

The European



Delphi Conference

26 maggio 2017 - ROMA



REST API Fantastiche e dove trovarle

Marco Breveglieri



The European Delphi Conference



Marco Breveglieri

Software & Web Developer @ABLS Team

Blogger (www.compilaquindiva.com)

Host @Delphi Podcast (www.delhipodcast.com)

and sushi eater! 🍣



Chi sono



The European Delphi Conference



Premessa sulle Web API



Le sfide di oggi

- Internet Scale Application
- Cloud-based Application
- Tipologie variegiate di device e di "form factor"
- Sistemi operativi eterogenei



Cosa intendiamo per Web API?

***"Non è tutto oro
ciò che è HTTP..."***

**REST
Web API**



Web API *non* è un Web Service

POST <http://www.myservice.com/getApplePrices>

```
POST /item HTTP/1.1
Host: 189.123.255.239
Content-Type: text/plain
Content-Length: 200
```

```
<?xml version="1.0"?>

<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
  soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

  <soap:Body>
    <m:GetPrice xmlns:m="https://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>
</soap:Envelope>
```

Web API *non* è (solo) JSON

POST <http://www.myservice.com/getGlossaryTerms>

```
{
  "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "SortAs": "SGML",
          "GlossTerm": "Standard Generalized Markup
Language",
          "Acronym": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language, used to
create markup languages such as DocBook."
          },
          "GlossSeeAlso": ["GML", "XML"]
        },
        "GlossSee": "markup"
      }
    }
  }
}
```

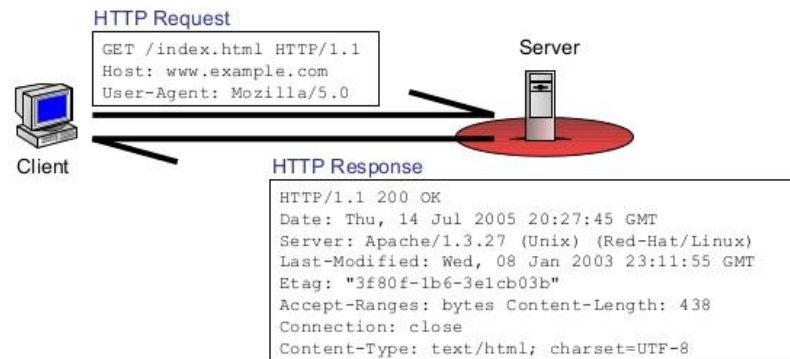
```
<!DOCTYPE glossary PUBLIC "-//OASIS//DTD DocBook
V3.1//EN">
<glossary><title>example glossary</title>
  <GlossDiv><title>S</title>
    <GlossList>
      <GlossEntry ID="SGML" SortAs="SGML">
        <GlossTerm>Standard Generalized Markup
Language</GlossTerm>
        <Acronym>SGML</Acronym>
        <Abbrev>ISO 8879:1986</Abbrev>
        <GlossDef>
          <para>A meta-markup language, used to
create markup
          languages such as DocBook.</para>
          <GlossSeeAlso OtherTerm="GML">
            <GlossSeeAlso OtherTerm="XML">
              </GlossDef>
            <GlossSee OtherTerm="markup">
              </GlossEntry>
            </GlossList>
          </GlossDiv>
        </glossary>
```

Il protocollo HTTP

- HTTP è un application layer protocol di prima classe
- Nasce per lo scambio di una singola informazione (a differenza di altri protocolli)
- E' facilmente leggibile poiché richiesta/risposta sono in formato testuale
- Su tale protocollo si fonda il servizio più usato nel mondo: il Web!

Sample HTTP exchange

- Browser = client, Web server = server



Accesso alle risorse

- HTTP non è un protocollo RPC (Remote Procedure Call)
- HTTP usa gli URI per identificare le risorse

`http://theagency.com:8080/agents?id=1`

Schema Host Port Resource Query

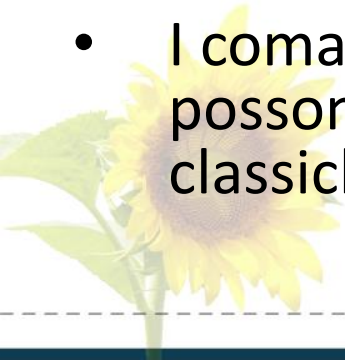


Verbi HTTP

- HTTP definisce un insieme di verbi che introducono il concetto di "azione" su una risorsa
- Nel pacchetto della richiesta, il verbo precede l'URI della risorsa
- I comandi più usati si possono mappare alle classiche operazioni CRUD

