

Applicazioni Web, HTTP e REST

Matteo Vaccari

<http://matteo.vaccari.name/>

Milano XP User Group, 3 ottobre 2007



sourcesense

Applicazioni Web?

Applicazione Web: un'applicazione client-server in cui il client è un semplice web browser

Perché Web Applications?

- Portabilità
- Niente da installare sul client
- Centralizzazione

Le applicazioni Web sono oggi la *maniera standard* di sviluppare un'applicazione

La ragnatela globale

Tim Berners-Lee inventa il World Wide Web circa 1989

Iper testo: testo con riferimenti (link) ad altri testi

Prima del web:

- formati di documenti incompatibili
- sistemi operativi incompatibili
- documenti inaccessibili

Con il web: documenti accessibili attraverso

- diversi sistemi operativi
- diverse applicazioni
- diverse reti

Il Web è client-server

Client: *browser o script*

- open a TCP connection
- send HTTP request
- receive response; parse; present/use

Server: *web server + docs, scripts*

- listen on port 80 TCP
- answer concurrent requests
- serve *documents* or execute *scripts*

Il Web è basato su *standard*

promulgati dal W3 Consortium
(<http://www.w3.org>)

HTTP

URI

HTML



RFC2616

The Hypertext Transfer Protocol (HTTP) is an **application-level** protocol for **distributed**, collaborative, hypermedia information systems. It is a generic, **stateless**, protocol ...
A feature of HTTP is the **typing** and **negotiation** of data representation,...

HTTP 0.9

Client request	Server response
GET /hello.txt	Hello, world!

HTTP 1.1

Client request	Server response
<pre>GET /hello.txt HTTP/1.1 host: www.example.com</pre>	<pre>HTTP/1.1 200 OK Content-Type: text/plain Hello, world!</pre>

URI

Uniform **Resource** Identifier

Il *nome* di una *risorsa*

Che cos'è una *risorsa*?

- La versione 22.1 del software rilasciato
- L'ultima versione del software rilasciato
- Il primo post del blog del 2 ottobre 2006
- Il post dedicato a “RESTful Web Services”
- Una mappa di Varese
- Informazioni sui facoceri
- Il primo numero primo > 1024
- I primi 5 numeri primi > 1024
- La lista dei banchi aperti

Esempi da Richardson, Ruby *RESTful Web Services*

Esempi di URI

- <http://www.example.com/software/releases/22.1.tgz>
- <http://www.example.com/software/releases/latest.tgz>
- <http://www.example.com/blog/2007/10/2/0>
- <http://www.example.com/blog/restful-web-services>
- <http://www.example.com/maps/IT/VA>
- <http://www.example.com/search?q=facoceri>
- <http://www.example.com/nextprime/1024>
- <http://www.example.com/next-5-primes/1024>
- <http://www.example.com/bugs/by-state/open>

Esempi da Richardson, Ruby *RESTful Web Services*

Addressability

Un'applicazione Web è *indirizzabile* se *espone* gli aspetti interessanti dei suoi dati come *risorse*

URI

scheme:*scheme-specific-part*

mailto:vaccari@pobox.com

http://www.example.com/**pages**/hello.txt

telnet://192.0.2.16:80/

news:rec.arts.int-fiction

http uri

http://<host><path>

http://www.example.com/

scheme: http

host: www.example.com

path: /

http uri

http://<host><path>

http://www.example.com/books/list.html

scheme: http

host: www.example.com

path: /books/list.html

http uri

http://<host><path>[<query>]

http://www.example.com/search?q=facoceri&start=10

scheme: http

host: www.example.com

path: /search

query: q=facoceri&start=10

RFC2616

The Hypertext Transfer Protocol (HTTP) is an **application-level** protocol for **distributed**, collaborative, hypermedia information systems. It is a generic, **stateless**, protocol ...
A feature of HTTP is the **typing** and **negotiation** of data representation,...

