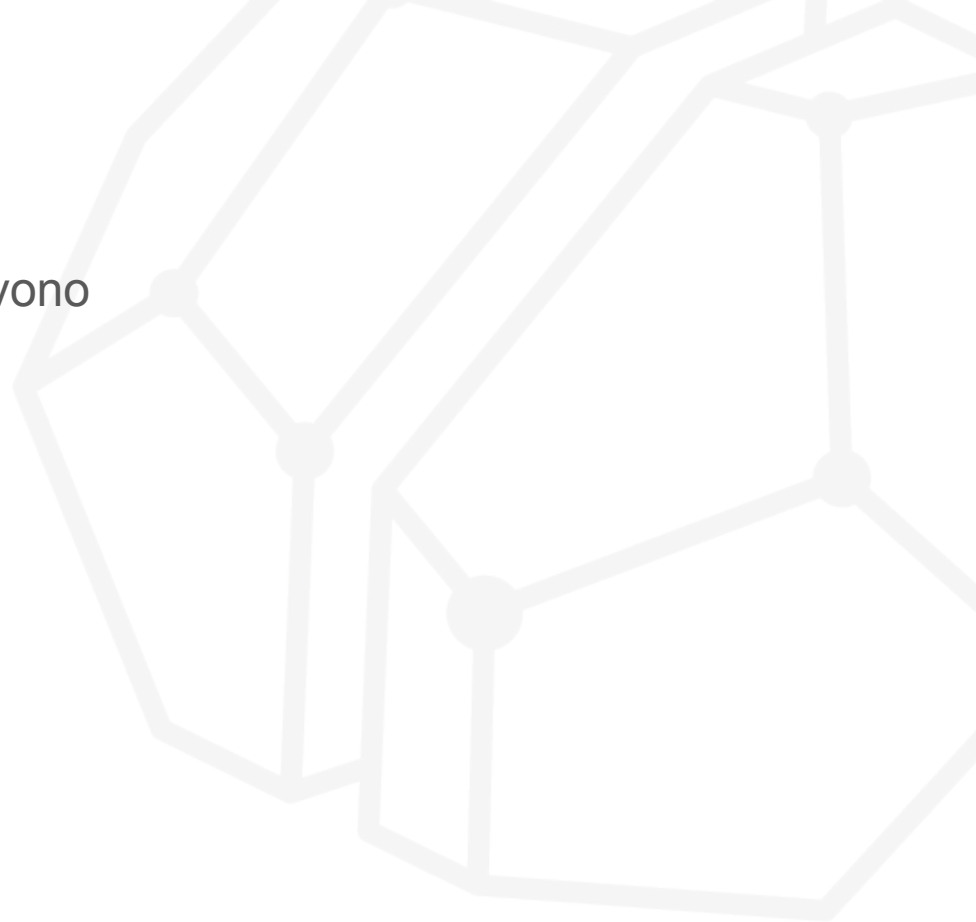


REST API e Json

Overview

Outline

- REST API - cosa sono e a cosa servono
- Formati di risposta
- Il formato JSON
- REST client



