Alternanza scuola lavoro
Blulink – Blaise Pascal

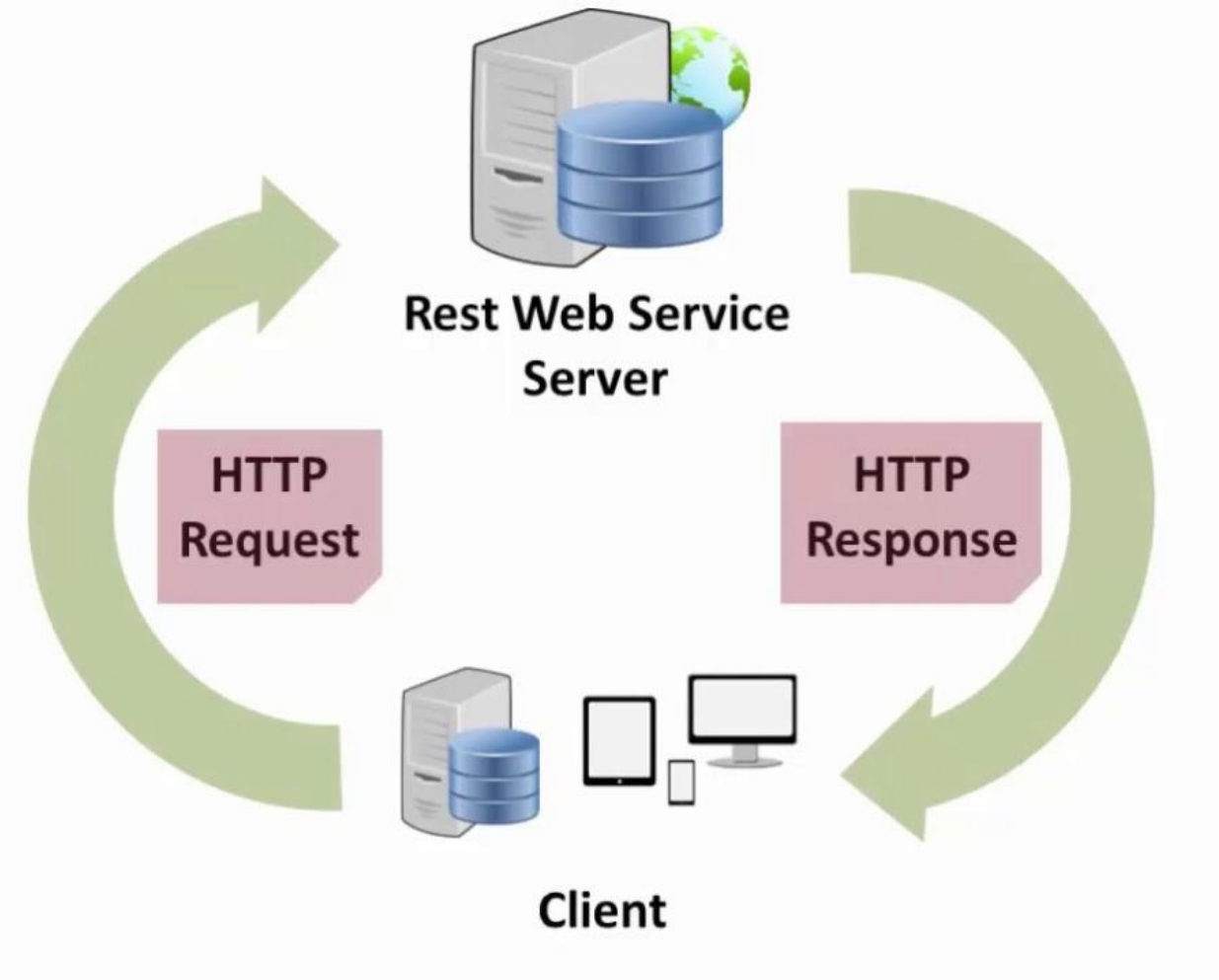
Ing. Claudio Soli

Claudio Soli

- 35 anni
- 2001 : Diploma di Perito Tecnico in Elettronica e Telecomunicazioni Presso I.I.S. Fermo Corni
- 2007 : Laurea Magistrale in Ingegneria Informatica presso UNIMORE
- 2007 : in Blulink come software developer
- Oggi : in Blulink coordino il team di sviluppo

