

# REST

by THE  
SIMPSONS



# REPRESENTATIONAL STATE TRANSFER

Roy Thomas Fielding (2000)



**R**EOURCE ORIENTED  
**E**ASY & SIMPLE  
**S**TANDARD & **S**TATELESS  
**T**ECHNOLOGY INDEPENDENT

Stateless = Sessionless  $\neq$  Cookieless

# REPRESENTATIONAL STATE TRANSFER

Roy Thomas Fielding (2000)



A "real" Web service is a service designed to embrace the "Web" from the ground up.

The Web has become the most successful distributed computing platform ever developed, thanks to years of investment and usage.

**REST** is not tied to any particular technology or platform – it's simply a way to design things to work like the Web.

# Open Letter to W3C on Web of Services for Enterprise Computing

Nick Gall (VP Gartner)

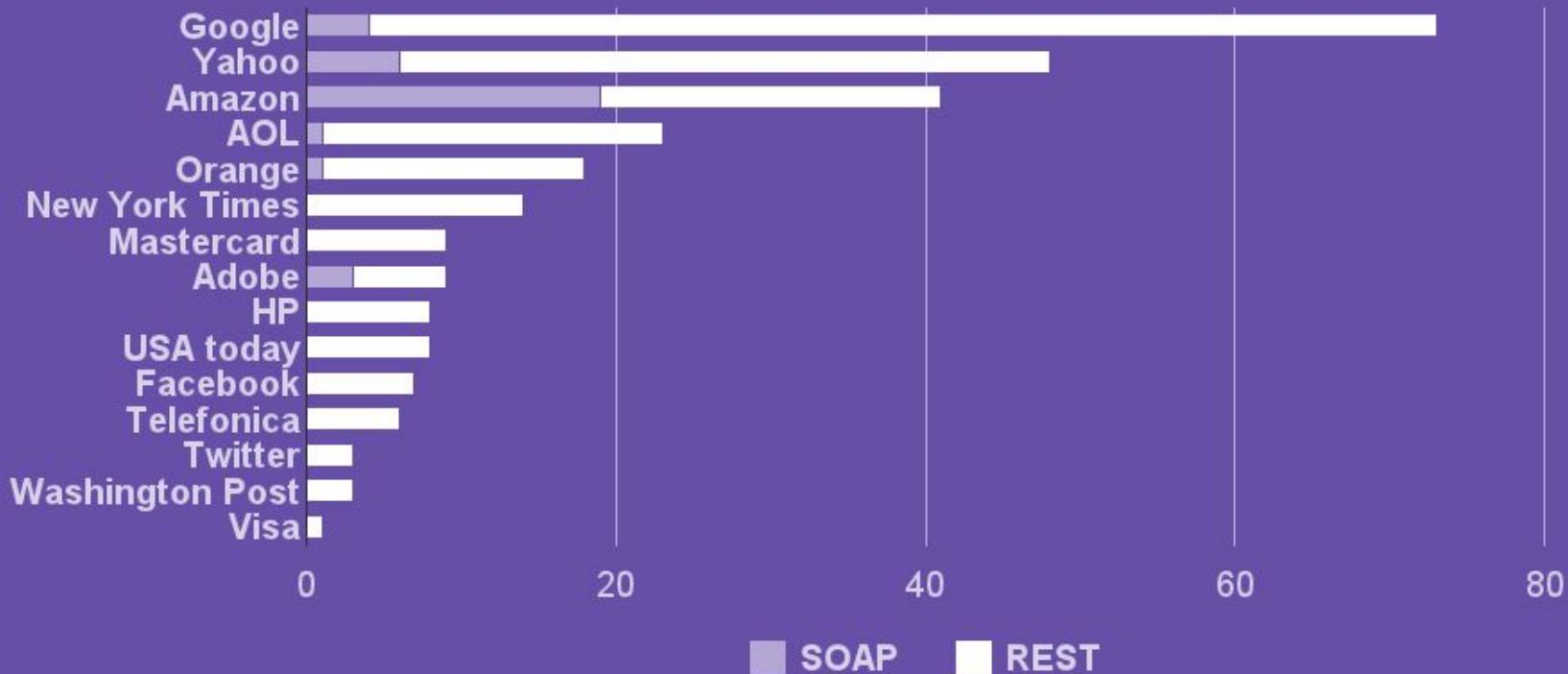
The W3C should leave the work on standardizing the WS-\* middleware architecture to the middleware \$vendors\$ and shift its focus to standardizing aspects of Web architecture that make it easier to apply to "application to application" scenarios.



