发出一组声明.

发行人标识符

核查的标识符为发行人.发卡机构标识是使用包含配置，主机，以及可选的端口号和路径组件，没有查询或片段组成部分https方案区分大小写的URL .

信息

请求或一个OpenID依赖方和OpenID提供者之间的反应.

OpenID提供者（ OP ）

的OAuth 2.0授权服务器，能够进行身份验证的最终用户，并提供索赔的依赖方对认证活动和最终用户的.

请求对象

智威汤逊，它包含了一组请求参数作为其索赔.

请求URI

引用包含请求对象的资源的URL .请求URI的内容必须是可检索的授权服务器.

成对假名标识符（ PPID ）

标识符标识实体的依赖方无法在其他依赖方相关联的实体的PPID .

个人可识别信息（PII ）

信息是：（a ）可以用来识别自然人向谁这样的信息涉及，或（ b）是或可能是直接或间接与自然人为之此类信息涉及.

依赖方（ RP ）

OAuth的2.0客户端应用程序需要最终用户验证，并从一个OpenID提供商索赔.

部门识别码

主机使用的依托党的组织，是一个输入的成对主题的标识符，依靠党的计算URL的组成部分.

自发行OpenID提供者

个人，自托管OpenID提供者颁发自签名的ID令牌.

主题标识

本地唯一和永远不会重新分配的标识符在发行人内部的最终用户，其目的是通过客户端使用.

使用者资讯端点

受保护的资源，通过客户端呈现一个访问令牌时，将返回有关最终用户通过相应的授权代表格兰特授权信息.使用者资讯端点URL必须使用https方案，并可能包含端口，路径和查询参数的组件.

验证

工艺打算建立一个结构的稳健性或正确性.

验证

流程旨在测试或证明的事实或价值的真实性或准确性.

自愿索赔

由客户指定的作为是有用的，但不是必须由最终用户所要求的特定的任务索赔.

重要提示给读者：本节中的术语定义本规范的规范性部分，在实施征收的要求.所有的大写单词本说明书中的文字，如“发行人标识符” ，引用这些定义的术语.每当读者遇到他们，在本节中找到它们的定义必须遵循的.

对于一些使用的术语的更多背景信息，请参阅互联网安全术语，第2版[ RFC4949 ] ， ISO / IEC 29115实体认证保证[ ISO29115 ] ，和ITU-T X.1252 [ X.1252 ] .

1.3.概观

OpenID的连接协议，在抽象的，遵循以下步骤.

对RP（客户端）发送一个请求到OpenID提供者（OP）.

OP的验证最终用户并获得授权.

OP的响应一个ID标记，通常一个访问令牌.

RP可以发送一个请求访问令牌的使用者资讯端点.

使用者资讯端点传回最终用户索赔.

这些步骤示于下图：

+--------+ +--------+

| | | |

| |---------(1) AuthN Request-------->| |

| | | |

| | +--------+ | |

| | | | | |

| | | End- |<--(2) AuthN & AuthZ-->| |

| | | User | | |

| RP | | | | OP |

| | +--------+ | |

| | | |

| |<--------(3) AuthN Response--------| |

| | | |

| |---------(4) UserInfo Request----->| |

| | | |

| |<--------(5) UserInfo Response-----| |

| | | |

+--------+ +--------+

2 . ID令牌

主要的扩展，支持OpenID使得连接到的OAuth 2.0 ，使最终用户进行身份验证是ID令牌的数据结构.该ID令牌是一个包含最终用户通过一个授权服务器的身份验证声明使用一个客户端，以及潜在的其他索赔要求时，安全令牌.该ID令牌被表示为一个JSON的Web令牌（ JWT ） [智威汤逊] .

下面的权利要求的ID令牌中所使用的OpenID连接所有的OAuth 2.0流量使用：

ISS

必需的.发行人的标识符响应的发行人.国际空间站值是使用包含配置，主机，以及可选的端口号和路径组件，没有查询或片段组成部分https方案区分大小写的URL .

子

必需的.主题标识符.本地唯一和永远不会重新分配的标识符在发行人内部的最终用户，其目的是通过客户端使用，例如， 24400320或AItOawmwtWwcT0k51BayewNvutrJUqsvl6qs7A4 .它必须不超过255个ASCII字符.子值是一个字符串，区分大小写.

AUD

必需的.观众（ s）表示，这个ID令牌是用于.它必须包含的OAuth 2.0的client\_id依赖方的作为一个观众的价值.它可能还包含标识符其他观众.在一般情况下，澳元的价值是区分大小写的字符串数组.在当有一个观众共同的特殊情况下，澳元的价值可以是单个字符串，区分大小写.

EXP

必需的.到期时间或以后该ID令牌必须不被接受处理.此参数的处理要求当前日期/时间必须在值中列出的截止日期/时间前.实施者可以提供一些小的回旋余地，超过几分钟平时没有更多的，占时钟歪斜.它的值是代表秒数从1970年-01- 01T0数量的JSON数：作为衡量UTC直到日期/时间0时00分Z的.请参阅RFC 3339 [ RFC3339 ]香港对一般和UTC在特定的日期/时间的详细信息.

IAT

必需的.时间点的智威汤逊发出.它的值是代表秒数从1970年-01- 01T0数量的JSON数：作为衡量UTC直到日期/时间0时00分Z的.

auth\_time

最终用户的身份验证发生的时间.它的值是代表秒数从1970年-01- 01T0数量的JSON数：作为衡量UTC直到日期/时间0时00分Z的.当MAX\_AGE请求时或当auth\_time要求作为了基本要求，那么这种说法是必需的，否则，将其列入是可选的. （该auth\_time索赔语义对应的OpenID 2.0 PAPE [ OpenID.PAPE ] auth\_time响应参数. ）

随机数

字符串值用来与一个ID令牌客户端会话相关联，并减轻重放攻击.该值是通过未修改的认证请求的ID令牌.如果出现在ID令牌，客户端必须验证随机数索赔值等于在认证请求中发送的随机数参数的值.如果出现在认证请求，授权服务器必须包含一个随机数索赔在ID令牌与索赔值是在验证请求发送的随机数值.授权服务器应该对使用的nonce值，不执行其他处理.该nonce值是一个字符串，区分大小写.

ACR

可选的.验证上下文类参考.字符串指定的认证上下文类参考值，标识的认证进行满意的验证上下文类.值“0 ”表示最终用户的认证不符合ISO / IEC 29115 [ ISO29115 ] 1级的要求.使用长寿命的浏览器cookie验证，举例来说，是一个例子，其中使用“级别0 ”是恰当的.认证与0水平不应被用于授权访问任何货币价值的任何资源. （这相当于支持OpenID 2.0 PAPE [ OpenID.PAPE ] nist\_auth\_level 0 . ）的绝对URI或RFC 6711 [ RFC6711 ]注册名称应作为ACR值;注册名不能用不同的含义比使用哪个已注册.使用这种说法缔约方将需要同意所使用的值的含义，这可能是特定上下文. ACR值是一个字符串，区分大小写.

AMR

可选的.验证方法参考. JSON数组是标识符在身份验证中使用的身份验证方法的字符串.例如，值可能表明，这两个密码， OTP身份验证方法被使用.对AMR权利要求中使用的特定的值的定义是超出了本说明书的范围.使用这种说法缔约方将需要同意所使用的值的含义，这可能是特定上下文.在AMR值是区分大小写的字符串数组.

AZP

可选的.授权方 - 即本ID令牌发出的一方.如果存在，它必须包含这个党的OAuth的2.0客户端的ID .当ID令牌有一个单一的受众价值以及受众比授权方不同的这种说法是只有必要的.它可以被包括即使当授权方是相同的唯一的观众.该AZP值是包含StringOrURI值的字符串，区分大小写.

ID标记可能包含其他诉讼请求.所使用的任何索赔是不被理解必须被忽略.见本规范定义额外的索赔第3.1.3.6 ， 3.3.2.11 ，5.1和7.4 .

ID标记必须使用JWS [ JWS ]签署，并选择性地双方签字，然后使用JWS [ JWS ]和JWE [ JWE ]分别，从而提供认证性，完整性，不可抵赖性，以及可选，保密，每科16.14加密.如果ID令牌是加密的，它必须被签署，然后加密，其结果是一个嵌套智威汤逊，如[智威汤逊]中定义. ID令牌必须不能使用none作为ALG值，除非响应类型使用返回没有ID令牌的授权端点（使用授权码流时，如）与客户明确要求在注册时使用没有.

ID令牌不应该使用的JWS或JWE x5u ， X 5 C ， jku ，或JWK头参数字段.相反，引用的按键传达提前使用发现和注册参数，每节10 .

下面是一个ID令牌集合索赔（智威汤逊权利要求书）中的非规范性的例

{

"iss": "https://server.example.com",

"sub": "24400320",

"aud": "s6BhdRkqt3",

"nonce": "n-0S6\_WzA2Mj",

"exp": 1311281970,

"iat": 1311280970,

"auth\_time": 1311280969,

"acr": "urn:mace:incommon:iap:silver"

}

3 .认证

OpenID的连接进行认证登录的最终用户或确定已经登录的终端用户英寸的OpenID连接返回由服务器向客户端以安全的方式进行身份验证的结果，从而使客户可以依靠它。出于这个原因，客户端被称为依赖方（RP）在这种情况下。

认证结果在ID令牌返回，在第2节中定义。它具有债权表达这样的信息作为发行人，主体标识，当认证过期等。

认证可以遵循三条路径之一：授权码流（ RESPONSE\_TYPE =代码） ，隐流量（ RESPONSE\_TYPE = id\_token令牌或RESPONSE\_TYPE = id\_token ） ，或混合流（使用的OAuth 2.0中定义的其它响应类型的值多重响应类型编码实践[ OAuth.Responses ] ） 。在流量确定的ID令牌和访问令牌返回给客户端。

三流的特征总结在下面的非规范表。该表的目的是提供一些指导其流在特定情况下进行选择。

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Authorization Code Flow** | **Implicit Flow** | **Hybrid Flow** |
| All tokens returned from Authorization Endpoint | no | yes | no |
| All tokens returned from Token Endpoint | yes | no | no |
| Tokens not revealed to User Agent | yes | no | no |
| Client can be authenticated | yes | no | yes |
| Refresh Token possible | yes | no | yes |
| Communication in one round trip | no | yes | no |
| Most communication server-to-server | yes | no | varies |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **OpenID Connect Authentication Flows** 所用的流量是由包含在授权请求的RESPONSE\_TYPE值确定。这些RESPONSE\_TYPE值选择这些流量：   |  |  | | --- | --- | | **"response\_type" value** | **Flow** | | code | Authorization Code Flow | | id\_token | Implicit Flow | | id\_token token | Implicit Flow | | code id\_token | Hybrid Flow | | code token | Hybrid Flow | | code id\_token token | Hybrid Flow |      |  | | --- | | **OpenID Connect "response\_type" Values** | |

但所有的代码响应类型的值，这是由的OAuth2.0[RFC6749]中定义，在OAuth的2.0多重响应类型编码实践[OAuth.Responses]规范中定义。注意：当的OAuth2.0还定义了令牌响应类型值的隐流，OpenID的连接不使用此响应类型，因为没有ID令牌将被退回。

3.1 使用授权码流认证

本节将介绍如何使用授权码流进行认证。当使用授权码流，所有的令牌从令牌端点返回。

该授权码流返回一个授权码给客户端，这样就可以直接换取一个ID令牌和访问令牌。这提供了不暴露任何标记的用户代理和其他可能的恶意应用程序能够访问用户代理的利益。授权服务器也可以交换授权码的访问令牌之前验证客户端。该授权码流适用于客户端，可以牢固地保持自己和授权服务器之间的客户端秘密。

3.1.1 授权码流步骤

该授权码流经过以下步骤。

客户端准备包含所需的请求参数的认证请求。

客户端将请求发送到授权服务器。

授权服务器验证的最终用户。

授权服务器获取最终用户许可/授权。

授权服务器发送最终用户返回到客户端用一个授权码。

客户端请求使用授权码的令牌端点的响应。

客户端收到包含一个ID令牌和访问令牌的响应正文的响应。

客户端验证ID标记和检索最终用户的主题标识符。

3.1.2。授权端点

授权端点进行最终用户的认证。这是通过发送用户代理的授权服务器的授权端点进行验证和授权，利用OAuth的2.0定义的请求参数和其他参数和OpenID的连接定义的参数值来实现。

与授权端点的通信必须使用TLS。见16.17节有关使用TLS的详细信息。

3.1.2.1 。认证请求

认证请求是的OAuth 2.0授权请求，请求的最终用户通过授权服务器进行身份验证。

授权服务器必须支持使用HTTP的GET和授权端点在RFC 2616 [ RFC2616 ]定义的POST方法。客户端可以使用HTTP GET或POST方法来授权请求发送到授权服务器。如果使用HTTP GET方法，请求参数是使用URI查询字符串序列，每节13.1序列化。如果使用HTTP POST方法，请求参数均采用系列化形式连载，每节13.2 。

OpenID的连接使用与授权码流以下的OAuth 2.0请求参数：

范围

必需的。 OpenID的连接请求必须包含OpenID的范围值。如果OpenID的范围值不存在，该行为是完全不确定的。其他范围的值可能存在。使用范围的值不是由一个实现理解应该被忽略。见本规范定义额外的范围值第5.4和11 。

RESPONSE\_TYPE

必需的。 OAuth的2.0响应类型值，用于确定该授权处理的流程也可以使用，包括什么参数是从所使用的端点返回。当使用授权码流，这个值是代码。

CLIENT\_ID

必需的。 OAuth的2.0客户端标识符在授权服务器有效。

redirect\_uri

必需的。重定向的URI的响应将被发送。这个URI必须完全匹配的重定向URI值的客户端预注册的OpenID提供者之一，在[ RFC3986 ]中第6.2.1节中描述进行匹配（简单字符串比较） 。当使用该流程中，重定向URI应当使用https方案，但是，它可能会使用http方案，但前提是客户端类型是保密的，在的OAuth 2.0的第2.1节定义，并提供了OP允许使用HTTP重定向的URI在这种情况下。重定向的URI可能会使用一个替代方案，如一个旨在确定回调到一个本机应用程序。

态

推荐。不透明的值用于维护请求和回调之间的状态。通常情况下，跨站点请求伪造（CSRF ， XSRF ）缓解由加密绑定该参数的值与浏览器cookie来实现。

OpenID的连接也使用以下的OAuth 2.0请求参数，这是在的OAuth 2.0中定义的多重响应类型编码实践[ OAuth.Responses ] ：

response\_mode

可选的。通知用于返回从授权端点参数的机制的授权服务器。此使用这个参数时不建议使用，将被要求响应模式是为响应类型指定的默认??模式。

该规范还定义了以下请求参数：

随机数

可选的。字符串值用来与一个ID令牌客户端会话相关联，并减轻重放攻击。该值是通过未修改的授权请求的ID令牌。足够的熵必须包含在用于防止攻击者猜测值的随机数的值。为实现说明，请参见15.5.2节。

显示

可选的。 ASCII字符串值，指定授权服务器是如何显示的验证和同意，用户界面页面给最终用户。所定义的值是：

页面

授权服务器应该显示的验证和同意的UI采用了全用户代理页面观点是一致的。如果未指定显示参数，这是默认的显示模式。

弹出窗口

授权服务器应该显示的验证和同意UI并弹出用户代理窗口保持一致。在弹出的用户代理窗口应该是一个适当大小的登录为重点的对话，而不应掩盖它在弹出整个窗口。

触摸

授权服务器应该显示的验证和同意的UI与利用触摸界面的设备保持一致。

WAP

授权服务器应该显示的验证和同意的UI以“功能手机”类型的显示是一致的。

授权服务器也可以尝试检测用户代理的能力，并提出适当的显示。

提示

可选的。空间分隔的ASCII字符串值的情况下，指定授权服务器是否提示最终用户重新进行身份验证和同意敏感名单。所定义的值是：

没有

授权服务器不能显示任何认证或同意用户界面页。如果最终用户尚未认证或客户不具备所要求的索赔预先配置的同意或不符合其他条件来处理请求，则会返回错误。错误代码通常会被login\_required ， interaction\_required ，或在第3.1.2.6另一个代码定义。这可以作为一种方法来检查现有的认证和/或许可。

注册

授权服务器应提示最终用户重新进行身份验证。如果不能重新验证最终用户，它必须返回一个错误，通常login\_required 。

同意

授权服务器应该返回信息给客户端之前提示最终用户的同意。如果它不能获得批准，它必须返回一个错误，通常consent\_required 。

select\_account

授权服务器应提示最终用户选择一个用户帐户。这使最终用户谁拥有多个账户的授权服务器选择之间，他们可能有当前会话的多个帐户。如果它不能获得一个帐户选择的选择由最终用户做，它必须返回一个错误，通常account\_selection\_required 。

该提示参数可用于在客户端，以确保最终用户仍然是本作的当前会话或让人们关注到的请求。如果此参数包含未与任何其他值，则返回一个错误。

MAX\_AGE

可选的。最大验证年龄。指定允许经过的时间（秒）自从上一次最终用户积极认证的任择议定书。如果经过的时间大于该值，则OP必须尝试，积极重新验证最终用户。 （该MAX\_AGE请求参数对应的OpenID 2.0 PAPE [ OpenID.PAPE ] max\_auth\_age请求参数。 ）当MAX\_AGE时，该ID令牌返回必须包括auth\_time索赔的价值。

ui\_locales

可选的。最终用户的首选语言和用户界面脚本，表示为BCP47 [ RFC5646 ]语言标签值的空格分隔的列表，责令偏好。例如，值“ FR- CA FR烯”代表在加拿大使用偏爱法国，然后法国（没有区域的指定），其次是英语（没有区域的指定） 。如果某些要求的语言环境或全部不支持OpenID提供者的错误不应导致。

id\_token\_hint

可选的。先前由授权服务器发出的ID令牌传递作为对最终用户的当前或过去的身份验证会话的客户端提示。如果最终用户确定的ID令牌登录或者被记录在该请求，那么授权服务器返回一个肯定的答复，否则，它应该返回一个错误，如login\_required 。如果可能的话，一个id\_token\_hint应该存在时提示=没有被使用，如果它不是一个INVALID\_REQUEST错误可能被退回，但是，服务器应该尽可能响应成功，即使它不存在。授权服务器不需要被列为一个观众的ID令牌时，它被用作id\_token\_hint值。

如果从OP收到RP的ID令牌是加密的，使用它作为一个id\_token\_hint ，客户端必须解密包含在加密的ID令牌内签署的ID令牌。客户端可以在签订的ID令牌重新加密使用一个密钥，使得服务器能够解密的ID令牌认证服务器，并使用重新加密的ID标记为id\_token\_hint值。

login\_hint

可选的。提示有关登录标识符最终用户可以用它来登录（如果有必要）授权服务器。这个提示可以使用一个RP ，如果它首先要求的最终用户为他们的E-mail地址（或其他标识符） ，然后想通过该值作为提示，来发现的授权服务。它建议将提示值相匹配用于查找的值。该值也可以在为PHONE\_NUMBER索赔指定格式的电话号码。使用这个参数是左到OP的自由裁量权。

acr\_values

可选的。要求身份验证上下文类的参考值。空格分隔的字符串，该字符串指定授权服务器被要求用于处理此认证请求，与出现在优先顺序的值的ACR值。认证上下文班进行认证支付返回的ACR声明值，如第2规定。 ACR的索赔请求为自愿索赔由该参数。

其他参数可能会被发送。见第3.2.2 ， 3.3.2 ， 5.2 ， 5.5 ， 6 ，和7.2.1本规范定义额外的授权请求参数和参数值。

以下是客户端，这将触发用户代理进行身份验证请求授权端点（带线包装内的值用于显示目的）的非规范性的例如HTTP 302重定向响应：

HTTP/1.1 302 Found

Location: https://server.example.com/authorize?

response\_type=code

&scope=openid%20profile%20email

&client\_id=s6BhdRkqt3

&state=af0ifjsldkj

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

以下是将由用户代理发送，以响应由上述客户端的HTTP 302重定向响应（带线包装内的值用于显示目的）授权服务器的非规范性的要求：

GET /authorize?

response\_type=code

&scope=openid%20profile%20email

&client\_id=s6BhdRkqt3

&state=af0ifjsldkj

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb HTTP/1.1

Host: server.example.com

3.1.2.2 。验证请求验证

授权服务器必须验证收到的请求如下：

授权服务器必须根据的OAuth 2.0规范验证所有的OAuth 2.0的参数。

验证作用域参数存在，并且包含OpenID的范围值。 （如果没有OpenID的范围值存在，请求可能仍然是一个有效的OAuth 2.0的要求，但不是一个OpenID的连接请求。 ）

授权服务器必须验证所有必需的参数是否存在以及它们的使用符合本规范。

如果子（主题）索赔请求与该ID标记一个特定的值，授权服务器必须只发出一个积极的回应，如果最终用户确定该子值有一个活跃的会话授权服务器或已被验证为该请求的结果。针对不同的用户有一个ID令牌或访问令牌的授权服务器不能回答，即使他们有活动会话的授权服务器。这样的要求可以使用id\_token\_hint参数或通过请求一个特定的索赔值，如5.5.1节所述，如果索赔参数是支持的实施作。

在OAuth的2.0 [ RFC6749 ]中指定，授权服务器应该忽略无法识别的请求参数。

如果授权服务器遇到任何错误，它必须返回一个错误响应，每节3.1.2.6 。

3.1.2.3 。授权服务器验证最终用户

如果请求有效，授权服务器尝试验证最终用户或确定最终用户是否通过认证，这取决于使用的请求参数值。所使用的授权服务器进行身份验证的最终用户（如用户名和密码，会话Cookie等）的方法超出了本规范的范围。身份验证的用户界面可以由授权服务器被显示，这取决于所使用的请求参数值和所使用的认证方法。

授权服务器必须尝试验证最终用户在下列情况下：

最终用户尚未经过身份验证。

该认证请求中包含的值的登录提示参数。在这种情况下，授权服务器必须重新验证最终用户即使最终用户已经通过身份验证。

授权服务器不能与最终用户在下列情况下进行交互：

该认证请求中包含的值没有任何提示参数。在这种情况下，如果最终用户尚未经过身份验证或无法验证的默默授权服务器必须返回一个错误。

当与最终用户交互，授权服务器必须采用对跨站请求伪造和点击劫持的适当措施，在第10.12和OAuth 2.0 [ RFC6749 ]的10.13描述。

3.1.2.4 。授权服务器获取最终用户许可/授权

一旦最终用户进行身份验证，授权服务器必须发布信息依赖方之前获得一个授权决定。当使用的请求参数允许的，这可能是通过与最终用户，使得它清楚什么是被通过处理请求或其他方式（例如条件同意或同意建立一个交互式的对话通过以前的管理做了，同意） 。第2和5.3描述的信息发布机制。

3.1.2.5 。成功的认证响应

认证响应是一个OAuth的从OP的授权端点响应由RP发送的授权请求消息2.0授权响应消息中返回。

当使用授权码流，授权响应必须使用application/x-www-form-新增为查询参数在授权请求中指定的redirect\_uri返回中的OAuth 2.0 [ RFC6749 ]中第4.1.2节中定义的参数urlencoded格式，除非指定了不同的反应模式。

下面是使用这个流量（带线值内包仅用于显示目的），非规范性的例子成功的响应：

HTTP/1.1 302 Found

Location: https://client.example.org/cb?

code=SplxlOBeZQQYbYS6WxSbIA

&state=af0ifjsldkj

有关授权码的内容实现说明，请参见第15.5.1 。

3.1.2.6 。验证错误响应

身份验证错误响应是一个OAuth的从OP的授权端点响应由RP发送的授权请求消息2.0授权错误响应消息中返回。

如果最终用户拒绝请求或最终用户身份验证失败，则OP （授权服务器）通过使用的OAuth 2.0 [ RFC6749 ]的4.1.2.1节定义的错误响应参数通知RP （客户端） 。 （无关RFC 6749 HTTP错误使用相应的HTTP状态代码返回给用户代理。 ）