Application-level

HTTP	<i>Application layer</i>
TCP	<i>Transport layer</i>
IP	<i>Network layer</i>
Network-specific link	<i>Data link layer</i>

Application-level

HTTP	<i>Application layer</i>	SMTP
TCP	<i>Transport layer</i>	TCP
IP	<i>Network layer</i>	IP
Network-specific	<i>Data link layer</i>	Network-specific

Stateless

<http://www.google.com/search?q=facoceri&start=15>

http://www.google.com/search?q=facoceri&start=15

Web Images Video News Maps Gmail more ▾

matteo.vaccari@gmail.com | [My Notebooks](#) | [My Account](#) | [Sign out](#)

Google™

facoceri

Search

[Advanced Search](#)
[Preferences](#)

New! [View and manage your web history](#)

Web Results **16 - 25** of about **12,400** for **facoceri**. (0.03 seconds)

[7Cervelli.com - Dialetto Perugino](#) - [[Translate this page](#)]

... ce l'metti pe l'alito???? sinnò co sta manata de robba stendi i **facoceri!** ... ce l'metti pe l'alito???? sinnò co sta manata de robba stendi i **facoceri!** ...

[www.7cervelli.com/modules.php?file=viewtopic&name=Forums&t=1409](#) - 89k -

[Cached](#) - [Similar pages](#) - [Note this](#)

[IZEYE's WORLD](#) - [[Translate this page](#)]

Tracked from **facoceri** 07/04/08 20:11 x. 제목: **facoceri. facoceri facoceri facoceri facoceri**
[http://craigiwa.prohosts.org/img/styles/facoceri.htm](#) **facoceri** ...

[imtl.skku.ac.kr/~izeye/tt/index.php?pl=260&ct1=1](#) - 428k - [Cached](#) - [Similar pages](#) - [Note this](#)

[Rayman Raving Rabbids PS2 recensione su Yahoo! Giochi - Guida ai ...](#) - [[Translate this page](#)]

La demenzialità dei vari minigiochi è quindi dominante: si passa dalle corse in groppa ai **facoceri** alla caccia al maiale sul dorso di un pipistrello, ...

[it.videogames.games.yahoo.com/ps2/recensioni/rayman-raving-rabbids/f4a6a5.html](#) - 23k -

[Cached](#) - [Similar pages](#) - [Note this](#)

Http Messages

Client request		Server response
GET /hello.txt HTTP/1.1	<i>Start line</i>	HTTP/1.1 200 OK
Host: www.example.com Accept: text/* Accept-language: en, it	<i>Headers</i>	Content-type: text/plain Content-length: 14
	<i>Body</i>	Hello, world!

GET /books/list.html HTTP/1.1

Host: localhost:3001

User-Agent: Mozilla/5.0 (Intel Mac OS X; en-US; rv:1.8.0.7) Gecko

Accept: text/xml,application/xml,text/html;q=0.9,*/*

Accept-Language: en-us,en

Accept-Encoding: gzip,deflate

Accept-Charset: ISO-8859-1,utf-8

HTTP/1.1 200 OK

Date: Fri, 29 Sep 2006 17:00:03 GMT

Content-Type: text/html; charset=UTF-8

Server: WEBrick/1.3.1 (Ruby/1.8.4/2005-12-24)

Content-Length: 5552

<html>

<head>

<title>La lista dei libri</title>

</head>

<body>

....

Esercizio

Osservare richieste e risposte http con
netcat

Status codes

1xx	informational
2xx	success
3xx	redirection
4xx	client error
5xx	server error

Per esempio

200 OK

303 See Other

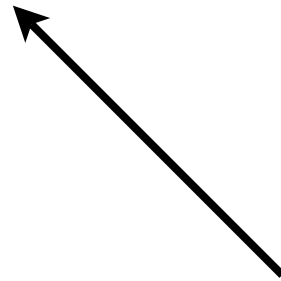
410 Unauthorized

404 Not Found

500 Internal Server Error

Request line

GET /foo/bar.html HTTP/1.1



method

Http methods

GET Get a document from the server.

POST Send data to the server for processing.

HEAD Get just the headers for a document from the server.

PUT Store the body of the request on the server.

