5. Issuing an Access Token

Eğer erişim belirteci isteği geçerli ve yetkilendirilmişse, yetkilendirme sunucusu **Bölüm 5.1'de** açıklandığı şekilde bir erişim belirteci ve isteğe bağlı olarak bir yenileme belirteci oluşturur. Eğer istek istemci kimlik doğrulamasında başarısız olursa veya geçersizse, yetkilendirme sunucusu **Bölüm 5.2'de** açıklandığı şekilde bir hata yanıtı döner.

`5.1. Successful Response`

Bu bölüm, başarılı bir erişim belirteci yanıtının nasıl oluşturulduğunu ve neleri içerdiğini detaylandırır. İşte açıklaması:

Erişim Belirteci ve Yenileme Belirteci

Yetkilendirme sunucusu, başarılı bir istek sonrasında şu bilgileri içeren bir HTTP yanıtı oluşturur:

1. **access_token (Gerekli):**
    - Yetkilendirme sunucusu tarafından verilen erişim belirtecidir. Bu belirteç, API gibi korunan kaynaklara erişimde kullanılır.
2. **token_type (Gerekli):**
    - Verilen belirtecin türüdür (örneğin, "Bearer"). Bu, **Bölüm 7.1'de** açıklanmıştır ve büyük/küçük harfe duyarsızdır.
3. **expires_in (Tavsiye Edilir):**
    - Erişim belirtecinin geçerlilik süresini (saniye olarak) belirtir. Örneğin, `3600` değeri, erişim belirtecinin bir saat geçerli olacağını gösterir. Bu parametre sağlanmazsa, yetkilendirme sunucusu başka yollarla son kullanma bilgisini sağlamalıdır veya varsayılan bir süreyi dokümante etmelidir.
4. **refresh_token (Opsiyonel):**
    - Yenileme belirteci, mevcut erişim belirtecinin süresi dolduğunda yeni bir belirteç almak için kullanılır. Bu işlem, **Bölüm 6'da** açıklanmıştır.
5. **scope (Opsiyonel):**
    - Eğer istemci tarafından talep edilen kapsamla aynıysa bu parametreye gerek yoktur. Farklıysa bu parametre sağlanmalıdır. Bu, erişim belirtecinin hangi izinlere sahip olduğunu belirtir.

JSON Formatı

Yetkilendirme sunucusu, bu parametreleri **JSON** formatında HTTP yanıt gövdesine dahil eder.

- Parametreler, JSON yapısında en üst seviyede yer alır.
- İsimler ve metin değerleri, JSON dizesi (string) olarak kodlanır.
- Sayısal değerler, JSON sayıları olarak yer alır.
- Parametrelerin sıralaması önemli değildir ve değişebilir.

Örnek Yanıt

Aşağıdaki yanıt, başarılı bir erişim belirteci isteğine örnek gösterir:

```
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"
}
```

Güvenlik Önlemleri

Yetkilendirme sunucusu, güvenlik için şu başlıkları her yanıtında içermelidir:

1. **Cache-Control:**
    - `no-store` değeri ile yanıtın tarayıcı veya ara önbellekler tarafından depolanmasını engeller.
2. **Pragma:**
    - `no-cache` değeri ile önceki önbelleğe alınmış verilerin kullanılmasını engeller.

İstemciye Yönelik Notlar

- İstemci, yanıtta tanımadığı parametre adlarını görmezden gelmelidir.
- Erişim belirteci veya diğer parametrelerin boyutu tanımlanmamıştır; bu nedenle istemci, belirli bir boyut beklentisiyle hareket etmemelidir.
- Yetkilendirme sunucusu, verdiği parametrelerin boyutunu dokümante etmelidir.

Bu süreç, güvenliğin sağlanması ve istemcilerin yetkilendirme sunucusundan aldıkları belirteçleri doğru şekilde işlemesi için önemlidir.

