API Manager APIMBBONE

Manuale per Sottoscrizione API

VERSIONE	PARAGRAFO O PAGINA	DESCRIZIONE DELLA VARIAZIONE
V01	Tutto il documento	Versione iniziale del documento

1 Sommario

1 API Management

Le soluzioni di API Management costituiscono l'elemento abilitante per arricchire e personalizzare l'interazione tra le applicazioni che richiedono l'accesso ad API e i servizi di business che espongono le informazioni utili per la composizione delle applicazioni stesse. L'introduzione di un layer di astrazione tra i servizi che espongono funzionalità (API Providers) e le applicazioni che li consumano (API Consumers) semplificano gli sviluppi e favoriscono il disaccoppiamento tra i due livelli.

L'API Manager è la soluzione che permette di centralizzare il punto di ingresso per le chiamate, applicare politiche di throttling efficienti, monitorare le risorse utilizzate, tracciare le chiamate dei fruitori delle API di business ed ultimo ma non meno importante securizzare i servizi API.

Gli obiettivi principali dell'API Manager possono essere riassunti come segue:

- Gestione della sicurezza: garantisce l'accesso soltanto agli utenti/sistemi autorizzati ed evita l'uso improprio delle risorse protette. Sono disponibili diversi framework di sicurezza, lo scenario attuale implementato prevedere l'adozione di OAuth2
- Gestione del traffico: performance gestite in maniera puntuale e dinamica per ciascuna API con limitazione del traffico in ingresso (rate limiting), applicazione di politiche di accesso diversificate in base sistema chiamante (throttling), routing e cache dei messaggi. Filtraggio del traffico in ottica di identificazione e neutralizzazione di minacce
- Gestione del ciclo di vita delle API: fasi di sviluppo, test, produzione e dismissione, nonché versionamento. Questa funzionalità garantisce la coerenza di differenti versioni dell'API consentendo il suo utilizzo da parte di diverse tipologie di utenti, in ambienti diversi e con diversi gradi di maturità
- Audit del sistema: tracciamento attività e statistiche di utilizzo.

Le componenti principali dell'API Manager sono le seguenti:

- **API Store,** Store dedicato agli sviluppatori e alle terze parti che intendono integrare le API nelle loro applicazioni, il portale ospita la documentazione di supporto, monitorare l'utilizzo delle API nonché accedere ad altri strumenti di comunicazione e condivisione.
- **Publisher,** applicazione web che permette agli API Providers di creare e pubblicare le API. Nel Publisher si definiscono tutti gli aspetti di configurazione dell'API, compresi endpoint dei servizi di back-end e politiche di throttling e rate limiting.
- **API Gateway,** componente di runtime dell'API Manager che ha come finalità essenziale di esporre i servizi messi a disposizione dall'intero sistema in maniera sicura, facilmente fruibile e controllata. L'API Gateway, dal punto di vista architetturale, è un proxy dei servizi esposti dai sistemi di back-end, in modo tale che tutti i sistemi fruitori debbano effettuare l'accesso a servizi e risorse attraverso questo componente. Dal punto di vista funzionale, il Gateway, riceve le richieste per accedere alle API ed attua le politiche di controllo di accessi, applica le regole di rate limiting e throttling e instrada le richieste verso i sistemi di back-end.
- **Key Manager,** componente che ha il compito di gestire tutte le questioni relative alla sicurezza e alle chiavi. Tutte le richieste di generazione di nuovi access token sono gestite da questo componente che effettua la validazione di tutti i parametri inviati nella richiesta (client_id, client_secret, username, password, ecc...).

- **Traffic Manager**, componente che si occupa di regolare il traffico di ciascuna API secondo le politiche che sono state definite in fase di definizione dell'API e in fase di sottoscrizione alla stessa. Il motore di elaborazione del Traffic Manager elabora le politiche di throttling in real time, incluse le politiche di rate limiting delle chiamate alle API.

2 Gestione della sicurezza

Open Authorization 2 (OAuth2) è lo standard de facto supportato dall'API Manager per l'autorizzazione a risorse (API) protette da autenticazione senza necessità di dover condividere pubblicamente tali credenziali. Il protocollo OAuth permette di autorizzare una terza parte a gestire sezioni riservate di una risorsa senza vincolare il sistema a mettere in circolazione le credenziali per accedervi. I vantaggi offerti da questa tecnologia derivano direttamente dalla possibilità di avere un accesso in autenticazione mediante generazione di un token di autorizzazione apposito.