除非重定向URI是无效的，授权服务器返回客户端的URI的授权请求中指定的相应的错误和状态参数的重定向。其他参数不应该被返回。

除的OAuth 2.0的4.1.2.1节定义的错误代码，该规范还定义了下面的错误代码：

interaction\_required

授权服务器需要某种形式进行最终用户交互。当在认证请求的提示参数值是没有这个错误可能被退回，但认证请求无法不显示用户界面，最终用户交互完成。

login\_required

授权服务器需要最终用户的身份验证。当在认证请求的提示参数值是没有这个错误可能被退回，但认证请求无法不显示用户界面，最终用户认证完成。

account\_selection\_required

最终用户需要选择一个会话的授权服务器。最终用户可以在授权服务器与不同的相关帐户进行身份验证，但最终用户并没有选择一个会话。当在认证请求的提示参数值是没有这个错误可能是回来了，但验证请求不能没有显示用户界面提示输入一个会话使用完毕。

consent\_required

授权服务器要求最终用户同意。当在认证请求的提示参数值是没有这个错误可能被退回，但认证请求无法不显示用户界面，最终用户同意完成。

invalid\_request\_uri

在授权请求的REQUEST\_URI返回一个错误或包含无效数据。

invalid\_request\_object

请求参数包含无效的请求对象。

request\_not\_supported

该任择议定书不支持使用在第6节中定义的请求参数。

request\_uri\_not\_supported

该任择议定书不支持使用在第6节中定义的REQUEST\_URI参数。

registration\_not\_supported

该任择议定书不支持使用在7.2.1节中定义的注册参数。

错误响应参数如下：

错误

必需的。错误代码。

error\_description

可选的。人类可读的ASCII编码错误的文字说明。

error\_uri

可选的。一个网页，其中包含有关该错误的附加信息的URI。

态

的OAuth 2.0的状态值。如果所需的授权请求包括状态参数。设置为从客户端接收的值。

当使用授权码流，误差响应参数添加到重定向URI的查询部分，除非指定了不同的反应模式。

下面是使用这个流量（带线值内包仅用于显示目的），非规范性的错误响应：

HTTP/1.1 302 Found

Location: https://client.example.org/cb?

error=invalid\_request

&error\_description=

Unsupported%20response\_type%20value

&state=af0ifjsldkj

3.1.2.7 。验证响应验证

当使用授权码流，客户端必须根据RFC 6749验证响应，特别是第4.1.2和10.12 。

3.1.3 。令牌端点

为了获得访问令牌，一个ID令牌和可选的刷新令牌，对RP （客户端）发送一个令牌请求令牌端点获得令牌响应，如2.0的OAuth [ RFC6749 ]第3.2节所述，当使用授权码流。

与令牌端点的通信必须使用TLS 。见16.17节有关使用TLS的详细信息。

3.1.3.1 。令牌请求

客户端发出请求令牌使用grant\_type值authorization\_code展示其授权格兰特（在一个授权码的形式）到令牌端点，如的OAuth 2.0 [ RFC6749 ]中第4.1.3节中描述。如果客户端是一个机密客户，那么它必须使用注册为它的client\_id的身份验证方法验证令牌端点，如第9节。

使用HTTP POST方法和形式序列化，每节13.2 ，在OAuth的2.0 [ RFC6749 ]中第4.1.3节中描述的客户端发送的参数令牌端点。

下面是一个令牌请求的非规范性的例子（带线值的范围内包裹仅用于显示目的） ??：

POST /token HTTP/1.1

Host: server.example.com

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

Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

grant\_type=authorization\_code&code=SplxlOBeZQQYbYS6WxSbIA

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

3.1.3.2 。令牌请求验证

授权服务器必须验证令牌请求如下：

验证客户端是否发出客户端凭证，或者如果它使用另一个客户端身份验证方法，每科9 。

确保授权码颁发给经过身份验证的客户端。

验证授权码是有效的。

如果可能的话，请验证授权码以前尚未使用。

确保redirect\_uri参数值是相同的，已包含在初始授权请求的redirect\_uri参数值。如果在只有一个注册redirect\_uri值redirect\_uri参数值不存在，则授权服务器可能会返回一个错误（因为客户端应包括参数），也可以继续操作没有错误（因为OAuth的2.0许可证，可以省略该参数在这种情况下） 。

验证授权码用于在响应一个连接的OpenID认证请求发出（这样，一个ID令牌将从令牌端点返回） 。

3.1.3.3 。成功的令牌响应

接收并验证从客户端有效，并授权令牌请求之后，授权服务器返回一个成功的响应，其中包含一个ID令牌和访问令牌。在成功响应的参数在的OAuth 2.0 [ RFC6749 ]中第4.1.4节中定义。该反应使用的应用程序/ json的媒体类型。

OAuth的2.0 token\_type响应参数值必须是不记名，在OAuth的2.0承载令牌用法[ RFC6750 ]中指定的，除非另一个令牌类型已经协商与客户。服务器应该支持承载令牌类型，使用其他令牌类型是本规范的范围之内。

除了通过OAuth的2.0规定的响应参数，下面的参数必须被包含在响应中：

id\_token

通过验证的会话ID令牌值相关联。

包含令牌，秘密或其他敏感信息的所有令牌响应必须包括以下HTTP响应头字段和值：

|  |  |
| --- | --- |
| **Header Name** | **Header Value** |
| Cache-Control | no-store |
| Pragma | no-cache |

|  |
| --- |
| **HTTP Response Headers and Values** |

下面是一个成功的响应令牌的非规范性的例子。该ID令牌签名的例子可以在附录A.7重点进行验证。

HTTP/1.1 200 OK

Content-Type: application/json

Cache-Control: no-store

Pragma: no-cache

{

"access\_token": "SlAV32hkKG",

"token\_type": "Bearer",

"refresh\_token": "8xLOxBtZp8",

"expires\_in": 3600,

"id\_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc

yI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4Mjg5

NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiAibi0wUzZ

fV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDEzMTEyODA5Nz

AKfQ.ggW8hZ1EuVLuxNuuIJKX\_V8a\_OMXzR0EHR9R6jgdqrOOF4daGU96Sr\_P6q

Jp6IcmD3HP99Obi1PRs-cwh3LO-p146waJ8IhehcwL7F09JdijmBqkvPeB2T9CJ

NqeGpe-gccMg4vfKjkM8FcGvnzZUN4\_KSP0aAp1tOJ1zZwgjxqGByKHiOtX7Tpd

QyHE5lcMiKPXfEIQILVq0pc\_E2DzL7emopWoaoZTF\_m0\_N0YzFC6g6EJbOEoRoS

K5hoDalrcvRYLSrQAZZKflyuVCyixEoV9GfNQC3\_osjzw2PAithfubEEBLuVVk4

XUVrWOLrLl0nx7RkKU8NXNHq-rvKMzqg"

}

在OAuth的2.0[RFC6749]中指定，客户端应该忽略无法识别的响应参数。

3.1.3.4。令牌错误响应

如果令牌请求是无效的或未经授权，授权服务器构建的错误响应。令牌错误响应的参数定义为的OAuth2.0[RFC6749]的5.2节。 HTTP响应体采用与400 HTTP响应代码的应用程序/ json的媒体类型。

下面是一个非规范的例子令牌错误响应：

HTTP/1.1 400 Bad Request

Content-Type: application/json

Cache-Control: no-store

Pragma: no-cache

{

"error": "invalid\_request"

}

3.1.3.5 。令牌响应验证

客户端必须验证令牌响应如下：

按照验证规则在RFC 6749 ，尤其是第5.1和10.12 。

按照第3.1.3.7标识令牌的验证规则。

按照第3.1.3.8访问令牌的验证规则。

3.1.3.6 。 ID令牌

该ID标记的内容是如第2节所述。当使用授权码流，理由如下： ID令牌声明这些额外的要求：

at\_hash

可选的。访问令牌的散列值。它的值是ACCESS\_TOKEN值，其中使用的哈希算法是在ID令牌的JWS [ JWS ]标头的ALG参数中使用的哈希算法的ASCII码表示的八位位组的散列最左边的一半base64url编码。举例来说，如果ALG是RS256 ，散列ACCESS\_TOKEN值与SHA-256 ，然后采取最左边的128位和base64url编码它们。该at\_hash值是一个字符串，区分大小写。

3.1.3.7 。 ID令牌验证

客户端必须验证ID令牌的令牌响应下列方式：

如果ID令牌进行加密，使用该客户端注册过程中指定的密钥和算法的运算是使用到的ID令牌进行加密解密。如果加密进行了谈判与OP的报名时间和ID令牌是不加密的， RP应拒绝它。

发行人的标识符OpenID提供者（这是典型的发现过程中获得的）必须在国际空间站（发行人）索赔的值完全匹配。

客户端必须验证澳元（观众）索赔包括确定国际空间站（发行人）索赔作为一个观众在发行人注册了CLIENT\_ID价值。澳元（观众）索赔可能包含多个元素的数组。该ID令牌必须被拒绝如果ID令牌没有列出客户端作为有效的观众，或者如果它包含不是由受信任的客户端附加的观众。

如果ID令牌包含多个观众，客户应确认的AZP索赔存在。

如果AZP （授权方）索赔存在，客户应确认其CLIENT\_ID是索赔的价值。

如果ID令牌是通过客户端和令牌端点（它是在这个流程）之间的直接通信接收时， TLS服务器验证可能会被用来验证发行人到位检查令牌签名。客户端必须根据JWS [ JWS ]使用在智威汤逊ALG头参数中指定的算法验证所有其他的ID令牌的签名。客户端必须使用由发行人提供的密钥。

该ALG值应为RS256的默认或注册过程中的id\_token\_signed\_response\_alg参数通过客户端发送的算法。

如果智威汤逊ALG头参数使用基于MAC的算法，如HS256 ， HS384 ， HS512或对应于包含在澳币的client\_id的CLIENT\_SECRET的UTF-8表示的八位位组（观众）索赔被用作验证密钥签名。对于基于MAC的算法，该行为是不确定的，如果澳元是多值或者如果AZP值存在比澳元的价值不同。

当前时间必须由EXP索赔所代表的时间之前。

IAT的权利要求可以被用来拒绝发出太远离当前时间标记，限制的时间随机数需要被存储，以防止攻击的量。可接受的范围是特定的客户端。

如果一个nonce值被发送的认证请求，一个随机数索赔必须存在且值检查，以确认它是相同的值被发送的认证请求的人。客户应检查重放攻击的随机数值。用于检测重放攻击的精确的方法是客户端特有的。

如果请求的ACR索赔，客户应检查所声称的索赔值是合适的。 ACR索赔值的含义和处理超出范围本规范。

如果请求的auth\_time索赔，无论是通过这种索赔的具体要求或使用的max\_age参数，客户端应该检查auth\_time索赔的价值，并要求重新验证，如果它决定了太多的时间，因为最后的最终用户认证已过。

3.1.3.8 。访问令牌验证

当使用授权码流，如果ID令牌包含at\_hash索赔，客户端可以使用它来验证访问令牌以同样的方式作为隐流，如第3.2.2.9定义，但使用ID令牌和访问令牌的令牌端点返回。

3.2 。使用隐式认证流程

本节将介绍如何使用隐式流量进行认证。当使用隐式流程，所有的令牌从授权端点返回;令牌端点不使用。

隐流主要用于通过使用脚本语言在浏览器中实现的客户端。访问令牌和ID令牌是直接返回给客户端，这可能会暴露他们的最终用户，并且有机会获得最终用户的用户代理应用程序。授权服务器不执行客户端身份验证。

3.2.1 。隐流步骤

内隐流遵循以下步骤：

客户端准备包含所需的请求参数的认证请求。

客户端将请求发送到授权服务器。

授权服务器验证的最终用户。

授权服务器获取最终用户许可/授权。

授权服务器发送最终用户返回到一个ID令牌的客户端，如果请求，访问令牌。

客户端验证ID标记和检索最终用户的主题标识符。

3.2.2 。授权端点

当使用隐流，授权端点中使用的相同的方式进行授权码流，如在3.1.2节中定义，除本节中指明的差异。

3.2.2.1 。认证请求

如第3.1.2.1定义的认证请求被提出，不同的是这些验证请求参数的使用方法如下：

RESPONSE\_TYPE

必需的。 OAuth的2.0响应类型值，用于确定该授权处理的流程也可以使用，包括什么参数是从所使用的端点返回。当使用隐流，此值id\_token令牌或id\_token 。这两个值的含义中的OAuth 2.0多重响应类型编码实践[ OAuth.Responses ]中定义。当值id\_token没有访问令牌返回。

注意：当的OAuth 2.0还定义了令牌响应类型值的隐流， OpenID的连接不使用此响应类型，因为没有ID令牌将被退回。

redirect\_uri

必需的。重定向的URI的响应将被发送。这个URI必须完全匹配的重定向URI值的客户端预注册的OpenID提供者之一，在[ RFC3986 ]中第6.2.1节中描述进行匹配（简单字符串比较） 。当使用该流程中，重定向的URI必须不使用http方案，除非客户是本机应用程序，在这种情况下，它可能会使用http:scheme用localhost作为主机名。

随机数

必需的。字符串值用来与一个ID令牌客户端会话相关联，并减轻重放攻击。该值是通过未修改的认证请求的ID令牌。足够的熵必须包含在用于防止攻击者猜测值的随机数的值。为实现说明，请参见15.5.2节。

下面是一个使用将由用户代理发送到响应的客户端相应的HTTP 302重定向响应（带线包装内的值用于显示目的）授权服务器的隐流的非规范性的要求：

GET /authorize?

response\_type=id\_token%20token

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile

&state=af0ifjsldkj

&nonce=n-0S6\_WzA2Mj HTTP/1.1

Host: server.example.com

3.2.2.2 。验证请求验证

当使用隐流，该认证请求中相同的方式确认为授权码流，正如在第3.1.2.2中定义。

3.2.2.3 。授权服务器验证最终用户

当使用隐流，最终用户身份验证的相同的方式进行的授权码流，正如在第3.1.2.3中定义。

3.2.2.4 。授权服务器获取最终用户许可/授权

当使用隐流，以相同的方式，获得最终用户同意作为授权码流，如在第3.1.2.4节中定义。

3.2.2.5 。成功的认证响应

当使用隐式流程，认证响应均采用相同的方式作为授权码流，如第3.1.2.5规定，除本节规定的差异。

当使用隐式流程，所有的响应参数添加到重定向URI的片段组成部分，在2.0的OAuth指定多个响应类型的编码实践[ OAuth.Responses ] ，除非指定了不同的反应模式。

这些参数是从授权端点返回：

ACCESS\_TOKEN

2.0的OAuth访问令牌。这回，除非使用的RESPONSE\_TYPE值id\_token 。

token\_type

的OAuth 2.0令牌类型值??。该值必须是记名或客户已协商与授权服务器的另一个token\_type值。客户端实现此配置文件中必须支持的OAuth 2.0承载令牌用法[ RFC6750 ]规范。此配置文件仅描述了承载的令牌。这回在同一案件ACCESS\_TOKEN是。

id\_token

必需的。 ID令牌。

态

的OAuth 2.0的状态值。如果所需的状态参数出现在授权请求。客户必须确认状态值等于状态参数的授权请求的值。

expires\_in

可选的。访问令牌的以秒为到期时间，因为生成的响应。

每次的OAuth 2.0 [ RFC6749 ]第4.2.2 ，使用隐式流量时，不返回任何代码的结果。

下面是使用的隐式流量（用线缠绕仅用于显示的目的）的成功响应一个非规范的示例：

HTTP/1.1 302 Found

Location: https://client.example.org/cb#

access\_token=SlAV32hkKG

&token\_type=bearer

&id\_token=eyJ0 ... NiJ9.eyJ1c ... I6IjIifX0.DeWt4Qu ... ZXso

&expires\_in=3600

&state=af0ifjsldkj

3.2.2.6 。验证错误响应

当使用隐式流程，授权错误响应是由相同的方式作为授权码流，如第3.1.2.6规定，除本节规定的差异。

如果最终用户拒绝请求或最终用户认证失败，授权服务器必须返回重定向的URI ，如的OAuth 2.0的4.2.2.1定义[ RFC6749 ]和OAuth 2.0多重的片段组成部分的错误授权响应响应类型编码实践[ OAuth.Responses ] ，除非指定了不同的反应模式。

3.2.2.7 。重定向的URI片段处理

由于响应参数在重定向的URI片段返回值，客户需要有用户代理解析片段编码的值，并将其传递给到消费客户的处理逻辑。参见第15.5.3关于URI的片段处理实施说明。

3.2.2.8 。验证响应验证

当使用隐流，客户端必须验证回应如下：

验证响应顺应了[ OAuth.Responses ]第5 。

按照验证规则在RFC 6749 ，特别是那些在4.2.2节和10.12 。

按照第3.2.2.11的ID令牌验证规则。

按照访问令牌的验证规则在第3.2.2.9 ，除非使用的RESPONSE\_TYPE值id\_token 。

3.2.2.9 。访问令牌验证

为了验证从授权端点与一个ID令牌发出一个访问令牌，客户应做到以下几点：

哈希与JWA在ID令牌的JWS [ JWS ]标头中的标头ALG参数指定[ JWA ]散列算法的ACCESS\_TOKEN的ASCII表示的字节。例如，如果ALG是RS256 ，所使用的散列算法是SHA-256 。

就拿哈希的最左边的一半， base64url对其进行编码。

at\_hash在ID标记的值必须匹配在先前步骤中所产生的值。

3.2.2.10 。 ID令牌

该ID标记的内容是如第2节所述。当使用隐式流程，有以下的ID令牌声明这些额外的要求：

随机数

使用随机数的索赔需要这个流程。

at\_hash

访问令牌的散列值。它的值是ACCESS\_TOKEN值的ASCII表示，其中使用的哈希算法是在ID令牌的JWS [ JWS ]的ALG头参数中使用的哈希算法的八位字节的哈希值的最左边的一半base64url编码头。举例来说，如果ALG是RS256 ，散列ACCESS\_TOKEN值与SHA-256 ，然后采取最左边的128位和base64url编码它们。该at\_hash值是一个字符串，区分大小写。

如果ID令牌是从端点授权颁布的一个ACCESS\_TOKEN价值，这是id\_token令牌RESPONSE\_TYPE值的情况下，这是必需的，没有访问令牌发出时，它可能无法使用，这对于RESPONSE\_TYPE值的情况下id\_token 。

3.2.2.11 。 ID令牌验证

当使用隐式流程，其ID令牌的内容必须被以同样的方式作为授权码流进行验证，如第3.1.3.7规定，除本节规定的差异。

客户端必须根据JWS [ JWS ]使用在智威汤逊头的ALG参数中指定的算法验证ID令牌的签名。

随机数索赔的值必须进行检查，以确认它是相同的值被发送的认证请求的人。客户应检查重放攻击的随机数值。用于检测重放攻击的精确的方法是客户端特有的。

3.3 。使用混合身份验证流程

本节将介绍如何使用混合流程进行认证。当使用混合流水，有的令牌是从授权端点返回，其他人都从令牌端点返回。返还在混合流程令牌的机制在的OAuth 2.0指定多个响应类型编码实践[ OAuth.Responses ] 。

3.3.1 。混合流程步骤

混合流程遵循以下步骤：

客户端准备包含所需的请求参数的认证请求。

客户端将请求发送到授权服务器。

授权服务器验证的最终用户。

授权服务器获取最终用户许可/授权。

授权服务器发送最终用户返回到客户端用一个授权码，并根据响应类型，一个或多个附加参数。

客户端请求使用授权码的令牌端点的响应。

客户端收到包含一个ID令牌和访问令牌的响应正文的响应。

客户端验证的ID令牌和获取最终用户的主题标识符。

3.3.2 。授权端点

当使用混合流，在授权端点中使用的相同的方式进行授权码流，如在3.1.2节中定义，除本节中指明的差异。

3.3.2.1 。认证请求

如第3.1.2.1定义的认证请求被提出，不同的是这些验证请求参数的使用方法如下：

RESPONSE\_TYPE

必需的。 OAuth的2.0响应类型值，用于确定该授权处理的流程也可以使用，包括什么参数是从所使用的端点返回。当使用混合流量，这个值是代码id\_token ，代码标记，或代码id\_token令牌。这些值的含义中的OAuth 2.0多重响应类型编码实践[ OAuth.Responses ]中定义。

下面是一个使用将由用户代理发送到响应的客户端相应的HTTP 302重定向响应（带线包装内的值用于显示目的）授权服务器的混合流水非规范性的要求：

GET /authorize?

response\_type=code%20id\_token

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile%20email

&nonce=n-0S6\_WzA2Mj

&state=af0ifjsldkj HTTP/1.1

Host: server.example.com

3.2.2.6 。验证错误响应

当使用隐式流程，授权错误响应是由相同的方式作为授权码流，如第3.1.2.6规定，除本节规定的差异。

如果最终用户拒绝请求或最终用户认证失败，授权服务器必须返回重定向的URI ，如的OAuth 2.0的4.2.2.1定义[ RFC6749 ]和OAuth 2.0多重的片段组成部分的错误授权响应响应类型编码实践[ OAuth.Responses ] ，除非指定了不同的反应模式。

3.2.2.7 。重定向的URI片段处理

由于响应参数在重定向的URI片段返回值，客户需要有用户代理解析片段编码的值，并将其传递给到消费客户的处理逻辑。参见第15.5.3关于URI的片段处理实施说明。

3.2.2.8 。验证响应验证

当使用隐流，客户端必须验证回应如下：

验证响应顺应了[ OAuth.Responses ]第5 。

按照验证规则在RFC 6749 ，特别是那些在4.2.2节和10.12 。

按照第3.2.2.11的ID令牌验证规则。

按照访问令牌的验证规则在第3.2.2.9 ，除非使用的RESPONSE\_TYPE值id\_token 。

3.2.2.9 。访问令牌验证

为了验证从授权端点与一个ID令牌发出一个访问令牌，客户应做到以下几点：

哈希与JWA在ID令牌的JWS [ JWS ]标头中的标头ALG参数指定[ JWA ]散列算法的ACCESS\_TOKEN的ASCII表示的字节。例如，如果ALG是RS256 ，所使用的散列算法是SHA-256 。

就拿哈希的最左边的一半， base64url对其进行编码。

at\_hash在ID标记的值必须匹配在先前步骤中所产生的值。

3.2.2.10 。 ID令牌

该ID标记的内容是如第2节所述。当使用隐式流程，有以下的ID令牌声明这些额外的要求：

随机数

使用随机数的索赔需要这个流程。

at\_hash

访问令牌的散列值。它的值是ACCESS\_TOKEN值的ASCII表示，其中使用的哈希算法是在ID令牌的JWS [ JWS ]的ALG头参数中使用的哈希算法的八位字节的哈希值的最左边的一半base64url编码头。举例来说，如果ALG是RS256 ，散列ACCESS\_TOKEN值与SHA-256 ，然后采取最左边的128位和base64url编码它们。该at\_hash值是一个字符串，区分大小写。

如果ID令牌是从端点授权颁布的一个ACCESS\_TOKEN价值，这是id\_token令牌RESPONSE\_TYPE值的情况下，这是必需的，没有访问令牌发出时，它可能无法使用，这对于RESPONSE\_TYPE值的情况下id\_token 。

3.2.2.11 。 ID令牌验证

当使用隐式流程，其ID令牌的内容必须被以同样的方式作为授权码流进行验证，如第3.1.3.7规定，除本节规定的差异。

客户端必须根据JWS [ JWS ]使用在智威汤逊头的ALG参数中指定的算法验证ID令牌的签名。

随机数索赔的值必须进行检查，以确认它是相同的值被发送的认证请求的人。客户应检查重放攻击的随机数值。用于检测重放攻击的精确的方法是客户端特有的。

3.3 。使用混合身份验证流程

本节将介绍如何使用混合流程进行认证。当使用混合流水，有的令牌是从授权端点返回，其他人都从令牌端点返回。返还在混合流程令牌的机制在的OAuth 2.0指定多个响应类型编码实践[ OAuth.Responses ] 。