Chi sono

Stefano Bianchini

mail: stefano.bianchini@simplenetworks.it

Founder & CTO

Simplenetworks SRL

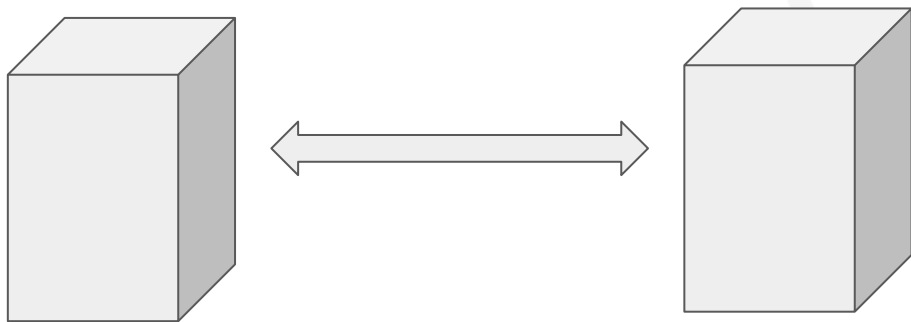


Il protocollo HTTP



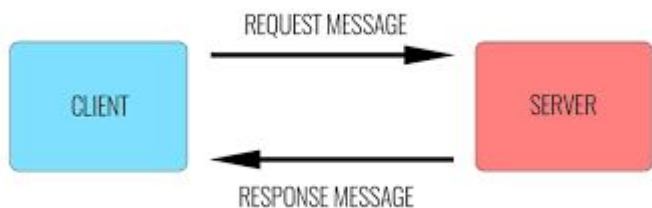
Cos'è un protocollo

- Un protocollo può essere visto come le parole e la grammatica di una lingua.
- È una modalità di comunicazione
- Client e server usano la stessa lingua (protocollo) e quindi possono comprendersi.



Il protocollo HTTP

- Si basa su richiesta e risposta
- Il messaggio di richiesta è composto di quattro parti:
 - riga di richiesta (request line);
 - composta da metodo (GET, POST, HEAD, PUT, DELETE, OPTIONS, PATCH...), URI e versione del protocollo.
 - sezione header (informazioni aggiuntive);
 - riga vuota (CRLF: i 2 caratteri carriage return e line feed);
 - body (corpo del messaggio).
- Il messaggio di risposta è composto da quattro parti:
 - riga di stato (status-line);
 - sezione header;
 - riga vuota (CRLF: i 2 caratteri carriage return e line feed);
 - body (contenuto della risposta).



HTTP status codes



1XX
INFORMATIONAL

2XX
SUCCESS

3XX
REDIRECTION

4XX
CLIENT ERROR

5XX
SERVER ERROR

Esempi di Header HTTP

Request Headers

User-Agent: l'identificazione del tipo di client che stiamo usando)

Accept-Language: la lingua desiderata nel messaggio di risposta

Accept: la tipologia di messaggio desiderato in risposta: HTML, XML, JSON, ecc.