Per la generazione degli access token si possono utilizzare diversi grant types OAuth 2.0 in base ai casi d'uso applicativi, di seguito i grant type supportati:

- client credentials
- authorization code
- refresh token
- resource owner password

Per le specifiche dei grant type si rimanda al sito ufficiale della OAuth 2.0 industry-standard protocol for authorization https://oauth.net/2/grant-types/.

3 Fruizione di API esposte su API Manager

Le API pubblicate sull' API Manager sono disponibili sullo Store al seguente indirizzo:

https://api-piemonte-store.csi.it/

Per poter accedere alle API è necessario che il fruitore completi i seguenti passi operativi:

- Accreditamento allo Store
- Creazione dell'applicazione contenitore delle API
- Sottoscrizione dell'API di interesse
- Utilizzo delle chiavi OAuth2 per accedere all'API

3.1 Accreditamento allo Store

Il primo passo, per chi vuole utilizzare le API, è l'accreditamento allo Store delle API.

Lo Store supporta le seguenti tipologie di utenti:

- Aziende fornitori di un ente
- Aziende accreditate presso un ente
- CSI-Piemonte e i suoi fornitori
- Funzionari di un ente

Le aziende con sede in Italia è necessario che dispongano di credenziali SPID per l'accesso allo Store API, nel caso l'azienda sia straniera e non possa disporre di credenziali SPID, l'accesso sarà garantito attraverso credenziali locali allo Store generati dall'amministratore della piattaforma.

CSI-Piemonte e i suoi fornitori, se dispongono di un accesso VPN, possono accreditarsi mediante Identity Provider "Unified Communication".

I funzionari degli enti pubblici possono accedere attraverso Identity Provider Rupar Sistema Piemonte.

Ogni credenziale di accesso allo Store sarà verificata e validata dall'amministratore della piattaforma che contatterà il richiedente per la corretta attribuzione del ruolo all'interno dello Store.

L'utente, prima di accreditarsi all'API Store, deve segnalare al referente delle API che intende sottoscrivere i seguenti dati:

- Nome Cognome dell'utente che si accrediterà al portale API Store
- Codice Fiscale dell'utente
- Indirizzo mail
- Nome Azienda (nel caso di funzionario PA indicare l'ente di appartenenza)
- Nome del gruppo di lavoro (opzionale): permette la condivisione delle informazioni contenute nell'applicazione che fruirà delle API, quali ad esempio le chiavi di per generare il token di access. Il nome del gruppo di lavoro dovrà, salvo casistiche da gestire puntualmente, essere composto dal nome dell'applicazione concatenando il suffisso **_team.**

Ricevuto il riscontro positivo da parte del referente API, sarà possibile accedere all'API Store utilizzando l'Identity Provider di competenza. Al primo accreditamento da parte dell'utente, l'amministratore dell'API Store confermerà alla casella di posta indicata l'autorizzazione all'accesso al portale.

3.2 Creazione dell'applicazione fruitrice delle API

La sottoscrizione di API richiede che l'utente, dopo aver effettuato il login sullo Store, effettui il censimento di una applicazione virtuale che costituisce il contenitore di sottoscrizioni ad una o più API che fanno capo ad un progetto di fruizione.

Gli attributi che sono richiesti in fase di creazione dell'applicazione sono:

- **Nome dell'applicazione**: nome parlante che riconduca il più possibile al nome dell'applicazione client che utilizza le API. Il nome dell'applicazione può avere al massimo 70 caratteri e non deve contenere caratteri speciali ne spazi e deve essere in lowercase.
  Nello scenario di applicazioni realizzate dal CSI Piemonte il nome dell'applicazione dovrà essere valorizzato con i dati censiti su Anagrafica Prodotti ovvero dal codice prodotto, codice componente e linea cliente. Il formato sarà quindi:
  {codice prodotto}_{codice_componente}_{linea cliente}
  Nello scenario di applicazioni realizzate fuori del contesto CSI Piemonte, il nome dell'applicazione dovrà avere il suffisso **-ext.**
- **Gruppi**: elenco dei gruppi di lavoro con cui condividere le informazioni dell'applicazione nello store. Il campo è preimpostato con i gruppi di lavoro di appartenenza dell'utente definiti in fase di accreditamento. Gli utenti che fanno parte del gruppo di lavoro possono visualizzare le chiavi di production e sandbox, generare i token e sottoscrivere API.

