
API REST

capirle

progettarle

provarle

documentarle

Loris Tissino • 26/10/2019 • Linux Day Pordenone
@loristissino

API

Application Programming Interface

- una persona chiede, un'altra risponde
 - una persona chiede, un programma risponde
 - un programma chiede, un altro programma risponde
-

API

Application Programming Interface

- ~~una persona chiede, un'altra risponde~~
 - ~~una persona chiede, un programma risponde~~
 - un programma chiede, un altro programma risponde
-

API

Esempi

- “Sistema operativo, forniscimi l'elenco dei file in */tmp/images*”
 - “Server aziendale, dimmi se l'utente *pippo* è autorizzato a...”
 - “Twitter, mandami la lista degli ultimi 20 tweet di *@pippo*”
-

REST

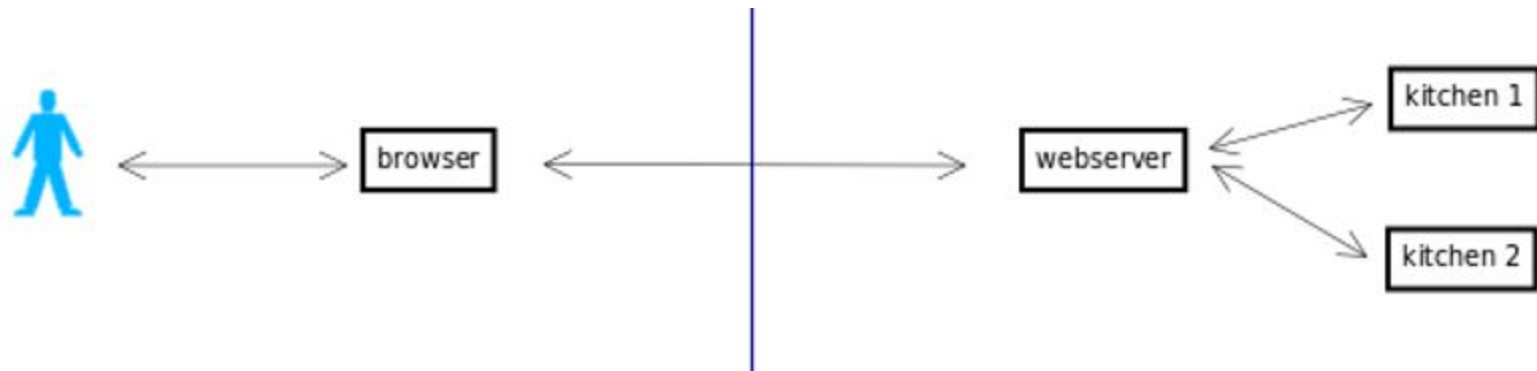
Representational State Transfer

Architettura software per la gestione di risorse in sistemi di rete

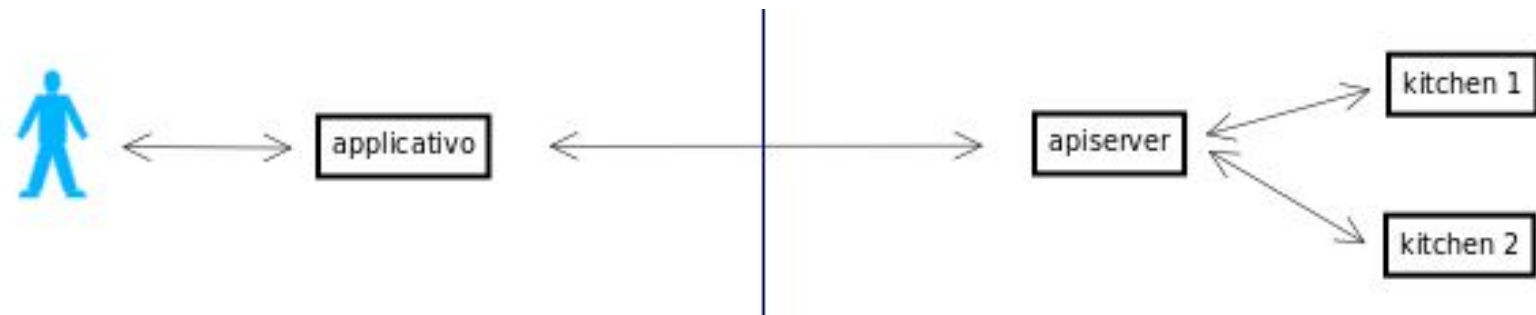
Definita nella tesi di dottorato di Roy Fielding (2000)