Traditional middleware styles include **RPC** (Remote Procedure Call, eg **DCE**), **MOM** (Message-Oriented Middleware, eg **MQSeries**), and **DOM** (Distributed Object Model, eg **CORBA** and **DCOM**). All are in production at many major enterprises. But all have fallen far short of initial expectations. Most have resulted in complex, tightly coupled, and difficult to extend integration networks.

Unfortunately, Web Services, at least the WS-\* style, are "Web" in name only. While WS-\* enables tunneling over HTTP (used merely as an XML message transport), in almost every important aspect, WS-\* violates (or at best ignores) the architectural principles of the Web.

# REST USAGE AND THE REAL WORLD



## ¿Quién es Roy Fielding?

Un tipo inteligente.

Ayudó a hacer los primeros servidores web, se dedicó a la investigación y colaboró en la definición del protocolo sobre el que funciona la web.

edna



## ¿Cómo funciona la web?

Eh... es bastante sorprendente. Y lo gracioso es que está muy subestimada.

El protocolo de la web, HTTP, es capaz de muchas cosas que la gente ignora.

## http, ¿lo que escribo en el navegador?

Sí, es el protocolo más importante de la historia de la informática...

¡Por qué?

hawkins

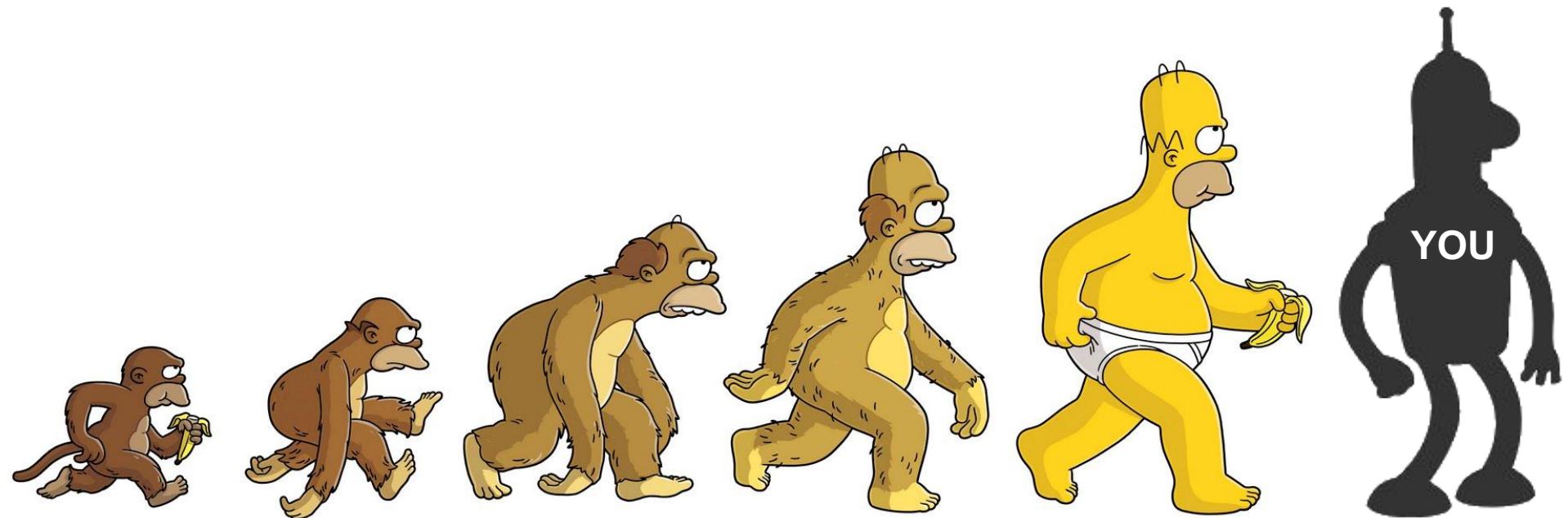


Por que no se lo dices?

Quizás lo haga...