OBIETTIVI

- Capire,
- Esercitarsi,
- Sviluppare



Client per siti/app web: Browser

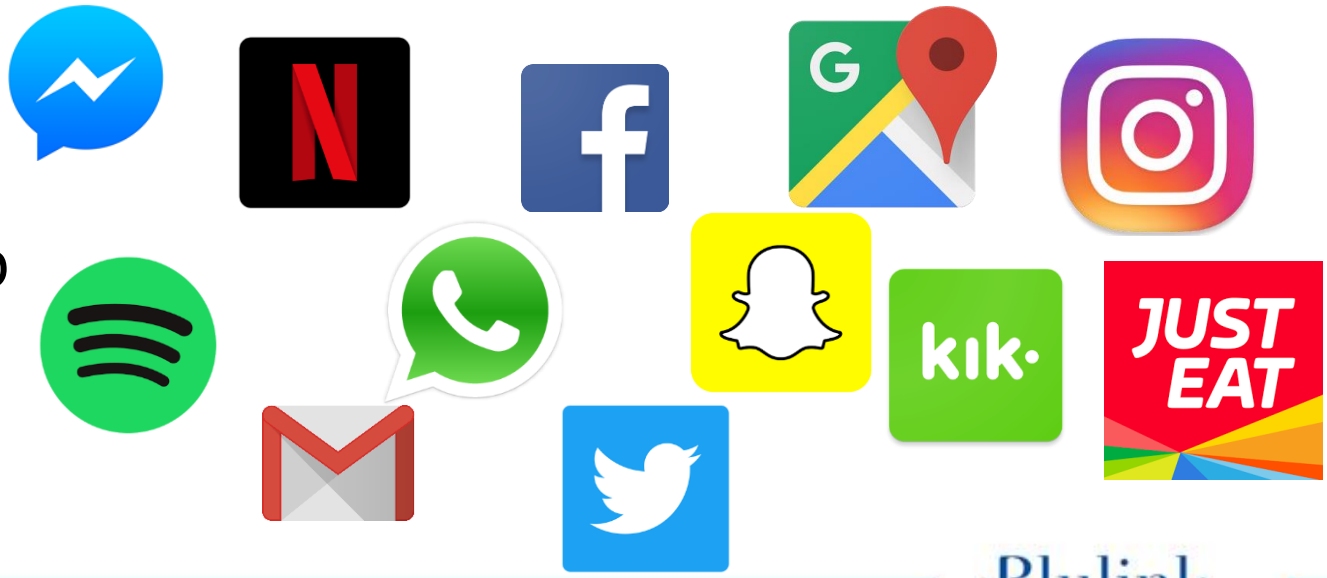
- applicazione (software) per il recupero, la presentazione e la navigazione di risorse (fonti di informazione) sul web.

