Python Biella Group

FastAPI framework
Introduzione alle API

a cura di: **Andrea e Mario**

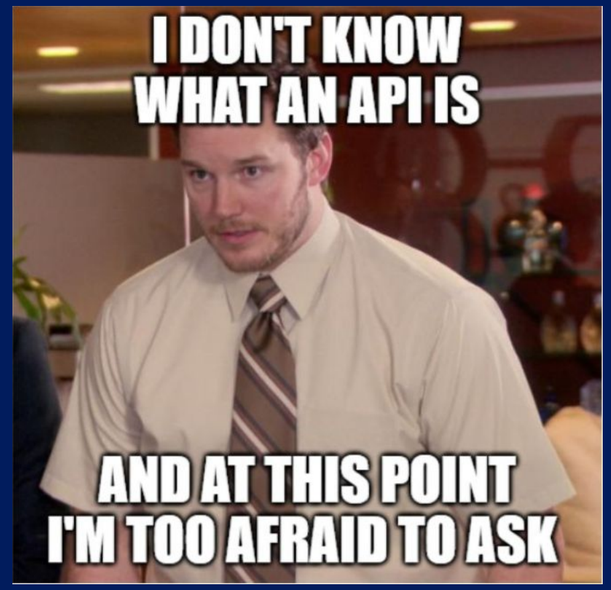
Introduzione alle API

Niente di “magico” ne’ troppo moderno

API (Application Program Interface)

Set di definizioni e protocolli con i quali vengono realizzati e integrati software applicativi (anche eterogenei)

Permettono di comunicare con servizi senza sapere come sono implementati



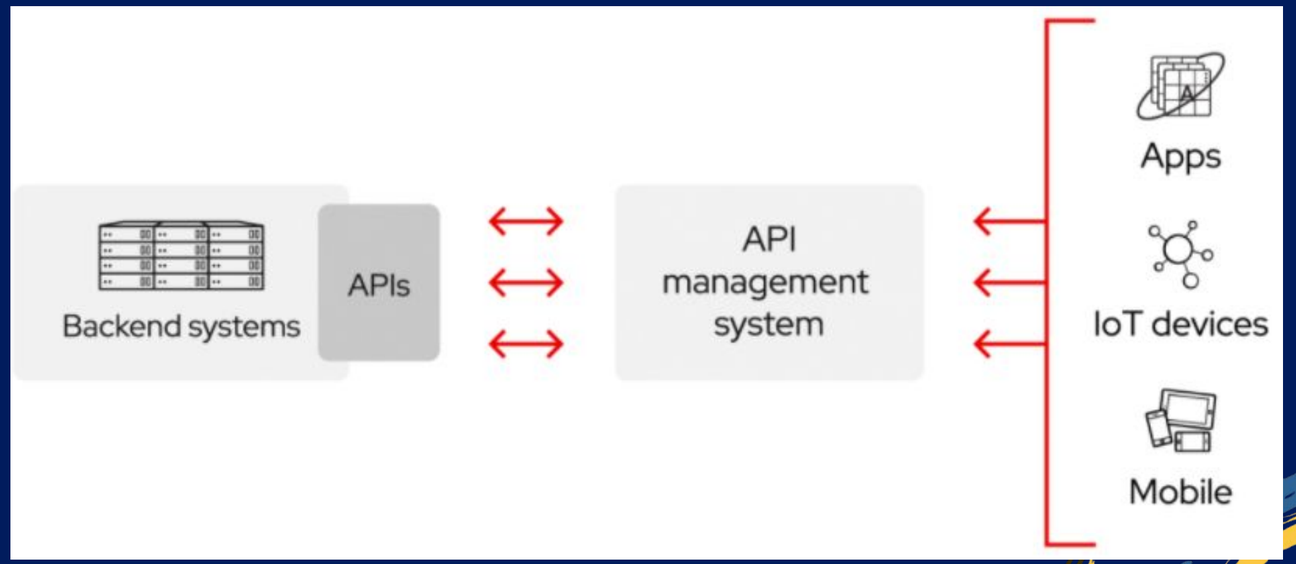
Tipi di API

SDK: Funzioni e librerie di interfaccia a framework / software proprietary

OS API: specifiche per colloquiare con il device (ex Android e la camera)

WEB API: Web Services
Network oriented
XML Soap (Simple Object Access Protocol)

OPEN API, Rest API
JSON, rappresentazione testuale
Astrazione alta



Web API / Open API

Comunicazioni via HTTP con una lista specifica di «verbi»

1. ***GET*** *: "Gets" the specified resources from an endpoint. This results in a response from the server without any change in the state of the server*
2. ***POST*** *: Sends some data to an endpoint, typically resulting in an action that in turn changes the state of the server*
3. ***PUT*** *: Replaces some data on a server. Like* ***POST*** *but different in the sense that* ***PUT*** *requests will always produce the same result*
4. ***DELETE*** *: The* ***DELETE*** *method deletes the specified resource from server*
5. ***HEAD*** *: The* ***HEAD*** *method asks for a response like that of a GET request, but only the status line and header section*
6. ***CONNECT*** *: The* ***CONNECT*** *method establishes a tunnel to the server identified by the target resource*
7. ***OPTIONS*** *: The* ***OPTIONS*** *method is used to describe the communication options for the target resource.*
8. ***TRACE*** *: The* ***TRACE*** *method performs a message loop-back test along the path to the target resource.*
9. ***PATCH*** *: The* ***PATCH*** *method is used to apply partial modifications to a resource.*