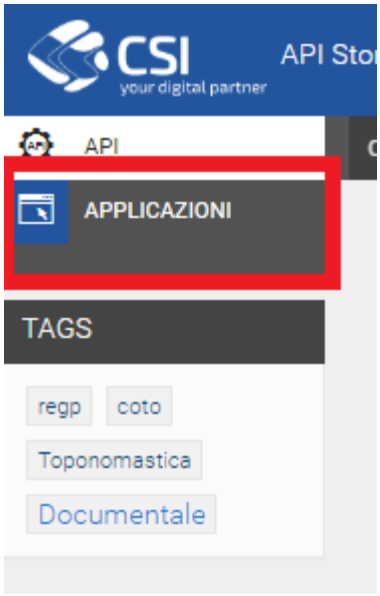
- **Per Token Quota**: limite massimo di richieste alle API per ciascun token generato. Il valore è preimpostato ad Unlimited e non è modificabile
- **Descrizione**: descrizione significativa dell'applicazione che ne permetta il riconoscimento e lo scopo. Nella descrizione occorre riportare l'Ente di riferimento dell'applicazione e l'azienda/ente di appartenenza dell'utente che sta creando l'applicazione.
- **Tipo di Token:** la tipologia di token è OAuth2, il valore è preimpostato e non modificabile

Al termine della creazione l'applicazione è in stato INACTIVE, è necessario attendere la validazione da parte dell'amministratore della piattaforma per poter procedere con la fase successiva. Il fruitore riceverà una mail non appena l'applicazione sarà validata.

A titolo di esempio seguono i passi per la creazione dell'applicazione.

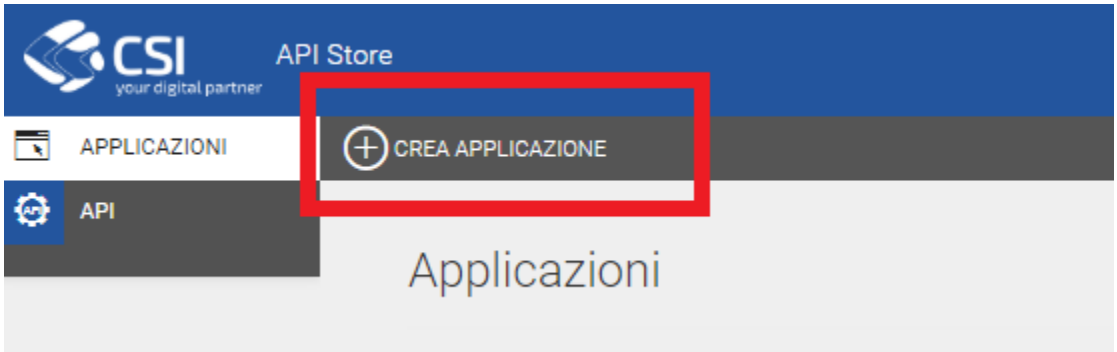
Accedere al portale store all'indirizzo https://api-piemonte-store.csi.it/store ed effettuare il login