3.3.1 。混合流程步骤

混合流程遵循以下步骤：

客户端准备包含所需的请求参数的认证请求。

客户端将请求发送到授权服务器。

授权服务器验证的最终用户。

授权服务器获取最终用户许可/授权。

授权服务器发送最终用户返回到客户端用一个授权码，并根据响应类型，一个或多个附加参数。

客户端请求使用授权码的令牌端点的响应。

客户端收到包含一个ID令牌和访问令牌的响应正文的响应。

客户端验证的ID令牌和获取最终用户的主题标识符。

3.3.2 。授权端点

当使用混合流，在授权端点中使用的相同的方式进行授权码流，如在3.1.2节中定义，除本节中指明的差异。

3.3.2.1 。认证请求

如第3.1.2.1定义的认证请求被提出，不同的是这些验证请求参数的使用方法如下：

RESPONSE\_TYPE

必需的。 OAuth的2.0响应类型值，用于确定该授权处理的流程也可以使用，包括什么参数是从所使用的端点返回。当使用混合流量，这个值是代码id\_token ，代码标记，或代码id\_token令牌。这些值的含义中的OAuth 2.0多重响应类型编码实践[ OAuth.Responses ]中定义。

下面是一个使用将由用户代理发送到响应的客户端相应的HTTP 302重定向响应（带线包装内的值用于显示目的）授权服务器的混合流水非规范性的要求：

HTTP/1.1 302 Found

Location: https://client.example.org/cb#

code=SplxlOBeZQQYbYS6WxSbIA

&id\_token=eyJ0 ... NiJ9.eyJ1c ... I6IjIifX0.DeWt4Qu ... ZXso

&state=af0ifjsldkj

3.3.2.6 。验证错误响应

当使用混合流程，授权错误响应是由相同的方式作为授权码流，如第3.1.2.6规定，除本节规定的差异。

如果最终用户拒绝请求或最终用户认证失败，授权服务器必须返回重定向的URI ，如的OAuth 2.0的4.2.2.1定义[ RFC6749 ]和OAuth 2.0多重的片段组成部分的错误授权响应响应类型编码实践[ OAuth.Responses ] ，除非指定了不同的反应模式。

3.3.2.7 。重定向的URI片段处理

当使用混合流程，申请重定向的URI片段参数处理相同的要求来做为隐流，如第3.2.2.7规定。也见第15.5.3关于URI的片段处理实施说明。

3.3.2.8 。验证响应验证

当使用混合流，客户端必须验证回应如下：

验证响应顺应了[ OAuth.Responses ]第5 。

按照验证规则在RFC 6749 ，特别是那些在4.2.2节和10.12 。

按照ID令牌的验证规则在第3.3.2.12时使用的RESPONSE\_TYPE值是代码id\_token或代码id\_token令牌。

按照访问令牌的验证规则在第3.3.2.9时使用的RESPONSE\_TYPE值码标记或代码id\_token令牌。

按照授权码验证规则在第3.3.2.10时使用的RESPONSE\_TYPE值码id\_token ，代码标记，或代码id\_token令牌。

3.3.2.9 。访问令牌验证

当使用混合流量，访问令牌的授权端点返回的相同方法进行了验证作为隐流，正如在第3.2.2.9中定义。

3.3.2.10 。授权码验证

为了验证从授权端点与一个ID令牌发出授权码，客户端应该做到以下几点：

哈希与JWA在ID令牌的JWS [ JWS ]标头中的标头ALG参数指定[ JWA ]散列算法代码的ASCII表示的字节。例如，如果ALG是RS256 ，所使用的散列算法是SHA-256 。

就拿哈希的最左边的一半， base64url对其进行编码。

c\_hash在ID标记的值必须匹配在先前步骤中所产生的值，如果c\_hash存在于ID令牌。

3.3.2.11 。 ID令牌

该ID标记的内容是如第2节所述。当使用混合Flow ，以下ID令牌索赔申请一个ID令牌的授权端点返回这些额外的要求：

随机数

使用随机数的索赔需要这个流程。

at\_hash

访问令牌的散列值。它的值是ACCESS\_TOKEN值的ASCII表示，其中使用的哈希算法是在ID令牌的JWS [ JWS ]的ALG头参数中使用的哈希算法的八位字节的哈希值的最左边的一半base64url编码头。举例来说，如果ALG是RS256 ，散列ACCESS\_TOKEN值与SHA-256 ，然后采取最左边的128位和base64url编码它们。该at\_hash值是一个字符串，区分大小写。

如果ID令牌是从端点授权颁布的一个ACCESS\_TOKEN值，这对于RESPONSE\_TYPE值的代码id\_token的道理，这是必需的情况下，否则，将其列入是可选的。

c\_hash

代码的哈希值。它的价值是代码值的ASCII表示，其中使用的哈希算法是在ID令牌的JWS [ JWS ]的ALG头参数中使用的哈希算法的八位字节的哈希值的最左边的一半base64url编码头。举例来说，如果ALG是HS512 ，散列码值与SHA -512 ，然后采取最左边的256位和base64url编码它们。该c\_hash值是一个字符串，区分大小写。

如果ID令牌是从授权端点与一个代码，这对于RESPONSE\_TYPE值的代码id\_token和代码id\_token令牌，这是必需的情况下发出的，否则，将其列入是可选的。

3.3.2.12 。 ID令牌验证

当使用混合流，一个ID令牌的内容，从授权端点返回必须以同样的方式作为隐流进行验证，如第3.2.2.11所定义。

3.3.3 。令牌端点

当使用混合流水，令牌端点使用相同的方式作为授权码流，如第3.1.3节规定，除本节规定的差异。

3.3.3.1 。令牌请求

当使用混合Flow ，令牌的请求是由以相同的方式作为本授权码流，正如在第3.1.3.1中定义。

3.3.3.2 。令牌请求验证

当使用混合Flow ，令牌请求被以相同的方式确认为授权码流，正如在第3.1.3.2中定义。

3.3.3.3 。成功的令牌响应

当使用混合Flow ，令牌的响应是由以相同的方式作为本授权码流，正如在第3.1.3.3中定义。

3.3.3.4 。令牌错误响应

当使用混合流水，令牌错误响应是由相同的方式作为授权码流，如第3.1.3.4规定。

3.3.3.5 。令牌响应验证

当使用混合Flow ，令牌响应都以相同的方式确认为授权码流，正如在第3.1.3.5中定义。

3.3.3.6 。 ID令牌

当使用混合流水，一个ID令牌的内容从令牌端点返回的都是相同的ID令牌的授权端点返回，如第3.3.2.11规定，除本节规定的差异。

如果一个ID令牌是由双方的授权端点和从令牌端点，这对RESPONSE\_TYPE值的代码id\_token和代码id\_token令牌的情况下返回，国际空间站和子索赔值必须在这两个ID令牌是相同的。关于目前无论是在身份验证事件的所有索赔应存在于两者。如果任一ID令牌包含有关最终用户声明，任何存在于两个应该有两个相同的值。请注意， OP可以选择返回对最终用户的授权端点索赔较少，例如，隐私的原因。该at\_hash和c\_hash权利要求可以从ID令牌可以省略从令牌端点返回即使这些说法都存在在ID令牌的授权端点返回，因为ID令牌和访问令牌值从令牌端点返回的已加密绑定一起由令牌端点进行TLS加密。

3.3.3.7 。 ID令牌验证

当使用混合流，一个ID令牌的内容，从令牌返回端点必须以同样的方式作为授权码流进行验证，正如在第3.1.3.7中定义。

3.3.3.8 。访问令牌

如果访问令牌同时从授权端点和从令牌端点，这对于RESPONSE\_TYPE值的代码标记和代码id\_token令牌的情况下返回的，它们的值可以是相同的或者它们可以是不同的。需要注意的是不同的访问令牌可能由于两个端点和寿命的不同的安全特性和访问给予他们的资源也可能返回不同的是。

3.3.3.9 。访问令牌验证

当使用混合流，该访问令牌从令牌端点返回以相同的方式被验证为授权码流，正如在第3.1.3.8中定义。

4 。由第三方发起登录

在某些情况下，登录流是由一个OpenID提供者或另一方，而不是依赖方发起。在这种情况下，启动重定向到RP在其登录开始端点，它要求的RP发送认证请求到指定的任择议定书。此登录启动端点可以在该RP地址的默认登陆页面一个深层链接，而不是。 RP的支持OpenID的动态连接客户端注册1.0 [ OpenID.Registration ]寄存器使用initiate\_login\_uri注册参数这个端点值。

党发起登录请求这样做重定向到RP的登录起始端点，通过以下参数：

ISS

必需的。发行人标识符的RP是身份验证请求发送到OP 。它的值必须使用https方案的URL。

login\_hint

可选的。提示有关登录标识符最终用户可以用它来登录。如果客户端收到该字符串值参数的值，它必须将其包含在认证请求的login\_hint参数值的授权服务器。

target\_link\_uri

可选的。该RP被要求身份验证后重定向到的URL 。 RP上必须验证target\_link\_uri的值，以防止被用作开放重定向到外部网站。

这些参数可以被传递使用HTTP GET方法的查询参数或传递的是自动提交的用户代理HTML表单的值，从而通过HTTP POST方法传递。

其他参数可以被发送，如果通过扩展定义。未了解使用任何参数必须由客户端被忽略。

客户应采用框架破坏和其他技术来防止最终用户被记录在由不知情的情况下第三方网站通过攻击，如点击劫持。参见[ RFC6819 ]的4.4.1.9节的更多细节。

5 。索赔

本部分规定了客户端如何获得有关最终用户和身份验证事件索赔。它还定义了一套标准的基本轮廓索赔。索赔的预定义集可以使用特定的范围值，被要求或个人索赔可以使用索赔请求参数要求。该声明可以直接来自OpenID提供者或分布式的数据源。

5.1 。标准索赔

该规范定义了一套标准的索赔。他们可以要求无论是在使用者资讯响应中返回，每第5.3.2节，或在ID令牌，每节2 。

|  |  |  |
| --- | --- | --- |
| **Member** | **Type** | **Description** |
| sub | string | Subject - Identifier for the End-User at the Issuer. |
| name | string | End-User's full name in displayable form including all name parts, possibly including titles and suffixes, ordered according to the End-User's locale and preferences. |
| given\_name | string | Given name(s) or first name(s) of the End-User. Note that in some cultures, people can have multiple given names; all can be present, with the names being separated by space characters. |
| family\_name | string | Surname(s) or last name(s) of the End-User. Note that in some cultures, people can have multiple family names or no family name; all can be present, with the names being separated by space characters. |
| middle\_name | string | Middle name(s) of the End-User. Note that in some cultures, people can have multiple middle names; all can be present, with the names being separated by space characters. Also note that in some cultures, middle names are not used. |
| nickname | string | Casual name of the End-User that may or may not be the same as the given\_name. For instance, a nickname value of Mikemight be returned alongside a given\_name value of Michael. |
| preferred\_username | string | Shorthand name by which the End-User wishes to be referred to at the RP, such as janedoe or j.doe. This value MAY be any valid JSON string including special characters such as @, /, or whitespace. The RP MUST NOT rely upon this value being unique, as discussed in [**Section 5.7**](http://openid.net/specs/openid-connect-core-1_0.html#ClaimStability). |
| profile | string | URL of the End-User's profile page. The contents of this Web page SHOULD be about the End-User. |
| picture | string | URL of the End-User's profile picture. This URL MUST refer to an image file (for example, a PNG, JPEG, or GIF image file), rather than to a Web page containing an image. Note that this URL SHOULD specifically reference a profile photo of the End-User suitable for displaying when describing the End-User, rather than an arbitrary photo taken by the End-User. |
| website | string | URL of the End-User's Web page or blog. This Web page SHOULD contain information published by the End-User or an organization that the End-User is affiliated with. |
| email | string | End-User's preferred e-mail address. Its value MUST conform to the [**RFC 5322**](http://openid.net/specs/openid-connect-core-1_0.html#RFC5322) [RFC5322] addr-spec syntax. The RP MUST NOT rely upon this value being unique, as discussed in [**Section 5.7**](http://openid.net/specs/openid-connect-core-1_0.html#ClaimStability). |
| email\_verified | boolean | True if the End-User's e-mail address has been verified; otherwise false. When this Claim Value is true, this means that the OP took affirmative steps to ensure that this e-mail address was controlled by the End-User at the time the verification was performed. The means by which an e-mail address is verified is context-specific, and dependent upon the trust framework or contractual agreements within which the parties are operating. |
| gender | string | End-User's gender. Values defined by this specification are female and male. Other values MAY be used when neither of the defined values are applicable. |
| birthdate | string | End-User's birthday, represented as an [**ISO 8601:2004**](http://openid.net/specs/openid-connect-core-1_0.html#ISO8601-2004) [ISO8601?2004] YYYY-MM-DD format. The year MAY be 0000, indicating that it is omitted. To represent only the year, YYYY format is allowed. Note that depending on the underlying platform's date related function, providing just year can result in varying month and day, so the implementers need to take this factor into account to correctly process the dates. |
| zoneinfo | string | String from zoneinfo [**[zoneinfo]**](http://openid.net/specs/openid-connect-core-1_0.html#zoneinfo) time zone database representing the End-User's time zone. For example, Europe/Paris orAmerica/Los\_Angeles. |
| locale | string | End-User's locale, represented as a [**BCP47**](http://openid.net/specs/openid-connect-core-1_0.html#RFC5646) [RFC5646] language tag. This is typically an [**ISO 639-1 Alpha-2**](http://openid.net/specs/openid-connect-core-1_0.html#ISO639-1) [ISO639?1] language code in lowercase and an [**ISO 3166-1 Alpha-2**](http://openid.net/specs/openid-connect-core-1_0.html#ISO3166-1) [ISO3166?1] country code in uppercase, separated by a dash. For example, en-US or fr-CA. As a compatibility note, some implementations have used an underscore as the separator rather than a dash, for example, en\_US; Relying Parties MAY choose to accept this locale syntax as well. |
| phone\_number | string | End-User's preferred telephone number. [**E.164**](http://openid.net/specs/openid-connect-core-1_0.html#E.164) [E.164] is RECOMMENDED as the format of this Claim, for example, +1 (425) 555-1212 or +56 (2) 687 2400. If the phone number contains an extension, it is RECOMMENDED that the extension be represented using the [**RFC 3966**](http://openid.net/specs/openid-connect-core-1_0.html#RFC3966) [RFC3966] extension syntax, for example, +1 (604) 555-1234;ext=5678. |
| phone\_number\_verified | boolean | True if the End-User's phone number has been verified; otherwise false. When this Claim Value is true, this means that the OP took affirmative steps to ensure that this phone number was controlled by the End-User at the time the verification was performed. The means by which a phone number is verified is context-specific, and dependent upon the trust framework or contractual agreements within which the parties are operating. When true, the phone\_number Claim MUST be in E.164 format and any extensions MUST be represented in RFC 3966 format. |
| address | JSON object | End-User's preferred postal address. The value of the address member is a JSON [**[RFC4627]**](http://openid.net/specs/openid-connect-core-1_0.html#RFC4627) structure containing some or all of the members defined in [**Section 5.1.1**](http://openid.net/specs/openid-connect-core-1_0.html#AddressClaim). |
| updated\_at | number | Time the End-User's information was last updated. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. |

|  |
| --- |
| **Table 1: Registered Member Definitions** |

5.1.1 。地址声明

该地址声明表示物理邮寄地址。实现可以返回一个地址字段的一个子集，根据现有的资料和最终用户的隐私偏好。例如，国家和地区可能不返回更细粒度的地址信息被返回。

实现可以只返回完整的地址，在格式化子场一个字符串，也可以使用其它子字段只返回单个组件领域，也可能同时返回。如果两个变量返回，他们应该是描述同一个地址，与地址格式化指示组件字段的方式结合起来。

格式化

详细通讯地址，格式化为一个邮寄标签展示和使用。此字段可能包含多行，用换行符分隔。换行符可以表示无论是作为一个回车符/换行符对（ “\ r \ N”的） ，或作为单个换行符（ “\ N” ） 。

street\_address

全街道地址组件，其中可能包括门牌号码，街道名称，邮政信箱，和多线延伸的街道地址信息。此字段可能包含多行，用换行符分隔。换行符可以表示无论是作为一个回车符/换行符对（ “\ r \ N”的） ，或作为单个换行符（ “\ N” ） 。

地方

城市或地区的组成部分。

区域

国家，省，州，区组成。

postal\_code

邮政编码或邮政编码的组成部分。

国家

国家名称的组成部分。

5.1.2 。其他索赔

虽然此规范只定义了一小套索赔标准索赔，索赔等可以与标准的索赔一起使用。当使用这样的索赔，这是推荐使用的名称，如权利要求耐冲撞的名字，如JSON的Web令牌（ JWT ） [智威汤逊]规范描述。可替换地，私人权利要求名称可以安全使用时的命名冲突是不太可能出现，如在JWT说明书中描述。或者，如果特定的额外索赔将有广泛和普遍的适用性，它们可以注册登记索赔名称，每智威汤逊规范。

5.2 。索赔语言和脚本

引用人类可读的值可读的索赔值和索赔值可以在多语言和文字来表示。要指定语言和文字， BCP47 [ RFC5646 ]语言标记被添加到成员名称，以＃字符分隔。例如， FAMILY\_NAME ＃ JA -假名-JP表达了姓的片假名在日本，这是常用的指数和代表表示为FAMILY\_NAME ＃ JA -哈尼族-JP相同的汉字代表的音标。再举一个例子，这两个网站和网站＃去声明值可能会返回，引用网站在未指定的语言和德语网站。

由于索赔名称是区分大小写的，所以强烈推荐要求名称使用的语言变量值与使用它们登记在IANA语言子标记注册[ IANA.Language ]字符大小写拼写。特别是，通常语言名称拼写与小写字符，地区名称的拼写与大写字符和脚本的拼写与大小写混合的字符。然而，由于BCP47语言标签值不区分大小写，实现应该解释在不区分大小写的方式提供的语言变量值。

每在BCP47的建议，对于索赔的语言变量值只应作为具体是必要的。例如，使用FR可能足以在许多情况下，而不是FR- CA或FR- FR 。如果可能的话，OPS应尽量匹配要求索赔的语言环境与索赔有。例如，如果客户要求用de（德语）语言标记和运算索赔有标记DE-CH （瑞士德语），并没有通用的德国值的值，这将是适当的OP返回瑞士德国的价值给客户。 （此故意移动尽可能多的语言标记匹配的复杂性到OP越好，以简化客户端。 ）

OpenID的连接定义了如下的授权请求参数来启用指定首选的语言和文字用于返回的索赔：

claims\_locales

可选的。最终用户的首选语言以及被退回索赔脚本，表示为BCP47 [ RFC5646 ]语言标签值的空格分隔的列表，责令偏好。如果某些要求的语言环境或全部不支持OpenID提供者的错误不应导致。

当OP的决定，无论是通过claims\_locales参数，或通过其他方式，使最终用户和客户机请求中只有一组的语言和文字声明，它建议的OP退货索赔没有语言标签时，他们使用这一语言和脚本。此外，还建议该客户端写入的方式，他们可以处理和使用语言标记利用索赔。

5.3 。使用者资讯端点

使用者资讯端点是返回有关身份验证的最终用户声明支持OAuth 2.0受保护的资源。要获得有关最终用户所请求的索赔，客户端通过使用连接的OpenID认证获得访问令牌请求的使用者资讯端点。这些说法是由包含名称和值对索赔的集合的JSON对象通常代表。

与使用者资讯端点的通信必须使用TLS 。见16.17节有关使用TLS的详细信息。

使用者资讯终端必须支持使用在RFC 2616 [ RFC2616 ]定义的HTTP GET和HTTP POST方法。

使用者资讯端点必须接受访问令牌作为的OAuth 2.0承载令牌用法[ RFC6750 ] 。

使用者资讯端点应支持使用交叉地资源共享（ CORS ） [ CORS ]和或其他方法（如适用） ，以使Java脚本客户端访问端点。

5.3.1 。使用者资讯需求

客户端发送使用HTTP GET或HTTP POST使用者资讯需求。从一个OpenID身份验证连接请求获得访问令牌必须被作为一个承载令牌，每OAuth的2.0承载令牌用法[ RFC6750 ]第2 。

推荐的做法是请求使用HTTP GET方法和访问令牌使用Authorization头域被发送。

下面是一个使用者资讯请求的非规范性例子：

GET /userinfo HTTP/1.1

Host: server.example.com

Authorization: Bearer SlAV32hkKG

5.3.2。成功的UserInfo响应

除非客户注册时要求签名或加密的响应使用者资讯索赔必须返回一个JSON对象的成员。在第5.1节中定义的索赔可以返回，因为没有规定可以有额外的索赔。

出于保护隐私的原因，OpenID的提供者可以选择不返回值的一些要求索赔。

如果没有返回一个索赔，该索赔的名称应该从代表申索JSON对象被省略，它不应该存在一个null或空字符串值。

子（主题）索赔必须始终在使用者资讯的响应返回。

注：由于令牌替换攻击（见16.11）的可能性，使用者资讯的反应是不能保证对最终用户确定的子标识令牌的（主题）的元素。本次索赔中的UserInfo反应必须核实准确的子索赔在ID令牌相匹配，如果不匹配，使用者资讯的响应值不能被使用。

在收到的UserInfo请求，使用者资讯端点必须返回的UserInfo响应作为在HTTP响应正文第13.3节的JSON序列化，除非是注册[OpenID.Registration]中指定不同的格式。使用者资讯端点必须返回一个Content-Type头指示正在返回的格式。内容类型的HTTP响应必须应用/ JSON如果响应主体是文本的JSON对象，响应主体应该使用UTF-8编码。

如果使用者资讯的响应进行签名和/或加密的，那么索赔返回在智威汤逊和内容类型必须是应用程序/智威汤逊。该反应可能是还没有被签名加密。如果两个签名和加密被请求时，应答必须签订再加密，其结果是一个嵌套智威汤逊，如[智威汤逊]中定义。

如果签署，使用者资讯响应应包含索赔ISS（发行人）和澳元（观众）的成员。国际空间站的值应该是OP的发行人标识符URL。澳元值应为或包含RP的客户端ID值。

下面是一个使用者资讯响应的非规范性的例子：

HTTP/1.1 200 OK

Content-Type: application/json

{

"sub": "248289761001",

"name": "Jane Doe",

"given\_name": "Jane",

"family\_name": "Doe",

"preferred\_username": "j.doe",

"email": "janedoe@example.com",

"picture": "http://example.com/janedoe/me.jpg"

}

5.3.3。使用者资讯错误响应

当错误情况发生时，使用者资讯端点返回一个错误响应中的的OAuth2.0承载令牌用法[RFC6750]第3节中定义。 （无关RFC6750 HTTP错误使用相应的HTTP状态代码返回给用户代理。）

以下是使用者资讯错误响应的非规范性的例子：

HTTP/1.1 401 Unauthorized

WWW-Authenticate: error="invalid\_token",

error\_description="The Access Token expired"

5.3.4 。使用者资讯响应验证

客户端必须验证的UserInfo回应如下：

验证所回应的OP是通过TLS服务器证书检查的预期入伙，根据RFC 6125 [ RFC6125 ] 。

如果客户已注册过程中提供一个userinfo\_encrypted\_response\_alg参数，使用注册时指定的解密密钥的使用者资讯响应。

如果响应签署，客户应根据JWS [ JWS ]验证签名。

5.4 。使用范围值，要求索赔

OpenID的连接客户端使用的范围值，如2.0的OAuth [ RFC6749 ]的3.3节定义的，指定什么访问被请求访问令牌的特权。与访问令牌相关联的范围确定哪些资源可当它们被用来访问的OAuth 2.0保护的端点。受保护的资源端点可以执行不同的操作，并返回基于范围值，并要求所提出的访问令牌时使用的其他参数的不同信息。

对于OpenID的连接，示波器可以用来请求特定的信息集被提供作为索赔值。

通过以下范围要求索赔是由授权服务器视为自愿索赔。

OpenID的连接定义了用于请求索赔以下范围的值：

轮廓

可选的。这个范围值请求访问最终用户的默认配置文件索赔，其中有：姓名， FAMILY\_NAME ， GIVEN\_NAME ， middle\_name ，昵称， preferred\_username ，轮廓，图片，网站，性别，出生日期，时区信息，语言环境和的updated\_at 。

电子邮件

可选的。这个范围值请求访问电子邮件和email\_verified索赔。

地址

可选的。这个范围值请求访问的地址索赔。

电话

可选的。这个范围值请求访问PHONE\_NUMBER和phone\_number\_verified索赔。

多个范围值可以用于创建分隔的空格， ASCII码范围的值区分大小写列表。

通过配置文件，电子邮件，地址和电话范围值要求索赔是从使用者端点返回，如第5.3.2节中所述，当一个RESPONSE\_TYPE值用于该结果在一个访问令牌被签发。然而，当没有访问令牌发出（这对于RESPONSE\_TYPE值id\_token的情况下） ，由此产生的索赔是在ID令牌返回。

