Documentazione tecnica

Accedere alle API protette di servizi web utilizzando access tokens

Istituto Zooprofilattico Sperimentale dell'Abruzzo e del Molise (IZSAM)

Versione: 1.1	*Data Emissione: 23 Agosto 2018*
Redatto da: Andrea Bucciacchio	*Emesso da : Marco Secone*

Indice

Indice delle immagini

1 Accedere alle API protette di servizi web utilizzando access tokens

1.1 Concetti generali

L'infrastruttura dei servizi del Sistema Informativo Nazionale della Farmacosorveglianza, relativamente alla parte di autenticazione e autorizzazione, è stata realizzata basandosi sul framework OAuth2, definito nell'rfc6749 (https://tools.ietf.org/html/rfc6749).
OAuth2 definisce quattro ruoli che partecipano al flusso di autenticazione e autorizzazione:

- User (Resource Owner): l'end-user che desidera invocare un servizio web;
- Application (Client): l'applicazione che effettua l'invocazione del servizio web per conto dell'end-user;
- Resource Server: i server che implementano i servizi web;
- Authorization Server: il server che si occupa dell'autenticazione e dell'autorizzazione.

Questa sezione descrive come ottenere un token per utilizzare ai servizi web utilizzando il flusso OAuth2 "resource owner password credentials" (https://tools.ietf.org/html/rfc6749#section-4.3).

Tale flusso di autenticazione ed autorizzazione è descritto dalla seguente immagine:

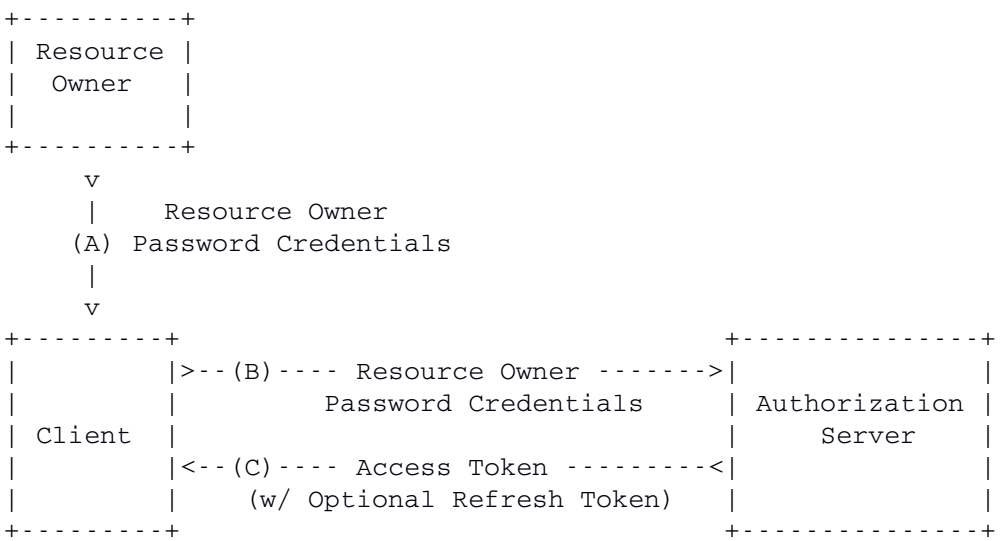

Figura 1: flusso di autenticazione ed autorizzazione

Il flusso descritto dall'immagine è il seguente:

1. L'utente fornisce al client le proprie credenziali (username, password);
2. Il cliente richiede un token per accedere ai servizi, fornendo le proprie credenziali (client_id, client_secret) e quelle dell'utente (username, password);
3. Se le credenziali inviate risultano corrette un token da utilizzare per accedere ai servizi web è restituito.

2 Come richiedere un access token

Per richiedere un token, come indicato nell'rfc6749, è necessario specificare i seguenti parametri:

- **client_id**, **client_secret**: sono le credenziali dell'applicativo tramite il quale si vuole accedere alle risorse; sono codificate nell'header http *Authorization* (vedi paragrafo dedicato). Nei seguenti esempi: '123456', 'abcdfeg'. Encoded in base64: MTIzNDU2OmFiY2RmZWc=
- **username**, **password**: sono le credenziali dell'utente. Nei seguenti esempi: 'user1234', 'password1234'
- **grant_type**: indica il grant type OAuth2; in questo caso deve essere sempre valorizzato con "password"

E' inoltre necessario indicare i seguenti parametri:

- **scope**: identifica un sottoinsieme delle risorse o delle funzionalità alle quali il token richiesto permetterà l'accesso; l'insieme dei possibili valori deve essere indicato dal service provider. Nei seguenti esempi è utilizzato lo scope 'FAR'.