Cliccare sul link APPLICAZIONI

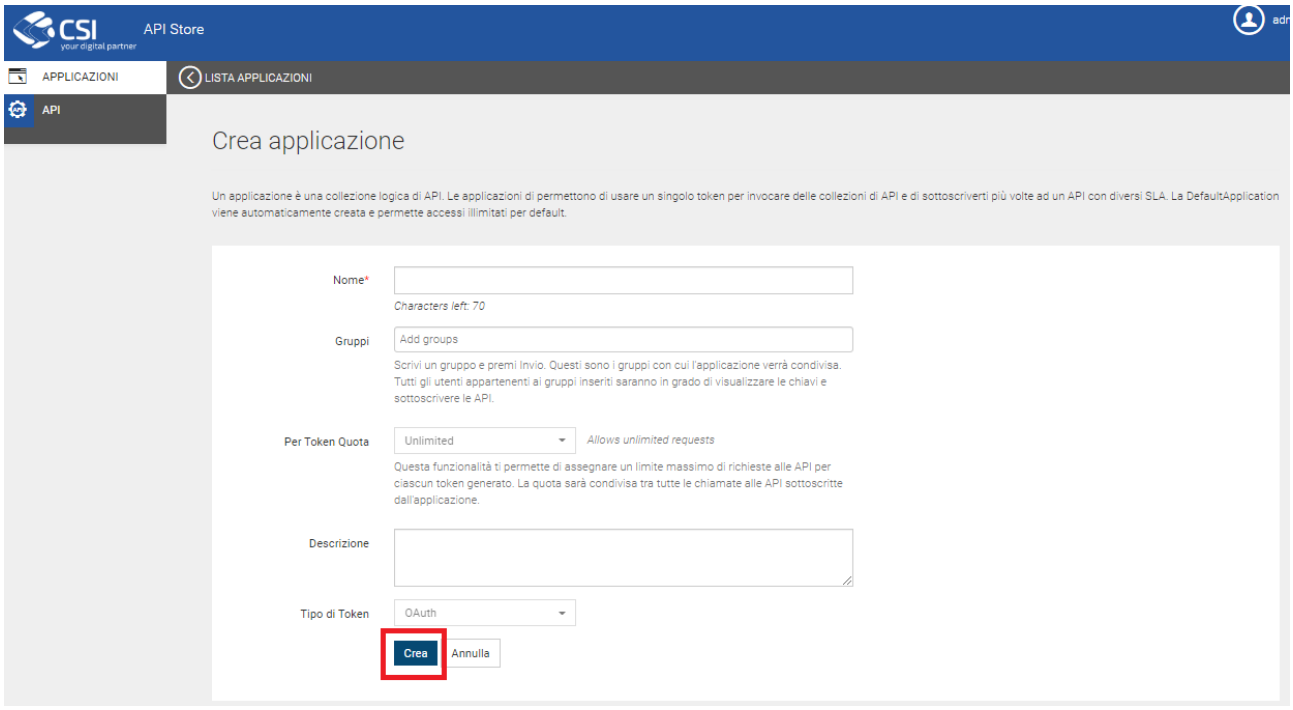


verrà visualizzata la lista di applicazioni censite

Cliccare sul link CREA APPLICAZIONE



Valorizzare i campi presenti nella schermata come descritto precedentemente



Cliccare sul pulsante Crea.

3.3 Sottoscrizione dell'API di interesse

L'utente fruitore dal portale API Store può richiedere la sottoscrizione di una API non appena dispone di un Applicazione il cui stato è ACTIVE. Concluso l'iter di sottoscrizione da parte dell'utente, seguirà un iter di autorizzazione all'accesso da parte dell'amministratore di piattaforma. L'utente riceverà comunicazione via mail sull'esito dell'operazione.

Gli stati della sottoscrizione di un'API sono:

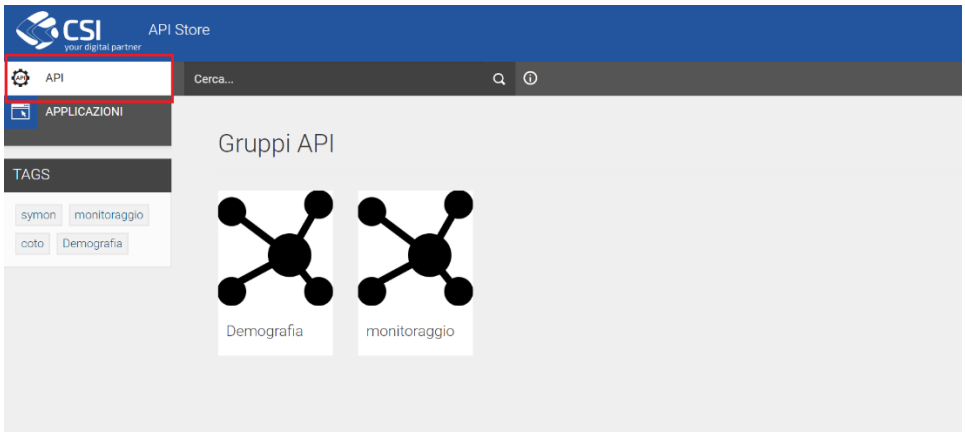
- ONHOLD: in attesa di approvazione
- UNBLOCKED: approvata
- BLOCKED: bloccata dall'owner dell'API
- REJECTED: rifiutata dal gruppo di amministrazione dell'API Manager

Al termine della procedura la sottoscrizione è in stato ONHOLD, ottenuta l'autorizzazione lo stato muterà in "UNBLOCKED" e sarà possibile per il fruitore effettuare le chiamate all'API sottoscritta.

A titolo di esempio seguono i passi da eseguire per effettuare la sottoscrizione.

