5. 颁发访问令牌

5. 颁发访问令牌

5.颁发访问令牌

如果访问令牌请求是有效的且被授权，授权服务器如5.1节所述颁发访问令牌以及可选的刷新令牌。如果请求因客户端身份验证失败或无效，授权服务器如5.2节所述的返回错误响应。

5. Issuing an Access Token

```
   If the access token request is valid and authorized, the   authorization server issues an
access token and optional refresh   token as described in Section 5.1.  If the request failed
client
   authentication or is invalid, the authorization server returns an
   error response as described in Section 5.2.
```

5.1. 成功的响应(5.1. Successful Response)

5.1. 成功的响应

授权服务器颁发访问令牌和可选的刷新令牌，通过向HTTP响应实体正文中添加下列参数并使用200（OK）状态码构造响应：

- access_token

  必需的。权服器颁发的访问令牌。
- token_type

  需。7.1节如述的颁发的令牌的类型。值是大小写不敏感的。
- expires_in

  推荐的。以秒为单位的访问令牌生命周期。例如，值“3600”表示访问令牌将在从生成响应时的1小时后到期。如果省略，则授权服务器应该通过其他方式提供过期时间，或者记录默认值。
- refresh_token

  选的。6 [illegible]新的访问令牌。
- scope

  选，3.3节[illegible]相同；否则，必需的。如

参数 RFC4627定义的“application/json”媒体类型包含在HTTP响应实体正文中。通过将每个参数添加到最高结构级别，参数被序列化为JavaScript对象表示法（JSON）的结构。参数

在任何包含令牌、凭据或其他敏感信息的响应中，授权服务器必须在其中 [illegible] RFC2616 “no-store” [illegible] HTTP “Cache-Control” [illegible] “Pragma” [illegible] RFC2616 [illegible] 例如：

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

```
Pragma: no-cache
{
  "access_token":"2YotnFZFEjr1zCsicMWpAA",
  "token_type":"example",
  "expires_in":3600,
  "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
  "example_parameter":"example_value"}
```

5.1. Successful Response

```
   The authorization server issues an access token and optional refresh   token, and constructs
the response by adding the following parameters   to the entity-body of the HTTP response with a
200 (OK) status code:
   access_token
         REQUIRED.  The access token issued by the authorization server.
   token_type
         REQUIRED.  The type of the token issued as described in         Section 7.1.  Value is
case insensitive.
   expires_in
         RECOMMENDED.  The lifetime in seconds of the access token.  For         example, the
value "3600" denotes that the access token will         expire in one hour from the time the
response was generated.
         If omitted, the authorization server SHOULD provide the         expiration time via
other means or document the default value.
Hardt                        Standards Track                   [Page 43]
```

```
RFC 6749                        OAuth 2.0                   October 2012
   refresh_token
         OPTIONAL.  The refresh token, which can be used to obtain new         access tokens
using the same authorization grant as described
         in Section 6.
```

scope

OPTIONAL, if identical to the scope requested by the client; otherwise, REQUIRED. The scope of the access token as

described by Section 3.3.

The parameters are included in the entity-body of the HTTP response using the "application/json" media type as defined by [RFC4627]. The parameters are serialized into a JavaScript Object Notation (JSON) structure by adding each parameter at the highest structure level.

Parameter names and string values are included as JSON strings. Numerical values are included as JSON numbers. The order of

parameters does not matter and can vary.

The authorization server MUST include the HTTP "Cache-Control" response header field [RFC2616] with a value of "no-store" in any response containing tokens, credentials, or other sensitive

information, as well as the "Pragma" response header field [RFC2616] with a value of "no-cache".

For example:

```
  HTTP/1.1 200 OK
  Content-Type: application/json;charset=UTF-8
  Cache-Control: no-store
  Pragma: no-cache
  {
    "access_token":"2YotnFZFEjr1zCsicMWpAA",
    "token_type":"example",
    "expires_in":3600,
    "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
    "example_parameter":"example_value"
  }
```

The client MUST ignore unrecognized value names in the response. The sizes of tokens and other values received from the authorization server are left undefined. The client should avoid making

assumptions about value sizes. The authorization server SHOULD

document the size of any value it issues.