在某些情况下，最终用户将给予有OpenID提供者拒绝提供所要求的RP部分或全部信息的选项。为了尽量减少信息的最终用户正在被要求披露的金额，一个RP可以选择只请求的信息，可从使用者端点的一个子集。

下面是一个非编码范围请求的非规范性的例子：

scope=openid profile email phone

5.5 。使用“索赔”请求参数要求索赔

OpenID的连接定义了如下的授权请求参数来启用，要求个人债权及订明适用于要求索赔参数：

索赔

可选的。此参数用来请求该特定权利要求被返回。该值是列出要求索赔的JSON对象。

声明身份验证请求参数请求特定的权利要求书从使用者端点和/或在ID令牌返回。它表示为含有权利要求从这些位置被请求列表的JSON对象。被请求的债权属性也可以指定。

支持索赔参数是可选的。应该由一个操作不支持该参数和RP使用它，它认为将是有益的RP和使用启发式任何它认为适当的最终用户的任择议定书应该返回一组声明的RP 。该claims\_parameter\_supported发现结果表明入伙是否支持该参数。

索赔参数值是代表在支持OAuth 2.0的要求为UTF -8编码的JSON （它结束了的时候作为一个OAuth的参数传递是窗体-urlencoded ） 。当在一个请求对象的值，每6.1节中使用的JSON被用作权利要求中元件的值。

索赔请求JSON对象的顶级成员名单如下：

用户信息

可选的。所列出的个人索赔从使用者端点返回的请求。如果存在，上市索赔被要求加入到正在使用的范围值要求的任何索赔。如果不存在，从使用者端点被请求的索赔只有那些使用范围值要求。

当用户信息构件时，该请求也必须使用一个RESPONSE\_TYPE值，结果在一个访问令牌被在使用者资讯端点发给客户端使用。

id\_token

可选的。所列出的个人索赔在ID令牌返回请求。如果存在，上市索赔被要求加入到在ID令牌默认索赔。如果不存在，则默认ID令牌的债权请求，按照第2节中的ID标记的定义和每个附加的每个流的ID令牌要求在第3.1.3.6 ， 3.2.2.10 ， 3.3.2.11和3.3.3.6 。

其他成员可能存在。所使用的任何成员不被理解必须被忽略。

一个例子如权利要求要求如下：

{

"userinfo":

{

"given\_name": {"essential": true},

"nickname": null,

"email": {"essential": true},

"email\_verified": {"essential": true},

"picture": null,

"http://example.info/claims/groups": null

},

"id\_token":

{

"auth\_time": {"essential": true},

"acr": {"values": ["urn:mace:incommon:iap:silver"] }

}

}

注意，权利要求，是不是在5.1节中，（例如）定义的标准集http://example.info/claims/groups权利要求，被请求了。使用要求的参数是请求的标准集之外索赔的唯一途径。它也要求将不能使用范围的值所指定的标准的权利要求的特定组合的唯一途径。

5.5.1。个人索赔要求

该用户信息和权利要求书的id\_token成员要求都是与被请求的成员名称的个人索赔的名称JSON对象。成员值必须是下列之一：

零

表明，这种说法是被请求的默认方式。特别是，这是一个自愿性索赔。例如，索赔要求：

请在默认方式GIVEN\_NAME索赔。

JSON对象

用于提供有关被请求索赔的附加信息。该规范定义了以下成员：

必要

可选的。表示被请求的索赔是否是一个不可或缺的索赔。如果该值为true，则表明索赔是一个不可或缺的索赔。例如，索赔要求：

"auth\_time": {"essential": true}

可以用来指定它是必要的，返回一个auth\_time权利要求值。

如果该值为false，则表明它是一个自愿性索赔。默认值为false。

通过请求索赔的必要权利要求书，RP表示对最终用户是释放这些索赔将确保由最终用户所要求的特定任务的顺利授权。请注意，即使索赔不可用，因为最终用户并没有授权他们释放或它们不存在，则授权服务器不能产生一个错误时，不会返回索赔，无论是基本的或自愿的，除非另有规定具体的要求的描述。

值

可选的。该索赔与一个特定的值返回的请求。例如索赔要求：

"sub": {"value": "248289761001"}

可用于指定要求适用于最终用户与主题标识符248289761001。

价值成员的值必须为被请求的索赔有效值。个人索赔的定义可以包括如何以及是否值限定符，要求索赔时要使用的要求。

值

可选的。该索赔与一组值中的一个被退回，与出现在优先顺序的值的请求。例如索赔要求：

"acr": {"essential": true,

"values": ["urn:mace:incommon:iap:silver",

"urn:mace:incommon:iap:bronze"]}

指定它是至关重要的ACR索赔与任一值瓮返回：狼牙棒：incommon：IAP：银或URN：狼牙棒：incommon：IAP：铜牌。

该成员的值数组中的值必须是被请求的索赔有效值。个人索赔的定义可以包括如何以及是否值限定符，要求索赔时要使用的要求。

其他成员可以定义为提供有关索赔请求的附加信息。所使用的任何成员不被理解必须被忽略。

需要注意的是支持该索赔请求参数时，即要求索赔的范围值，如5.4节中定义，是有效的速记方法，要求台个人索赔。例如，使用范围值的OpenID电子邮件和RESPONSE\_TYPE返回一个访问令牌相当于使用范围值OpenID和个人索赔以下要求。

相当于使用电子邮件的范围值：

{

"userinfo":

{

"email": null,

"email\_verified": null

}

}

5.5.1.1。要求“ACR”索赔

如果ACR索赔要求作为ID令牌的了基本要求与值参数请求特定的身份验证上下文类参考价值和实现支持索赔参数，授权服务器必须返回一个ACR声明值相匹配的请求的值之一。授权服务器可以要求最终用户与其他因素，以满足这一要求重新进行身份验证。如果这是一个不可或缺的索赔，并要求不能被满足，则授权服务器必须把这一结果作为认证尝试失败。

请注意，RP可以通过使用acr\_values??请求参数或不包括“基本”要求索赔ACR作为一个自愿索赔：在真正的个人ACR索赔请求。如果索赔是没有必要的，不能提供所要求的值，授权服务器应该返回会话的当前ACR为ACR索赔的价值。如果索赔是不是必需的，该授权服务器时不需要提供此索赔在它的响应。

如果客户端请求同时使用acr\_values??请求参数，并为ID令牌个人ACR索赔要求列出具体要求值的ACR索赔，由此产生的行为是不确定的。

5.5.2。语言和文字为个人索赔

正如5.2节所述，引用人类可读的值可读的索赔值和索赔值可以在多语言和文字来表示。在个人索赔请求，要求语言和特定债权脚本可以通过包括含有＃分隔BCP47 [RFC5646]语言标签的索赔要求索赔名称要求，采用5.2节中规定的索赔名称语法。例如，一个姓片假名在日本可以使用Claim名称FAMILY\_NAME＃可以使用Claim名称FAMILY\_NAME＃JA-哈尼族-JP要求JA-假名太平绅士及的姓在日本一个汉字代表提出要求。一位德国语言网站可以要求与索赔网站名称＃去。

如果OP接收到对一个语言和脚本，它不具有人类可读的索赔请求，任何版本的索赔退换不使用所要求的语言和文字，应在索赔名称使用语言标记。

5.6。声明类型

声明值的三个代表是由本规范定义：

普通债权

索赔是直接断言的OpenID提供者。

合计索赔

是由一个声明提供比OpenID提供其他断言，但债权是由OpenID提供者返回。

分布式索赔

是由一个声明提供比OpenID提供其他断言，但债权返回由OpenID提供参考。

正常的索赔必须得到支持。支持聚合索赔和分布式号称是可选的。

5.6.1。普通债权

普通债权中表示为一个JSON对象的成员。索赔名称是成员名称和索赔值是成员值。

下面是一个包含普通债权的非规范性的反应：

{

"name": "Jane Doe",

"given\_name": "Jane",

"family\_name": "Doe",

"email": "janedoe@example.com",

"picture": "http://example.com/janedoe/me.jpg"

}

5.6.2 。汇总和分布式索赔

聚合和分发债权是通过使用特殊的\_claim\_names和包含索赔的JSON对象的\_claim\_sources成员代表。

\_claim\_names

JSON对象，其成员名称的索赔名称的汇总和分布式索赔。成员值均指在\_claim\_sources成员从实际索赔值可以被检索到的成员名称。

\_claim\_sources

JSON对象，其成员名称由\_claim\_names成员的成员值引用。成员值包含几组聚合索赔或参考位置的分布式索赔。成员值可以有以下格式之一取决于它是否是提供聚合或分散索赔：

合计索赔

必须包含在智威汤逊成员，其值是JSON对象一个JWT [智威汤逊] ，它必须包含在\_claim\_names对象，引用相应的\_claim\_sources成员的所有索赔。其他成员可能存在。所使用的任何成员不被理解必须被忽略。

智威汤逊

必需的。智威汤逊含索赔值。

智威汤逊不应该包含一个子（主题）索赔，除非它的值是一个标识符最终用户的声明提供程序（而不是为OpenID提供者或其他方），这通常意味着一个子索赔不应提供。

分布式索赔

JSON对象，它包含以下成员和值：

端点

必需的。从相关的索赔可以被检索的OAuth 2.0资源端点。端点URL必须返回索赔作为智威汤逊。

ACCESS\_TOKEN

可选的。访问令牌能够从端点URL索赔的检索使用的OAuth 2.0承载令牌用法[ RFC6750 ]协议。索赔应通过授权请求报头字段和索赔提供商必须支持此方法要求。如果访问令牌不可用时， RP上可能需要获取访问令牌带外，或使用访问令牌被预先商定的声明提供程序和RP ，或声明提供方可能重新验证的最终用户和/或重新授权RP了。

一个子（主题）申索不应从声明提供程序返回的，除非它的值是一个标识符最终用户的声明提供程序（而不是为OpenID提供者或其他方），这通常意味着一个子索赔不应来提供。

一般情况下，它是由OP时，它是适合使用聚合索赔和索赔分布。在某些情况下，什么时候使用什么类型的索赔可能会协商出带的RP与OPS之间的信息。

5.6.2.1 。聚合权利要求的示例

在这种不规范的例子，从声明提供理赔相结合，与由OpenID提供商举行，从声明提供一个被返回的聚合索赔索赔其他诉讼请求。

在这个例子中，大约李四这些说法已发出声明提供一个：

{

"address": {

"street\_address": "1234 Hollywood Blvd.",

"locality": "Los Angeles",

"region": "CA",

"postal\_code": "90210",

"country": "US"},

"phone\_number": "+1 (310) 123-4567"

}

声明提供一个签署的JSON索赔，代表他们在一个签名的智威汤逊：jwt\_header.jwt\_part2.jwt\_part3。正是这种JWT所使用的OpenID提供。

在这个例子中，这个智威汤逊含声明提供一个Jane Doe的聚合索赔是结合其他普通债权，并返回作为索赔的以下设置：

{

"name": "Jane Doe",

"given\_name": "Jane",

"family\_name": "Doe",

"birthdate": "0000-03-22",

"eye\_color": "blue",

"email": "janedoe@example.com",

"\_claim\_names": {

"address": "src1",

"phone\_number": "src1"

},

"\_claim\_sources": {

"src1": {"JWT": "jwt\_header.jwt\_part2.jwt\_part3"}

}

}

5.6.2.2。分布式权利要求的示例

在这种不规范的例子，OpenID提供结合了普通声称，它与引用由两个不同的声明提供，B和C，纳入参考一些由B和C为分布式债权持有的债权持有的债权持有。

在这个例子中，大约Jane Doe的这些说法是持有声明提供B（Jane Doe的银行）：

{

"shipping\_address": {

"street\_address": "1234 Hollywood Blvd.",

"locality": "Los Angeles",

"region": "CA",

"postal\_code": "90210",

"country": "US"},

"payment\_info": "Some\_Card 1234 5678 9012 3456",

"phone\_number": "+1 (310) 123-4567"

}

此外，在这个例子中，这个索赔约李四是持有声明提供C（信用机构）：

{

"credit\_score": 650

}

在OpenID提供者返回Jane Doe的索赔以及引用的声明提供B与发送的访问令牌，并从中分布式索赔可以检索位置的URL声明提供C中的分布式索赔：

{

"name": "Jane Doe",

"given\_name": "Jane",

"family\_name": "Doe",

"email": "janedoe@example.com",

"birthdate": "0000-03-22",

"eye\_color": "blue",

"\_claim\_names": {

"payment\_info": "src1",

"shipping\_address": "src1",

"credit\_score": "src2"

},

"\_claim\_sources": {

"src1": {"endpoint":

"https://bank.example.com/claim\_source"},

"src2": {"endpoint":

"https://creditagency.example.com/claims\_here",

"access\_token": "ksj3n283dke"}

}

}

5.7 。权利要求稳定性和唯一性

子（主题）和ISS （发行人）权利要求书，一起使用，是一个RP可以根据依赖作为一个稳定的标识符最终用户唯一的索赔，因为索赔子必须是本地唯一与发行人之内从未进行重新分配一个特定的最终用户，如第2节所述。因此，对于一个给定的最终用户的唯一保证唯一标识符是ISS索赔和副索赔的组合。

所有其他索赔携带稳定性跨越了用户的时间或独特性条款在不同发行人没有这样的保证，发行人可申请当地限制和政策。例如，发行人可在时间重复使用在不同的最终用户的电子邮件声明值在不同的点，且所主张的电子邮件地址对于一个给定的最终用户可能会随时间而改变。因此，其他的索赔，如电子邮件， PHONE\_NUMBER和preferred\_username并且不能被用来作为最终用户的唯一标识符。

6 。通过请求参数作为JWTs

OpenID的连接定义了如下的授权请求的参数，以使认证请求进行签名和加密可选：

要求

可选的。此参数使OpenID的连接请求被通过在一个单一的，自包含的参数，并进行选择性签名和/或加密。该参数的值是请求对象的值，如第6.1节规定。它所代表的请求作为智威汤逊的声明是请求参数。

REQUEST\_URI

可选的。此参数使OpenID的连接请求通过引用传递，而不是通过值。该REQUEST\_URI值是使用HTTPS模式引用包含一个请求对象的值，它是一个包含请求参数的智威汤逊的资源的URL。

使用这些参数的请求表示为JWTs ，这是按值或按引用进行传递。通过引用传递请求的能力是大的要求特别有用。如果使用这些参数中的一个，其他的不能在同一个请求中使用。

6.1 。按值传递一个请求对象

请求授权请求参数启用OpenID的连接请求被通过在一个单一的，自包含的参数，并进行选择性签名和/或加密。它所代表的请求作为智威汤逊的声明是在第3.1.2节中指定的请求参数。此智威汤逊被称为请求对象。

支持请求参数是可选的。该request\_parameter\_supported发现结果表明入伙是否支持该参数。应该由一个操作不支持该参数和RP使用它，则OP必须返回request\_not\_supported错误。

当请求参数时，包含在智威汤逊的OpenID的连接请求参数值取代那些使用的OAuth 2.0请求语法通过。然而，参数也可以使用，即使一个请求对象所使用的2.0的OAuth请求语法过去了，这通常会做，以使缓存，预签（也可能是预加密）请求对象的值包含要使用的固定请求参数，同时可以与每个请求，如状态和随机数发生变化的参数，被传递的OAuth 2.0的参数。

因此，该请求是有效的OAuth 2.0授权请求，必须使用的OAuth 2.0请求语法包括为RESPONSE\_TYPE和CLIENT\_ID参数的值，因为它们是由OAuth的2.0必需的。这些参数的值必须匹配那些在请求对象，如果存在的话。

即使一个作用域参数出现在请求对象的值，一个作用域参数必须始终使用含OpenID的范围值，以指示到底层的OAuth 2.0的逻辑，这是一个OpenID连接请求的OAuth 2.0请求语法通过。

请求对象可以是有符号或无符号（明文） 。当它是纯文本的，这是由在JWS头使用没有算法[ JWA ]的指示。如果签字，请求对象应包含索赔ISS （发行人）和澳元（观众）的成员。国际空间站的值应该是RP的客户端ID ，除非它是由不同的政党比RP签署。澳元值应为或包括业务方案的发行人标识符URL 。

请求对象也可以使用JWE [ JWE ]加密的，并且可能还没有被签名进行加密。如果两个签名和加密被执行，它必须被签署，然后加密，其结果是一个嵌套智威汤逊，如[智威汤逊]中定义。

请求和REQUEST\_URI参数必须不包含在请求对象。

以下是base64url编码和签署之前请求对象的索赔的非规范性的例子：

{

"iss": "s6BhdRkqt3",

"aud": "https://server.example.com",

"response\_type": "code id\_token",

"client\_id": "s6BhdRkqt3",

"redirect\_uri": "https://client.example.org/cb",

"scope": "openid",

"state": "af0ifjsldkj",

"nonce": "n-0S6\_WzA2Mj",

"max\_age": 86400,

"claims":

{

"userinfo":

{

"given\_name": {"essential": true},

"nickname": null,

"email": {"essential": true},

"email\_verified": {"essential": true},

"picture": null

},

"id\_token":

{

"gender": null,

"birthdate": {"essential": true},

"acr": {"values": ["urn:mace:incommon:iap:silver"]}

}

}

}

与RS256算法结果到此Request对象中的值（用线缠绕值范围内仅用于显示目的）??签署了：

eyJhbGciOiJSUzI1NiIsImtpZCI6ImsyYmRjIn0.ew0KICJpc3MiOiAiczZCaGRSa3

F0MyIsDQogImF1ZCI6ICJodHRwczovL3NlcnZlci5leGFtcGxlLmNvbSIsDQogInJl

c3BvbnNlX3R5cGUiOiAiY29kZSBpZF90b2tlbiIsDQogImNsaWVudF9pZCI6ICJzNk

JoZFJrcXQzIiwNCiAicmVkaXJlY3RfdXJpIjogImh0dHBzOi8vY2xpZW50LmV4YW1w

bGUub3JnL2NiIiwNCiAic2NvcGUiOiAib3BlbmlkIiwNCiAic3RhdGUiOiAiYWYwaW

Zqc2xka2oiLA0KICJub25jZSI6ICJuLTBTNl9XekEyTWoiLA0KICJtYXhfYWdlIjog

ODY0MDAsDQogImNsYWltcyI6IA0KICB7DQogICAidXNlcmluZm8iOiANCiAgICB7DQ

ogICAgICJnaXZlbl9uYW1lIjogeyJlc3NlbnRpYWwiOiB0cnVlfSwNCiAgICAgIm5p

Y2tuYW1lIjogbnVsbCwNCiAgICAgImVtYWlsIjogeyJlc3NlbnRpYWwiOiB0cnVlfS

wNCiAgICAgImVtYWlsX3ZlcmlmaWVkIjogeyJlc3NlbnRpYWwiOiB0cnVlfSwNCiAg

ICAgInBpY3R1cmUiOiBudWxsDQogICAgfSwNCiAgICJpZF90b2tlbiI6IA0KICAgIH

sNCiAgICAgImdlbmRlciI6IG51bGwsDQogICAgICJiaXJ0aGRhdGUiOiB7ImVzc2Vu

dGlhbCI6IHRydWV9LA0KICAgICAiYWNyIjogeyJ2YWx1ZXMiOiBbInVybjptYWNlOm

luY29tbW9uOmlhcDpzaWx2ZXIiXX0NCiAgICB9DQogIH0NCn0.nwwnNsk1-Zkbmnvs

F6zTHm8CHERFMGQPhos-EJcaH4Hh-sMgk8ePrGhw\_trPYs8KQxsn6R9Emo\_wHwajyF

KzuMXZFSZ3p6Mb8dkxtVyjoy2GIzvuJT\_u7PkY2t8QU9hjBcHs68PkgjDVTrG1uRTx

0GxFbuPbj96tVuj11pTnmFCUR6IEOXKYr7iGOCRB3btfJhM0\_AKQUfqKnRlrRscc8K

ol-cSLWoYE9l5QqholImzjT\_cMnNIznW9E7CDyWXTsO70xnB4SkG6pXfLSjLLlxmPG

iyon\_-Te111V8uE83IlzCYIb\_NMXvtTIVc1jpspnTSD7xMbpL-2QgwUsAlMGzw

下面的RSA公钥，在JWK格式表示，可以用来验证请求对象的签名本和随后的请求对象实例（有包线内的值用于显示目的）：

{

"kty":"RSA",

"kid":"k2bdc",

"n":"y9Lqv4fCp6Ei-u2-ZCKq83YvbFEk6JMs\_pSj76eMkddWRuWX2aBKGHAtKlE5P

7\_vn\_\_PCKZWePt3vGkB6ePgzAFu08NmKemwE5bQI0e6kIChtt\_6KzT5OaaXDF

I6qCLJmk51Cc4VYFaxgqevMncYrzaW\_50mZ1yGSFIQzLYP8bijAHGVjdEFgZa

ZEN9lsn\_GdWLaJpHrB3ROlS50E45wxrlg9xMncVb8qDPuXZarvghLL0HzOuYR

adBJVoWZowDNTpKpk2RklZ7QaBO7XDv3uR7s\_sf2g-bAjSYxYUGsqkNA9b3xV

W53am\_UZZ3tZbFTIh557JICWKHlWj5uzeJXaw",

"e":"AQAB"

}

6.1.1。请使用“请求”请求参数

客户端发送的授权请求授权端点。

下面是使用的请求参数（与线值的范围内包裹仅用于显示目的）??授权请求的非规范性的例子：

https://server.example.com/authorize?

response\_type=code%20id\_token

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid

&state=af0ifjsldkj

&nonce=n-0S6\_WzA2Mj

&request=eyJhbGciOiJSUzI1NiIsImtpZCI6ImsyYmRjIn0.ew0KICJpc3MiOiA

iczZCaGRSa3F0MyIsDQogImF1ZCI6ICJodHRwczovL3NlcnZlci5leGFtcGxlLmN

vbSIsDQogInJlc3BvbnNlX3R5cGUiOiAiY29kZSBpZF90b2tlbiIsDQogImNsaWV

udF9pZCI6ICJzNkJoZFJrcXQzIiwNCiAicmVkaXJlY3RfdXJpIjogImh0dHBzOi8

vY2xpZW50LmV4YW1wbGUub3JnL2NiIiwNCiAic2NvcGUiOiAib3BlbmlkIiwNCiA

ic3RhdGUiOiAiYWYwaWZqc2xka2oiLA0KICJub25jZSI6ICJuLTBTNl9XekEyTWo

iLA0KICJtYXhfYWdlIjogODY0MDAsDQogImNsYWltcyI6IA0KICB7DQogICAidXN

lcmluZm8iOiANCiAgICB7DQogICAgICJnaXZlbl9uYW1lIjogeyJlc3NlbnRpYWw

iOiB0cnVlfSwNCiAgICAgIm5pY2tuYW1lIjogbnVsbCwNCiAgICAgImVtYWlsIjo

geyJlc3NlbnRpYWwiOiB0cnVlfSwNCiAgICAgImVtYWlsX3ZlcmlmaWVkIjogeyJ

lc3NlbnRpYWwiOiB0cnVlfSwNCiAgICAgInBpY3R1cmUiOiBudWxsDQogICAgfSw

NCiAgICJpZF90b2tlbiI6IA0KICAgIHsNCiAgICAgImdlbmRlciI6IG51bGwsDQo

gICAgICJiaXJ0aGRhdGUiOiB7ImVzc2VudGlhbCI6IHRydWV9LA0KICAgICAiYWN

yIjogeyJ2YWx1ZXMiOiBbInVybjptYWNlOmluY29tbW9uOmlhcDpzaWx2ZXIiXX0

NCiAgICB9DQogIH0NCn0.nwwnNsk1-ZkbmnvsF6zTHm8CHERFMGQPhos-EJcaH4H

h-sMgk8ePrGhw\_trPYs8KQxsn6R9Emo\_wHwajyFKzuMXZFSZ3p6Mb8dkxtVyjoy2

GIzvuJT\_u7PkY2t8QU9hjBcHs68PkgjDVTrG1uRTx0GxFbuPbj96tVuj11pTnmFC

UR6IEOXKYr7iGOCRB3btfJhM0\_AKQUfqKnRlrRscc8Kol-cSLWoYE9l5QqholImz

jT\_cMnNIznW9E7CDyWXTsO70xnB4SkG6pXfLSjLLlxmPGiyon\_-Te111V8uE83Il

zCYIb\_NMXvtTIVc1jpspnTSD7xMbpL-2QgwUsAlMGzw

6.2 。按引用传递对象请求

该REQUEST\_URI授权请求参数启用OpenID的连接请求通过引用传递，而不是通过值。此参数用于相同的请求参数，比请求对象的值从资源取得在指定的URL ，而不是按值传递等。