5.2. Error Response

Bu bölüm, erişim belirteci isteği sırasında oluşabilecek hatalar ve bu hatalar karşısında yetkilendirme sunucusunun nasıl bir yanıt oluşturması gerektiğini açıklar.

Hata Yanıtının Temelleri

- Yetkilendirme sunucusu, bir hata durumunda genellikle **HTTP 400 (Bad Request)** durum kodu döner (aksi belirtilmedikçe).
- Yanıt, aşağıdaki parametreleri içerir ve JSON formatında iletilir.

Hata Parametreleri

1. **error (Gerekli):**
    - Hatanın türünü belirten bir ASCII hata kodu içerir. Bu kodlar aşağıdaki gibidir:
        - **invalid_request:**
            - İstek aşağıdaki nedenlerden dolayı geçersizdir:
                - Gerekli bir parametre eksik.
                - Desteklenmeyen bir parametre değeri kullanılmış.
                - Parametre tekrar edilmiş.
                - Birden fazla kimlik bilgisi veya kimlik doğrulama mekanizması kullanılmış.
                - İstek başka bir şekilde hatalı oluşturulmuş.
        - **invalid_client:**
            - İstemcinin kimlik doğrulaması başarısız olmuştur. Örneğin:
                - İstemci tanınmıyor.
                - İstemci kimlik doğrulama bilgileri sağlanmamış.
                - Desteklenmeyen bir kimlik doğrulama yöntemi kullanılmış.
            - Yetkilendirme sunucusu, **HTTP 401 (Unauthorized)** durum kodu dönebilir ve desteklenen kimlik doğrulama şemalarını belirtmek için `WWW-Authenticate` başlığını içermelidir.
        - **invalid_grant:**
            - Sağlanan yetkilendirme yetkisi (örneğin, yetkilendirme kodu, kaynak sahibi kimlik bilgileri) veya yenileme belirteci:
                - Geçersiz, süresi dolmuş, iptal edilmiş.
                - Yetkilendirme isteğinde kullanılan yönlendirme URI'siyle eşleşmiyor.
                - Başka bir istemci için verilmiş.
        - **unauthorized_client:**
            - Kimliği doğrulanmış istemci, bu yetkilendirme yetkisini kullanmaya yetkili değil.
        - **unsupported_grant_type:**
            - Kullanılan yetkilendirme türü, yetkilendirme sunucusu tarafından desteklenmiyor.
        - **invalid_scope:**
            - Talep edilen kapsam:
                - Geçersiz, bilinmiyor, hatalı biçimlendirilmiş.
                - Kaynak sahibinin verdiği kapsamı aşıyor.
2. **error_description (Opsiyonel):**
    - İnsan tarafından okunabilir bir açıklama sağlar.
    - Hatanın neden meydana geldiğini anlaması için istemci geliştiricisine yardımcı olur.

3. **error_uri (Opsiyonel):**
    - Hata hakkında daha fazla bilgi içeren, okunabilir bir web sayfasını tanımlayan URI'yi içerir.

JSON Formatı

- Tüm parametreler, JSON formatında yanıt gövdesine eklenir.
- Parametre isimleri ve değerleri JSON dizesi olarak ifade edilir.
- Parametrelerin sıralaması önemli değildir.
- Örnek hata yanıtı:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "error": "invalid_request"
}
```

Güvenlik Notları

- Yetkilendirme sunucusu, yanıtlarında aşağıdaki başlıkları içermelidir:
    - **Cache-Control:** `no-store` ile duyarlı bilgilerin önbelleğe alınmasını engeller.
    - **Pragma:** `no-cache` ile önbellekten yanlışlıkla bilgi alınmasını önler.

İstemciye Öneriler

- İstemci, tanımadığı `error` parametresi değerlerini yok saymalıdır.
- Hata mesajları, istemci geliştiricisinin hatayı anlamasına ve çözmesine yardımcı olacak şekilde tasarlanmalıdır.