5.2. 错误响应(5.2. Error Response)

5.2. 错误响应

授权服务器使用HTTP 400（错误请求）状态码响应，在响应中包含下列参数：

- error 必需的。下列ASCII[USASCII]错误代码之一：
    - invalid_request
    - 请求缺少□□的参数、包含□支持的参数值（除了许可类型）、重复参数、包含多个凭据、采用超过一种客户端身份验证机制或其他不规范的格式。
    - invalid_client

      客户端身份验证失败（例如，未知的客户端，不包含客户端身份验证，或不支持的身份验证方法）。授权服务器可以返回HTTP 401（未授权）状态码来指出支持的HTTP身份验证方案。如果客户端
    - invalid_grant

      提供的授权许可（如授权码、资源所有者凭据）或刷新令牌无效、过期、吊销、与在授权请求使用的重定向URI不匹配或颁发给另一个客户端。
    - unauthorized_client
    - 进行身份验证的客户□没有□授权使用这□授□许可类型。
    - unsupported_grant_type
    - 授□许可□□不被授权服务器支持。
    - invalid_scope

  “error”参数的值不能包含集合%x20-21 /%x23-5B /%x5D-7E以外的字符。
- error_description

  可选的。提供额外信息的人类可读的ASCII[USASCII]文本，用于协助客户端开发人员理解所发生的错误。“error_description”参数的值不能包含集合%x20-21 /%x23-5B
- error_uri

  可选的。指向带有有关错误的信息的人类可读网页的URI，用于提供客户端开发人员关于该错误的额外信息。“error_uri”参数值必须符合URI参考语法，因此不能包含集合%x21/%x23-5B /%

□□参数□□

RFC4627定义的“application/json”媒体类型包含在HTTP响应实体正文中。通过将每个参数添加到最高结构级别，参数被序列化为JavaScript对象表示法（JSON）的结构。参数

```
HTTP/1.1 400 Bad Request

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

Cache-Control: no-store
```

```
Pragma: no-cache
{
  "error":"invalid_request"}
```

5.2. Error Response

```
   The authorization server responds with an HTTP 400 (Bad Request)   status code (unless
specified otherwise) and includes the following
   parameters with the response:
   error
         REQUIRED.  A single ASCII [USASCII] error code from the
         following:
         invalid_request              The request is missing a required parameter, includes
an
               unsupported parameter value (other than grant type),              repeats a
parameter, includes multiple credentials,               utilizes more than one mechanism for
authenticating the
               client, or is otherwise malformed.
         invalid_client               Client authentication failed (e.g., unknown client,
no
               client authentication included, or unsupported              authentication
method).  The authorization server MAY               return an HTTP 401 (Unauthorized) status
code to indicate               which HTTP authentication schemes are supported.  If
the
               client attempted to authenticate via the "Authorization"               request
header field, the authorization server MUST               respond with an HTTP 401
(Unauthorized) status code and               include the "WWW-Authenticate" response header
field
               matching the authentication scheme used by the client.
         invalid_grant               The provided authorization grant (e.g.,
authorization
               code, resource owner credentials) or refresh token is               invalid,
expired, revoked, does not match the redirection               URI used in the authorization
```

request, or was issued to
another client.
unauthorized_client The authenticated client is not authorized to use this
authorization grant type.
unsupported_grant_type The authorization grant type is not supported by the
authorization server.

invalid_scope The requested scope is invalid, unknown, malformed, or
exceeds the scope granted by the resource owner.
Values for the "error" parameter MUST NOT include characters outside the set %x20-21 / %x23-5B / %x5D-7E.

error_description

OPTIONAL. Human-readable ASCII [USASCII] text providing additional information, used to assist the client developer in understanding the error that occurred.

Values for the "error_description" parameter MUST NOT include characters outside the set %x20-21 / %x23-5B / %x5D-7E.

error_uri

OPTIONAL. A URI identifying a human-readable web page with information about the error, used to provide the client developer with additional information about the error.

Values for the "error_uri" parameter MUST conform to the URI-reference syntax and thus MUST NOT include characters
outside the set %x21 / %x23-5B / %x5D-7E.

The parameters are included in the entity-body of the HTTP response using the "application/json" media type as defined by [RFC4627]. The parameters are serialized into a JSON structure by adding each parameter at the highest structure level. Parameter names and string
values are included as JSON strings. Numerical values are included as JSON numbers. The order of parameters does not matter and can

```
vary.
For example:
  HTTP/1.1 400 Bad Request
  Content-Type: application/json;charset=UTF-8
  Cache-Control: no-store
  Pragma: no-cache
  {
    "error":"invalid_request"
  }
```