GET
POST
PUT
DELETE
PATCH

HEAD
OPTIONS
TRACE
CONNECT



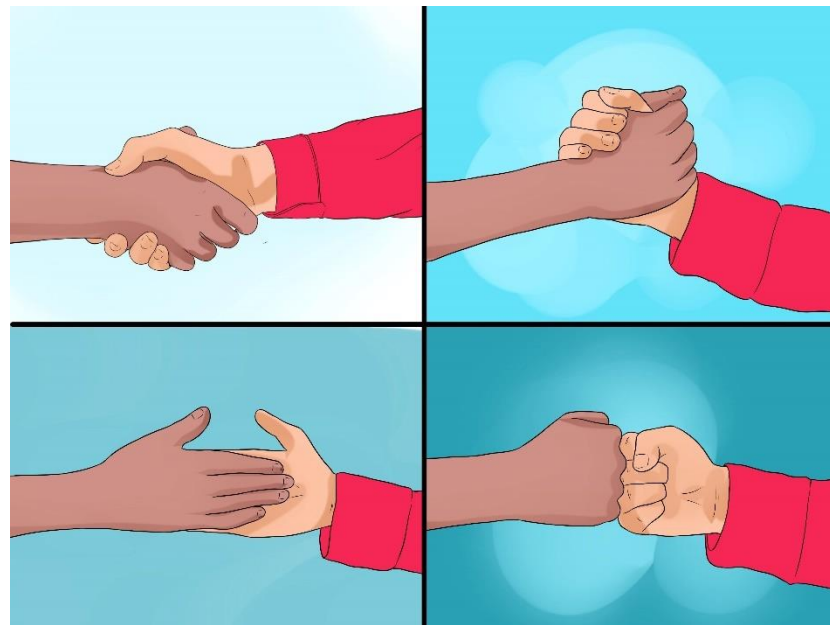
Status Code

- I codici di stato (*Status Code*) descrivono il risultato dell'azione nel tentativo (da parte del server) di soddisfare la richiesta
- Vengono restituiti nel pacchetto di risposta assieme a una descrizione e a un'eventuale contenuto
- Sono suddivisi per tipologia

1xx	<i>Informational</i>
2xx	<i>Success</i>
3xx	<i>Redirection</i>
4xx	<i>Client Error</i>
5xx	<i>Server Error</i>

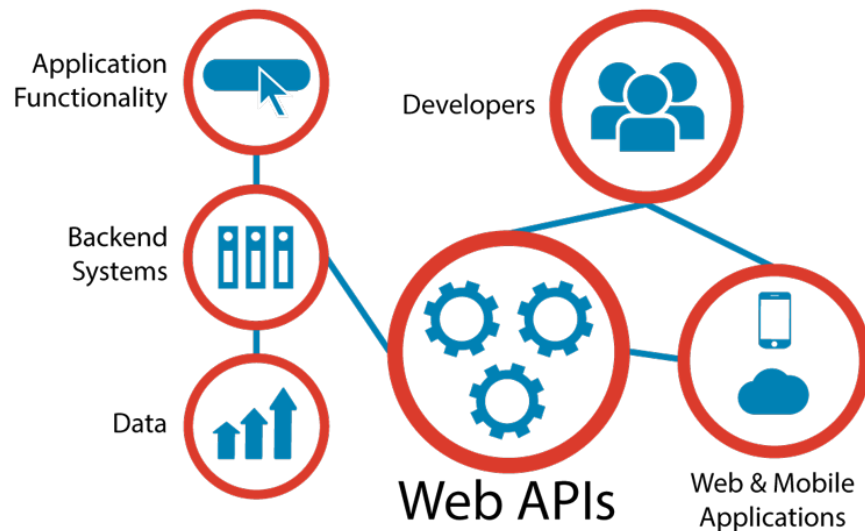
Content Negotiation

- HTTP è nato per trasferire ipertesti, ovvero documenti HTML
- Le pagine HTML contengono riferimenti a risorse esterne (immagini, video, ecc.)
- HTTP consente al client di specificare nella richiesta il formato (o i formati) attesi nella risposta usando lo standard MIME
- HTTP supporta diverse intestazioni (*Accept-**) e diverse modalità (es. *querystring*) per selezionare le caratteristiche della risposta