Client per web service

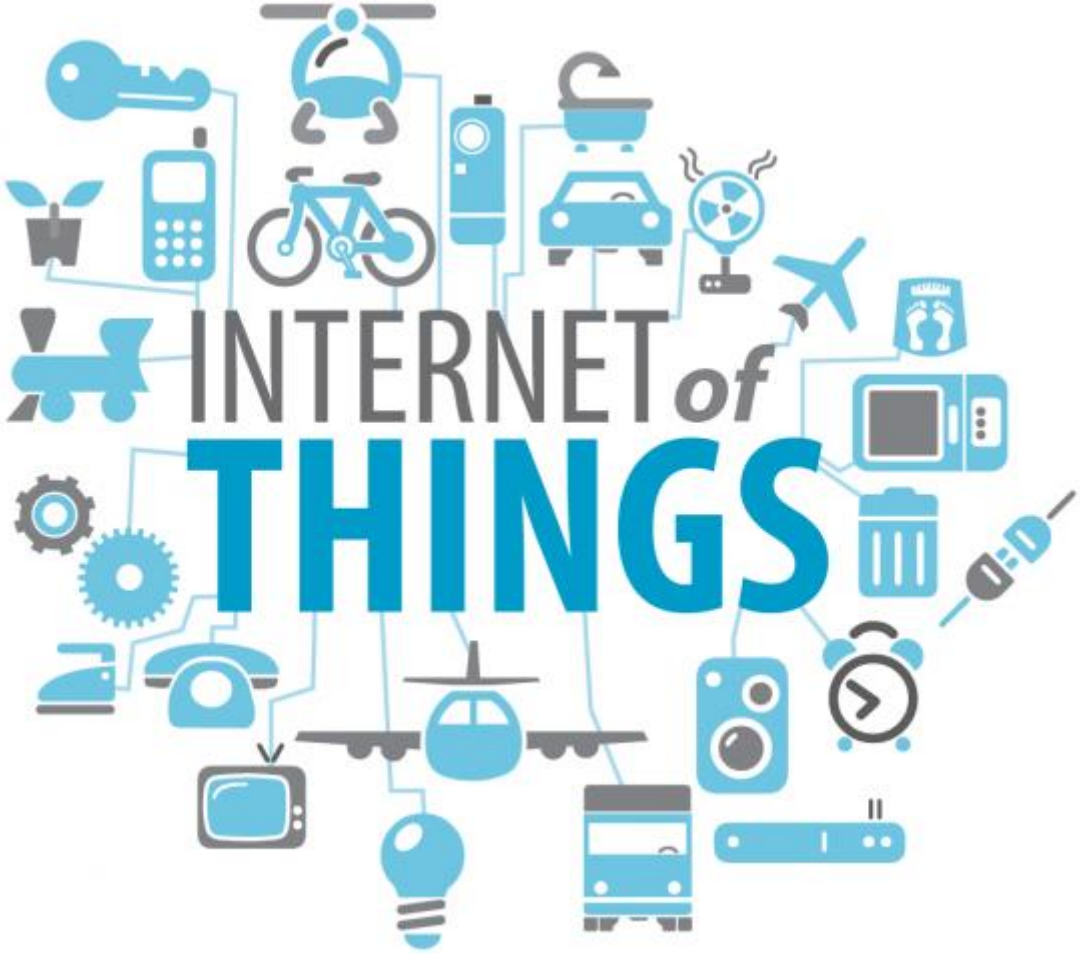
- Qualsiasi tipo di software che è in grado di scambiare (inviare/ricevere) informazioni con un server

- Browser:

- App:
  - Mobile
  - Desktop
  - Console



Client: In quali dispositivi?



Rest Web Service Server

- Server:

Qualsiasi software in grado di soddisfare le richieste dei client fornendo un **servizio** tramite l'accesso a **risorse**

- Rest Service (Web API o Rest API):

l'accesso alle informazioni avviene via Web mediante l'utilizzo del solo protocollo HTTP

- Risorse:

rappresentano le fonti di informazioni. Sono accessibili attraverso URL

Protocollo di rete

- Un **protocollo** è un insieme di regole formalmente descritte che definiscono le modalità di comunicazione tra una o più **entità**.

- Se le due entità sono remote all'interno di una rete informatica (internet), si parla di **protocollo di rete**

- Nel nostro caso le entità sono rappresentate dai **servizi/risorse** esposti dal server

Protocollo di rete HTTP

- Acronimo di HyperText Transfer Protocol
- Iperteso: più documenti di testo tra di loro connessi (link tra le pagine web). La caratteristica principale è la possibilità di lettura
- Ipermedia: non solo testo ma anche immagini, audio, video

Richiesta HTTP

- Principali Richieste:
  - GET → Per il recupero di risorse
  - POST → Per la creazione di una nuova risorsa
  - PUT → Per la modifica di una risorsa
  - DELETE → Per la cancellazione di una risorsa
- Header: Informazioni aggiuntive sulla richiesta (tipo di contenuto nel body..)
- Body: (opzionale) messaggio dati

Risposta HTTP

- Status: codice esito della richiesta
  - Successo (2xx) : la richiesta è stata ricevuta, capita e processata correttamente
  - Client Error (4xx): La richiesta presenta errori
  - Server Error (5xx) - Il server fallisce nel processare la richiesta
- Header: Informazioni aggiuntive sulla risposta (tipo di contenuto nel body..)
- Body: contiene i dati, le informazioni

Il formato dati

- I dati, le risorse, le informazioni vengono tipicamente rappresentate in un formato dati
- Json è attualmente il formato più diffuso (standard) restituito da un Rest Service http://json.org/json-it.html
- HTML è invece il formato dati standard restituito da un sito o applicazione web