2.1 Esempio di richiesta token

2.1.1 Server di autenticazione e credenziali client

Per ottenere gli endpoint di autenticazione relativi agli ambienti di test e di produzione e per far richiesta delle credenziali client/applicativo è possibile far riferimento alla relativa sezione del manuale utente (https://wstest.izs.it/help/farmaco/help/integrazione#OAuth2).
Nei seguenti esempi è utilizzato, come server di autenticazione, l'host *authtest.izs.it.*

2.1.2 Esempio richiesta (HTTP)

```
POST /j_test_auth/oauth/token HTTP/1.1
Host: authtest.izs.it
Authorization: Basic MTIzNDU2OmFiY2RmZWc=
Content-Type: application/x-www-form-urlencoded

grant_type=password&username=user1234&password=password1234&scope=FAR
```

2.1.3 Esempio richiesta (cURL)

```
curl -X POST \
  https://authtest.izs.it/j_test_auth/oauth/token \
  -H 'Authorization: Basic MTIzNDU2OmFiY2RmZWc=' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
   -d 'grant_type=password&username=user1234&password=password1234&scope=FAR'
```

2.1.4 Risposta positiva – access token rilasciato

In caso di esito positivo, il server di autenticazione rilascia un access token e (a seconda della configurazione del client) un refresh token.

Il codice della risposta HTTP avrà valore 200 ed nel body saranno specificati:

- access_token: il token utilizzabile per accedere ai servizi web
- token_type: sempre valorizzato come "bearer", indica la tipologia di token
- expires_in: validità del token, expressa in secondi.

- scope: scope del token – indica l'applicativo cui il token è associato; dipende dal client per cui il token è stato richiesto.

Esempio:


```
{
 "access_token": "1d00cb6e-d4aa-42ce-b609-79280082a904",
 "token_type": "bearer",
 "expires_in": 600,
 "scope": "FAR"
}
```


2.1.5 Risposta negativa – access token non rilasciato

In caso di esito negativo, il codice http della richiesta avrà valore maggiore o uguale a 400 e nel body sarà contenuta una descrizione dell'errore.

Esempio:


```
{
  "error": "unauthorized",
  "error_description": "wrong credentials"
}
```


2.2 Encoding della credenziali del client

Le credenziali del client (client_id, client_secret) devono essere inviate utilizzando la modalità di autenticazione “Basic” di HTTP, ovvero utilizzando un header Authorization costruito in questo modo:

1. client_id e client_secret sono concatenati utilizzando il carattere “:” come separatore (“client_id:client_secret”)
2. Il risultato è codificato con base64
3. Il metodo di autorizzazione (Basic) e uno spazio sono inseriti all'inizio della stringa codificata.

Esempio:
client_id = “pippo”, client_secret = “aaardm” → “pippo:aaardm” → cGlwcG86YWFhcmRt → “Authorization: Basic cGlwcG86YWFhcmRt”

3 Come utilizzare un access token

E’ possibile utilizzare l'access token appena ottenuto (nei seguenti esempi *1d00cb6e-d4aa-42ce-b609-79280082a904*) per invocare i servizi web desiderati aggiungendo nelle richieste il seguente header di autenticazione:

```
Authorization: Bearer  1d00cb6e-d4aa-42ce-b609-79280082a904
```

così come descritto nell’rfc6750, paragrafo 2.1.

Ogni token ha una validità limitata (indicata dal campo “*expires_in*” al momento della generazione) e durante il periodo di validità è possibile riutilizzarlo un numero illimitato di volte.

3.1 Esempio invocazione web service (HTTP)

```
POST /demo_farmaco_test/lov/ricettafornitura/ws/search/byNumeroAndPin/ HTTP/1.1
Host: wstest.izs.it
Content-Type: application/x-www-form-urlencoded
Accept: application/json
Authorization: Bearer 1d00cb6e-d4aa-42ce-b609-79280082a904

numero=123456789&pin=1234&saiId=123456
```

3.2 Esempio invocazione web service (cURL)

```
curl -X POST \

https://wstest.izs.it/demo_farmaco_test/lov/ricettafornitura/ws/search/byN
umeroAndPin/ \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer 1d00cb6e-d4aa-42ce-b609-79280082a904' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -d 'numero=123456789&pin=1234&saiId=123456'
```

Per la documentazione dei web services è possibile far riferimento alla relativa sezione del manuale utente (https://wstest.izs.it/help/farmaco/help/api_servizi).

3.3 Esempio risposta negativa (access token non valido oppure scaduto)

Nel caso in cui il token sia non valido, il codice http della risposta avrà valore 401 e nel body sarà contenuta una descrizione dell'errore.

```
{
    "error": "invalid_token",
    "error_description": "123456780-d4aa-42ce-b609-1234567897"
}
```