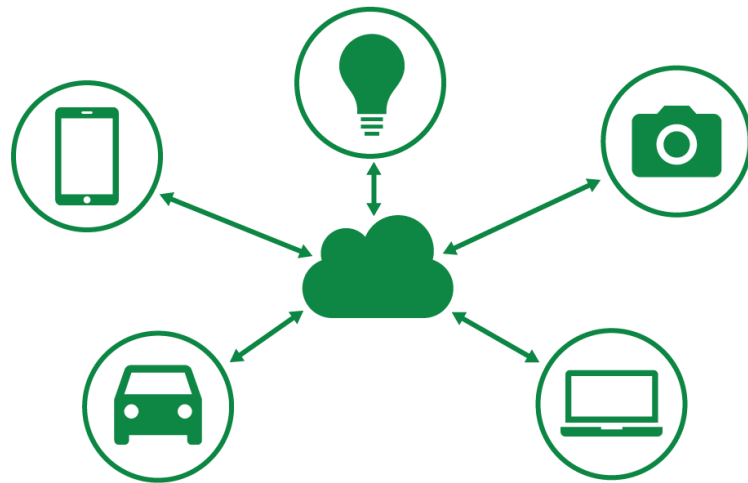
Quindi una Web API...

- Si basa sul protocollo HTTP e ne sfrutta le caratteristiche
- Espone dati sottoforma di risorse accessibili tramite URI
- Consente di eseguire operazioni sui dati tramite i verbi HTTP
- Può essere interrogata tramite AJAX dalle pagine Web e dalle SPA (Single Page Application)



Le Web API sono accessibili da

- Applicazioni desktop
- Applicazioni mobile
- Web API e applicazioni Web (proprie o anche di terze parti)
- Qualsiasi cosa che parli HTTP(S)





JSON: JavaScript Object Notation

- Rappresenta lo standard di fatto per la serializzazione di dati da/al server
- Viene spesso preferito all'XML perché leggero e compatto
- Struttura e tantissimi client sul sito ufficiale:
<http://json.org/>

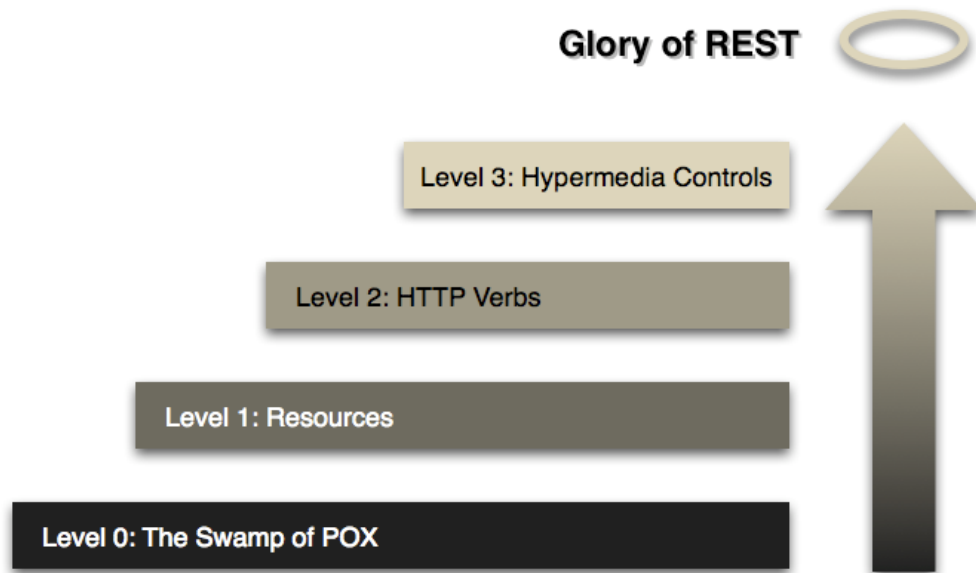


Quanto è REST una Web API?