Demo: richiesta verso google.it e risposta

Name	x	Headers	Preview	Response	Initiator	Timing	Cookies
www.google.it		General					
m-cdos.dpf.hsm.jsa.d.csl		Request URL: https://www.google.it/					
ADea4I7azMzqatKGveA270...		Request Method: GET					
googlelogo_color_272x92dp...		Status Code: 200					
data:image/gif;base...		Remote Address: 216.58.205.67:443					
data:image/png;base...		Referer Policy: strict-origin-when-cross-origin					
rs=AA2YrTvxBuWAKOAS...							
rs=AA2YrTu_vEVoK98iK9...		Response Headers					
desktop_searchbox_sprites...		accept-ch: Sec-CH-UA-Platform-Version					
gen_204?s=webhp&ta=af&at...		accept-ch: Sec-CH-UA-Full-Version					
search?q&c=0&client=gws-...		accept-ch: Sec-CH-UA-Arch					
m=CnSW2d,DhPYme.EkexV...		accept-ch: Sec-CH-UA-Model					
client_204?atyp=i&biw=245...		accept-ch: Sec-CH-UA-Bitness					
cb=google.loaded_0		alt-svc: h3="443"; ma=2592000, h3-29="443"; ma=2592000, h3-0850="443"; ma=2592000, h3-0846="443"; ma=2592000, h3-0843="443"; ma=2592000, quic="443"; ma=2592000; v="46,43"					
cid7prid=19027093&pgid=19...		bfcache-opt-in: unload					
get?rt=j&sourceid=538		cache-control: private, max-age=0					
m=DPrE.LiQuz.IXOQxe.KQv...		content-encoding: br					
gen_204?atyp=i&el=u77qYe...		content-length: 44434					
client_204?cs=1		content-type: text/html; charset=UTF-8					
log?format=json&hasfast=tru...		date: Fri, 21 Jan 2022 14:10:03 GMT					
m=aLUIP7xjs=82		expires: -1					
LXwMfBb_v4CVleJnB6QvRp...		p3p: CP="This is not a P3P policy! See g.co/p3phelp for more info."					
log?format=json&hasfast=tru...		server: gws					
m=b_b_tp		set-cookie: SEARCH_SAME_SITE=Cg0IXZ0B; expires=Wed, 20-Jul-2022 14:10:03 GMT; path=/; domain=.google.it; SameSite=strict					
gsa_supper_g-64.gif		set-cookie: NID=511-NG02A-4FkgVazidfdu7z3bLgpeE43GeW8csAgacj9f3y2dAdzXQd7wZkqN18GLHSovq6Pkt0g73ct1Yd3C_rZceTNGttVcrgvrd09_lhnoQvW0h_gbnJ0Tzrgb3liJnrg30a_BK312h3fI0NjXJB					
u4abrhNHxJiGduGoTOiLU...		oqlHRFUB88t5JgaTdcRwLQZEGlaptCJDH096L4T3sC573lVPg_6cHGmGT82SLguhw7E6TfMqK0RfghL9nZ7ZJndf-IPbJxL6j5pTtUjTlfp2; expires=Sat, 23-Jul-2022 14:10:03 GMT; path=/; domain=					
KF0mCnqde92Fm4mxK...		oogle.it; Secure; HttpOnly; SameSite=none					
m=byfTOJJaVmc.LEkZe		strict-transport-security: max-age=31536000					
m=n73qwf,ws9Tic.IZT63.e5g...		x-frame-options: SAMEORIGIN					
log?format=json&hasfast=tru...		x-xss-protection: 0					
gen_204?use_corp=on&atyp...							
m=Wb0yJ.hhhU8.FCpbqb.W...		Request Headers					
log?format=json&hasfast=tru...		:authority: www.google.it					
gen_204?atyp=csi&el=u77q...		:method: GET					
ui		:path: /					
log?format=json&hasfast=tru...		:scheme: https					
manifest?psa=webhp		accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9					
favicon.ico		accept-encoding: gzip, deflate, br					
log?format=json&hasfast=tru...		accept-language: it-IT, it;q=0.9, en-US;q=0.8, en;q=0.7					
log?format=json&hasfast=tru...		cookie: SID=GAJE4AnIqbbw0hwrIOSVINQXIA9rC5FN5s5PLuvEzQV7_1jGxpAP41F10H_hwFozkQA.; __Secure-1PSID=GAJE4AnIqbbw0hwrIOSVINQXIA9rC5FN5s5PLuvEzQV7_1CPebh1JAQVj82TSLJY3m8w.; __					
log?format=json&hasfast=tru...		ecure-3PSID=GAJE4AnIqbbw0hwrIOSVINQXIA9rC5FN5s5PLuvEzQV7_1evzkw0GBX4MYTnXLU0X7.; HSID=AzY8231dNgTwh0dht; SSID=ACr8_pxqf-s-dcdk0; APISID=TY1PFP2R_xpCOR/AZGHP_qa6M9nK8BI					
gen_204?atyp=i&el=u77qYe...		SAPISID=VE3pq7PrZY5vMdg/AaV2PZq_yn5-NR3g3; __Secure-1PAPISID=VE3pq7PrZY5vMdg/AaV2PZq_yn5-NR3g3; __Secure-3PAPISID=VE3pq7PrZY5vMdg/AaV2PZq_yn5-NR3g3; NID=511-mm0PvaJyuXayMf					
log?hasfast=true&auth=SAPI...		fhna1St3ud3181fenM_QdXvjYlM1a4WiaZ70jegCVqKs4EN12240vWgoIntpQThySC6wJq0f1FdzT_Yqw4xfCrtFk5GatVxvFksmlZ8Rgd0wFasvyM1tT93c3EHMnVevSLvs6evvYIDu8rTvZP0T9BYXCmHontDmPsezi					
log?hasfast=true&auth=SAPI...		At1u72ptwCphUzjZj-J0Hf3SxxX9252a8BK7U2GfEzYqK8iInzqM_Vxg					
app?orig=https%3A%2F%...		dnt: 1					
m=b_b_tp							