Accedere al portale store all'indirizzo https://api-piemonte-store.csi.it/store ed effettuare il login

Cliccare su API

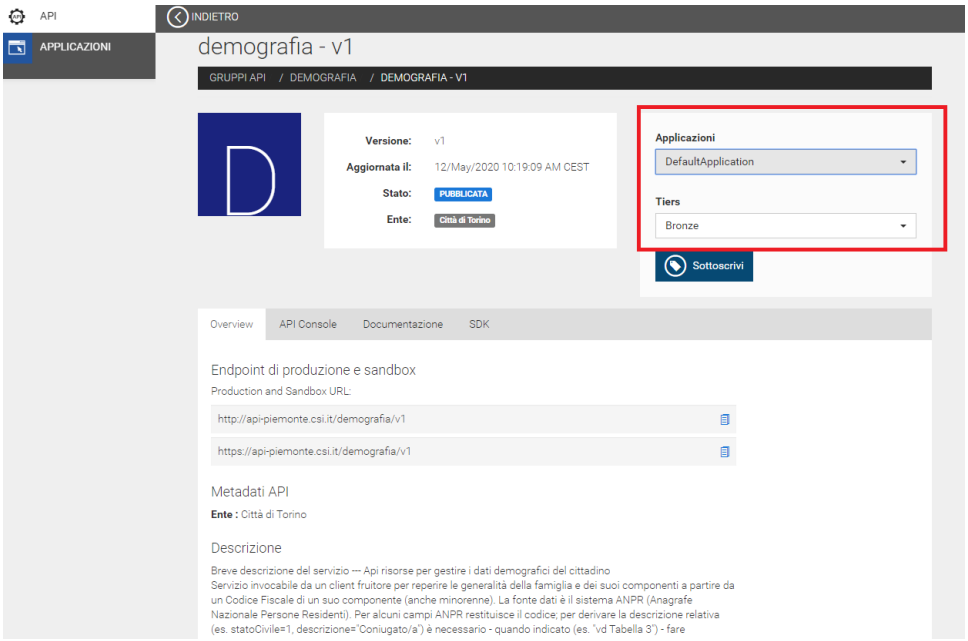


cliccare sull'API che si vuole sottoscrivere. Si aprirà una pagina di dettaglio con le informazioni descrittive dell'API. In alto a destra selezionare l'applicazione che si vuole utilizzare per sottoscrivere l'API e il Tiers di richieste da utilizzare.

Il Tiers rappresenta il numero di richieste per minuto e dispone di configurazioni predefinite dall'owner dell'API.

I tier disponibili di default sono:

- Unlimited (numero illimitato di richieste)
- Gold (5000 richieste al minuto)
- Silver (2000 richieste al minuto)
- Bronze (1000 richieste al minuto)



3.4 Generazione e utilizzo delle chiavi di accesso

Attraverso lo Store per ciascuna applicazione è possibile generare le chiavi (consumer key e consumer secret) per la generazione dei token.

Le applicazioni hanno due tipi di chiavi:

- chiavi di produzione
- chiavi di sandbox

Le chiavi di produzione permettono di generare access token per chiamare gli endpoint di produzione delle API. Le chiavi di sandbox permettono di generare access token con cui accedere, se disponibili, agli endpoint di test delle API, utili per la fase preliminare di integrazione. Indipendentemente dalle chiavi in uso (produzione o sandbox), l'endpoint API è lo stesso, l'API Manager in funzione delle chiavi è in grado di indirizzare al back-end di riferimento (produzione o test di integrazione).

Per ciascuna coppia di chiavi (sia di produzione che di sandbox) è possibile decidere quali grant types utilizzare per la generazione degli access token. I grant types che la piattaforma è in grado di supportare sono i seguenti standard OAuth:

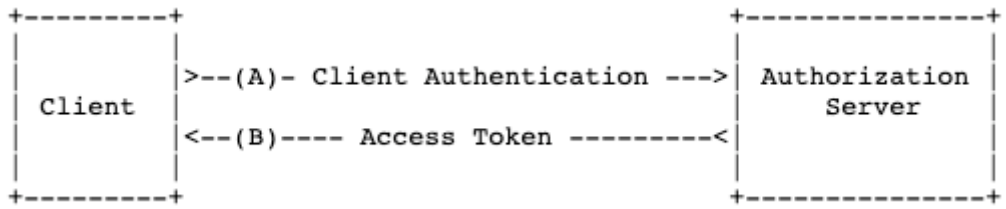
- Refresh Token
- Client Credentials
- Password
- Code