Esempio (da cmd):

curl -i -X GET "https://api.coindesk.com/v1/bpi/currentprice.json"

HEADER

```
Connection: keep-alive
Access-Control-Allow-Origin: *
Cache-Control: max-age=15
Date: Mon, 13 Sep 2021 15:37:51 GMT
Expires: Mon, 13 Sep 2021 15:38:07 UTC
Server: nginx/1.18.0
X-Powered-By: Fat-Free Framework
X-Cache: Hit from cloudfront
Via: 1.1 2ba5677785db2f66bc73820b2a261477.cloudfront.net (CloudFront)
X-Amz-Cf-Pop: CDG50-P2
X-Amz-Cf-Id: 5MjY2mrONNOqy9Xzl5XZrkDroMtik64FOecNHHl8xwQo7Nl-rUrmZA==
Age: 3
```

BODY


```
{
    "time": {
      "updated": "Sep 13, 2021 15:37:00 UTC",
      "updatedISO": "2021-09-13T15:37:00+00:00",
      "updateduk": "Sep 13, 2021 at 16:37 BST"
    },
    "disclaimer": "This data was produced from the CoinDesk Bitcoin Price Index (USD). Non-USD currency data converted using hourly conversion rate from
openexchangerates.org",
    "chartName": "Bitcoin",
    "bpi": {
      "USD": {
            "code": "USD",
            "symbol": "&#36;",
            "rate": "44,446.6668",
            "description": "United States Dollar",
            "rate_float": 44446.6668
      },
      "GBP": {
            "code": "GBP",
            "symbol": "&pound;",
            "rate": "32,094.5825",
            "description": "British Pound Sterling",
            "rate_float": 32094.5825
      },
      "EUR": {
            "code": "EUR",
            "symbol": "&euro;",
            "rate": "37,636.0151",
            "description": "Euro",
            "rate_float": 37636.0151
      }
    }
```

JSON
JavaScript Object Notation

Composizione di una serie di coppie chiave-valore

Formato standard de-facto, leggero

In Python facilmente importabile in un *dictionary*

API – Alcuni use cases

Integrare applicazioni: due applicazioni eterogenee devono scambiarsi dati e “parlarsi”

Costruire applicazioni multi-piattaforma: servizi comuni con diversi tipi di client
Es: Facebook da Android, Apple, browser

Migliorare la funzionalità di applicazioni
Es: integrazione con Oauth di Google

Fornire accesso esterno/servizio ad una tua applicazione

Disaccoppiare BackEnd e FrontEnd dell’applicazione

Rest (REpresentational State Transfer) API

Pattern architetturale

Tipicamente usa solo 4 verbi

- ***GET*** *: Used to select or retrieve data from a server. Can be used to same limited data as well.*
- ***POST*** *: Used to send or write data to the server. Typically used to send sensitive information such as credentials, financial data, or large data sets such as files*
- ***PUT*** *: Used to update data that is already present on the server, e.g. updating database entries, replacing files, etc*
- ***DELETE*** *: Used to delete existing data from the server*

Stateless: le richieste client devono contenere tutte le info necessarie per eseguirla; il server non deve memorizzare lo stato del client tra una richiesta e l'altra

Risorsa: elemento di interesse nel dominio dell'applicazione

REST – Metodi request

Table 14-1. HTTP request methods in RESTful APIs

Request method	Target	Description	HTTP response status code
GET	Individual resource URL	Obtain the resource.	200
GET	Resource collection URL	Obtain the collection of resources (or one page from it if the server implements pagination).	200
POST	Resource collection URL	Create a new resource and add it to the collection. The server chooses the URL of the new resource and returns it in a `Location` header in the response.	201
PUT	Individual resource URL	Modify an existing resource. Alternatively, this method can also be used to create a new resource when the client can choose the resource URL.	200 or 204
DELETE	Individual resource URL	Delete a resource.	200 or 204
DELETE	Resource collection URL	Delete all resources in the collection.	200 or 204

REST – Errori HTTP

Table 14-2. HTTP response status codes typically returned by APIs

HTTP status code	Name	Description
200	OK	The request was completed successfully.
201	Created	The request was completed successfully and a new resource was created as a result.
202	Accepted	The request was accepted for processing, but it is still in progress and will run asynchronously.
204	No Content	The request was completed successfully and there is no data to return in the response.
400	Bad Request	The request is invalid or inconsistent.
401	Unauthorized	The request does not include authentication information or the credentials provided are invalid.
403	Forbidden	The authentication credentials sent with the request are insufficient for the request.
404	Not Found	The resource referenced in the URL was not found.
405	Method Not Allowed	The method requested is not supported for the given resource.
500	Internal Server Error	An unexpected error occurred while processing the request.

Adesso parliamo di:

- Come configurare il proprio IDE (VSCode)

- Come creare il proprio virtualenv (con poetry)

- Concetti:
  - Async
  - Pydantic (dataclasses)
  - type hints

- Un piccolo esempio per iniziare