# EVOLUTION



ONC-RPC  
1988 1989

CORBA  
1991

RMI  
EJB 97-98  
SOAP 1999  
REST 2000  
1996

JAX-RS  
2008

???  
2015

HTTP

WWW

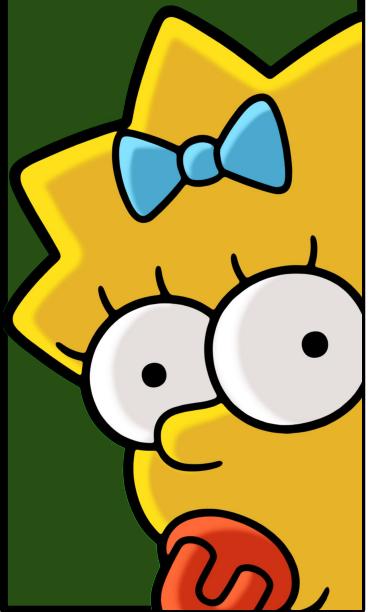
0.9

1.0

1.1

2.0

ONE  
RESOURCE  
URL



TWO  
HTTP  
METHODS



THREE  
MULTIPLE  
REPRESEN-  
TATIONS



FOUR  
RESPONSE  
CODES



FIVE  
USE THE  
HEADERS



A cartoon illustration of Maggie Simpson, a yellow baby with a blue bow tie and a red pacifier. She is looking towards the right side of the frame.

ONE  
RESOURCE URL

# ASIGNA URLs A TUS RECURSOS

<http://www.simpsons.com/characters/maggie/father>

/mother

# GOOD URL



Fácil de leer/escribir, decir y recordar

Describen el recurso

Utilizan nombres en lugar de verbos

Todos los niveles/paths de la URL son recursos

Asigna un **<id>** (mejor universal) a tus recursos

Filtrá recursos mediante **?query-params**

ID Universal: estandarizado/normalizado

<http://www.simpsons.com/characters/>



/homer  
/marge  
/bart  
/lisa  
/burns  
/moe  
/nelson  
/skinner  
/milhouse

# ALIAS vs. IDENTIFIERS

## ALIAS:VALUE

/characters/**marion-anthony**

TABLE

IDENTIFIER (PK)

/characters/**nickname:fat-**

tony TABLE

ALIAS (UI)

PK: Primary Key

UI: Unique Index



# URL PATTERNS

MANYtoONE · ONETO MANY

/characters/**maggie**/brothers/**bart**/cousins/**lin**  
M TO O            O TO M            M TO O            O TO M            M TO O  
TABLE            PK            RELATION            PK            RELATION            PK



# URL PATTERNS

## ONEToONE



abraham



# URL FILTERS

MANY?FILTER

M?F  
/characters?surname=Flanders  
TABLE WHERE



# URL FILTERS

MANY?FILTER

MTOO      OTOM      M?F  
/characters/**maggie**/brothers?  
TABLE      **sex=female**      PK      RELATION      WHERE



# URL FILTERS

Operator	Name suffix	Value prefix
equals	name=Lisa	
distinct	surname.not=Flanders	surname!=Flanders
in	age=9&age=10	age=9,10
greather	age.from=10	age=>10
less	age.to=15	age=<15
like	name.contains=a name.starts=a name.ends=a	name=*a* name=a* name=*a
order	order=name&order=-surname	order=name,-surname
paginate	from=20&size=10(&count=true)	