REST API Overview



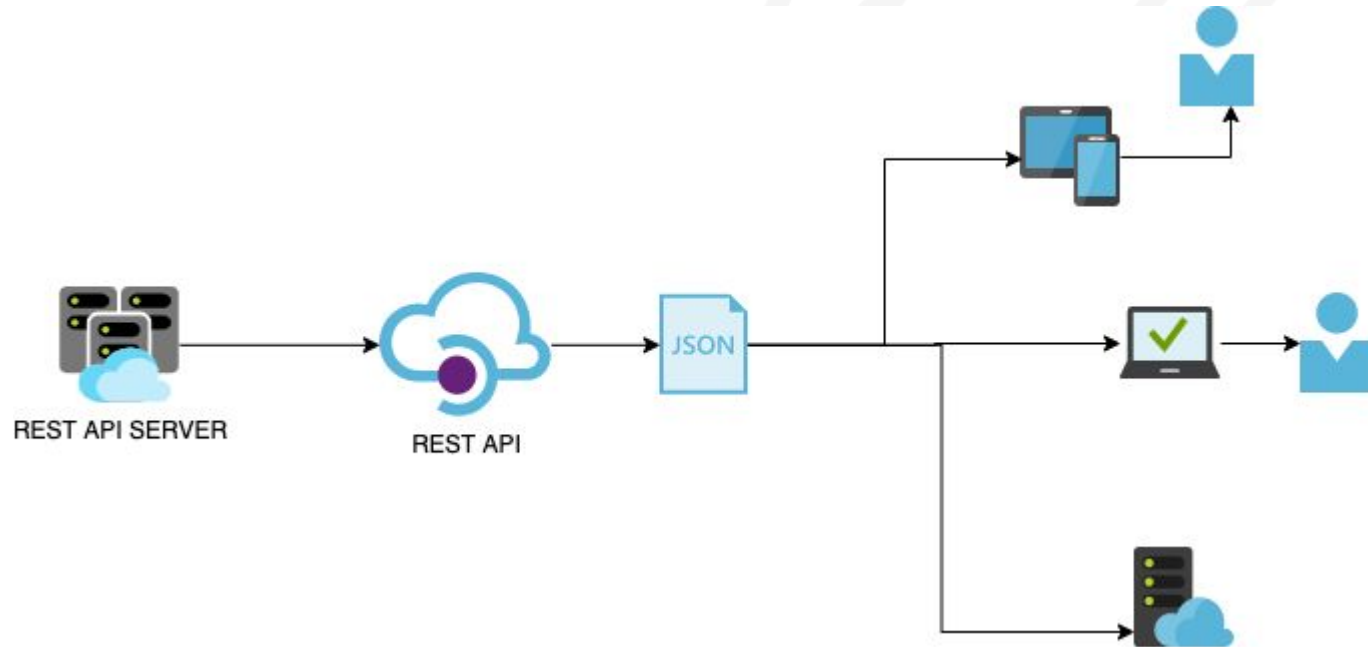
REST API

- Acronimo di **R**epresentational **S**tate **T**ransfer
- Non è uno standard, è uno stile architetturale
- È un protocollo di comunicazione
 - Stateless
 - Client-server
 - Cacheable
- Utilizza le richieste HTTP per le **operazioni sui dati** (lettura, creazione, modifica, cancellazione)
- Definisce un insieme di metodi di richiesta per indicare l'azione desiderata relativamente alla risorsa
- È un sistema leggero
- «Nouns as URI, verbs as HTTP method»
- Indipendente da specifici linguaggi/tecnologie

Architettura orientata alle risorse

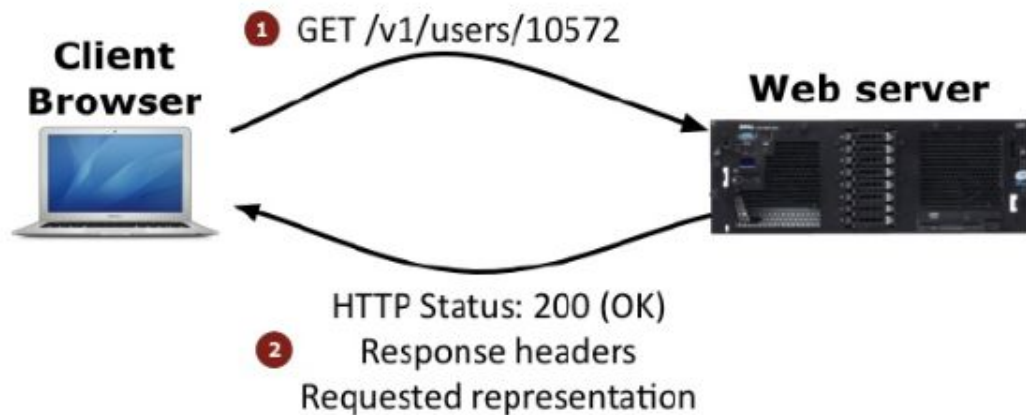
- Le risorse sono identificate dalle URI
- Le azioni dai metodi HTTP
 - HTTP GET, PUT, POST, DELETE
- Senza stato
 - Non viene mantenuto uno stato tra le invocazioni delle API REST
 - Lo stato non viene salvato sul server
 - Ogni richiesta HTTP alle API REST viene eseguita a se stante
 - Non usa quindi sessioni e cookies

