API (application programming interface)

Le API (Application Programming Interface) svolgono un ruolo cruciale nell'era dell'informatica e della condivisione dei dati. Esse fungono da intermediari che consentono a diverse applicazioni o servizi di comunicare e scambiare dati tra loro in modo standardizzato. Ecco alcune aspetti comuni a tutte le API:

- **Autenticazione dell'Utente**: Molte API richiedono un processo di autenticazione dell'utente per garantire la sicurezza e il controllo degli accessi. Gli utenti possono ottenere una chiave API o un token di autenticazione che viene utilizzato per identificare e autorizzare le richieste API. Questa autenticazione è fondamentale per proteggere i dati e le risorse dell'utente.

- **Richiesta di Accesso ai Dati o Pubblicazione di Contenuti**: Gli sviluppatori di applicazioni utilizzano le API per inviare richieste ai server che ospitano i dati o i servizi desiderati. Queste richieste possono essere di diversi tipi, come richieste di lettura (per ottenere dati) o richieste di scrittura (per pubblicare dati o effettuare modifiche).

- **Risposta del Server all'Utente**: Il server che ospita l'API riceve la richiesta dell'utente, la elabora e restituisce una risposta. Questa risposta può contenere i dati richiesti o confermare che l'azione richiesta è stata completata con successo. In caso di errore, il server può restituire un messaggio di errore.

Tipi di API

Open API: Queste API sono pubbliche e accessibili a chiunque abbia l'autorizzazione. Sono destinate a essere condivise con il pubblico e consentono a sviluppatori esterni di creare applicazioni di terze parti che interagiscono con i dati o i servizi offerti tramite l'API. Tuttavia, potrebbero essere applicati limiti sul numero di richieste che gli utenti possono effettuare per evitare abusi o sovraccarichi del sistema.

Partner API: Queste API sono solitamente limitate a partner o sviluppatori autorizzati. Sono utilizzate per integrazioni più specifiche o per consentire a partner commerciali di accedere a determinate funzionalità o dati dell'API. Questo tipo di API può essere utilizzato per creare partnership commerciali e offrire servizi aggiuntivi a clienti o utenti.

Internal API: Queste API sono destinate all'uso interno all'azienda o all'organizzazione che le ha create. Sono accessibili solo da sviluppatori o operatori all'interno dell'azienda stessa e consentono di accedere ai dati del backend o alle funzionalità aziendali. Le API interne sono utilizzate per automatizzare processi aziendali, condividere dati tra divisioni aziendali e creare strumenti personalizzati per l'uso aziendale.

	Public API	Partner API	Internal API
Availability	Anyone can use it	Restricted to selected users	Restricted users within an organization
Subscription	Limits the number of requests for free users sometimes	Requests limits may vary depending on the subscription plan	No subscription plan
Authentication	May require an OAuth or an API key	May require an access token	May not require any authentication tokens
Use Case	Business-to-Consumer	Business-to-Consumer & Business-to-Business	Business-to-Employee, Application-to-Application, Business-to-Business, Business-to-Consumer

Rest API

Utilizzano il protocollo rest . Effettuano richieste HTTP (GET/POST/DELETE metodi) e ricevono risposte strutturate dal server in formato XML o JSON. Il protocollo HTTP usato per la connessione è stateless , ovvero non tiene conto dello storico delle richieste ed ognuna è indipendente dalle precedenti e successive.

Il messaggio strutturato in formato **JSON** (JavaScript Object Notation) è una rappresentazione dei dati che utilizza una sintassi chiara e semplice basata su coppie chiave-valore.

La chiamata REST è costituita da :

- tipo di protocollo : le chiamate REST usano il protocollo HTTP o HTTPS
- server da contattare : specifica l'URL del server a cui inviare la richiesta
- risorsa richiesta : specifica la risorsa a cui si desidera accedere sul server
- parametro : includi nella richiesta per definire le condizioni o le informazioni richieste (id=5)

Posso utilizzare diversi strumenti per effettuare richieste REST , come per esempio curl o python.

Python:

```python
import json
from urllib.request import urlopen as uRequest
page = uRequest('https://jsonplaceholder.typicode.com/comments?id=5')
content = json.load(page) #Python dictionary
print('Name:',content[0]['name'])
print('Body:',content[0]['body'])
```

```
Name: vero eaque aliquid doloribus et culpa
Body: harum non quasi et ratione
tempore iure ex voluptates in ratione
harum architecto fugit inventore cupiditate
voluptates magni quo et
```

Curl :

Show users:

```
curl --request GET --url
'https://api.twitter.com/1.1/users/show.json?user_id=1705602238746435584'
--header 'Authorization: Bearer Bearer AAAAAAAAAAAAAAAAAAAAA……'
```

Autenticazioni

- **API Keys**: Le chiavi API sono stringhe di caratteri uniche assegnate a un'applicazione o a un utente per autenticare le richieste API. Queste chiavi vengono solitamente inviate come parte delle richieste API e consentono ai server di identificare e autorizzare l'accesso. Le chiavi API sono spesso utilizzate per il controllo degli accessi e per tenere traccia delle richieste API.

- **Basic Authentication**: L'autenticazione di base richiede l'invio di username e password come parte dell'header della richiesta HTTP. Tuttavia, questa modalità di autenticazione non è la più sicura, poiché le credenziali viaggiano in chiaro sulla rete. Di solito, è preferibile utilizzare metodi di autenticazione più sicuri, come OAuth.

- **HMAC (Hash-based Authorization Control)**: HMAC è un metodo di autenticazione basato su hash. Un messaggio viene codificato e firmato con una chiave segreta, creando una firma univoca. Questa firma viene inviata insieme al messaggio. Il server riceve la firma, decodifica il messaggio e verifica la firma utilizzando la chiave segreta. Se la firma corrisponde, l'accesso viene autorizzato.

- **OAuth (Open Authorization)**: OAuth è un protocollo di autorizzazione aperto e standardizzato che consente a un'applicazione di ottenere l'accesso a risorse in nome di un utente senza condividere le credenziali di accesso dirette. Tramite OAuth le credenziali non vengono date al servizio a cui si vuole accedere ma ci si affida all'autenticazione tramite account presenti in servizi di terze parti. OAuth consente alle applicazioni di terze parti di accedere ai dati degli utenti da un sito web attendibile (come Gmail) senza fornire le credenziali dell'utente alle applicazioni di terze parti. OAuth è un framework di autorizzazione. Tutte le richieste di autorizzazione da parte di terze parti sono gestite dal server OAuth.