# <http://simpsons/characters/homer>

id: **homer**

name: Homer Jay Simpson

age: 38

parent: Abraham Simpson

mother: Mona Simpson

married: Marge Simpson

children: Bart, Lisa, Maggie

brothers: Herb, Abbie



# <http://simpsons/characters/bart>



id: bart

name: Bartholomew Simpson

age: 10

parent: Homer Simpson

mother: Marge Simpson

brothers: Lisa, Maggie

cousins: Ling

<http://simpsons/characters/abraham>

ANYtoONE RELATIONS



/mona/couple



/homер/mother/couple



/herb/father



/marge/father-inlaw



/bart/grandfather



/lisa/father/father

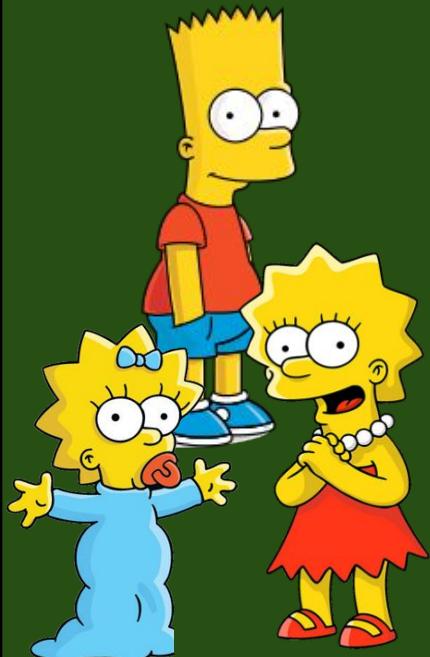


/maggie/mother/couple/father



<http://simpsons/characters/homer/children>

## ONEtoMANY RELATIONS



/marge/children



/selma/nephews



/patty/sisters/marge/children



/abraham/grandchildren



/jackeline/children-inlaw/homer/children



/ling/cousins

/characters?parent=homer

# TWO HTTP METHODS

INTERACTÚA CON  
MÉTODOS HTTP

GET /places/moe's-bar/duff



# HTTP METHODS

Method	Description	Type
GET	Obtener recurso o colección	safe & idempotent
POST	Crear sin ID (Location) Actualizar uno/más recursos	non-idempotent
PUT	Crear con ID Actualizar completamente	idempotent
DELETE	Eliminar recurso o colección	idempotent
PATCH (*)	Actualizar parcialmente	idempotent

PATCH (\*) is proposed, but at moment is not included into HTTP spec



# POST METHOD

Usa **POST** cuando no encaje ningún otro método:

**GET**: obtener/recuperar recursos

**PUT**: actualizar/reemplazar totalmente un recurso

**DELETE**: eliminar/borrar un recurso

**PATCH**: actualizar parcialmente un recurso (\*)

Casi todas actualizaciones serán parciales, salvo en ficheros/docs:

**PATCH** queda desaconsejado hasta que se incluya en HTTP

**PUT** queda “relegado” a crear recursos con URL/ID conocido

**El 99% de las acciones que actualicen tus recursos son POST**

POST can't retrieve (GET) or remove (DELETE) resources



# SQL SIMILARITIES

HTTP	SQL
GET	SELECT
POST	INSERT / UPDATE
PUT (*)	INSERT / UPDATE
DELETE	DELETE (no DROP)
<ID>	PRIMARY KEY
<ALIAS>:<VALUE>	UNIQUE INDEX
URL/LINK	TABLES/RELATIONS

PUT can INSERT with known ID or UPDATE completely (with all columns)