该request\_uri\_parameter\_supported发现结果表明入伙是否支持该参数。应该由一个操作不支持该参数和RP使用它，则OP必须返回request\_uri\_not\_supported错误。

当REQUEST\_URI参数时，包含在引用的智威汤逊的OpenID的连接请求参数值取代那些使用的OAuth 2.0请求语法通过。然而，参数也可以使用即使当REQUEST\_URI使用的OAuth的2.0请求语法传递，这将典型地进行，使一个缓存，预先签署（以及可能的预加密）请求对象的值可以使用含有固定的请求参数，而参数可以与每个请求而变化，如国家和nonce ，被传递的OAuth 2.0的参数。

因此，该请求是有效的OAuth 2.0授权请求，必须使用的OAuth 2.0请求语法包括为RESPONSE\_TYPE和CLIENT\_ID参数的值，因为它们是由OAuth的2.0必需的。这些参数的值必须匹配那些在请求对象，如果存在的话。

即使一个作用域参数存在于引用请求对象，作用域参数必须始终使用含OpenID的范围值，以指示到底层的OAuth 2.0的逻辑，这是一个OpenID连接请求的OAuth 2.0请求语法通过。

服务器可能被缓存请求的URI引用的资源的内容。如果引用的资源的内容可以不断变化， URI应当包括编码的引用资源的内容作为URI的片段组成部分的SHA-256散列的base64url 。如果用于一个URI改变片段的价值，这预示着与旧片段值的URI的任何缓存的值不再有效的服务器。

请注意，客户端可以使用OpenID的动态连接客户端注册1.0 [ OpenID.Registration ]规范的第2.1节中定义的request\_uris参数预先登记REQUEST\_URI值。有机磷农药可以要求使用REQUEST\_URI的值被预先注册的require\_request\_uri\_registration发现参数。

整个请求URI必须不超过512个ASCII字符。

由URL引用的资源的内容必须是一个请求对象。在REQUEST\_URI值所使用的计划，必须是https的，除非目标请求对象的方式，是可验证的授权服务器签名。该REQUEST\_URI值必须是可访问的授权服务器，而且应该是可访问的客户端。

以下是一个请求对象的资源，可以通过REQUEST\_URI引用（带线值的范围内包裹仅用于显示目的）??内容的非规范性的例子：

eyJhbGciOiJSUzI1NiIsImtpZCI6ImsyYmRjIn0.ew0KICJpc3MiOiAiczZCaGRSa3

F0MyIsDQogImF1ZCI6ICJodHRwczovL3NlcnZlci5leGFtcGxlLmNvbSIsDQogInJl

c3BvbnNlX3R5cGUiOiAiY29kZSBpZF90b2tlbiIsDQogImNsaWVudF9pZCI6ICJzNk

JoZFJrcXQzIiwNCiAicmVkaXJlY3RfdXJpIjogImh0dHBzOi8vY2xpZW50LmV4YW1w

bGUub3JnL2NiIiwNCiAic2NvcGUiOiAib3BlbmlkIiwNCiAic3RhdGUiOiAiYWYwaW

Zqc2xka2oiLA0KICJub25jZSI6ICJuLTBTNl9XekEyTWoiLA0KICJtYXhfYWdlIjog

ODY0MDAsDQogImNsYWltcyI6IA0KICB7DQogICAidXNlcmluZm8iOiANCiAgICB7DQ

ogICAgICJnaXZlbl9uYW1lIjogeyJlc3NlbnRpYWwiOiB0cnVlfSwNCiAgICAgIm5p

Y2tuYW1lIjogbnVsbCwNCiAgICAgImVtYWlsIjogeyJlc3NlbnRpYWwiOiB0cnVlfS

wNCiAgICAgImVtYWlsX3ZlcmlmaWVkIjogeyJlc3NlbnRpYWwiOiB0cnVlfSwNCiAg

ICAgInBpY3R1cmUiOiBudWxsDQogICAgfSwNCiAgICJpZF90b2tlbiI6IA0KICAgIH

sNCiAgICAgImdlbmRlciI6IG51bGwsDQogICAgICJiaXJ0aGRhdGUiOiB7ImVzc2Vu

dGlhbCI6IHRydWV9LA0KICAgICAiYWNyIjogeyJ2YWx1ZXMiOiBbInVybjptYWNlOm

luY29tbW9uOmlhcDpzaWx2ZXIiXX0NCiAgICB9DQogIH0NCn0.nwwnNsk1-Zkbmnvs

F6zTHm8CHERFMGQPhos-EJcaH4Hh-sMgk8ePrGhw\_trPYs8KQxsn6R9Emo\_wHwajyF

KzuMXZFSZ3p6Mb8dkxtVyjoy2GIzvuJT\_u7PkY2t8QU9hjBcHs68PkgjDVTrG1uRTx

0GxFbuPbj96tVuj11pTnmFCUR6IEOXKYr7iGOCRB3btfJhM0\_AKQUfqKnRlrRscc8K

ol-cSLWoYE9l5QqholImzjT\_cMnNIznW9E7CDyWXTsO70xnB4SkG6pXfLSjLLlxmPG

iyon\_-Te111V8uE83IlzCYIb\_NMXvtTIVc1jpspnTSD7xMbpL-2QgwUsAlMGzw

6.2.1。 URL引用的请求对象

客户在本地或远程存储请求对象的资源在URL中的服务器可以访问。这个URL是请求URI，REQUEST\_URI。

如果请求对象包括索赔请求的值，它不能透露给任何人，但该授权服务器。因此，REQUEST\_URI必须有适当的熵的寿命。建议如果它是已知的，其将不会被再次或一个合理的超时之后使用，除非采取访问控制措施，将其除去。

下面是一个请求URI值的非规范性的例子（带线值的范围内包裹仅用于显示目的）??：

https://client.example.org/request.jwt#

GkurKxf5T0Y-mnPFCHqWOMiZi4VS138cQO\_V7PZHAdM

6.2.2。要求使用“REQUEST\_URI”请求参数

客户端发送的授权请求授权端点。

下面是一个使用REQUEST\_URI参数（与线值的范围内包裹仅用于显示目的）??授权请求的非规范性的例子：

https://server.example.com/authorize?

response\_type=code%20id\_token

&client\_id=s6BhdRkqt3

&request\_uri=https%3A%2F%2Fclient.example.org%2Frequest.jwt

%23GkurKxf5T0Y-mnPFCHqWOMiZi4VS138cQO\_V7PZHAdM

&state=af0ifjsldkj&nonce=n-0S6\_WzA2Mj

&scope=openid

6.2.3。授权服务器取指请求对象

在收到请求后，授权服务器必须发送一个HTTP GET请求给REQUEST\_URI检索引用的请求对象，除非它已经被缓存，并解析它重新授权请求参数。

需要注意的是RP应使用唯一的URI使用不同的参数，每个请求，或以其他方式阻止授权服务器从缓存REQUEST\_URI。

以下是此提取过程的非规范性的例子：

GET /request.jwt HTTP/1.1

Host: client.example.org

6.2.4 。 “ REQUEST\_URI ”的理由

有迹象表明，人们可能会选择使用REQUEST\_URI参数几个原因：

所述一组请求参数可能会变得很大，并且可能会超过浏览器URI的大小限制。通过引用传递的请求参数可以解决这个问题。

传递REQUEST\_URI值，而不是通过值一个完整的请求，可以减少请求延迟。

大多数请求从一个RP索赔是不变的。该REQUEST\_URI是创造，有时也签名和加密一个常数一套事先请求参数的方法。 （该REQUEST\_URI值变为一个“神器”代表一个特定组固定的请求参数。 ）

预登记一套固定的请求参数在报名时间使老年退休金计划的报名时间缓存和预验证的请求参数，这意味着他们需要不被检索在请求的时候。

预登记一套固定的请求参数在报名时间使有机磷农药审批从保护消费者和其他信息点意见，请求的内容，无论是自己或通过利用第三方。

6.3 。智威汤逊验证基于请求

当请求或REQUEST\_URI授权请求参数的使用，额外的步骤必须执行，以验证超出了第3.1.2.2 ， 3.2.2.2 ， 3.3.2.2或指定的认证请求。这些步骤来验证包含请求对象的智威汤逊和验证请求对象本身。

6.3.1 。加密的请求对象

如果授权服务器已刊登在其发现文件[ OpenID.Discovery ]的request\_object\_encryption\_alg\_values\_supported和request\_object\_encryption\_enc\_values\_supported元素JWE加密算法，或已通过其他方式提供加密算法，这些所使用的客户端的智威汤逊加密。

授权服务器必须在智威汤逊解密按照JSON的Web加密[ JWE ]规范。其结果可以是符号或无符号（明文）请求对象。在前一种情况下，如第6.3.2节中定义签名验证必须执行。

如果解密失败的授权服务器必须返回一个错误。

6.3.2 。签署请求对象

要执行签名验证，在JWS头的ALG参数必须匹配request\_object\_signing\_alg客户注册[ OpenID.Registration ]或者是预注册通过其他方式的值时设定的值。签名必须与相应的键为CLIENT\_ID和算法进行验证。

如果签名验证失败时，授权服务器必须返回一个错误。

6.3.3 。请求参数组装和验证

授权服务器必须装配一套授权请求参数从请求对象的价值和使用OAuth的2.0授权请求参数（减去请求或REQUEST\_URI参数） 。如果相同的参数存在无论是在请求对象和OAuth的授权请求参数，请求对象的参数使用。使用装配一套授权请求参数，授权服务器然后验证正在使用的流量的正常方式下，如在章节3.1.2.2 ， 3.2.2.2 ， 3.3.2.2或指定的请求。

7 。自发行OpenID提供者

支持OpenID Connect支持自颁发的OpenID提供商 - 即签发自签名的ID令牌个人，自托管的有机磷农药。自颁发有机磷农药使用特殊的发行人识别https://self-issued.me 。

使用自颁发的OP通信的消息大多是那些相同用来与其他有机磷进行通信。规范使用的一些额外的参数，并在自行颁发情况下某些参数的值在本节中定义。

7.1 。自发行OpenID提供者发现

如果搜索过程的输入标识符包含域自issued.me ，不进行动态发现。相反，然后将下面的静态配置值用于：

{

"authorization\_endpoint":

"openid:",

"issuer":

"https://self-issued.me",

"scopes\_supported":

["openid", "profile", "email", "address", "phone"],

"response\_types\_supported":

["id\_token"],

"subject\_types\_supported":

["pairwise"],

"id\_token\_signing\_alg\_values\_supported":

["RS256"],

"request\_object\_signing\_alg\_values\_supported":

["none", "RS256"]

}

注意： OpenID基金会计划举办的OpenID提供者网站https://self-issued.me/ ，包括其的WebFinger服务，使在其上执行发现返回上面的静态发现信息，使的RP不需要任何特殊处理发现自我发行的OP 。该网站将被托管在了实验依据。生产的实现不应该采取在它的依赖不通过OpenID基金会的后续承诺主办的网站在用于进行生产的方式。

 TOC

7.2 。自发行OpenID提供者报名

当使用自行颁发的OP ，不需要注册。该客户端可以进行无需注册，犹如注册的OP和获得以下客户端注册响应：

CLIENT\_ID

客户的redirect\_uri价值。

client\_secret\_expires\_at

0

注意： OpenID基金会计划举办的（??无状态的）端点https://self-issued.me/registration/1.0/返回的响应上面，使的RP不需要任何特殊的处理登记与自行颁发入伙。该网站将被托管在了实验依据。生产的实现不应该采取在它的依赖不通过OpenID基金会的后续承诺主办的网站在用于进行生产的方式。

 TOC

7.2.1 。与“注册”请求参数提供信息

OpenID的连接定义了如下的授权请求参数，使客户能够以自行颁发的OpenID提供商提供额外的注册信息：

注册

可选的。这个参数是由客户提供其自身信息，通常会动态客户端注册期间提供到OP自行颁发的任择议定书。该值是一个包含客户端的元数据值的JSON对象，如OpenID的动态连接客户端注册1.0 [ OpenID.Registration ]规范的第2.1节中定义。当OP是不是自发放入伙登记参数不应该被使用。

没有这些信息是由自发行的OP必需的，所以使用这个参数是可选的。

注册参数值表示在的OAuth 2.0的要求为UTF- 8编码的JSON对象（它结束了的时候作为一个OAuth的参数传递是窗体-urlencoded ） 。当请求对象的值，每节6.1使用JSON对象作为注册成员的值。

这通常会在要求自行颁发有机磷农药使用的注册参数policy\_uri ， tos\_uri和logo\_uri 。如果客户端使用一个以上的重定向URI，那么redirect\_uris参数将被用来注册它们。最后，如果客户端请求加密的反应，它通常会使用jwks\_uri ， id\_token\_encrypted\_response\_alg和id\_token\_encrypted\_response\_enc参数。

 TOC

7.3 。自发行OpenID提供者请求

客户端发送认证请求到端点授权使用以下参数：

范围

必需的。 scope参数值，如第3.1.2节规定。

RESPONSE\_TYPE

必需的。常量字符串值id\_token 。

CLIENT\_ID

必需的。对于客户端，在这种情况下，包含客户端的redirect\_uri值客户ID值。由于客户的redirect\_uri URI值是作为沟通客户的身份证件， redirect\_uri参数不是必需的也被包含在请求中。

id\_token\_hint

可选的。 id\_token\_hint参数值，如第3.1.2规定。如果ID令牌进行加密，自发行入伙，已签署的ID标记的子（主体）必须被作为了JWE的孩子（密钥ID ） 。加密内容自发布有机磷农药时， OP的JWK密钥类型为RSA和使用的加密算法是RSA1\_5目前只支持。

索赔

可选的。声称参数值，如5.5节规定。

注册

可选的。这个参数是由客户提供其自身信息，通常会动态客户端注册过程中提供给OP，如第7.2.1节中指定一个自发行入伙。

要求

可选的。 Request对象的值，如第6.1节规定。请求对象可以通过客户端进行加密，以自发行入伙。在这种情况下，先前发出的ID标记为这个客户端的子（主体）必须被作为了JWE的孩子（密钥ID ） 。加密内容自发布有机磷农药时， OP的JWK密钥类型为RSA和使用的加密算法是RSA1\_5目前只支持。

其他参数可能会被发送。请注意，所有索赔在ID令牌返回。

整个URL不能超过2048个ASCII字符。

以下是客户端的非规范性的例如HTTP 302重定向响应，从而引发用户代理进行身份验证请求自发行OpenID提供者（与线值的范围内包裹仅用于显示目的） ??：

HTTP/1.1 302 Found

Location: openid://?

response\_type=id\_token

&client\_id=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile

&state=af0ifjsldkj

&nonce=n-0S6\_WzA2Mj

&registration=%7B%22logo\_uri%22%3A%22https%3A%2F%2F

client.example.org%2Flogo.png%22%7D

7.4 。自发行OpenID提供者响应

OpenID的连接定义了以下索赔在自颁发OpenID提供者使用的回应：

sub\_jwk

必需的。公钥用于检查一个ID令牌由一个自发行OpenID提供者签发的签名，在第7节中指定。最关键的是一个裸露的关键在JWK [ JWK ]格式（不是X.509证书值） 。该sub\_jwk值是一个JSON对象。当OP是不是自发放使用sub\_jwk索赔不推荐。

自发行OpenID提供者的反应是一样的，与以下改进正常的隐式流量响应。因为它是一个隐式流量响应，响应参数将URL中的片段组成部分退还，除非指定了不同的反应模式。

国际空间站（发行人）宣告值是https://self-issued.me 。

一个sub\_jwk索赔存在，它的值是用来检查ID令牌的签名的公共密钥。

子（主题）索赔值是base64url编码的SHA-256散列的sub\_jwk索赔的base64url编码键值的UTF -8表示的八位位组的串联。当KTY值是RSA ，密钥值N和E被连接的顺序。当KTY值是乳油，键值CRV， x和y被连接的顺序。

没有访问令牌是用于访问的UserInfo端点返回，所以返回的所有索赔必须在ID令牌。

 TOC

7.5 。自发行标识令牌验证

为了验证令牌收到的ID ，客户端必须做到以下几点：

客户端必须验证空间站（发行人）索赔的值是https://self-isued.me 。如果ISS包含不同的值时，ID令牌是不是自发放，而是它必须根据第3.1.3.7验证。

客户端必须验证澳元（观众）索赔包含在认证请求的客户端发送作为一个观众的redirect\_uri的价值。

客户端必须根据JWS [ JWS ]使用在智威汤逊头[智威汤逊]的ALG参数中指定的算法，利用密钥的sub\_jwk索赔在验证ID令牌的签名;关键是一个光秃秃的关键在JWK格式（不是X.509证书值） 。

该ALG值应为RS256的默认值。这也可能是ES256 。

客户端必须验证子（主题）索赔值是base64url编码的SHA-256散列的sub\_jwk索赔的base64url编码键值的UTF -8表示的八位位组的串联。当KTY值是RSA ，密钥值N和E被连接的顺序。当KTY值是乳油，键值CRV， x和y被连接的顺序。

当前时间必须由EXP索赔（可能允许一些小的余地占时钟歪斜）所表示的时间之前。

IAT的权利要求可以被用来拒绝发出太远离当前时间标记，限制的时间随机数需要被存储，以防止攻击的量。可接受的范围是特定的客户端。

如果一个nonce值被发送的认证请求，一个随机数索赔必须存在和其价值的检查，以确认它是相同的值被发送的认证请求的人。客户应检查重放攻击的随机数值。用于检测重放攻击的精确的方法是客户端特有的。

以下是base64url解码自行颁发的ID令牌（带线包装内的值用于显示目的）的非规范性的例子：

{

"iss": "https://self-issued.me",

"sub": "wBy8QvHbPzUnL0x63h13QqvUYcOur1X0cbQpPVRqX5k",

"aud": "https://client.example.org/cb",

"nonce": "n-0S6\_WzA2Mj",

"exp": 1311281970,

"iat": 1311280970,

"sub\_jwk": {

"kty":"RSA",

"n": "0vx7agoebGcQSuuPiLJXZptN9nndrQmbXEps2aiAFbWhM78LhWx

4cbbfAAtVT86zwu1RK7aPFFxuhDR1L6tSoc\_BJECPebWKRXjBZCiFV4n3oknjhMs

tn64tZ\_2W-5JsGY4Hc5n9yBXArwl93lqt7\_RN5w6Cf0h4QyQ5v-65YGjQR0\_FDW2

QvzqY368QQMicAtaSqzs8KJZgnYb9c7d0zgdAZHzu6qMQvRL5hajrn1n91CbOpbI

SD08qNLyrdkt-bFTWhAI4vMQFh6WeZu0fM4lFd2NcRwr3XPksINHaQ-G\_xBniIqb

w0Ls1jF44-csFCur-kEgU8awapJzKnqDKgw",

"e":"AQAB"

}

}

8 。主题标识符类型

一个主题标识符是本地唯一永不重新分配的标识符在发行人内部的最终用户，其目的是通过客户端使用。两个主题标识符类型由本规范定义：

公

这提供了相同的子（主题）价值给所有的客户。这是默认的，如果提供者在它的发现文件没有subject\_types\_supported元素。

配对

这提供了一个不同的子值给每个客户端，以免使客户能够私自关联最终用户的活动。

OpenID提供探知文件应列出其支持的主题标识符类型的subject\_types\_supported元素。如果是数组中列出了多个类型，客户可以选择提供注册时使用subject\_type参数其首选标识符类型。

 TOC

8.1 。成对标识算法

当两两主题标识符的使用， OpenID提供者必须计算出一个独特的分（主题）值对每个扇区标识符。主题标识符值不能是可逆的任何一方比OpenID提供者除外。

使用成对的子值，并支持动态客户端注册商[ OpenID.Registration ]应该使用sector\_identifier\_uri参数。它提供了一种方式的一组常见的管理控制是有独立的个人域名一致的成对子值的网站。它还提供了一种方法，为客户更改redirect\_uri域，而无需重新注册所有的用户。

如果客户没有提供一个值，在动态客户注册[ OpenID.Registration ] sector\_identifier\_uri ，该部门标识符用于成对标识符的计算是注册redirect\_uri的主机组件。如果在注册redirect\_uris多个主机名，客户端必须注册一个sector\_identifier\_uri 。

当提供一个sector\_identifier\_uri ，该URL的主机组件作为该部门的标识符成对的标识符计算。该sector\_identifier\_uri的值必须使用指向包含redirect\_uri值的数组JSON文件https方案的URL。注册redirect\_uris的值必须被包含在数组中的元素。

任何算法具有以下属性可以使用的OpenID提供商来计算成对的主题标识符：

主题标识符值不能是可逆的任何一方比OpenID提供者除外。

不同的扇区标识符值必须导致不同的主题标识符值。

该算法必须是确定性的。

三个例子的方法有：

该扇区标识符可以连接使用本地帐户ID和一个由供应商秘而不宣的盐值。连接后的字符串，然后用适当的算法进行散列。

算子= SHA- 256 （ sector\_identifier | | local\_account\_id | |盐） 。

该扇区标识符可以连接使用本地帐户ID和一个由供应商秘而不宣的盐值。连接后的字符串，然后用适当的算法进行加密。

算子= AES-128 （ sector\_identifier | | local\_account\_id | |盐） 。

发行人在对扇区标识符和本地帐户ID和存储此值创建一个全局唯一标识符（GUID ） 。

 TOC

9 。客户端身份验证

本节定义了一组客户端身份验证方法，这些方法使用令牌端点时所使用的客户端验证到授权服务器。在客户端注册， RP （客户端）可以注册一个客户端身份验证方法。如果没有办法注册，默认方法是client\_secret\_basic 。

这些客户端身份验证方法是：

client\_secret\_basic

已经收到了CLIENT\_SECRET值从授权服务器与客户端的授权服务器验证按照的OAuth 2.0 [ RFC6749 ]第3.2.1使用HTTP基本身份验证方案。

client\_secret\_post

已经收到了CLIENT\_SECRET值从授权服务器的客户端，验证与根据的OAuth 2.0 [ RFC6749 ]通过在请求主体的客户端凭据的第3.2.1节的授权服务器。

client\_secret\_jwt

已经收到了CLIENT\_SECRET值从授权服务器客户端创建一个智威汤逊使用HMAC SHA算法，如HMAC SHA-256 。使用HMAC （基于散列的消息认证码）是使用CLIENT\_SECRET的UTF-8表示的八位位组作为共享密钥计算。

按照JSON的Web令牌（ JWT）个人资料的OAuth 2.0客户端身份验证和授权资助[ OAuth.JWT ] ，并断言框架的OAuth 2.0客户端身份验证和授权资助[ OAuth.Assertions ]客户端进行身份验证。智威汤逊必须包含以下所需的索赔值，可能包含以下可选索赔值：

ISS

必需的。发行人。这必须包含的OAuth客户端的CLIENT\_ID 。

子

必需的。主题。这必须包含的OAuth客户端的CLIENT\_ID 。

AUD

必需的。听众。澳元（观众）索赔。值，标识授权服务器作为目标受众。授权服务器必须验证它是目标读者的令牌。观众应在授权服务器的令牌端点的URL 。

JTI

必需的。智威汤逊的ID 。的唯一标识符的标记，该标记可以用来防止令牌的再利用。这些记号必须只能使用一次，除非再利用条件乃经订约各方磋商;任何这样的谈判是超出了本规范的范围。