Esercitazioni : Disclaimer

- L’utilizzo della quasi totalità dei Rest Web Service pubblici (Google, Apple, Microsoft compresi) richiede una chiave di autenticazione rilasciata a seguito della creazione di un account.
  Molti servizi sono a pagamento per cui non addentratevi ne nella creazione di nessun tipo di account ne nella richiesta di chiave di accesso

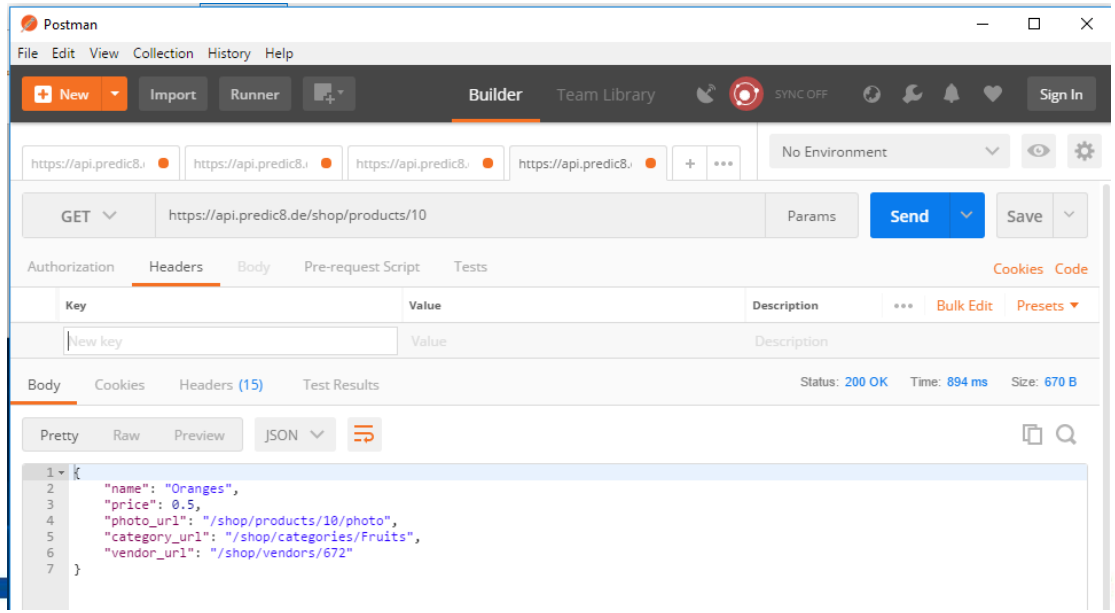
Esercitazione 1

- Obiettivo: inviare richieste HTTP ad un Rest Web Service Server

- Cosa serve:
  - Client
  - Connessione Internet
  - Rest Web Service Server

Esercitazione 1 : il Client

- Postman:
  - client che consente di formulare richieste HTTP
  - Utilizzato per supportare lo sviluppo, il test e la documentazione di Rest Service
  - https://www.getpostman.com/postman



Esercitazione 1 : alcuni Server

- Fruit Shop
  - URL Base: https://api.predic8.de/shop
  - Documentazione: https://api.predic8.de/shop/docs#
- Star Wars API
  - Url Base: https://swapi.co/api/
  - Documentazione: https://swapi.co/documentation
- Open Air Quality Data
  - Url Base : https://api.openaq.org/v1
  - Documentazione: https://docs.openaq.org/

Esercitazione 2 – sviluppiamo il Client

- Obiettivo : sviluppare un client "console" che invia richieste HTTP ad un Rest Web Service Server. Farlo poi evolvere a piacere

- Cosa serve:
  - Linguaggio di programmazione: C#
  - IDE : Visual Studio Code
  - SDK : .NET CORE SDK
  - Connessione Internet
  - Rest Web Service Server

Esercitazione 2 : Client Tutorial

- **Passo 1** : SEGUIRE SCRUPOLOSAMENTE la guida: https://docs.microsoft.com/it-it/dotnet/core/tutorials/with-visual-studio-code

Verificando che il programma "Hello World" funzioni (anche in debug)