Più quelli custom:

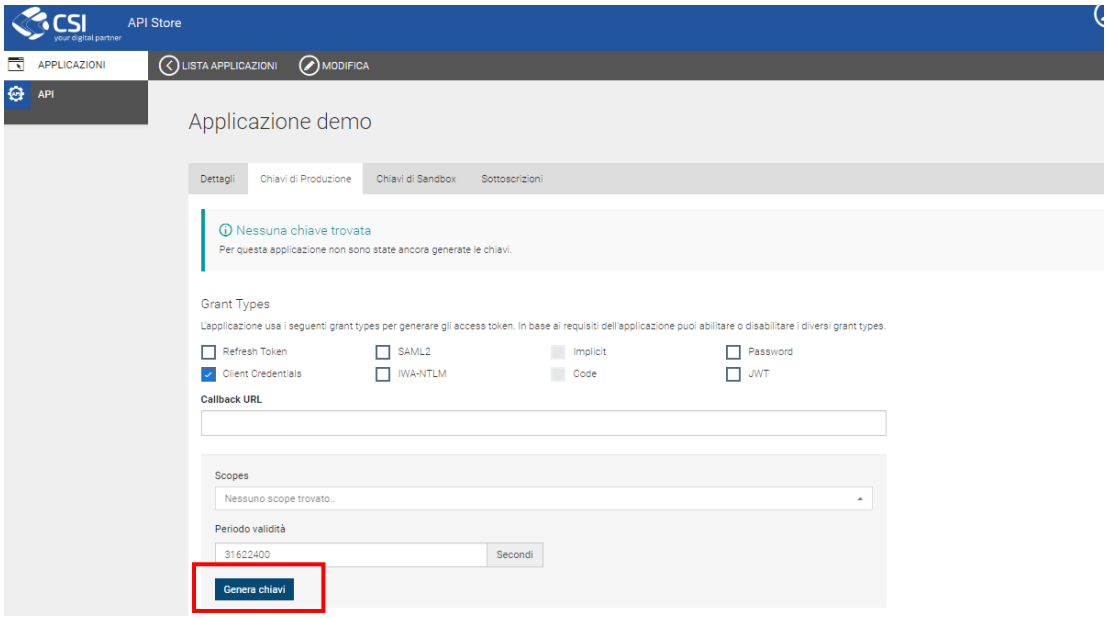
- SAML2
- IWA-NTLM
- JWT

Per una corretta configurazione dell'applicazione bisogna abilitare **solamente** i grant types strettamente necessari in base al tipo di applicazione che si sta sviluppando.

Ad oggi è abilitato esclusivamente Il grant type **client credentials grant type**. Di seguito il flow descritto dalla Internet Engineering Task Force (IETF) nella RFC che descrive il framework Oauth (https://tools.ietf.org/html/rfc6749#page-40).



Per procedere alla generazione delle chiavi selezionare il Grant Type Client Credentials e cliccare sul pulsante Genera chiavi come illustrato nell'immagine seguente



3.4.1 Generazione Token

Questo paragrafo illustra come generare il token utilizzando il grant type client credentials. Un access token generato con questo token identifica solamente l'applicazione chiamante.

Effettuare una chiamata alla Token API con i seguenti parametri:

```
host: api-piemonte.csi.it
path: /token
method: POST
headers:
Content-Type: application/x-www-form-urlencoded
Authorization: Basic base64encode(consumer_key:consumer_secret)
body: grant_type=client_credentials
```

La risposta ritornata in caso di esito positivo è nel seguente formato:

```
http status code: 200
headers
Content-Type: application/json
body:
{
    "scope":"<space separated list of scopes>",
    "token_type":"Bearer",
    "expires_in":<seconds>,
    "access_token":"<access token>"
}
```

Il messaggio restituito contiene un nuovo access token.