DELETE Remove a document from the server.

Safe methods

Un metodo è *sicuro* (safe) se non provoca effetti visibili sullo stato dell'applicazione

GET vs. POST

GET	POST
safe	not safe
bookmarkable	not bookmarkable
query string nella URI	query string nel BODY

Get vs Post

Get: appropriato per *leggere* informazioni

Post: appropriato per *modificare* informazioni

Esercizio:

Fabbricare una *form html* ed osservare la richiesta effettuata tramite *netcat*

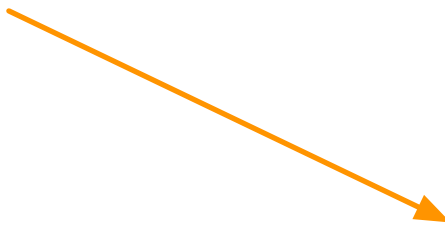
Redirections

301 Moved Permanently

302 Found

303 See Other

GET /books/new



200 OK

...

<form action='/books/new'>

</form>

GET /books/new

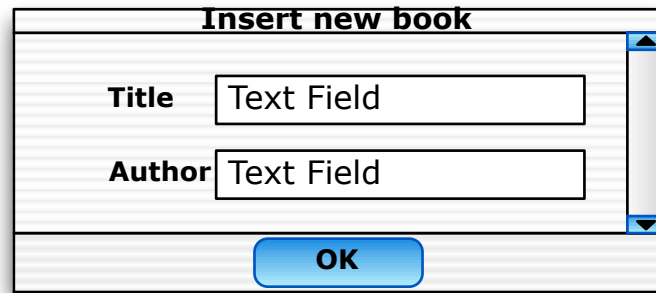
The image shows a graphical user interface for a web form. The form has a title bar that says "Insert new book". Below the title bar, there are two text input fields. The first field is labeled "Title" and the second is labeled "Author". Both fields contain the text "Text Field". At the bottom of the form, there is a blue button with the text "OK".

200 OK

...

<form action='/books/new'>
</form>

GET /books/new



Insert new book

Title Text Field

Author Text Field

OK

200 OK

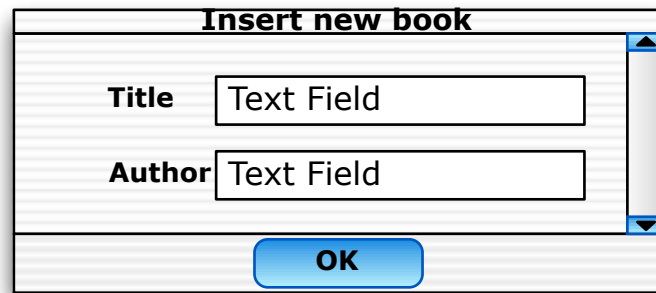
...
<form action='/books/new'>
</form>

POST /books/new

...

title=Ragione e sentimento
author=Jane Austen

GET /books/new



The image shows a web browser window with a form titled "Insert new book". The form contains two text input fields: "Title" and "Author". Below the fields is a blue button labeled "OK".

200 OK

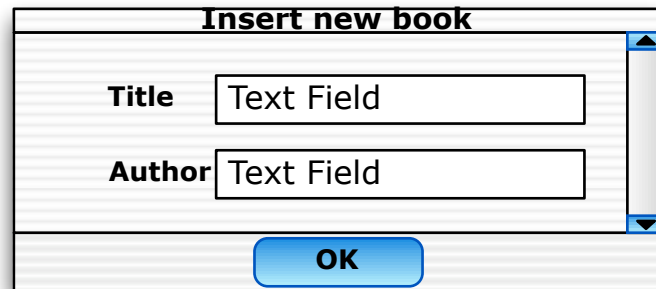
...
<form action='/books/new'>
</form>

POST /books/new

...
title=Ragione e sentimento
author=Jane Austen

303 See other
Location: /books/show/1234

GET /books/new



The image shows a web browser window with a form titled "Insert new book". The form contains two text input fields: "Title" and "Author". Below the fields is a blue button labeled "OK".