- **Passo 2** : Installare Libreria HTTPClient
  - Da VS Code Selezionare View→Integrated Terminal
  - Lanciare il comando:

    dotnet add package Microsoft.AspNet.WebApi.Client --version 5.2.4-preview1
  - Lanciare il comando:

    dotnet restore
- **Passo 3**: Usare la libreria httpClient installata al Passo 2 per interrogare I servizi
  - Esempio Semplice di utilizzo:

    https://gist.github.com/dantheman213/a39684b7824e30f445e5
  - Esempio Avanzato di utilizzo:

    https://docs.microsoft.com/it-it/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-client
- **Passo 4**: Personalizzare e far evolvere l'applicazione con creatività interrogando I servizi della slide seguente
- **Documentazione di Base su C#:**
- https://docs.microsoft.com/it-it/dotnet/csharp/language-reference/

Esercitazione 2 : alcuni Server

- Fruit Shop (USARE QUESTO)
  - URL Base: https://api.predic8.de/shop
  - Documentazione: https://api.predic8.de/shop/docs#
- Star Wars API
  - Url Base: https://swapi.co/api/
  - Documentazione: https://swapi.co/documentation
- Open Air Quality Data
  - Url Base : https://api.openaq.org/v1
  - Documentazione: https://docs.openaq.org/

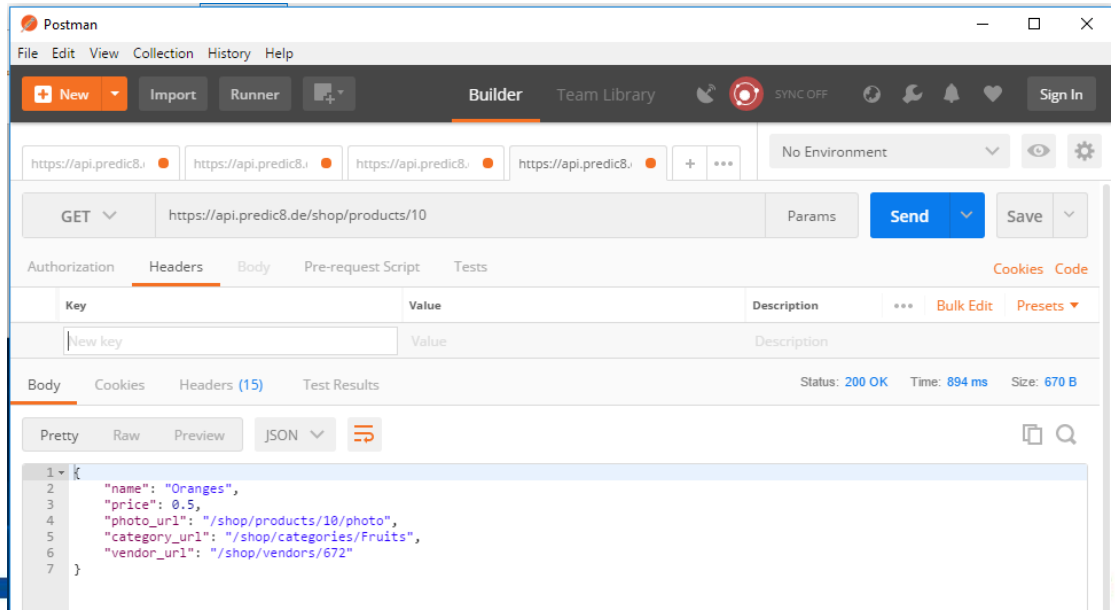
Esercitazione 3 – sviluppiamo il Server

- Obiettivo: sviluppare un Rest Web Service Server e testarlo con postman

- Cosa serve:
  - Postman come client
  - Connessione Internet
  - Linguaggio di programmazione + IDE + Framework per lo sviluppo del Rest Web Service Server

Esercitazione 3 : il Client

- Postman:
  - client che consente di formulare richieste HTTP
  - Utilizzato per supportare lo sviluppo, il test e la documentazione di Rest Service
  - https://www.getpostman.com/postman



Esercitazione 3 : Linguaggio e IDE

- Integrated Development Environment:
  Visual Studio Code:
- Framework per Rest API: ASP.NET Core
- Linguaggio C#

Esercitazione 3 : il server

- Seguire la guida

https://docs.microsoft.com/it-it/aspnet/core/tutorials/web-api-vsc