Al centro: concetti di *risorsa* e di sua *rappresentazione*

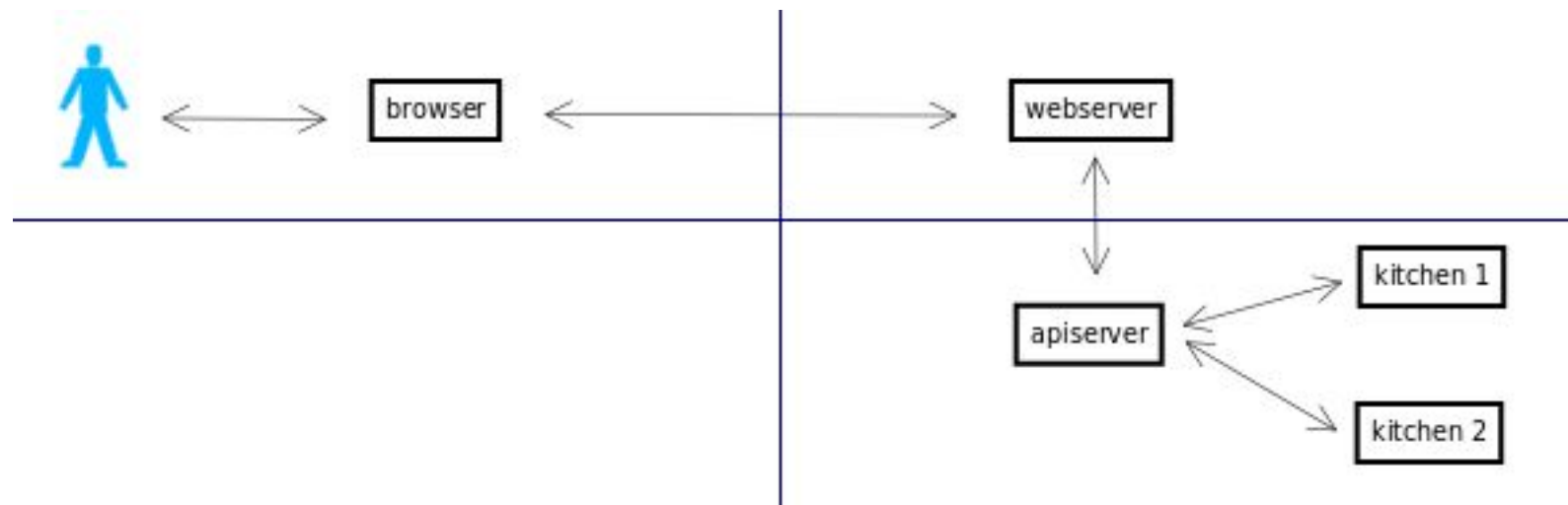
HTTP



HTTP



HTTP



HTTP

```
GET /kitchens/1/cookers/4 HTTP/1.1  
Host: kitchen.example.com  
User-Agent: simpleBrowser v.1
```

HTTP - API versioning

```
GET /v1/kitchens/1/cookers/4 HTTP/1.1  
Host: kitchen.example.com  
User-Agent: simpleBrowser v.1
```

HTTP - tipo di rappresentazione

```
GET /v1/kitchens/1/cookers/4.html HTTP/1.1  
Host: kitchen.example.com  
User-Agent: simpleBrowser v.1
```

HTTP - risorsa

```
GET /v1/kitchens/1/cookers/4.html HTTP/1.1  
Host: kitchen.example.com  
User-Agent: simpleBrowser v.1
```

HTTP - risorsa

```
GET /v2/kitchens/1/cookers/4.json HTTP/1.1  
Host: kitchen.example.com  
User-Agent: simpleBrowser v.1
```

HTTP - URI ed Endpoint

```
GET /v1/kitchens/1/cookers/4 HTTP/1.1
Host: kitchen.example.com
User-Agent: simpleBrowser v.1
```

URI

```
https://kitchen.example.com/v1/kitchens/1/cookers/4
```

ENDPOINT

```
https://kitchen.example.com/v1
```

HTTP - tipo di rappresentazione

```
GET /v2/kitchens/1/cookers/4 HTTP/1.1  
Host: kitchen.example.com  
Accept: application/json  
User-Agent: simpleBrowser v.1
```

HTTP - POST

```
POST /v2/kitchens/1/cookers HTTP/1.1
```

```
Host: kitchen.example.com
```

```
Content-Type: application/json
```

```
Accept: application/json
```

```
User-Agent: simpleBrowser v.1
```

```
{ "size": 20, "type": "induction", "status": "off" }
```

HTTP - PUT

```
PUT /v2/kitchens/1/cookers/8 HTTP/1.1
```

```
Host: kitchen.example.com
```

```
Accept: application/json
```

```
User-Agent: simpleBrowser v.1
```

```
{ "size": 18, "type": "gas", "status": "off" }
```

HTTP - PATCH

```
PATCH /v2/kitchens/1/cookers/8 HTTP/1.1
Host: kitchen.example.com
Accept: application/json
Content-Type: application/json
User-Agent: simpleBrowser v.1
```

```
{ "status": "off" }
```