EXP

必需的。到期时间或以后该ID令牌必须不被接受处理。

IAT

可选的。时间点的智威汤逊发出。

智威汤逊可以包含其他诉讼请求。所使用的任何索赔是不被理解必须被忽略。

身份验证令牌必须发送为[ OAuth.Assertions ] client\_assertion参数的值。

在[ OAuth.Assertions ] client\_assertion\_type参数的值必须是“金塔： IETF ： params：一个OAuth的：客户端断言类型：智威汤逊旗手” ，按[ OAuth.JWT ] 。

private\_key\_jwt

已注册的公钥标志一个JWT使用该密钥的客户端。按照JSON的Web令牌（ JWT）个人资料的OAuth 2.0客户端身份验证和授权资助[ OAuth.JWT ] ，并断言框架的OAuth 2.0客户端身份验证和授权资助[ OAuth.Assertions ]客户端进行身份验证。智威汤逊必须包含以下所需的索赔值，可能包含以下可选索赔值：

ISS

必需的。发行人。这必须包含的OAuth客户端的CLIENT\_ID 。

子

必需的。主题。这必须包含的OAuth客户端的CLIENT\_ID 。

AUD

必需的。听众。澳元（观众）索赔。值，标识授权服务器作为目标受众。授权服务器必须验证它是目标读者的令牌。观众应在授权服务器的令牌端点的URL 。

JTI

必需的。智威汤逊的ID 。的唯一标识符的标记，该标记可以用来防止令牌的再利用。这些记号必须只能使用一次，除非再利用条件乃经订约各方磋商;任何这样的谈判是超出了本规范的范围。

EXP

必需的。到期时间或以后该ID令牌必须不被接受处理。

IAT

可选的。时间点的智威汤逊发出。

智威汤逊可以包含其他诉讼请求。所使用的任何索赔是不被理解必须被忽略。

身份验证令牌必须发送为[ OAuth.Assertions ] client\_assertion参数的值。

在[ OAuth.Assertions ] client\_assertion\_type参数的值必须是“金塔： IETF ： params：一个OAuth的：客户端断言类型：智威汤逊旗手” ，按[ OAuth.JWT ] 。

例如（用线缠绕值范围内仅用于显示目的） ??：

POST /token HTTP/1.1

Host: server.example.com

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

grant\_type=authorization\_code&

code=i1WsRn1uB1&

client\_id=s6BhdRkqt3&

client\_assertion\_type=

urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwt-bearer&

client\_assertion=PHNhbWxwOl ... ZT

没有

客户没有在令牌端点验证自己，可能是因为它仅使用隐式流量（所以不使用令牌端点），或者是因为它是一个公共客户，没有客户端秘密或其他身份验证机制。

 TOC

10 。签名和加密

根据通过它发送消息的传输，消息的完整性可能无法得到保证和消息的鼻祖可能不进行身份验证。为了降低这些风险， ID令牌，使用者资讯响应，请求对象和客户端身份验证智威汤逊值可以利用JSON的Web签名（ JWS ） [ JWS ]签上自己的内容。为了实现消息的机密性，这些值也可以使用JSON的Web加密（ JWE ） [ JWE ]他们的内容进行加密。

当消息被双方签名和加密，它必须先签名再加密，每节16.14 ，结果是一个嵌套智威汤逊，如[智威汤逊]中指定。请注意，所有JWE加密方法进行完整性检查。

OP的宣传其支持签名和加密算法在其发现文件，也可以通过其他方式提供这些信息。 RP的宣告其需要签名和加密算法中的动态注册请求，或者可以通过其他方式传达此信息。

该任择议定书通过其发现文档发布其公开密钥，也可以通过其他方式提供这些信息。对RP通过其动态注册请求宣告其公开密钥，或者可以通过其他方式传达此信息。

 TOC

10.1 。签名

签署方必须选择的基础上由收件人所支持的算法签名算法。

非对称签名

当使用RSA或ECDSA签名时， JWS头的ALG头参数值必须被设置为一个适当的算法以JSON的Web算法[ JWA ]中定义。用来签署内容私钥必须与用于发送方在其JWK设置文件发布签名验证公钥相关联。如果有引用JWK设置文档中有多个按键，一个孩子值必须在JWS头提供。各键的键用法必须支持签名。

对称的签名

当采用基于MAC的签名时， JWS头的ALG头参数值必须被设置为一个MAC算法，如JSON的Web算法[ JWA ]中定义。所使用的MAC密钥是CLIENT\_SECRET值的UTF-8表示的字节。见第16.19为为CLIENT\_SECRET熵值要求的讨论。因为他们无法保守秘密的对称特征，绝不能使用公共（非机密）客户端。

见16.20节关于是否需要签署的请求的安全注意事项。

 TOC

10.1.1 。不对称签名密钥轮换

签名密钥的旋转可以完成以下办法。签名者公布其键在JWK设置在其jwks\_uri位置，其中包括在每个消息的JWS头的签名密钥的孩子，以指示哪些键是用来验证签名的验证。键可以通过在jwks\_uri位置定期添加新键的JWK设置滚存。签名者可以开始使用新的密钥在其自由裁量权和信号使用小子值的变化来验证者。验证者知道要回到jwks\_uri位置重新检索键时，看到一个陌生的孩子的价值。该JWK设置文件在jwks\_uri应该保留最近退役的签名密钥对一段合理的时间，以便顺利过渡。

 TOC

10.2 。加密

加密方还必须选择的基础上由收件人所支持的算法的加密算法。

非对称加密： RSA

将公共密钥的内容被加密，必须使用由收件人在其JWK设置文件公布加密公钥。如果有引用JWK设置文档中有多个按键，一个孩子值必须在JWE头提供。使用支持RSA加密算法来加密用于加密该签名JWT随机内容加密密钥。各键的键使用必须包括加密。

非对称加密：椭圆曲线

创建一个短暂的椭圆曲线公钥对JWE头的EPK元素。用于密钥协议计算的其他公共密钥必须由收件人在其JWK设置文件发布的公共密钥。如果有引用JWK设置文档中有多个按键，一个孩子值必须在JWE头提供。使用ECDH -ES算法以商定要使用的用于加密的签名JWT一个内容加密密钥。各键的键必须使用支持加密。

对称加密

对称加密密钥从CLIENT\_SECRET值通过使用左截断SHA- 2哈希CLIENT\_SECRET的UTF-8表示的八位位组的派生。对于256或更少的位密钥， SHA -256被使用;为257-384位的密钥， SHA -384被使用;为385-512位的密钥， SHA -512被使用。散列值必须留截断到适当的位长度的AES密钥包装或直接使用的加密算法，例如，截断的SHA -256散列到128位A128KW 。如果大于512比特的对称密钥是需要的，用于从所述CLIENT\_SECRET密钥的不同的方法将必须由一个扩展来定义。因为他们无法保守秘密的对称加密绝不能使用公共（非机密）客户端。

见16.21节关于是否需要加密的请求的安全注意事项。

 TOC

10.2.1 。非对称加密密钥的旋转

旋转的加密密钥一定使用不同的过程比一个签名密钥，因为加密方启动的过程，因此不能依靠孩子的变化作为一个信号，按键需要改变。加密方仍然使用在JWE小鬼头参数告诉解密方的私钥用于解密，然而，加密方需要首先选择从这些收件人的jwks\_uri位置提供在JWK设置最合适的关键。

转动钥匙，解密方可以在其jwks\_uri位置发布新的密钥，并从JWK集中删除那些正在退役。该jwks\_uri应包括包含的max-age指令在响应一个Cache-Control头，如RFC 2616 [ RFC2616 ]中定义的，它使党的加密安全缓存JWK集，而不必重新检索文档每一个加密的事件。解密方应该从JWK集由jwks\_uri引用删除退役键，但在内部保留他们的一些合理的时间内，配合缓存持续时间，通过允许加密方一些时间来获得新键，以便按键之间的平滑过渡。缓存持续时间也应与发放新的签名密钥协调，如第10.1.1节所述。

 TOC

11 。离线访问

OpenID的连接定义了以下范围的值来请求脱机访问：

offline\_access

可选的。可以用来获得访问令牌授权访问最终用户的使用者资讯端点，即使在??最终用户不存在，一个的OAuth 2.0刷新令牌发出此范围值的请求（没有登录） 。

当离线访问请求，同意提示参数值必须被使用，除非其他条件来处理请求允许脱机访问所请求的资源到位。该任择议定书必须始终获得同意返回一个刷新令牌，使离线访问所请求的资源。先前保存的用户的同意并不总是足够的，授予离线访问。

在收到包含offline\_access值，授权服务器的作用域参数：

必须确保及时参数包含的同意，除非其他条件来处理请求允许脱机访问所请求的资源是否到位;除非这些条件之一或两者都满足，那么它必须忽略offline\_access要求，

必须忽略offline\_access请求，除非客户端使用的是RESPONSE\_TYPE值，将导致在返回的授权码，

必须明确地收取或同意所有客户端时，注册application\_type是网络，

应该明确地收取或同意所有客户端时，注册application\_type是当地人。

使用刷新令牌是不是唯一的offline\_access用例。授权服务器可能会在其他情况下，超出本规范的范围内给予刷新令牌。

 TOC

12 。使用刷新令牌

一个请求令牌端点也可以使用刷新令牌使用grant\_type值refresh\_token ，如的OAuth 2.0 [ RFC6749 ]第6条所述。本节定义的行为OpenID的连接授权服务器时，刷新令牌的使用。

 TOC

12.1 。刷新请求

要刷新访问令牌，客户端必须使用注册为它的client\_id ，如第9节的验证方法验证令牌端点。使用表格序列，每节13.2通过HTTP POST将客户端发送的参数令牌端点。

下面是一个刷新请求（与线值的范围内包裹仅用于显示目的）??的非规范性的例子：

POST /token HTTP/1.1

Host: server.example.com

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

client\_id=s6BhdRkqt3

&client\_secret=some\_secret12345

&grant\_type=refresh\_token

&refresh\_token=8xLOxBtZp8

&scope=openid%20profile

授权服务器必须验证刷新令牌，必须验证它是否是发给客户端，并且必须验证客户端身份验证成功它有一个客户端身份验证方法。

 TOC

12.2 。成功刷新响应

待刷新令牌的验证成功，响应主体是3.1.3.3节不同的是它可能不包含id\_token的令牌响应。

如果一个ID令牌返回为一个令牌刷新请求的结果，应满足以下要求：

其ISS索赔值必须是相同的ID令牌签发的正本验证发生时，

其子索赔值必须相同，在ID令牌签发的正本验证发生时，

其IAT索赔必须代表新的ID令牌发出的时间，

其AUD索赔值必须是相同的ID标记的原始身份验证发生时发出，

如果ID令牌包含auth\_time索赔，其值必须代表原始身份验证的时间 - 而不是发出新的ID令牌的时候，

其AZP声明值必须出具原始身份验证时发生相同的ID标记，如果没有AZP索赔存在于原始ID令牌，人们不应出现在新的ID令牌，和

否则，同样的规则适用于原认证时发出的ID令牌时适用。

以下是刷新响应的非规范性的例子：

HTTP/1.1 200 OK

Content-Type: application/json

Cache-Control: no-store

Pragma: no-cache

{

"access\_token": "TlBN45jURg",

"token\_type": "Bearer",

"refresh\_token": "9yNOxJtZa5",

"expires\_in": 3600

}

12.3 。刷新错误响应

如果刷新请求是无效的或未经授权，授权服务器返回的令牌错误响应中的OAuth 2.0 [ RFC6749 ]的5.2节中定义。

 TOC

13 。序列化

消息使用下列方法之一连载：

查询字符串序列化

序列化形式

JSON序列化

本节介绍这些序列化方法的语法，其他各节描述的时候可以而且必须使用它们。请注意，并非所有的方法可用于所有消息。

 TOC

13.1 。查询字符串序列化

为了使用查询字符串序列化的参数序列化，客户端通过添加参数和值到URL中使用的应用程序/ x - www-窗体- urlencoded格式的查询组件由[ W3C.REC - 定义构造字符串HTML401 - 19991224 ] 。查询字符串序列化通常用于HTTP GET请求。将参数添加到URL的片段组成部分时相同的序列化方法也使用。

以下是此序列的非规范性的例子（带线值的范围内包裹仅用于显示目的）

GET /authorize?

response\_type=code

&scope=openid

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb HTTP/1.1

Host: server.example.com

13.2。序列化形式

参数和它们的值被序列化格式通过使用应用程序/ x-www-窗体-urlencoded格式所界定的[W3C.REC-HTML401-19991224]加入参数名称和值的HTTP请求的实体主体。形式序列通常用在HTTP POST请求。

以下是此序列的非规范性的例子（带线值的范围内包裹仅用于显示目的）??：

POST /authorize HTTP/1.1

Host: server.example.com

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

response\_type=code

&scope=openid

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

13.3。 JSON序列化

该参数由最高级别的结构将每个参数序列化为JSON对象的结构。参数名和字符串值表示为JSON字符串。数值表示为JSON的数字。布尔值表示为JSON布尔值。省略参数和参数没有值应该从物体被省略，而不是由一个JSON空值表示，除非另有规定。参数可以有一个JSON对象或JSON数组作为它的值。

以下是此序列的非规范性的例子：

{

"access\_token": "SlAV32hkKG",

"token\_type": "Bearer",

"expires\_in": 3600,

"refresh\_token": "8xLOxBtZp8"

}

14 。字符串操作

处理一些OpenID的连接信息，需要比较在消息值已知值。例如，使用者资讯端点返回的索赔名称可能是比较具体的索赔名称如子。比较Unicode字符串，但是，具有显著的安全隐患。

因此， JSON字符串和其他Unicode字符串之间的比较而必须执行如下规定：

删除任何JSON的应用逸出，产生的Unicode代码点的数组。

Unicode标准化[ USA15 ]不能在任何时候适用于任何JSON字符串或者是对要比较的字符串。

两个字符串之间的比较必须进行为Unicode代码点代码点相等比较。

在一些地方，这种规范使用字符串的空格分隔列表。在所有这些情况下，一个单一的ASCII空格字符（ 0X20 ）必须作为分隔符。

 TOC

15 。实施注意事项

该规范定义了使用这两种依赖方和OpenID提供者的功能。预计一些支持OpenID提供商将需要静态的，出带外配置RP上的使用它们，而其他人将支持动态使用情况的RP没有它们之间预先建立的关系。出于这个原因，强制到实现的功能为有机磷农药列举如下两类：第一为所有OP和第二个为“动态??”的OpenID提供商。

 TOC

15.1 。强制性地对所有的OpenID提供商特点

所有的OpenID提供商必须实现本规范中定义了以下功能。此列表扩充了一套已经在其他地方为“要求”上市或与“必须”进行了描述，所以是没有，本身一套完整的用于有机磷农药实施要求的功能。

签名ID令牌和RSA SHA- 256

有机磷农药必须支持签名ID令牌与RSA SHA-256算法（ RS256的ALG值） ，除非OP只支持从令牌端点返回的ID令牌（如对于授权码流的情况下） ，只允许客户登记指定none作为请求的ID令牌签名算法。

提示参数

有机磷农药必须支持的提示参数，如第3.1.2节中定义，包括指定用户界面的行为，如没有和登录。

显示参数

有机磷农药必须支持的显示参数，如第3.1.2节中定义。 （请注意，需要此参数支持的最低水平很简单，就是它的使用必须不会导致错误。 ）

首选语言环境

有机磷农药必须通过ui\_locales和claims\_locales请求参数支持首选语言，用户界面和脚本的索赔请求，如第3.1.2节中定义。 （请注意，所需要的这些参数支持的最低水平简直是让他们的使用不会导致错误。 ）

验证时间

有机磷农药必须支持要求时，如第2条所界定返回在该最终用户通过auth\_time索赔，经过身份验证的时间。

最大年龄验证

有机磷农药必须支持通过MAX\_AGE参数执行的最高认证的年龄，如第3.1.2节中定义。

验证上下文类参考

有机磷农药必须通过acr\_values??参数支持特定认证背景类参考值的要求，如第3.1.2节中定义。 （请注意，需要此参数支持的最低水平简直是有它的使用不会导致错误。 ）

 TOC

15.2 。强制实现动态的OpenID提供商特点

除了上面列出的功能，支持OpenID提供商支持动态建立与RP的关系，他们没有与必须在一个预先配置的关系也实现了??在这个定义和相关规范了以下功能。

响应类型

这些支持OpenID提供商必须支持id\_token响应类型和所有不属于自颁发有机磷农药也必须支持代码和id\_token令牌响应类型。

发现

这些有机磷农药必须支持的发现，如OpenID的连接愉1.0 [ OpenID.Discovery ]中定义。

动态注册

这些有机磷农药必须支持动态客户端注册，如OpenID的动态连接客户端注册1.0 [ OpenID.Registration ]中定义。

使用者资讯端点

那发出访问令牌的所有动态的OP必须支持的UserInfo端点，如第5.3节中定义。 （自颁发有机磷农药不发出访问令牌。 ）

公共密钥发布时间为裸钥匙

这些有机磷农药必须公布他们的公钥作为被动JWK键（这也可能伴随着这些密钥的X.509表示） 。

请求URI

这些有机磷农药必须支持使用从一个请求URI所提供的REQUEST\_URI参数，如第3.1.2节中定义一个检索请求对象的价值提出了要求。

 TOC

15.3 。发现和注册

一些OpenID的连接安装可以使用预配置设定的OpenID提供商和/或依赖方。在这种情况下，它可能没有必要支持有关标识或服务或客户端的动态注册信息动态发现。

然而，如果设备选择支持依赖方，而不必预先配置的关系的OpenID提供商之间的互动意料之外，应通过实施在OpenID的连接愉1.0 [ OpenID.Discovery ]和OpenID的连接动态客户注册中定义的设备完成，这1.0 [ OpenID.Registration ]规格。

 TOC

15.4 。强制实现的功能为信任方

一般情况下，它是由依赖方哪些功能使用OpenID提供商进行交互时，他们使用。然而，有些选择是由他们的OAuth客户的性质决定的，如是否是机密客户，能够保守秘密，在这种情况下，授权码流可能是合适的，或者它是否是一个公共客户端，例如中，一个用户代理为基础的应用程序或一个静态注册本地应用程序，在这种情况下，隐式流量可能是适当的。

当使用OpenID的连接功能，这些列为“需要”或与描述的“必须”是强制性实施，由依赖方使用时。同样，被描述为“可选”这些功能不需要使用或支持的，除非它们提供的价值在特定的应用环境。最后，支持探索的OpenID提供者进行交互时，则OP的发现文档可以用来动态地确定哪些是可以由RP使用OP功能。

 TOC

15.5 。实现说明

 TOC

15.5.1 。授权代码实现说明

当使用授权码或混合流，一个ID令牌从令牌端点响应使用授权码令牌请求返回。一些实现可能会选择在授权码值返回到状态编码有关的ID令牌。其他人可能使用的授权码值作为索引在数据库中存储此状态。

 TOC

15.5.2 。杜撰实现说明

随机数参数值需要包括每会话状态，并不可猜测的攻击者。一种方法来实现这个Web Server的客户端是存储经过加密的随机值作为HttpOnly会话cookie并使用值的加密哈希作为随机数参数。在这种情况下，返回的ID令牌的随机数是相对于会话cookie第三方检测ID令牌重播的哈希值。适用于JavaScript的客户端的相关方法是存储在HTML5的本地存储加密的随机值，并使用该值的加密哈希。

 TOC

15.5.3 。重定向的URI片段操作实现说明

当响应参数在重定向的URI片段返回值，客户需要有用户代理解析片段编码的值，并将其传递给到消费客户的处理逻辑。这直接访问加密API的用户代理可能能够自成体系，例如，与正在用JavaScript编写的所有客户端代码。

但是，如果客户端没有在用户代理，单程运行完全实现这一目标是张贴他们为验证一个Web服务器客户端。

下面是一个客户机可能承载在其redirect\_uri JavaScript文件的一个例子。这是通过从授权服务器的重定向加载。该片段组成部分进行分析，然后通过POST发送到URI ，将验证并使用接收到的信息。

下面是一个重定向的URI响应的非规范性的例子：

GET /cb HTTP/1.1

Host: client.example.org

HTTP/1.1 200 OK

Content-Type: text/html

<script type="text/javascript">

// First, parse the query string

var params = {}, postBody = location.hash.substring(1),

regex = /([^&=]+)=([^&]\*)/g, m;

while (m = regex.exec(postBody)) {

params[decodeURIComponent(m[1])] = decodeURIComponent(m[2]);

}

// And send the token over to the server

var req = new XMLHttpRequest();

// using POST so query isn't logged

req.open('POST', 'https://' + window.location.host +

'/catch\_response', true);

req.setRequestHeader('Content-Type',

'application/x-www-form-urlencoded');

req.onreadystatechange = function (e) {

if (req.readyState == 4) {

if (req.status == 200) {

// If the response from the POST is 200 OK, perform a redirect

window.location = 'https://'

+ window.location.host + '/redirect\_after\_login'

}

// if the OAuth response is invalid, generate an error message

else if (req.status == 400) {

alert('There was an error processing the token')

} else {

alert('Something other than 200 was returned')

}

}

};

req.send(postBody);

15.6。兼容性说明

 TOC

15.6.1。准决赛IETF规范

实施者应注意，本规范使用几个IETF规范尚未最终规格。这些规范是：

JSON的Web令牌（JWT）草案-13 [智威汤逊]

JSON的网络签名（JWS）草案-18 [JWS]

JSON的Web加密（JWE）草案-18 [JWE]

JSON的网络密钥（JWK）草案-18 [JWK]

JSON的Web算法草案-18 [JWA]

断言框架的OAuth 2.0客户端身份验证和授权资助草案-13 [OAuth.Assertions]

JSON的Web令牌（JWT）天寒的OAuth 2.0客户端身份验证和授权资助草案-07 [OAuth.JWT]

虽然将尽一切努力，以防止重大更改这些规格，就应该发生，OpenID的连接实现应该继续使用专门引用以上版本草案优先于最终版本，除非使用的是未来可能的OpenID Connect配置文件或规范的更新一些这些参考文献的部分或全部。

 TOC

15.6.2。谷歌“ISS”的价值

实现者可能想知道，作为写这篇文章的时候，谷歌的部署OpenID的连接执行问题的ID令牌是忽略从国际空间站（发行人）宣告值所需https://开头方案前缀。因此，有意与谷歌合作依赖方实现将需要有解决此代码，直至其执行更新的时间。任何这样的解决方法的代码应写在不会在这样的点谷歌会将缺少的前缀，其发行人值突破的方式。

 TOC

15.7。相关的规范和实施者指南

这些相关的可选规格可用于与本规范相结合，以提供额外的功能：

OpenID的连接愉1.0 [OpenID.Discovery] - 定义如何信任方的动态发现的OpenID提供商的信息

OpenID的动态连接客户端注册1.0 [OpenID.Registration] - 定义依赖方如何动态地使用OpenID提供商注册

OpenID的连接会话管理1.0 [OpenID.Session] - 定义了如何管理OpenID的连接会话，包括退出功能

的OAuth 2.0表单提交响应模式[OAuth.Post] - 定义了如何使用HTTP POST使用被自动提交用户代理HTML表单的值，返回的OAuth 2.0授权响应参数（包括连接的OpenID认证响应参数）

这些实施者指南的目的是作为基本的基于Web的依赖方的实施者自足参考：

OpenID的连接基本客户实施者指南1.0 [OpenID.Basic] - 实施者指南包含本规范正在使用OAuth的授权码流适用于使用基本的基于Web的信任方的一个子集

OpenID的隐式连接客户端实施者指南1.0 [OpenID.Implicit] - 实施者指南包含此规范，使用的是OAuth的隐式流程适用于使用基本的基于Web的依赖方的一个子集

 TOC

16。安全注意事项

本规范中引用的OAuth 2.0 [RFC6749]第10条所定义的安全考虑，以及OAuth的2.0承载令牌用法[RFC6750]第5。此外，OAuth的2.0威胁模型和安全注意事项[RFC6819]规范提供了一个广泛的适用于本规范，以及威胁和控制列表，因为它是基于OAuth的2.0。 ISO / IEC 29115 [ISO29115]还提供了威胁和控制，实现者需要考虑。实施者高度应详细阅读这些文献，并应用所描述的对策。

此外，攻击向量和补救措施的以下列表也算。

 TOC

16.1。要求披露

如果不采取适当的措施，请求可能会被披露给攻击者，冒充安全和隐私威胁。

除此之外，还有什么是在[RFC6819]中第5.1.1节所述，该标准提供了一种方法来提供请求结束的机密结束通过使用请求或REQUEST\_URI参数，其中请求的内容是加密智威汤逊用适当的密钥和密码。这可以保护甚至反对在间接要求的情况下被攻破的用户代理。