Richardson Maturity Model

- Level 0 - Uso di HTTP
- Level 1 - Risorse
- Level 2 - Verbi HTTP
- Level 3 - Hypermedia Controls

<https://martinfowler.com/articles/richardsonMaturityModel.html>



The European Delphi Conference

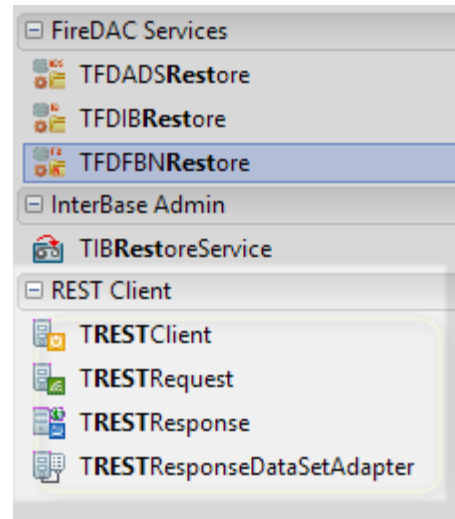


Consumo di Web API in Delphi



REST Client Components

Un set di componenti orientati al colloquio con Web API REST.



TRESTClient

*Implementa la
connessione all'endpoint
REST per l'invio delle
richieste.*



- Specificare valori per gli header HTTP
- Aggancia i sistemi di autenticazione richiesti
- Fornisce l'URL di base da utilizzare per tutte le richieste da inviare
- Recupera e salva le risposte ricevute
- Consente di configurare il proxy

TRESTRequest

Implementa la specifica richiesta HTTP da inviare all'endpoint.

- Replica molte delle proprietà del RESTClient
- Supporta la valorizzazione dei parametri da sostituire nell'URL
- Si può associare al componente TRESTResponse in cui memorizzare la risposta ricevuta
- E' possibile inviare richieste anche a designtime!



TRESTResponse

Memorizza il contenuto della risposta ricevuta a fronte della richiesta.

- Mantiene l'ultima risposta ricevuta dall'endpoint
- Indica il Content-Type comunicato dal server per indicare il formato dei dati.



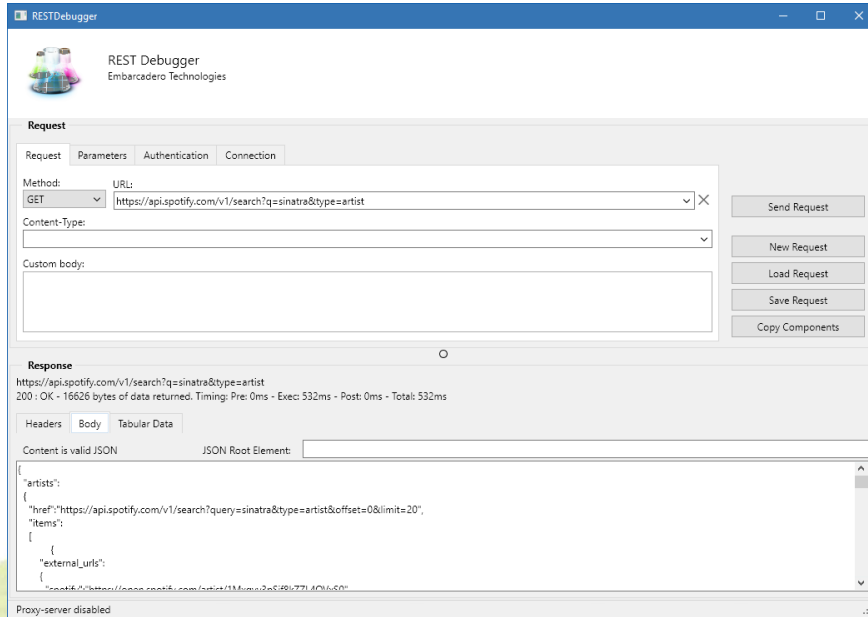
TRESTResponseDataSetAdapter

Converte la risposta ricevuta dalla REST API in una tipica struttura tabellare.



- Esegue il parsing del pacchetto JSON ricevuto
- Converte i dati JSON nei record di un DataSet
- Appiattisce all'occorrenza gli oggetti JSON per creare campi semplici
- Ideale per vedere dati live e reali a designtime (grazie a FDMemTable)