HTTP - DELETE

```
DELETE /v2/kitchens/1/cookers/8 HTTP/1.1  
Host: kitchen.example.com  
Accept: application/json  
User-Agent: simpleBrowser v.1
```

HTTP - Risposte

HTTP/1.1 200 OK

Date: Thu, 24 Oct 2019 19:30:21 GMT

Content-Type: application/json

Server: Apache/2.4.29

Content-Length: 61

```
{ "id": 8, "size": 20, "type": "induction", "status":  
"off" }
```

Codici HTTP? → evertpot.com

HTTP series

This article is part of a series about the HTTP protocol. Read them all here:

- [Series of posts on HTTP status codes](#) (2018-06-27)

Informational 1xx

- [100 Continue](#) (2018-06-27)
- [101 Switching Protocols](#) (2018-06-28)
- [102 Processing](#) (2018-06-29)
- [103 Early Hints](#) (2018-06-30)

Successful 2xx

- [200 OK](#) (2018-07-03)
- [201 Created](#) (2018-07-10)

07-17

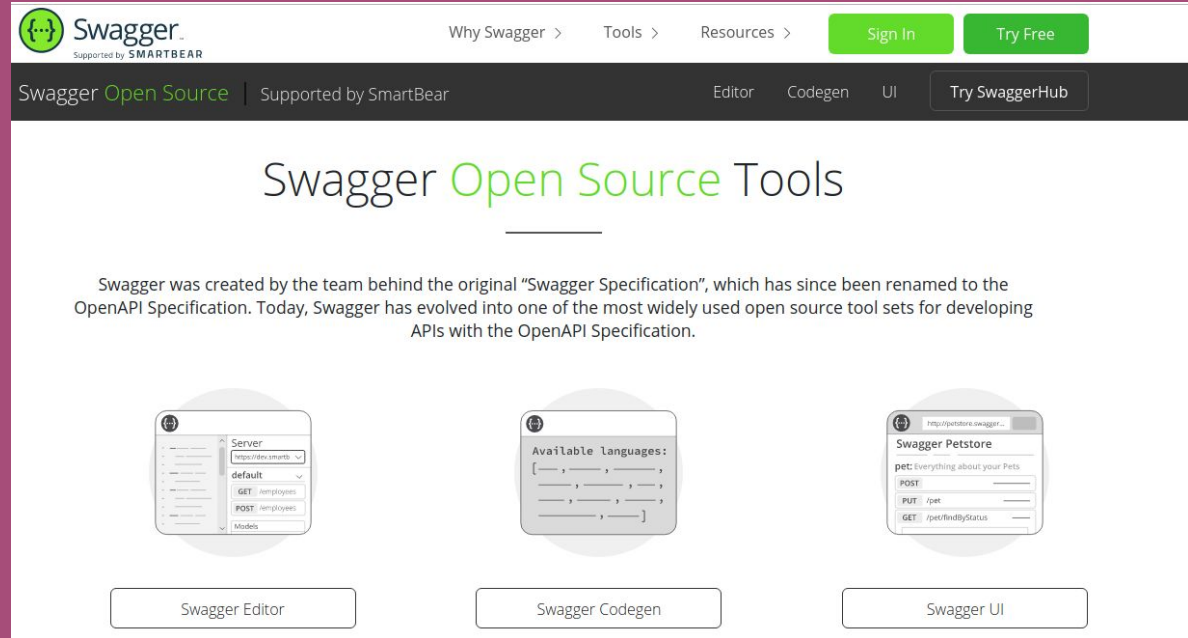
REST - Princìpi

- L'URI identifica risorse, non azioni
 - URI diversi possono identificare la medesima risorsa
 - Risorsa \neq Rappresentazione
 - Negoziazione del contenuto
 - Interfaccia uniforme
 - Codici HTTP di risposta significativi
 - Nessuna gestione di stato lato server
 - Possibilità di *caching*
 - *Hypermedia as the engine of application state* (HATEOAS)
-

OpenAPI → www.openapis.org




Swagger → swagger.io




The screenshot shows the Swagger.io website. At the top, there is a navigation bar with the Swagger logo (a green circle with a curly brace and dots) and the text "Swagger Supported by SMARTBEAR". To the right of the logo are links for "Why Swagger >", "Tools >", and "Resources >". Further right are two green buttons: "Sign In" and "Try Free". Below the navigation bar is a dark grey bar with "Swagger Open Source" in green, "Supported by SmartBear", and links for "Editor", "Codegen", "UI", and a "Try SwaggerHub" button.

Swagger Open Source Tools


Swagger was created by the team behind the original "Swagger Specification", which has since been renamed to the OpenAPI Specification. Today, Swagger has evolved into one of the most widely used open source tool sets for developing APIs with the OpenAPI Specification.