In caso di esito negativo verrà restituito un http status code diverso da 200 e nel body un oggetto json che descriverà l'errore, nel formato:

```
{
"error_description":"Client Authentication failed.",
"error":"invalid_client"
}
```

3.4.2 Esempio di chiamata cURL alla token API:

```
curl -k -d "grant_type=client_credentials" \
-H "Authorization: Basic
SVpzSWk2SERiQjVlOFZLZFpBblVpX2ZaM2Y4YTpHbTBiSjZvV1Y4ZkM1T1FMTGxDNmpzbEFDVzhh " \
-H "Content-Type: application/x-www-form-urlencoded" \
https://api-piemonte.csi.it/token
```

3.4.3 Esempio di codice java per ottenere l'access token

Esempio di codice java per ottenere l'access token

```
import javax.net.ssl.HttpsURLConnection;
import java.io.*;
import java.net.URL;
import java.util.Base64;

private static final String clientId = "….";//clientId
private static final String clientSecret = "….";//client secret
private static final String tokenUrl = "https://api-piemonte.csi.it/token";
private static final String auth = clientId + ":" + clientSecret;
private static final String authentication =
Base64.getEncoder().encodeToString(auth.getBytes());

public String getToken(){
String content = "grant_type=client_credentials";
BufferedReader reader = null;
HttpsURLConnection connection = null;
String returnValue = "";
try {
    URL url = new URL(tokenUrl);
    connection = (HttpsURLConnection) url.openConnection();
    connection.setRequestMethod("POST");
    connection.setDoOutput(true);
    connection.setRequestProperty("Authorization", "Basic " + authentication);
    connection.setRequestProperty("Content-Type","application/x-www-form-urlencoded");
    connection.setRequestProperty("Accept", "application/json");
    PrintStream os = new PrintStream(connection.getOutputStream());
    os.print(content);
```

```
    os.close();
    reader=new BufferedReader(new InputStreamReader(connection.getInputStream()));
    String line = null;
    StringWriter out = new StringWriter(connection.getContentLength() > 0 ?
connection.getContentLength() : 2048);
    while ((line = reader.readLine()) != null) {
            out.append(line);
    }
    returnValue = out.toString();
} catch (Exception e) {
    System.out.println("Error : " + e.getMessage());
} finally {
if (reader != null) {
            try {
                    reader.close();
            } catch (IOException e) {
    }
}
connection.disconnect();
}
}
```

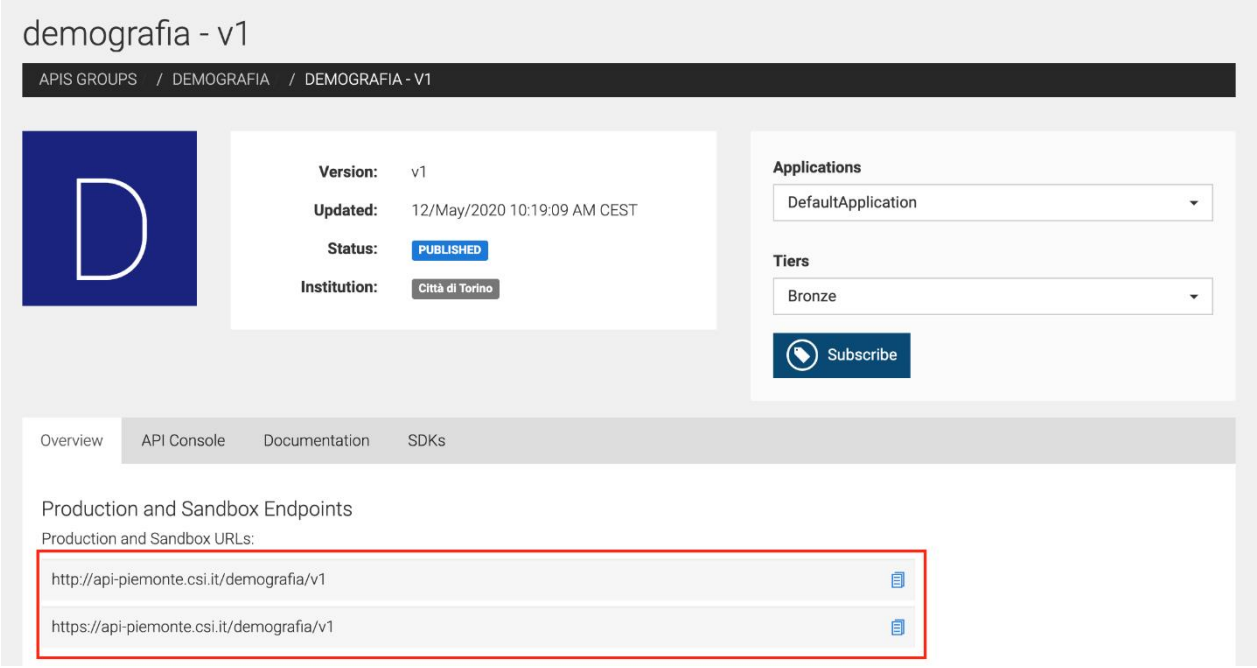
3.4.4 Esempio di codice phyton per ottenere l'access token