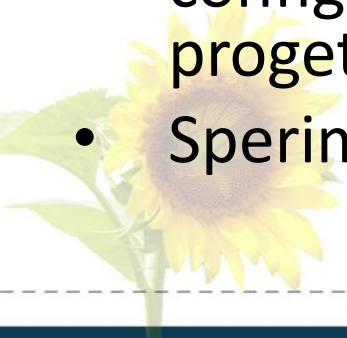
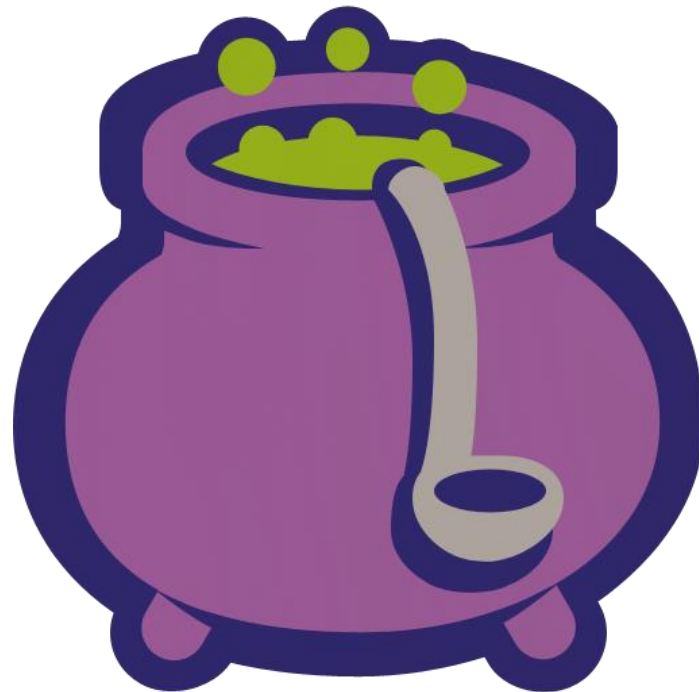
REST Debugger (companion tool)



- Consente di testare Web API, proprie o di altri
- Consente di definire il contenuto della richiesta, header compresi
- Visualizza la risposta ricevuta in formato JSON
- Utilizza i componenti REST Client per convertire JSON in un DataSet da visualizzare
- E' possibile copiare componenti e loro proprietà da incollare nell'IDE!

Ricetta per costruire un prototipo

- Leggere la documentazione
- Testare le chiamate all'API con REST Debugger
- Incollare la configurazione in un progetto Delphi
- Sperimentare



The European Delphi Conference



API!



Spotify



Spotify®

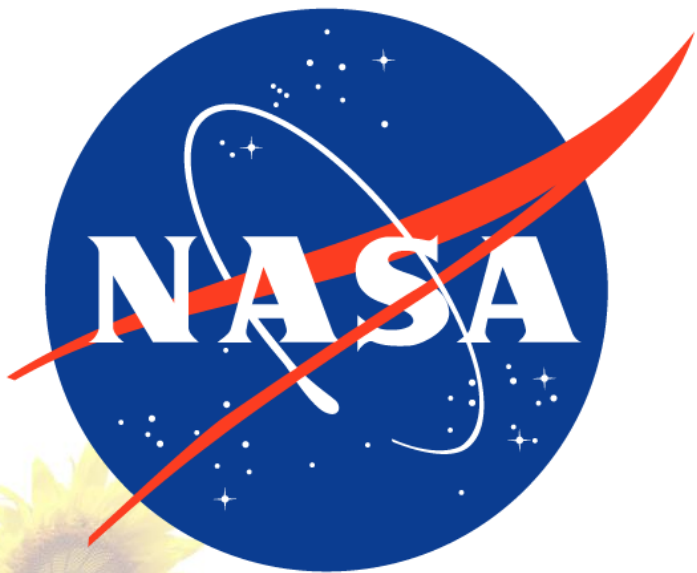
URL

<https://api.spotify.com>

- Consente di accedere alla base dati degli artisti, degli album e dei brani
- Consente di gestire i propri dispositivi



NASA Open Data



URL

<https://api.nasa.gov>

Vedi anche

<http://blog.marcocantu.com/blog/2017-january-fun-delphi-contest-nasa-api.html>

Media Wiki



MEDIAWIKI

URL

[https://en.wikipedia.org
/api/rest v1](https://en.wikipedia.org/api/rest_v1)

WordPress



URL

<http://v2.wp-api.org/>

Guida (reference):

<http://v2.wp-api.org/reference>

...e ancora

- 🌻 Google Maps (<http://maps.google.it>)
- 🌻 iTunes API
(<https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api/>)
- 🌻 JSON Placeholder (<https://jsonplaceholder.typicode.com/>)
- 🌻 HTTP Status... Cats (<https://http.cat/>)
- 🌻 REST Trading API (<https://labs.ig.com/rest-trading-api-guide>)

...e tante altre!



**KEEP
CALM
AND
ASK
QUESTIONS**



Thanks!

