# 5.1. [illegible](5.1. Successful Response)

## 5.1. [illegible]

[illegible]HTTP[illegible]200[illegible]OK[illegible]

- access_token
  [illegible]
- token_type
  [illegible] 7.1 [illegible]
- expires_in
  [illegible]“3600”[illegible]1[illegible]
- refresh_token
  [illegible] 6 [illegible]
- scope
  [illegible] 3.3 [illegible]

[illegible] RFC4627[illegible]“application/json”[illegible]HTTP[illegible] [illegible]JavaScript[illegible]JSON[illegible]

[illegible]RFC2616[illegible]“no-store”[illegible]HTTP“Cache-Control”[illegible]“Pragma”[illegible]“no-cache”[illegible]RFC2616[illegible]

```
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"}
```

[illegible]

# 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.

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.
```

Revision #1
Created Wed, Mar 25, 2020 11:04 PM by □□□
Updated Wed, Mar 25, 2020 11:05 PM by □□□