REST API



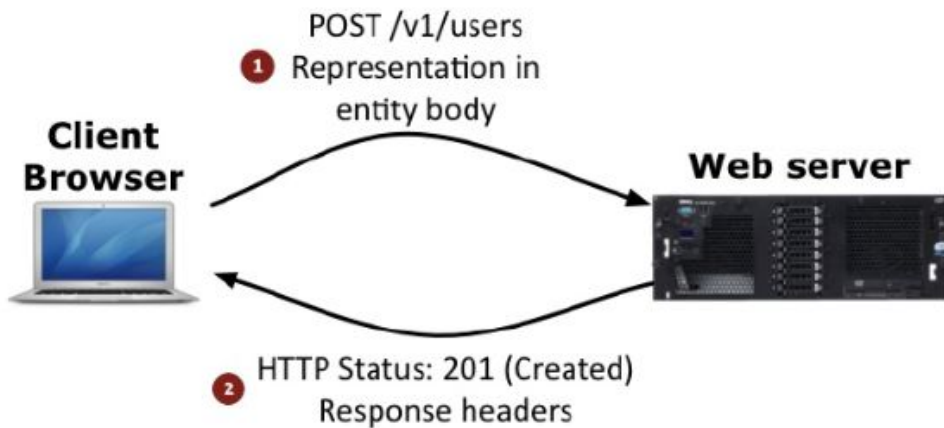
GET

- Per definizione, le GET sono operazioni a sola lettura
- Non comportano cambiamenti server side



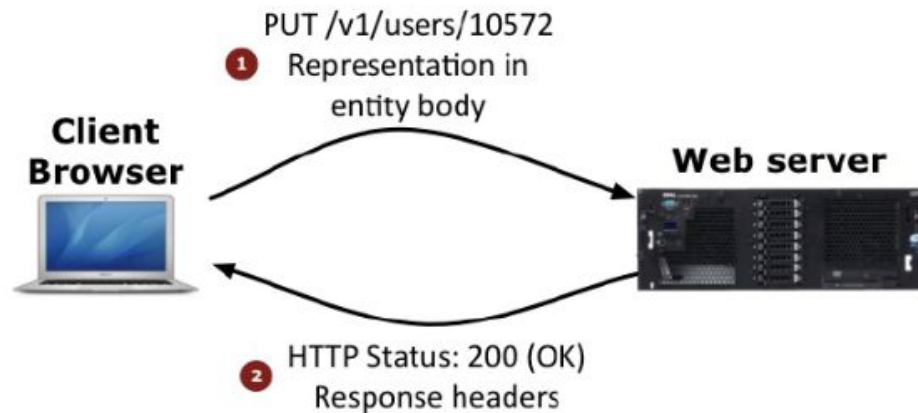
POST

- Per definizione, la POST è un'operazione di **creazione**
- Comporta cambiamenti server side



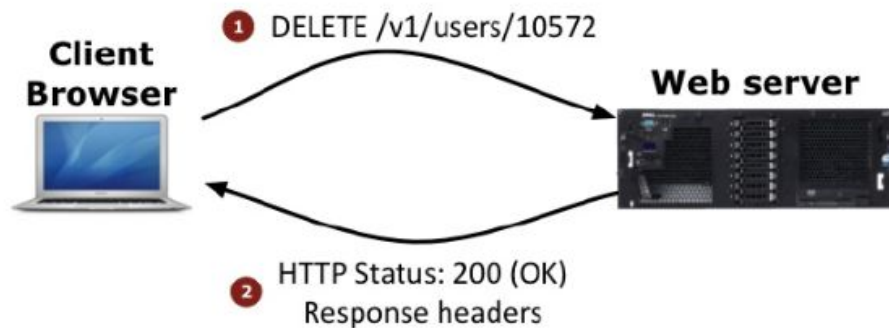
PUT / PATCH

- Per definizione, la PUT è un'operazione di **modifica**
- Comporta cambiamenti server side