 TOC

16.2。服务器伪装

恶意服务器可能伪装成利用各种手段合法的服务器。为了检测这种攻击，客户端需要认证服务器。

除此之外，还有什么是在[RFC6819]中第5.1.2节所述，该标准提供了一种方式或者通过使用签名或加密JWTs与合适的密钥和密码进行身份验证的服务器。

 TOC

16.3。令牌制造/修改

攻击者可能会生成一个假的令牌或修改现有的可解析令牌的令牌的内容（如索赔值或签名），导致RP给予不恰当的访问客户端。例如，攻击者可能会修改解析的令牌来延长有效期，客户端可能会修改解析的令牌来访问，他们不应该能够查看信息。

有两种方法来减轻这种攻击：

令牌可以通过OP进行数字签名。依赖方应验证数字签名，以验证它是由一个合法的OP发行。

该标记可以发送受保护的通道如TLS。见16.17节有关使用TLS的详细信息。在本说明书中，令牌总是发送通过TLS保护的通道。但是请注意，这项措施只针对第三方攻击的自卫，并不适用于如客户为攻击者的情况。

 TOC

16.4。访问令牌披露

访问令牌是用于访问受保护资源的凭据，如的OAuth 2.0 [RFC6749]的1.4节中定义。访问令牌代表了最终用户的认可，并不得暴露给未经授权的第三方。

 TOC

16.5。服务器响应披露

服务器的响应可能包含认证数据和索赔，其中包括敏感的客户信息。公开的响应内容可以使客户端易受其它类型的攻击。

服务器响应的披露可以减轻在以下两个方面：

使用代码响应类型。该响应发送一个TLS保护的通道，那里的客户端是由CLIENT\_ID和CLIENT\_SECRET验证。

对于其他类型的响应，签名的响应可以与客户端的公钥或共享密钥作为加密的智威汤逊用适当的密钥和密码进行加密。

 TOC

16.6。服务器响应赖账

的响应可能会被服务器拒绝接受，如果适当的机制不到位。例如，如果服务器没有数字签名响应，服务器可以声称，它没有通过服务器的服务产生。

为了减轻这一威胁，反应可能是由数位使用支持不可抵赖性的关键服务器签名。客户端应该验证数字签名，以验证它是由一个合法的服务器发行，其完整性是完整的。

 TOC

16.7。要求赖账

因为它是可能的妥协，或恶意的客户端将请求发送到错误的党，这是只使用一个承载令牌可以否定任何交易验证的客户端。

为了减轻这种威胁，服务器可能要求请求由客户端使用支持不可抵赖性的一个关键数字签名。服务器应该验证数字签名，以验证它是由一个合法的客户端发出和完整性是完整的。

 TOC

16.8。访问令牌重定向

攻击者使用一个资源所产生的访问令牌，以获得进入第二个资源。

为了减轻这一威胁，访问令牌应该是观众和范围的限制。实现它的一种方法是，包括人，这是因为生成的观众的资源的标识符。资源验证传入令牌包括其作为标记的观众标识符。

 TOC

16.9。令牌重用

攻击者试图使用一个一次性使用的令牌，如授权的代码，已经使用过一次与预期资源。为了减轻这种威胁，令牌应该包含一个时间戳和一个短效度一生。依赖方然后检查时间戳和生存期值，以确保令牌是当前有效的。

或者，服务器可以记录使用令牌的状态，并检查每个请求的状态。

 TOC

16.10。窃听或泄漏授权码（中学验证器捕捉）

除了在[RFC6819]中第4.4.1.1节中描述的攻击模式，一个授权码可以被捕获在用户代理那里的TLS会话时，如果用户代理是感染了恶意软件终止。然而，捕获它是没有用的，只要不是客户端身份验证或加密的响应被使用。

 TOC

16.11。标记替换

标记替换为一个恶意的用户交换各种令牌，包括交换一个授权码与另一个令牌，攻击者有一个合法用户的一类攻击。实现这一点的一种方法是攻击者复制令牌出一个会话，并用它在HTTP消息不同的会话，这是很容易做到当令牌是提供给浏览器，这被称为“切粘贴“攻击??。

的OAuth 2.0 [RFC6749]的隐流的目的不是要降低这种风险。在第10.16节，它规范性规定，任何使用授权过程的委派最终用户身份验证的客户端的形式绝不能使用隐式流量不采用额外的安全机制，使客户以确定是否ID令牌和访问令牌发行供其使用。

在OpenID的连接，这是通过通过ID令牌提供的机制缓解。该ID令牌是签名的安全令牌，提供索赔，如ISS（发行人），子（主题），澳元（观众），AZP（授权方），at\_hash（访问令牌哈希），和c\_hash（哈希代码）。使用ID令牌，客户能够检测的标记替换攻击。

在ID令牌的c\_hash使客户能够防止授权码替换。在ID令牌的at\_hash使客户能够阻止访问令牌替换。

此外，恶意用户可能会试图通过颠覆授权端点和客户端，或令牌端点和客户端之间的通信通道来模拟一个更高权限的用户，例如通过交换的授权码或重新排序的信息，说服令牌端点攻击者的权限授予对应于派代更特权用户的补助金。

对于本规范定义绑定在HTTP的响应令牌请求是在HTTP绑定到相应的请求通过消息顺序，既包含令牌和请求的响应是由TLS，这将检测并阻止数据包重新排序的保护。

当设计另一个绑定本规范的一种协议无能力强绑定令牌端点请求的响应，以解决此问题的其他机制必须加以利用。这样的机制之一可能是包括ID令牌，在令牌请求和响应c\_hash索赔。

 TOC

16.12。时序攻击

定时攻击可以使攻击者通过采取成功的和不成功的解密操作或消息的成功和不成功的签名验证的代码路径所经过的时间差，获得了不必要的大量信息。它可用于降低所用的密码的有效密钥长度。

实现不应该在发现错误的瞬间终止确认的过程，但应该继续运行，直到所有的字节，已处理，以避免这种攻击。

 TOC

16.13。其他加密相关的攻击

有根据用于加密和签名/完整性检查的方法不同加密相关的可能的攻击。实施者需要咨询的安全注意事项JWT [智威汤逊]规范和规范，它引用，以避免在这些规范中确定的漏洞。

 TOC

16.14。签名和加密订购

在签名加密文本不被视为有效在许多司法管辖区。因此，完整性和不可抵赖性，本规范要求签订纯文本的JSON索赔，当进行签约。如果两个签名和加密是理想的，它是在包含已签名的声明的JWS执行，结果是一个嵌套智威汤逊，如[智威汤逊]中指定。需要注意的是，因为所有的JWE加密算法提供完整性保护，就没有必要单独签名的加密内容。

 TOC

16.15。发行人标识符

OpenID的连接支持每个主机和端口组合的多个发行人。通过发现返回的发行人必须ISS中的ID标记的值完全匹配。

OpenID的连接对待任何发行URI的路径组件作为发行人标识符的一部分。例如，主题为“1234”与“https://example.com”的发行商标识符不等于主题为“1234”与“https://example.com/sales”的发行商标识符。

推荐的做法是每台主机上只有一个单一的发行人使用。但是，如果主机支持多个租户，可能需要多个发行人该主机。

 TOC

16.16。隐流威胁

在隐流量，访问令牌是通过HTTPS返回在客户端的redirect\_uri的片段组成部分，因此，它是OP和用户代理之间的保护，用户代理和RP之间。它可以被捕获的唯一的地方是用户代理那里的TLS会话被终止，这是可能的，如果用户代理是感染了恶意软件或恶意方的控制之下。

 TOC

16.17。 TLS的要求

实现必须支持TLS。哪个版本的（S）应该实施将随时间而变化，而且依赖于广泛的部署和已知的安全漏洞，在执行的时候。在写这篇文章的时候，TLS 1.2版[RFC5246]是最新的版本，但已经非常有限实际部署，并且可能不容易在执行工具包。 TLS 1.0版[RFC2246]是最广泛部署的版本，并给最广泛的互操作性。

为了防止信息泄露和篡改，机密性保护必须使用TLS用一个密码套件，它提供机密性和完整性保护的应用。

当使用TLS，TLS一个服务器证书检查必须进行，根据RFC 6125 [RFC6125]。

 TOC

16.18。访问令牌和刷新令牌的寿命

由授权服务器访问令牌可能无法撤销。访问令牌的寿命，因此应保持单一使用或很短的寿命。

如果持续存取使用者资讯端点或其他受保护的资源是必需的，一个刷新令牌都可以使用。然后，客户端可以交换刷新令牌的令牌端点的一个新的短暂访问令牌可以用来访问资源。

授权服务器应明确长期的拨款，给予用户授权时。授权服务器应该提供一种机制，用于最终用户取消访问令牌和刷新令牌到客户端授权。

 TOC

16.19。对称密钥熵

在第10.1节和第10.2节，钥匙都来源于CLIENT\_SECRET值。因此，对称签名和加密操作使用时，CLIENT\_SECRET值必须包含足够的熵产生强加密密钥。此外，CLIENT\_SECRET值也必须至少包含最小的所需所使用的特定算法的MAC密钥字节数的。因此，例如，对于HS256的CLIENT\_SECRET值必须至少包含32个字节（而且几乎可以肯定应该包含更多的，因为CLIENT\_SECRET值有可能使用受限制的字母）。

 TOC

16.20。需要签名请求

在某些情况下，客户端可能需要使用签名的请求，以确保所需的请求参数被传递到OP没有处于被篡改。例如，MAX\_AGE和acr\_values??提供关于何时在签名请求交付执行的认证的性质更有保证。

 TOC

16.21。需要加密的请求

在某些情况下，知道一个OpenID连接请求的内容可以在其本身，揭示了最终用户的敏感信息。例如，知道客户端请求一个特定的索赔或者说，它正在请求使用一个特定的身份验证方法可以揭示有关最终用户的敏感信息。 OpenID的连接请求，使得被加密的OpenID提供者，以防止这种潜在的敏感信息被泄露。

 TOC

17。隐私考量

 TOC

17.1。个人身份信息

使用者资讯的响应通常包含个人身份信息（PII）。因此，最终用户同意的信息，指定用途的释放应在或之前批准的时间按照有关规定获??得。使用目的一般是登记在与redirect\_uris关联。

只需要使用者资讯的数据应该存储在客户端和客户端应该接收到的数据与使用声明的目的相关联。

 TOC

17.2。数据访问监控

资源服务器应该提供给他们的最终用户的使用者资讯的访问日志，以便他们可以监控谁访问他们的数据。

17.3。相关

为了保护最终用户从客户端之间可能存在的关联，使用成对假名标识符（PPID）为子（主题）应予以考虑。

 TOC

17.4。离线访问

离线访问可以访问，当用户不存在债权，构成更大的隐私风险比债权转让时，该用户是否存在。因此，谨慎的做法是获得离线访问资源的明确同意。该规范要求使用的提示参数，以获得同意，除非它已经知道该请求符合用于处理各个司法管辖区的要求条件。

当一个访问令牌是通过使用隐式流或混合流的用户代理返回的，那里是它被暴露给攻击者，谁可以在以后使用它来访问的UserInfo端点的风险更大。如果访问令牌不启用脱机访问和服务器可以区分客户是否要求已脱机或联机，风险将大幅降低。因此，本规范要求忽视了离线访问请求时，访问令牌是通过用户代理发送。请注意，从服务器联机和脱机访问区分是很困难的特别是对于本地客户端。该服务器可能要依靠启发式。此外，曝光的访问令牌的风险通过用户代理交付的代码令牌和令牌的响应类型是一样的。因此，实现应做好检测访问令牌是否是通过用户代理或直接从令牌端点发出，并拒绝如果令牌是通过用户代理发行的离线访问。

需要注意的是，虽然这些规定需要明确同意通过对话提示参数，仅仅因为用户按下“接受”按钮等，可能不构成有效的同意。开发人员应该知道，对于同意是有效的行为，通常情况下，该条款的影响，必须由最终用户所理解的，同意必须是自愿给予的，而不是强迫（即其他选项必须可用） ，其条款必须公平和公正。一般情况下，它是最好的服务，以按照每个司法管辖区所需要的隐私原则，依靠其他条件的处理而非简单明确同意的请求，在线自助服务“明确同意”往往不形成一些有效的同意司法管辖区。

8。 IANA考虑

 TOC

18.1。 JSON网站令牌债权登记

本规范中注册第5.1节和在[智威汤逊]中定义的IANA JSON的Web令牌债权登记第2节规定的索赔。

 TOC

18.1.1。注册表内容

* Claim Name: name
* Claim Description: Full name
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: given\_name
* Claim Description: Given name(s) or first name(s)
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: family\_name
* Claim Description: Surname(s) or last name(s)
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: middle\_name
* Claim Description: Middle name(s)
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: nickname
* Claim Description: Casual name
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: preferred\_username
* Claim Description: Shorthand name by which the End-User wishes to be referred to
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: profile
* Claim Description: Profile page URL
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: picture
* Claim Description: Profile picture URL
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: website
* Claim Description: Web page or blog URL
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: email
* Claim Description: Preferred e-mail address
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: email\_verified
* Claim Description: True if the e-mail address has been verified; otherwise false
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: gender
* Claim Description: Gender
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: birthdate
* Claim Description: Birthday
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: zoneinfo
* Claim Description: Time zone
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: locale
* Claim Description: Locale
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: phone\_number
* Claim Description: Preferred telephone number
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: phone\_number\_verified
* Claim Description: True if the phone number has been verified; otherwise false
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: address
* Claim Description: Preferred postal address
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: updated\_at
* Claim Description: Time the information was last updated
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 5.1**](http://openid.net/specs/openid-connect-core-1_0.html#StandardClaims) of this document
* Claim Name: azp
* Claim Description: Authorized party - the party to which the ID Token was issued
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 2**](http://openid.net/specs/openid-connect-core-1_0.html#IDToken) of this document
* Claim Name: nonce
* Claim Description: Value used to associate a Client session with an ID Token
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 2**](http://openid.net/specs/openid-connect-core-1_0.html#IDToken) of this document
* Claim Name: auth\_time
* Claim Description: Time when the authentication occurred
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 2**](http://openid.net/specs/openid-connect-core-1_0.html#IDToken) of this document
* Claim Name: at\_hash
* Claim Description: Access Token hash value
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 2**](http://openid.net/specs/openid-connect-core-1_0.html#IDToken) of this document
* Claim Name: c\_hash
* Claim Description: Code hash value
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 3.3.2.11**](http://openid.net/specs/openid-connect-core-1_0.html#HybridIDToken) of this document
* Claim Name: acr
* Claim Description: Authentication Context Class Reference
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 2**](http://openid.net/specs/openid-connect-core-1_0.html#IDToken) of this document
* Claim Name: amr
* Claim Description: Authentication Methods References
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 2**](http://openid.net/specs/openid-connect-core-1_0.html#IDToken) of this document
* Claim Name: sub\_jwk
* Claim Description: Public key used to check the signature of an ID Token
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [**Section 7.4**](http://openid.net/specs/openid-connect-core-1_0.html#SelfIssuedResponse) of this document

18.2。 OAuth的参数注册

本规范登记在RFC6749[RFC6749]中定义的IANA的OAuth参数注册表中的以下参数。

18.2.1。注册表内容

* Parameter name: nonce
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2**](http://openid.net/specs/openid-connect-core-1_0.html#AuthorizationEndpoint) of this document
* Related information: None
* Parameter name: display
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2**](http://openid.net/specs/openid-connect-core-1_0.html#AuthorizationEndpoint) of this document
* Related information: None
* Parameter name: prompt
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2**](http://openid.net/specs/openid-connect-core-1_0.html#AuthorizationEndpoint) of this document
* Related information: None
* Parameter name: max\_age
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2**](http://openid.net/specs/openid-connect-core-1_0.html#AuthorizationEndpoint) of this document
* Related information: None
* Parameter name: ui\_locales
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2**](http://openid.net/specs/openid-connect-core-1_0.html#AuthorizationEndpoint) of this document
* Related information: None
* Parameter name: claims\_locales
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 5.2**](http://openid.net/specs/openid-connect-core-1_0.html#ClaimsLanguagesAndScripts) of this document
* Related information: None
* Parameter name: id\_token\_hint
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2**](http://openid.net/specs/openid-connect-core-1_0.html#AuthorizationEndpoint) of this document
* Related information: None
* Parameter name: login\_hint
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2**](http://openid.net/specs/openid-connect-core-1_0.html#AuthorizationEndpoint) of this document
* Related information: None
* Parameter name: acr\_values
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2**](http://openid.net/specs/openid-connect-core-1_0.html#AuthorizationEndpoint) of this document
* Related information: None
* Parameter name: claims
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 5.5**](http://openid.net/specs/openid-connect-core-1_0.html#ClaimsParameter) of this document
* Related information: None
* Parameter name: registration
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 7.2.1**](http://openid.net/specs/openid-connect-core-1_0.html#RegistrationParameter) of this document
* Related information: None
* Parameter name: request
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 6**](http://openid.net/specs/openid-connect-core-1_0.html#JWTRequests) of this document
* Related information: None
* Parameter name: request\_uri
* Parameter usage location: Authorization Request
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 6**](http://openid.net/specs/openid-connect-core-1_0.html#JWTRequests) of this document
* Related information: None
* Parameter name: id\_token
* Parameter usage location: Authorization Response, Access Token Response
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.3.3**](http://openid.net/specs/openid-connect-core-1_0.html#TokenResponse) of this document
* Related information: None

18.3。 OAuth的扩展错误登记

本规范登记在RFC6749[RFC6749]中定义的IANA的OAuth扩展错误注册表中的以下错误。

18.3.1。注册表内容

* Error name: interaction\_required
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document
* Error name: login\_required
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document
* Error name: account\_selection\_required
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document
* Error name: consent\_required
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document
* Error name: invalid\_request\_uri
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document
* Error name: invalid\_request\_object
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document
* Error name: request\_not\_supported
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document
* Error name: request\_uri\_not\_supported
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document
* Error name: registration\_not\_supported
* Error usage location: Authorization Endpoint
* Related protocol extension: OpenID Connect
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3.1.2.6**](http://openid.net/specs/openid-connect-core-1_0.html#AuthError) of this document

### 19.  References

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### 19.1. Normative References

|  |  |
| --- | --- |
| **[CORS]** | Opera Software ASA, “[**Cross-Origin Resource Sharing**](http://www.w3.org/TR/access-control/),” July 2010. |
| **[E.164]** | International Telecommunication Union, “[**E.164: The international public telecommunication numbering plan**](http://www.itu.int/rec/T-REC-E.164-201011-I/en),” 2010. |
| **[IANA.Language]** | Internet Assigned Numbers Authority (IANA), “[**Language Subtag Registry**](http://www.iana.org/assignments/language-subtag-registry),” 2005. |
| **[ISO29115]** | International Organization for Standardization, “[**ISO/IEC 29115:2013 -- Information technology - Security techniques - Entity authentication assurance framework**](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=45138),” ISO/IEC 29115, March 2013. |
| **[ISO3166-1]** | International Organization for Standardization, “[**ISO 3166-1:1997. Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes**](http://www.w3.org/WAI/ER/IG/ert/iso639.htm),” 1997. |
| **[ISO639-1]** | International Organization for Standardization, “ISO 639-1:2002. Codes for the representation of names of languages -- Part 1: Alpha-2 code,” 2002. |
| **[ISO8601-2004]** | International Organization for Standardization, “ISO 8601:2004. Data elements and interchange formats - Information interchange - Representation of dates and times,” 2004. |
| **[JWA]** | Jones, M., “[**JSON Web Algorithms (JWA)**](http://tools.ietf.org/html/draft-ietf-jose-json-web-algorithms),” draft-ietf-jose-json-web-algorithms (work in progress), November 2013 ([**HTML**](http://tools.ietf.org/html/draft-ietf-jose-json-web-algorithms-18)). |
| **[JWE]** | Jones, M., Rescorla, E., and J. Hildebrand, “[**JSON Web Encryption (JWE)**](http://tools.ietf.org/html/draft-ietf-jose-json-web-encryption),” draft-ietf-jose-json-web-encryption (work in progress), November 2013 ([**HTML**](http://tools.ietf.org/html/draft-ietf-jose-json-web-encryption-18)). |
| **[JWK]** | Jones, M., “[**JSON Web Key (JWK)**](http://tools.ietf.org/html/draft-ietf-jose-json-web-key),” draft-ietf-jose-json-web-key (work in progress), November 2013 ([**HTML**](http://tools.ietf.org/html/draft-ietf-jose-json-web-key-18)). |
| **[JWS]** | Jones, M., Bradley, J., and N. Sakimura, “[**JSON Web Signature (JWS)**](http://tools.ietf.org/html/draft-ietf-jose-json-web-signature),” draft-ietf-jose-json-web-signature (work in progress), November 2013 ([**HTML**](http://tools.ietf.org/html/draft-ietf-jose-json-web-signature-18)). |
| **[JWT]** | Jones, M., Bradley, J., and N. Sakimura, “[**JSON Web Token (JWT)**](http://tools.ietf.org/html/draft-ietf-oauth-json-web-token),” draft-ietf-oauth-json-web-token (work in progress), November 2013 ([**HTML**](http://tools.ietf.org/html/draft-ietf-oauth-json-web-token-13)). |
| **[OAuth.Assertions]** | Campbell, B., Mortimore, C., Jones, M., and Y. Goland, “[**Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants**](http://tools.ietf.org/html/draft-ietf-oauth-assertions),” draft-ietf-oauth-assertions (work in progress), December 2013 ([**HTML**](http://tools.ietf.org/html/draft-ietf-oauth-assertions-13)). |
| **[OAuth.JWT]** | Jones, M., Campbell, B., and C. Mortimore, “[**JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants**](http://tools.ietf.org/html/draft-ietf-oauth-jwt-bearer),” draft-ietf-oauth-jwt-bearer (work in progress), December 2013 ([**HTML**](http://tools.ietf.org/html/draft-ietf-oauth-jwt-bearer-07)). |
| **[OAuth.Responses]** | de Medeiros, B., Ed., Scurtescu, M., Tarjan, P., and M. Jones, “[**OAuth 2.0 Multiple Response Type Encoding Practices**](http://openid.net/specs/oauth-v2-multiple-response-types-1_0.html),” February 2014. |
| **[OpenID.Discovery]** | Sakimura, N., Bradley, J., Jones, M., and E. Jay, “[**OpenID Connect Discovery 1.0**](http://openid.net/specs/openid-connect-discovery-1_0.html),” February 2014. |
| **[OpenID.Registration]** | Sakimura, N., Bradley, J., and M. Jones, “[**OpenID Connect Dynamic Client Registration 1.0**](http://openid.net/specs/openid-connect-registration-1_0.html),” February 2014. |
| **[RFC2119]** | [**Bradner, S.**](mailto:sob@harvard.edu), “[**Key words for use in RFCs to Indicate Requirement Levels**](http://tools.ietf.org/html/rfc2119),” BCP 14, RFC 2119, March 1997 ([**TXT**](http://www.rfc-editor.org/rfc/rfc2119.txt), [**HTML**](http://xml.resource.org/public/rfc/html/rfc2119.html), [**XML**](http://xml.resource.org/public/rfc/xml/rfc2119.xml)). |
| **[RFC2246]** | [**Dierks, T.**](mailto:tdierks@certicom.com) and [**C. Allen**](mailto:callen@certicom.com), “[**The TLS Protocol Version 1.0**](http://tools.ietf.org/html/rfc2246),” RFC 2246, January 1999 ([**TXT**](http://www.rfc-editor.org/rfc/rfc2246.txt)). |
| **[RFC2616]** | [**Fielding, R.**](mailto:fielding@ics.uci.edu), [**Gettys, J.**](mailto:jg@w3.org), [**Mogul, J.**](mailto:mogul@wrl.dec.com), [**Frystyk, H.**](mailto:frystyk@w3.org), [**Masinter, L.**](mailto:masinter@parc.xerox.com), [**Leach, P.**](mailto:paulle@microsoft.com), and [**T. Berners-Lee**](mailto:timbl@w3.org), “[**Hypertext Transfer Protocol -- HTTP/1.1**](http://tools.ietf.org/html/rfc2616),” RFC 2616, June 1999 ([**TXT**](http://www.rfc-editor.org/rfc/rfc2616.txt), [**PS**](http://www.rfc-editor.org/rfc/rfc2616.ps), [**PDF**](http://www.rfc-editor.org/rfc/rfc2616.pdf), [**HTML**](http://xml.resource.org/public/rfc/html/rfc2616.html), [**XML**](http://xml.resource.org/public/rfc/xml/rfc2616.xml)). |
| **[RFC3339]** | [**Klyne, G., Ed.**](mailto:GK@ACM.ORG) and [**C. Newman**](mailto:chris.newman@sun.com), “[**Date and Time on the Internet: Timestamps**](http://tools.ietf.org/html/rfc3339),” RFC 3339, July 2002 ([**TXT**](http://www.rfc-editor.org/rfc/rfc3339.txt), [**HTML**](http://xml.resource.org/public/rfc/html/rfc3339.html), [**XML**](http://xml.resource.org/public/rfc/xml/rfc3339.xml)). |
| **[RFC3966]** | Schulzrinne, H., “[**The tel URI for Telephone Numbers**](http://tools.ietf.org/html/rfc3966),” RFC 3966, December 2004 ([**TXT**](http://www.rfc-editor.org/rfc/rfc3966.txt)). |
| **[RFC3986]** | [**Berners-Lee, T.**](mailto:timbl@w3.org), [**Fielding, R.**](mailto:fielding@gbiv.com), and [**L. Masinter**](mailto:LMM@acm.org), “[**Uniform Resource Identifier (URI): Generic Syntax**](http://tools.ietf.org/html/rfc3986),” STD 66, RFC 3986, January 2005 ([**TXT**](http://www.rfc-editor.org/rfc/rfc3986.txt),**[HTML](http://xml.resource.org/public/rfc/html/rfc3986.html)**, [**XML**](http://xml.resource.org/public/rfc/xml/rfc3986.xml)). |
| **[RFC4627]** | Crockford, D., “[**The application/json Media Type for JavaScript Object Notation (JSON)**](http://tools.ietf.org/html/rfc4627),” RFC 4627, July 2006 ([**TXT**](http://www.rfc-editor.org/rfc/rfc4627.txt)). |
| **[RFC5246]** | Dierks, T. and E. Rescorla, “[**The Transport Layer Security (TLS) Protocol Version 1.2**](http://tools.ietf.org/html/rfc5246),” RFC 5246, August 2008 ([**TXT**](http://www.rfc-editor.org/rfc/rfc5246.txt)). |
| **[RFC5322]** | [**Resnick, P., Ed.**](mailto:presnick@qualcomm.com), “[**Internet Message Format**](http://tools.ietf.org/html/rfc5322),” RFC 5322, October 2008 ([**TXT**](http://www.rfc-editor.org/rfc/rfc5322.txt), [**HTML**](http://xml.resource.org/public/rfc/html/rfc5322.html), [**XML**](http://xml.resource.org/public/rfc/xml/rfc5322.xml)). |
| **[RFC5646]** | Phillips, A. and M. Davis, “[**Tags for Identifying Languages**](http://tools.ietf.org/html/rfc5646),” BCP 47, RFC 5646, September 2009 ([**TXT**](http://www.rfc-editor.org/rfc/rfc5646.txt)). |
| **[RFC6125]** | Saint-Andre, P. and J. Hodges, “[**Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)**](http://tools.ietf.org/html/rfc6125),” RFC 6125, March 2011 ([**TXT**](http://www.rfc-editor.org/rfc/rfc6125.txt)). |
| **[RFC6711]** | Johansson, L., “[**An IANA Registry for Level of Assurance (LoA) Profiles**](http://tools.ietf.org/html/rfc6711),” RFC 6711, August 2012 ([**TXT**](http://www.rfc-editor.org/rfc/rfc6711.txt)). |
| **[RFC6749]** | Hardt, D., “[**The OAuth 2.0 Authorization Framework**](http://tools.ietf.org/html/rfc6749),” RFC 6749, October 2012 ([**TXT**](http://www.rfc-editor.org/rfc/rfc6749.txt)). |
| **[RFC6750]** | Jones, M. and D. Hardt, “[**The OAuth 2.0 Authorization Framework: Bearer Token Usage**](http://tools.ietf.org/html/rfc6750),” RFC 6750, October 2012 ([**TXT**](http://www.rfc-editor.org/rfc/rfc6750.txt)). |
| **[RFC6819]** | Lodderstedt, T., McGloin, M., and P. Hunt, “[**OAuth 2.0 Threat Model and Security Considerations**](http://tools.ietf.org/html/rfc6819),” RFC 6819, January 2013 ([**TXT**](http://www.rfc-editor.org/rfc/rfc6819.txt)). |
| **[USA15]** | [**Davis, M.**](mailto:markdavis@google.com), [**Whistler, K.**](mailto:ken@unicode.org), and M. Dürst, “Unicode Normalization Forms,” Unicode Standard Annex 15, 09 2009. |
| **[W3C.REC-html401-19991224]** | Raggett, D., Hors, A., and I. Jacobs, “[**HTML 4.01 Specification**](http://www.w3.org/TR/1999/REC-html401-19991224),” World Wide Web Consortium Recommendation REC-html401-19991224, December 1999 ([**HTML**](http://www.w3.org/TR/1999/REC-html401-19991224)). |
| **[zoneinfo]** | Public Domain, “[**The tz database**](http://www.twinsun.com/tz/tz-link.htm),” June 2011. |