Esempio di codice phyton per ottenere l'access token (utilizza la libreria Requests-OAuthlib - https://requests-oauthlib.readthedocs.io/):

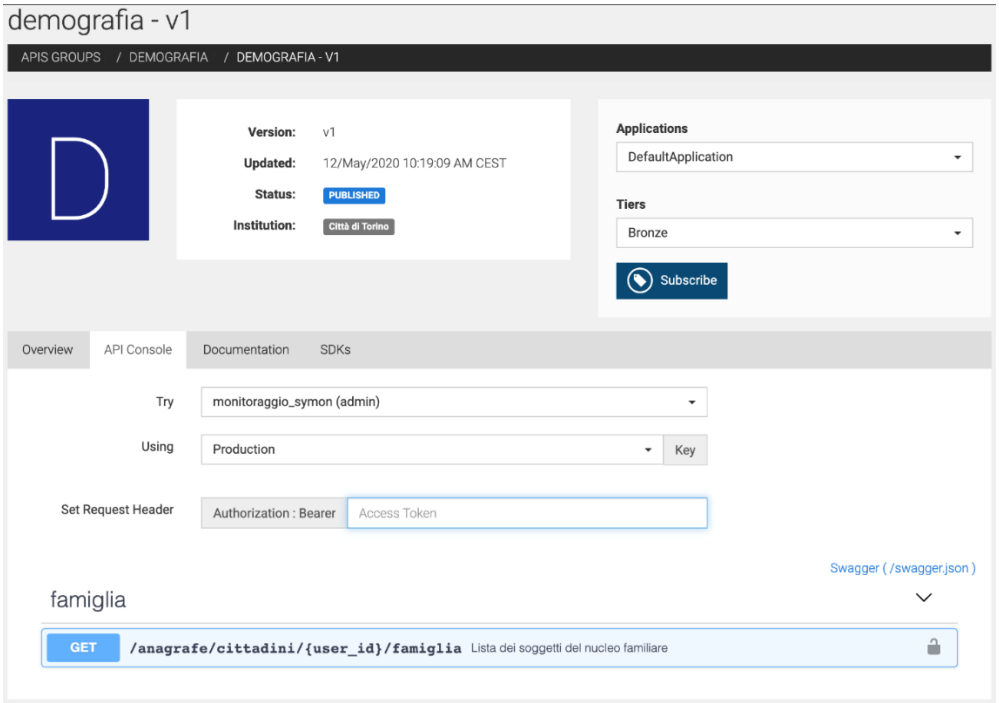
```
client_id = 'your_client_id'
client_secret = 'your_client_secret'


from oauthlib.oauth2 import BackendApplicationClient
from requests.auth import HTTPBasicAuth
auth = HTTPBasicAuth(client_id, client_secret)
client = BackendApplicationClient(client_id=client_id)
oauth = OAuth2Session(client=client)
token = oauth.fetch_token(token_url='https://api-piemonte.csi.it/token', auth=auth)
```

3.5 Invocare un API

Recuperato l'access token è possibile invocare le API sottoscritte. Gli endpoint delle API e la relativa documentazione sono visibili nella pagina di dettaglio di ciascuna API.



L'owner dell'API può decidere con quale trasporto http pubblicare l'API, se http e/o http con SSL (https). In tutti i casi, il trasporto http senza SSL può essere usato solamente dalle applicazioni che risiedono sulla rete interna del CSI. Nella tab "API Console" del dettaglio di una API è possibile vedere, nel caso di un API REST, tutte le risorse esposte con le relative interfacce di input e output.



Per invocare l'API quindi bisogna aggiungere l'header Authorization mettendo il bearer token generato.

Di seguito il formato di una chiamata cURL ad una risorsa GET di una API esposta sull'API Manager:

```
curl –H "Authorization: Bearer <access_token>" <api_endpoint>
```

Ad esempio:

```
curl –H "Authorization: Bearer d7252944-14c4-48d5-858b-6b80d8bf08b4" https://api-
piemonte.csi.it/demografia/v1/anagrafe/cittadini/XXXXXX45B23L219F/famiglia
```