200 OK

...
<form action='/books/new'>
</form>

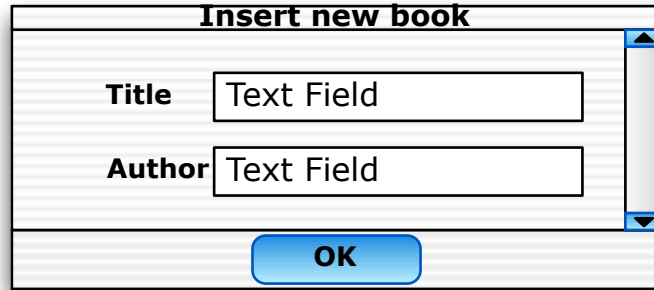
POST /books/new

...
title=Ragione e sentimento
author=Jane Austen

303 See other
Location: /books/show/1234

GET /books/show/1234

GET /books/new



The screenshot shows a web browser window with the title "Insert new book". Inside the window, there are two text input fields: one labeled "Title" and one labeled "Author", both containing the text "Text Field". Below the input fields is a blue button labeled "OK".

200 OK

...
<form action='/books/new'>
</form>

POST /books/new

...

title=Ragione e sentimento
author=Jane Austen

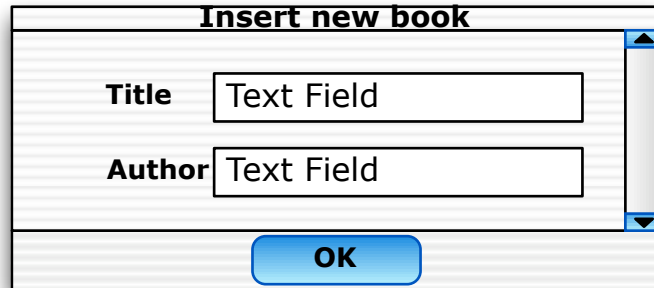
303 See other
Location: /books/show/1234

GET /books/show/1234

200 OK

...
<html>...</html>

GET /books/new



A screenshot of a web browser window showing a form titled "Insert new book". The form contains two text input fields: "Title" and "Author". Below the fields is a blue "OK" button. The browser window has a scrollbar on the right side.

200 OK

...
<form action='/books/new'>
</form>

POST /books/new

...
title=Ragione e sentimento
author=Jane Austen

303 See other
Location: /books/show/1234

GET /books/show/1234



200 OK

...
<html>...</html>

Mime types

text/plain

text/html

image/gif

<primary type>/<subtype>

image/jpeg

application/pdf

Mime types

text/html; charset=UTF-8

<primary type>/<subtype>; <parameters>

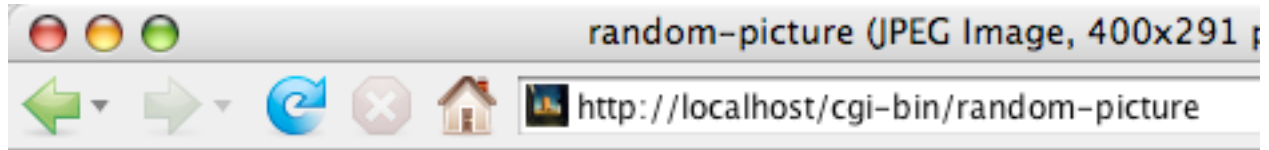
Il browser non usa il “suffisso”

```
file /var/www/cgi-bin/random-picture
```

```
#!/bin/bash
```

```
file=/tmp/g.de_chirico_la_torre_rossa.jpeg  
echo "Content-type: image/jpeg"  
echo ""  
cat $file
```

```
http://localhost/random-picture
```



Esercizio

1. Installare lo script di prima e *farlo funzionare*
2. Modificarlo per fargli restituire un immagine diversa ogni volta

Roy Fielding



rfc2616

Hypertext Transfer
Protocol -- HTTP/1.1

World-Wide-Web

HTTP

(X)HTML

URL

REpresentational State Transfer

UNIVERSITY OF CALIFORNIA, IRVINE

*Architectural Styles and
the Design of Network-based Software Architectures*

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Roy Thomas Fielding

2000

Nomi

<mailto:vaccari@pobox.com>

<http://matteo.vaccari.name/blog>

<http://matteo.vaccari.name/blog/123>

<http://matteo.vaccari.name/blog/2007-05>

GET [/blog](#) HTTP/1.1
Host: [matteo.vaccari.name](#)
Accept: [text/html](#)

Matteo Vaccari

 Search

Maximum programmer FUN!