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### 19.2. Informative References

|  |  |
| --- | --- |
| **[OAuth.Post]** | Jones, M. and B. Campbell, “[**OAuth 2.0 Form Post Response Mode**](http://openid.net/specs/oauth-v2-form-post-response-mode-1_0.html),” February 2014. |
| **[OpenID.2.0]** | OpenID Foundation, “OpenID Authentication 2.0,” December 2007 ([**TXT**](http://openid.net/specs/openid-authentication-2_0.txt), [**HTML**](http://openid.net/specs/openid-authentication-2_0.html)). |
| **[OpenID.Basic]** | Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, “[**OpenID Connect Basic Client Implementer's Guide 1.0**](http://openid.net/specs/openid-connect-basic-1_0.html),” February 2014. |
| **[OpenID.Implicit]** | Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, “[**OpenID Connect Implicit Client Implementer's Guide 1.0**](http://openid.net/specs/openid-connect-implicit-1_0.html),” February 2014. |
| **[OpenID.PAPE]** | [**Recordon, D.**](mailto:david@sixapart.com), [**Jones, M.**](mailto:mbj@microsoft.com), [**Bufu, J., Ed.**](mailto:johnny.bufu@gmail.com), [**Daugherty, J., Ed.**](mailto:cygnus@janrain.com), and [**N. Sakimura**](mailto:n-sakimura@nri.co.jp), “OpenID Provider Authentication Policy Extension 1.0,” December 2008 ([**TXT**](http://openid.net/specs/openid-provider-authentication-policy-extension-1_0.txt), [**HTML**](http://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html)). |
| **[OpenID.Session]** | Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “[**OpenID Connect Session Management 1.0**](http://openid.net/specs/openid-connect-session-1_0.html),” February 2014. |
| **[RFC4949]** | Shirey, R., “[**Internet Security Glossary, Version 2**](http://tools.ietf.org/html/rfc4949),” RFC 4949, August 2007 ([**TXT**](http://www.rfc-editor.org/rfc/rfc4949.txt)). |
| **[X.1252]** | International Telecommunication Union, “[**ITU-T Recommendation X.1252 -- Cyberspace security -- Identity management -- Baseline identity management terms and definitions**](http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.1252-201004-I!!PDF-E&type=items),” ITU-T X.1252, November 2010. |

附录A.授权示例

以下是授权请求不同RESPONSE\_TYPE的价值观和他们的反应（与行包内的值用于显示目的）的非规范性的例子：

### A.1.  Example using response\_type=code

GET /authorize?

response\_type=code

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile%20email

&nonce=n-0S6\_WzA2Mj

&state=af0ifjsldkj HTTP/1.1

Host: server.example.com

HTTP/1.1 302 Found

Location: https://client.example.org/cb?

code=Qcb0Orv1zh30vL1MPRsbm-diHiMwcLyZvn1arpZv-Jxf\_11jnpEX3Tgfvk

&state=af0ifjsldkj

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### A.2.  Example using response\_type=id\_token

GET /authorize?

response\_type=id\_token

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile%20email

&nonce=n-0S6\_WzA2Mj

&state=af0ifjsldkj HTTP/1.1

Host: server.example.com

HTTP/1.1 302 Found

Location: https://client.example.org/cb#

id\_token=eyJraWQiOiIxZTlnZGs3IiwiYWxnIjoiUlMyNTYifQ.ewogImlz

cyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4

Mjg5NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiAi

bi0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDEz

MTEyODA5NzAsCiAibmFtZSI6ICJKYW5lIERvZSIsCiAiZ2l2ZW5fbmFtZSI6

ICJKYW5lIiwKICJmYW1pbHlfbmFtZSI6ICJEb2UiLAogImdlbmRlciI6ICJm

ZW1hbGUiLAogImJpcnRoZGF0ZSI6ICIwMDAwLTEwLTMxIiwKICJlbWFpbCI6

ICJqYW5lZG9lQGV4YW1wbGUuY29tIiwKICJwaWN0dXJlIjogImh0dHA6Ly9l

eGFtcGxlLmNvbS9qYW5lZG9lL21lLmpwZyIKfQ.rHQjEmBqn9Jre0OLykYNn

spA10Qql2rvx4FsD00jwlB0Sym4NzpgvPKsDjn\_wMkHxcp6CilPcoKrWHcip

R2iAjzLvDNAReF97zoJqq880ZD1bwY82JDauCXELVR9O6\_B0w3K-E7yM2mac

AAgNCUwtik6SjoSUZRcf-O5lygIyLENx882p6MtmwaL1hd6qn5RZOQ0TLrOY

u0532g9Exxcm-ChymrB4xLykpDj3lUivJt63eEGGN6DH5K6o33TcxkIjNrCD

4XB1CKKumZvCedgHHF3IAK4dVEDSUoGlH9z4pP\_eWYNXvqQOjGs-rDaQzUHl

6cQQWNiDpWOl\_lxXjQEvQ

&state=af0ifjsldkj

The value of the id\_token parameter is the ID Token, which is a signed JWT, containing three base64url encoded segments separated by period ('.') characters. The first segment represents the JWT header. Base64url decoding it will result in the following set of Header Parameters:

{"kid":"1e9gdk7","alg":"RS256"}

The alg value represents the algorithm that was used to sign the JWT, in this case RS256, representing RSASSA-PKCS-v1\_5 using SHA-256. The kidvalue is a key identifier used in identifying the key to be used to verify the signature. If the kid value is unknown to the RP, it needs to retrieve the contents of the OP's JWK Set again to obtain the OP's current set of keys.

The second segment represents the Claims in the ID Token. Verifying and decoding the ID Token will yield the following Claims:

{

"iss": "http://server.example.com",

"sub": "248289761001",

"aud": "s6BhdRkqt3",

"nonce": "n-0S6\_WzA2Mj",

"exp": 1311281970,

"iat": 1311280970,

"name": "Jane Doe",

"given\_name": "Jane",

"family\_name": "Doe",

"gender": "female",

"birthdate": "0000-10-31",

"email": "janedoe@example.com",

"picture": "http://example.com/janedoe/me.jpg"

}

The third segment represents the ID Token signature, which is verified as described in [**[JWS]**](http://openid.net/specs/openid-connect-core-1_0.html#JWS).

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### A.3.  Example using response\_type=id\_token token

GET /authorize?

response\_type=id\_token%20token

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile%20email

&nonce=n-0S6\_WzA2Mj

&state=af0ifjsldkj HTTP/1.1

Host: server.example.com

HTTP/1.1 302 Found

Location: https://client.example.org/cb#

access\_token=jHkWEdUXMU1BwAsC4vtUsZwnNvTIxEl0z9K3vx5KF0Y

&token\_type=Bearer

&id\_token=eyJraWQiOiIxZTlnZGs3IiwiYWxnIjoiUlMyNTYifQ.ewogIml

zcyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ

4Mjg5NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiA

ibi0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDE

zMTEyODA5NzAsCiAiYXRfaGFzaCI6ICI3N1FtVVB0alBmeld0RjJBbnBLOVJ

RIgp9.F9gRev0Dt2tKcrBkHy72cmRqnLdzw9FLCCSebV7mWs7o\_sv2O5s6zM

ky2kmhHTVx9HmdvNnx9GaZ8XMYRFeYk8L5NZ7aYlA5W56nsG1iWOou\_-gji0

ibWIuuf4Owaho3YSoi7EvsTuLFz6tq-dLyz0dKABMDsiCmJ5wqkPUDTE3QTX

jzbUmOzUDli-gCh5QPuZAq0cNW3pf\_2n4zpvTYtbmj12cVcxGIMZby7TMWES

RjQ9\_o3jvhVNcCGcE0KAQXejhA1ocJhNEvQNqMFGlBb6\_0RxxKjDZ-Oa329e

GDidOvvp0h5hoES4a8IuGKS7NOcpp-aFwp0qVMDLI-Xnm-Pg

&state=af0ifjsldkj

Verifying and decoding the ID Token will yield the following Claims:

{

"iss": "http://server.example.com",

"sub": "248289761001",

"aud": "s6BhdRkqt3",

"nonce": "n-0S6\_WzA2Mj",

"exp": 1311281970,

"iat": 1311280970,

"at\_hash": "77QmUPtjPfzWtF2AnpK9RQ"

}

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### A.4.  Example using response\_type=code id\_token

GET /authorize?

response\_type=code%20id\_token

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile%20email

&nonce=n-0S6\_WzA2Mj

&state=af0ifjsldkj HTTP/1.1

Host: server.example.com

HTTP/1.1 302 Found

Location: https://client.example.org/cb#

code=Qcb0Orv1zh30vL1MPRsbm-diHiMwcLyZvn1arpZv-Jxf\_11jnpEX3Tgfvk

&id\_token=eyJraWQiOiIxZTlnZGs3IiwiYWxnIjoiUlMyNTYifQ.ewogIml

zcyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ

4Mjg5NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiA

ibi0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDE

zMTEyODA5NzAsCiAiY19oYXNoIjogIkxEa3RLZG9RYWszUGswY25YeENsdEE

iCn0.XW6uhdrkBgcGx6zVIrCiROpWURs-4goO1sKA4m9jhJIImiGg5muPUcN

egx6sSv43c5DSn37sxCRrDZZm4ZPBKKgtYASMcE20SDgvYJdJS0cyuFw7Ijp

\_7WnIjcrl6B5cmoM6ylCvsLMwkoQAxVublMwH10oAxjzD6NEFsu9nipkszWh

sPePf\_rM4eMpkmCbTzume-fzZIi5VjdWGGEmzTg32h3jiex-r5WTHbj-u5HL

7u\_KP3rmbdYNzlzd1xWRYTUs4E8nOTgzAUwvwXkIQhOh5TPcSMBYy6X3E7-\_

gr9Ue6n4ND7hTFhtjYs3cjNKIA08qm5cpVYFMFMG6PkhzLQ

&state=af0ifjsldkj

Verifying and decoding the ID Token will yield the following Claims:

{

"iss": "http://server.example.com",

"sub": "248289761001",

"aud": "s6BhdRkqt3",

"nonce": "n-0S6\_WzA2Mj",

"exp": 1311281970,

"iat": 1311280970,

"c\_hash": "LDktKdoQak3Pk0cnXxCltA"

}

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### A.5.  Example using response\_type=code token

GET /authorize?

response\_type=code%20token

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile%20email

&nonce=n-0S6\_WzA2Mj

&state=af0ifjsldkj HTTP/1.1

Host: server.example.com

HTTP/1.1 302 Found

Location: https://client.example.org/cb#

code=Qcb0Orv1zh30vL1MPRsbm-diHiMwcLyZvn1arpZv-Jxf\_11jnpEX3Tgfvk

&access\_token=jHkWEdUXMU1BwAsC4vtUsZwnNvTIxEl0z9K3vx5KF0Y

&token\_type=Bearer

&state=af0ifjsldkj

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### A.6.  Example using response\_type=code id\_token token

GET /authorize?

response\_type=code%20id\_token%20token

&client\_id=s6BhdRkqt3

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

&scope=openid%20profile%20email

&nonce=n-0S6\_WzA2Mj

&state=af0ifjsldkj HTTP/1.1

Host: server.example.com

HTTP/1.1 302 Found

Location: https://client.example.org/cb#

code=Qcb0Orv1zh30vL1MPRsbm-diHiMwcLyZvn1arpZv-Jxf\_11jnpEX3Tgfvk

&access\_token=jHkWEdUXMU1BwAsC4vtUsZwnNvTIxEl0z9K3vx5KF0Y

&token\_type=Bearer

&id\_token=eyJraWQiOiIxZTlnZGs3IiwiYWxnIjoiUlMyNTYifQ.ewogIml

zcyI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ

4Mjg5NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiA

ibi0wUzZfV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDE

zMTEyODA5NzAsCiAiY19oYXNoIjogIkxEa3RLZG9RYWszUGswY25YeENsdEE

iCn0.XW6uhdrkBgcGx6zVIrCiROpWURs-4goO1sKA4m9jhJIImiGg5muPUcN

egx6sSv43c5DSn37sxCRrDZZm4ZPBKKgtYASMcE20SDgvYJdJS0cyuFw7Ijp

\_7WnIjcrl6B5cmoM6ylCvsLMwkoQAxVublMwH10oAxjzD6NEFsu9nipkszWh

sPePf\_rM4eMpkmCbTzume-fzZIi5VjdWGGEmzTg32h3jiex-r5WTHbj-u5HL

7u\_KP3rmbdYNzlzd1xWRYTUs4E8nOTgzAUwvwXkIQhOh5TPcSMBYy6X3E7-\_

gr9Ue6n4ND7hTFhtjYs3cjNKIA08qm5cpVYFMFMG6PkhzLQ

&state=af0ifjsldkj

Verifying and decoding the ID Token will yield the following Claims:

{

"iss": "http://server.example.com",

"sub": "248289761001",

"aud": "s6BhdRkqt3",

"nonce": "n-0S6\_WzA2Mj",

"exp": 1311281970,

"iat": 1311280970,

"at\_hash": "77QmUPtjPfzWtF2AnpK9RQ",

"c\_hash": "LDktKdoQak3Pk0cnXxCltA"

}

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### A.7.  RSA Key Used in Examples

The following RSA public key, represented in JWK format, can be used to validate the ID Token signatures in the above examples (with line wraps within values for display purposes only):

{

"kty":"RSA",

"kid":"1e9gdk7",

"n":"w7Zdfmece8iaB0kiTY8pCtiBtzbptJmP28nSWwtdjRu0f2GFpajvWE4VhfJA

jEsOcwYzay7XGN0b-X84BfC8hmCTOj2b2eHT7NsZegFPKRUQzJ9wW8ipn\_aD

JWMGDuB1XyqT1E7DYqjUCEOD1b4FLpy\_xPn6oV\_TYOfQ9fZdbE5HGxJUzeku

GcOKqOQ8M7wfYHhHHLxGpQVgL0apWuP2gDDOdTtpuld4D2LK1MZK99s9gaSj

RHE8JDb1Z4IGhEcEyzkxswVdPndUWzfvWBBWXWxtSUvQGBRkuy1BHOa4sP6F

KjWEeeF7gm7UMs2Nm2QUgNZw6xvEDGaLk4KASdIxRQ",

"e":"AQAB"

}

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### Appendix B.  Acknowledgements

As a successor version of OpenID, this specification heavily relies on ideas explored in [**OpenID Authentication 2.0**](http://openid.net/specs/openid-connect-core-1_0.html#OpenID.2.0) [OpenID.2.0]. Please refer to Appendix C of OpenID Authentication 2.0 for the full list of the contributors for that specification.

In addition, the OpenID Community would like to thank the following people for their contributions to this specification:

Naveen Agarwal (naa@google.com), Google

Amanda Anganes (aanganes@mitre.org), MITRE

Casper Biering (cb@peercraft.com), Peercraft

John Bradley (ve7jtb@ve7jtb.com), Ping Identity

Tim Bray (tbray@textuality.com), Google

Johnny Bufu (jbufu@janrain.com), Janrain

Brian Campbell (bcampbell@pingidentity.com), Ping Identity

Blaine Cook (romeda@gmail.com), Independent

Breno de Medeiros (breno@google.com), Google

Pamela Dingle (pdingle@pingidentity.com), Ping Identity

Vladimir Dzhuvinov (vladimir@nimbusds.com), Nimbus Directory Services

George Fletcher (george.fletcher@corp.aol.com), AOL

Roland Hedberg (roland.hedberg@adm.umu.se), University of Umea

Ryo Ito (ryo.ito@mixi.co.jp), mixi, Inc.

Edmund Jay (ejay@mgi1.com), Illumila

Michael B. Jones (mbj@microsoft.com), Microsoft

Torsten Lodderstedt (t.lodderstedt@telekom.de), Deutsche Telekom

Nov Matake (nov@matake.jp), Independent

Chuck Mortimore (cmortimore@salesforce.com), Salesforce

Anthony Nadalin (tonynad@microsoft.com), Microsoft

Hideki Nara (hdknr@ic-tact.co.jp), Tact Communications

Axel Nennker (axel.nennker@telekom.de), Deutsche Telekom

David Recordon (dr@fb.com), Facebook

Justin Richer (jricher@mitre.org), MITRE

Nat Sakimura (n-sakimura@nri.co.jp), Nomura Research Institute, Ltd.

Luke Shepard (lshepard@fb.com), Facebook

Andreas ?kre Solberg (andreas.solberg@uninett.no), UNINET

Paul Tarjan (pt@fb.com), Facebook

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### Appendix C.  Notices

Copyright (c) 2014 The OpenID Foundation.

The OpenID Foundation (OIDF) grants to any Contributor, developer, implementer, or other interested party a non-exclusive, royalty free, worldwide copyright license to reproduce, prepare derivative works from, distribute, perform and display, this Implementers Draft or Final Specification solely for the purposes of (i) developing specifications, and (ii) implementing Implementers Drafts and Final Specifications based on such documents, provided that attribution be made to the OIDF as the source of the material, but that such attribution does not indicate an endorsement by the OIDF.

The technology described in this specification was made available from contributions from various sources, including members of the OpenID Foundation and others. Although the OpenID Foundation has taken steps to help ensure that the technology is available for distribution, it takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this specification or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any independent effort to identify any such rights. The OpenID Foundation and the contributors to this specification make no (and hereby expressly disclaim any) warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to this specification, and the entire risk as to implementing this specification is assumed by the implementer. The OpenID Intellectual Property Rights policy requires contributors to offer a patent promise not to assert certain patent claims against other contributors and against implementers. The OpenID Foundation invites any interested party to bring to its attention any copyrights, patents, patent applications, or other proprietary rights that may cover technology that may be required to practice this specification.

|  |
| --- |
| [**TOC**](http://openid.net/specs/openid-connect-core-1_0.html#toc) |

### Authors' Addresses

|  |  |
| --- | --- |
|  | Nat Sakimura |
|  | Nomura Research Institute, Ltd. |
| **Email:** | [**n-sakimura@nri.co.jp**](mailto:n-sakimura@nri.co.jp) |
| **URI:** | [**http://nat.sakimura.org/**](http://nat.sakimura.org/) |
|  |  |
|  | John Bradley |
|  | Ping Identity |
| **Email:** | [**ve7jtb@ve7jtb.com**](mailto:ve7jtb@ve7jtb.com) |
| **URI:** | [**http://www.thread-safe.com/**](http://www.thread-safe.com/) |
|  |  |
|  | Michael B. Jones |
|  | Microsoft |
| **Email:** | [**mbj@microsoft.com**](mailto:mbj@microsoft.com) |
| **URI:** | [**http://self-issued.info/**](http://self-issued.info/) |
|  |  |
|  | Breno de Medeiros |
|  | Google |
| **Email:** | [**breno@google.com**](mailto:breno@google.com) |
| **URI:** | [**http://stackoverflow.com/users/311376/breno**](http://stackoverflow.com/users/311376/breno) |
|  |  |
|  | Chuck Mortimore |
|  | Salesforce |
| **Email:** | [**cmortimore@salesforce.com**](mailto:cmortimore@salesforce.com) |
| **URI:** | [**https://twitter.com/cmort**](https://twitter.com/cmort) |