DELETE

- Per definizione, la DELETE è un'operazione di **eliminazione**
- Comporta cambiamenti server side
- Restituisce solitamente un codice 204 (diversamente dall'immagine in basso)



Esempio pratico

Task	Method	Path
Create a new customer	POST	/customers
Delete an existing customer	DELETE	/customers/{id}
Get a specific customer	GET	/customers/{id}
Search for customers	GET	/customers
Update an existing customer	PUT	/customers/{id}

Come comunicare con le REST API

- Usando i metodi HTTP
- Modificando e plasmando i dati inviati nella body request se i metodi utilizzati sono PUT e POST
- Modificando gli HTTP headers della request
 - Ad esempio specificando una autenticazione HTTP Basic

REST API

formato delle risposte

Risposta delle REST API

- Il server potrà specificarmi degli header HTTP di risposta, ad esempio
 - Potrebbe usare l'header **Content-Type**
 - Indica il formato della risposta (la tipologia) e aiuta il client a processare correttamente il corpo della risposta stessa.
 - Potrebbe usare l'header **Cache-Control**
 - Definisce, da parte del server, una policy di caching per la risposta
- Solitamente, la risposta è in formato XML oppure JSON

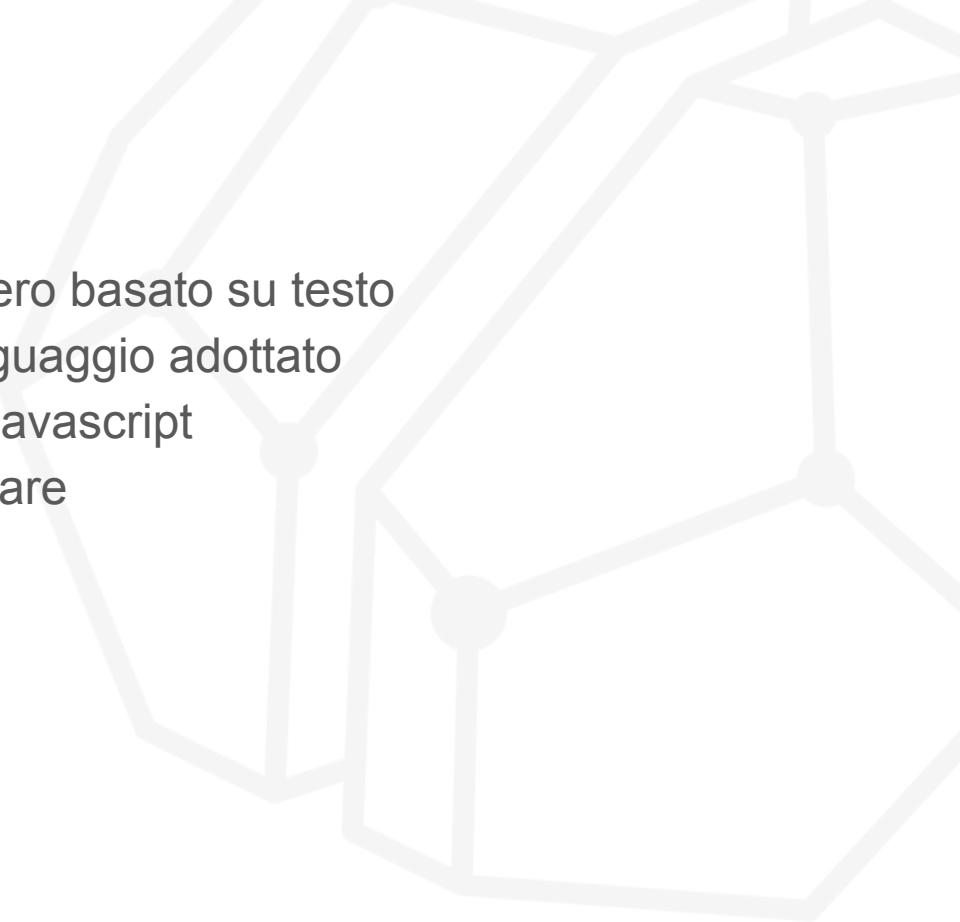
XML

- XML (sigla di eXtensible Markup Language) è un metalinguaggio per la definizione di linguaggi di markup, ovvero un linguaggio marcatore basato su un meccanismo sintattico che consente di definire e controllare il significato degli elementi contenuti in un documento o in un testo.

```
<?xml version="1.0" encoding="UTF-8"?>
<utenti>