Swagger Editor

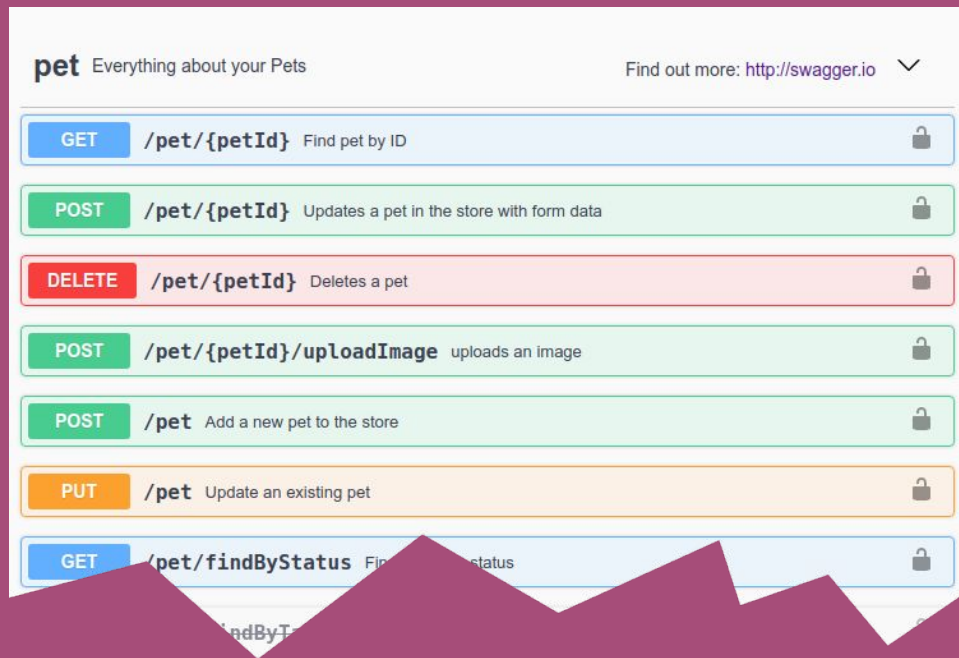


Swagger Codegen



Swagger UI

Swagger → petstore.swagger.io



The screenshot displays the Swagger UI for the petstore API. The title is "pet" with the subtitle "Everything about your Pets". A link "Find out more: http://swagger.io" is visible in the top right. The main content is a list of API endpoints, each with a colored button for the HTTP method, the endpoint path, a description, and a lock icon.

Method	Endpoint	Description
GET	/pet/{petId}	Find pet by ID
POST	/pet/{petId}	Updates a pet in the store with form data
DELETE	/pet/{petId}	Deletes a pet
POST	/pet/{petId}/uploadImage	uploads an image
POST	/pet	Add a new pet to the store
PUT	/pet	Update an existing pet
GET	/pet/findByStatus	Find pets by status

Esempio → LDE

The screenshot shows the LDE 0.9 interface. On the left is a sidebar with navigation options like 'info', 'Tags', 'Servers', 'Firms', 'accounts', and 'journal entries'. The main area is split into three panes:

- API Documentation:** Shows the 'LearnDoubleEntry.org API' with endpoints like `GET /firms` and `POST /firms/slug`.
- Terminal:** Displays a series of REST API calls and responses, including a successful `POST /firms/slug` call.
- Invoices Window:** A modal window titled 'Invoices' showing a table of transactions for the firm 'Impresa Demo Linux Day'. The table has columns for 'date', 'number', 'customer', 'amount', and 'je'.

date	number	customer	amount	je
2019-10-24	101/19	Rossi SpA	122	-1
2019-10-23	102/19	Gialli srl	244	0
2019-10-22	103/19	Neri srl	183	0
2019-10-21	104/19	Verdi SpA	1000	0
2019-10-20	105/19	Gialli srl	330	-1

Libro giornale

Visualizzazione 1-10 di 10 risultati.

S	Data	Descrizione	Conto	Dare	Avere	
	01/10/19	Inizio attività	Denaro in cassa	€ 250,00		
			Patrimonio netto		€ 250,00	
	01/10/19	Spese iniziali	Costi di impianto	€ 50,00		
			Denaro in cassa		€ 50,00	
	22/10/19	Invoice # 103/19	Crediti v/cienti @Neri srl	€ 183,00		
			IVA ns/debito		€ 33,00	
			Merci c/vendite		€ 150,00	
	23/10/19	Invoice # 102/19	Crediti v/cienti @Gialli srl	€ 244,00		
			IVA ns/debito		€ 44,00	
			Merci c/vendite		€ 200,00	

Grazie per l'attenzione

Domande?