May 25th, 2007

Finalmente in edicola il mio articoletto introduttivo a Ruby on Rails su [Linux & C.](#)! Chiaro che non c'è nulla di sconvolgentemente nuovo, ma è conciso e in italiano. E sono particolarmente contento di pubblicare su Linux & C., perché è un'ottima rivista.



Posted in [Uncategorized](#) | [Edit](#) | [3 Comments](#) »

Think first, code later

May 4th, 2007

About

I am an XP coach at [Sourcesense](#). I am a contract teacher at the [Università dell'Insubria](#). I do [consulting](#) work on software development. I'm a fan of [Extreme Programming](#). I helped organize the [European Summer School on Agile Programming](#)

Teaching

- » [Sistemi Operativi](#)
- » [Applicazioni Web](#)

Archives

- » [May 2007](#)
- » [April 2007](#)
- » [March 2007](#)
- » [February 2007](#)
- » [January 2007](#)
- » [December 2006](#)
- » [November 2006](#)
- » [October 2006](#)
- » [September 2006](#)
- » [August 2006](#)
- » [July 2006](#)
- » [June 2006](#)
- » [May 2006](#)
- » [April 2006](#)
- » [March 2006](#)
- » [February 2006](#)
- » [January 2006](#)

Categories

- » [Uncategorized](#) (69)

Blogroll

- » [Agile Day 2006](#)
- » [essap](#)
- » [PiratPartiet.it](#)

GET /blog HTTP/1.1

Host: matteo.vaccari.name

Accept: application/rss+xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generator="wordpress/2.0.2" -->
<rss version="2.0"
  xmlns:content="http://purl.org/rss/1.0/modules/content/"
  xmlns:wfw="http://wellformedweb.org/CommentAPI/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  >

<channel>
  <title>Matteo Vaccari</title>
  <link>http://matteo.vaccari.name/blog</link>
  <description>Extreme enthusiasm</description>
  <pubDate>Fri, 25 May 2007 07:57:02 +0000</pubDate>
  <generator>http://wordpress.org/?v=2.0.2</generator>
  <language>en</language>
  <item>
    <title>Maximum programmer FUN!</title>
    <link>http://matteo.vaccari.name/blog/?p=83</link>
    <comments>http://matteo.vaccari.name/blog/?p=83#comments</comments>
    <pubDate>Fri, 25 May 2007 07:52:21 +0000</pubDate>
    <dc:creator>matteo</dc:creator>

    <category>Uncategorized</category>
    <guid isPermaLink="false">http://matteo.vaccari.name/blog/?p=83</guid>
    <description><![CDATA[Finalmente in edicola il mio articolo introduttivo
a Ruby on Rails su Linux &#038; C.! Chiaro che non c&#8217;è nulla di
sconvolgentemente nuovo, ma è conciso e in italiano. E sono particolarmente
contento di pubblicare su Linux &#038; C., perché è un&#8217;ottima rivista.
```

Verbi

- GET
- POST
- PUT
- DELETE
- find
- create
- update
- destroy
- SELECT
- INSERT
- UPDATE
- DELETE

Tanti standard obsoleti

RMI

CORBA

DCOM

SOAP

WS-*

OSF DCE

RPC

```
s = Student.find(11245)  
s.addCourse("Applicazioni Web", 2007-05-28, "30/30")
```


REST

s = Student.find(11245)

c = Course.find("Applicazioni Web")

Vote.create(
 :date => 2007-05-28, :vote => "30/30",
 :student => **s**, :course => **c**)