  <utente anni="20">
    <nome>Ema</nome>
    <cognome>Princi</cognome>
    <indirizzo>Torino</indirizzo>
  </utente>
  <utente anni="54">
    <nome>Max</nome>
    <cognome>Rossi</cognome>
    <indirizzo>Roma</indirizzo>
  </utente>
</utenti>
```

JSON: cos'è?

- Un formato di interscambio dati leggero basato su testo
- Completamente indipendente dal linguaggio adottato
- Basato su un subset del linguaggio Javascript
- Facile da capire, manipolare e generare



JSON: cosa NON è?

- Non è complicato
- Non è un linguaggio di markup
- Non è un linguaggio di programmazione



JSON: cosa NON è?

- Non è complicato
- Non è un linguaggio di markup
- Non è un linguaggio di programmazione



JSON: perché usarlo?

- Facile da leggere
- Facile da creare e manipolare
- Può essere utilizzato da molti linguaggi (Javascript, Php)



JSON e XML

- Entrambi sono in formato «plain text»
- Entrambi sono facilmente leggibili
- Gerarchici: un valore può contenere una lista di oggetti o altri valori
- JSON è molto più veloce di XML
- Meno sintassi
- JSON usa oggetti tipizzati; in XML i valori devono essere interpretati a runtime

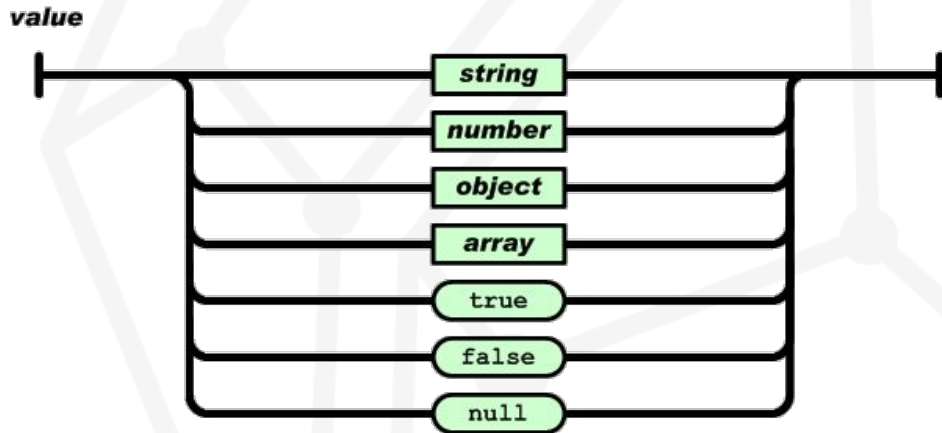
Il formato JSON

Sintassi JSON

- JSON è basato su due strutture:
 - Un insieme di coppie nome/valore. In diversi linguaggi, questo è realizzato come un oggetto, un record, uno struct, un dizionario, una tabella hash, un elenco di chiavi o un array associativo.
 - Un elenco ordinato di valori. Nella maggior parte dei linguaggi questo si realizza con un array, un vettore, un elenco o una sequenza.

Sintassi JSON

- Stringhe
 - Con escaping di backslash «/»
 - Racchiuse in doppi apici
- Numeri
 - Interi
 - Reali
- Booleani
- Null
- Oggetti - insieme di chiavi/valori racchiusi da {}
- Array – insieme di chiavi / valori racchiusi da []



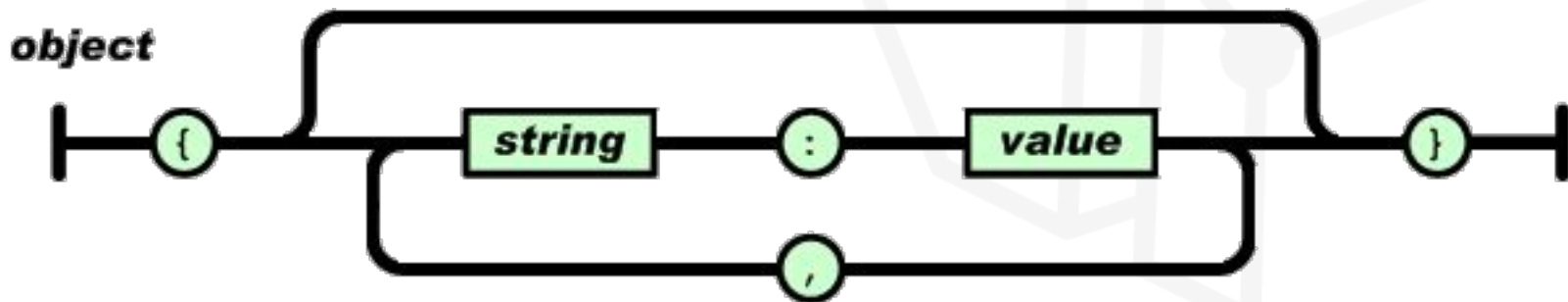
Esempio JSON

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        {
          "value": "New",
          "onclick": "CreateNewDoc()"
        },
        {
          "value": "Open",
          "onclick": "OpenDoc()"
        },
        {
          "value": "Close",
          "onclick": "CloseDoc()"
        }
      ]
    }
  }
}
```



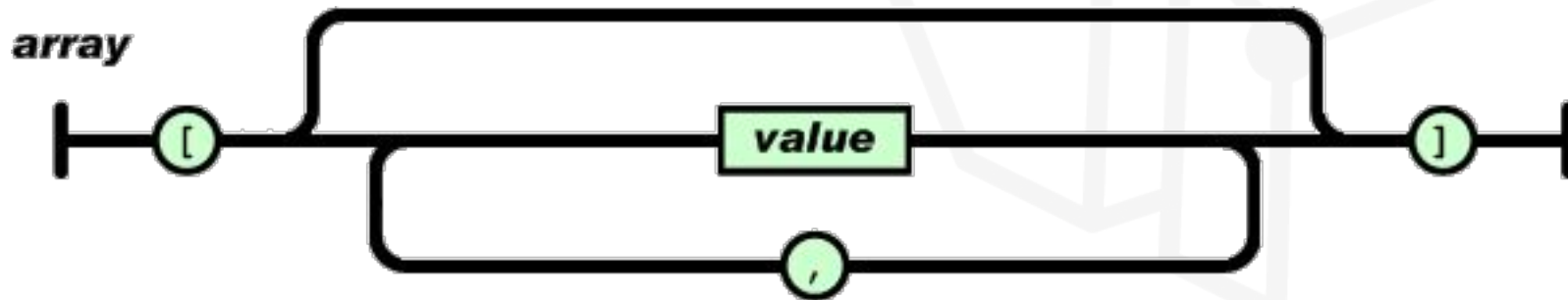
Gli oggetti

- Un oggetto è una serie non ordinata di nomi/valori. Un oggetto inizia con { (parentesi graffa sinistra) e finisce con } (parentesi graffa destra). Ogni nome è seguito da : (due punti) e la coppia di nome/valore sono separata da , (virgola).



Gli array in JSON

- Liste di valori / oggetti ordinati
- Inizia con [
- Termina con]
- Gli elementi sono separati da virgola (,)



REST client e risorse

Con cosa mi collego alle REST API?

- Principalmente, tramite Javascript (XMLHttpRequest) e i suoi framework (Angular, React ...) per interagirvi all'interno di Web Application
- Attraverso cURL, server side
- Per testing, attraverso dei REST client
 - Postman (<https://www.postman.com>)
 - Insomnia (<https://insomnia.rest>)

Risorse

- Fake API
 - <https://reqres.in>
 - <https://jsonplaceholder.typicode.com>
 - <https://fakerestapi.azurewebsites.net/>
- JSON Beautifier
 - <https://codebeautify.org/jsonviewer>





Q & A