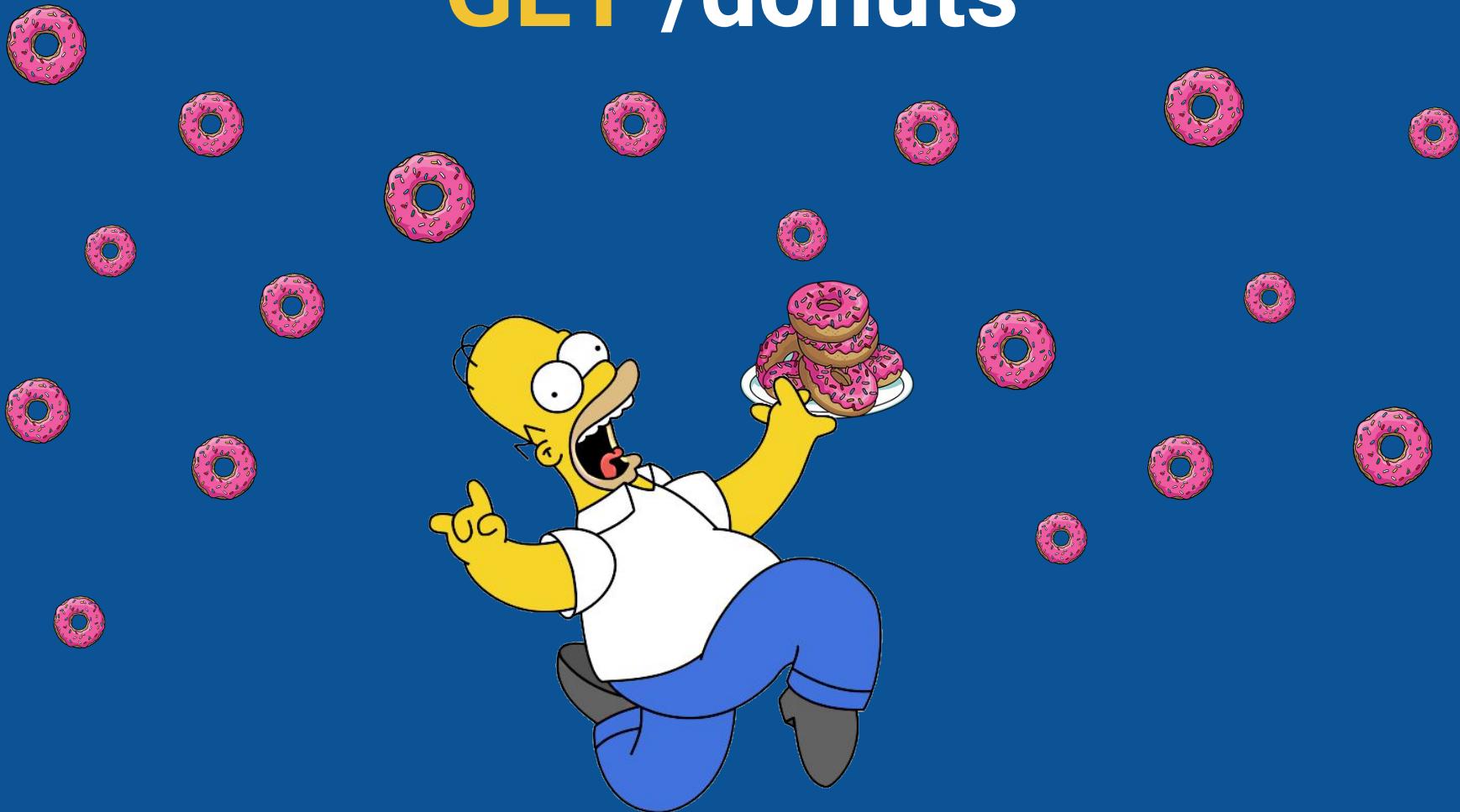
# GET /donuts/1



# DELETE /donuts/1



# GET /donuts



# DELETE /donuts



# POST /donuts



A cartoon character with orange hair and a blue suit is holding a large, round donut with white frosting and colorful sprinkles. He has a surprised or excited expression.

**PUT**  
**/super-donuts/greatwhite**

# DELETE /super-donuts/greatwhite





# THREE

## MULTIPLE REPRESENTATIONS

PERMITE MÚLTIPLES  
REPRESENTACIONES

XHTML • XML • JSON • CSV • ASCII  
PROPERTIES • SVG • PNG • MATHML

# MIME TYPES

	Mime-Type
SVG	image/svg+xml
XML	application/xml
JSON	application/json
PROPERTIES	text/plain
CSV	text/csv
TXT	text/plain
XHTML	application/xhtml+xml
MATHML	application/mathml+xml

# SVG • SCALABLE VECTOR GRAPHICS

No olvides enlazar  
unos recursos con  
otros



# XML • EXTENSIBLE MARKUP LANGUAGE

A cartoon illustration of Homer Simpson's face, showing his signature mustache and a slightly grumpy expression, positioned on the left side of the slide.

```
<?xml version="1.0" encoding="UTF-8"?>
<character xmlns:xlink="http://www.w3.org/1999/xlink"
    id="lisa"
    desc="Lisa Simpson"
    xlink:href="http://simpsons/characters/lisa">
    <name>Lisa</name>
    <surname>Simpson</surname>
    <nickname>nerd</nickname>
    <parent id="homer" desc="Homer Simpson" xlink:href="/characters/homer"/>
    <mother id="marge" desc="Marge Simpson" xlink:href="/characters/marge"/>
    <brothers xlink:href="/characters/lisa/brothers"/>
    <cousins xlink:href="/characters/lisa/cousins"/>
</character>
```

# JSON • JAVASCIPT OBJECT NOTATION

```
{ "id": "lisa", "desc": "Lisa Simpson", "href": "http://simpsons/characters/lisa",  
  "name": "Lisa",  
  "surname": "Simpson",  
  "nickname": "nerd",  
  "parent": { "id": "homer", "desc": "Homer Simpson" "href": "/characters/homer" },  
  "mother": { "id": "marge", "desc": "Marge Simpson", "href" : "/characters/marge" },  
  "brothers": { "/characters/lisa/brothers" },  
  "cousins": { "/characters/lisa/cousins" }  
}
```



# PROPERTIES

id=lisa

desc=Lisa Simpson

href=http://simpsons/characters/lisa

name=Lisa

surname=Simpson

nickname=nerd

parent.id=homer

parent.desc=Homer Simpson

parent.href=/characters/homer

mother.id=marge

mother.desc=Marge Simpson

mother.href=/characters/marge

brothers=/characters/lisa/brothers

cousins=/characters/lisa/cousins

# CSV • COMMA-SEPARATED VALUES

id,name,surname,parent.id,parent.desc,mother.desc,...

"lisa","Lisa","Simpson","homer","Homer Simpson","Marge Simpson",...

"bart","Bart","Simpson","homer","Homer Simpson","Marge Simpson",...

"maggie","Maggie","Simpson","homer","Homer Simpson","Marge Simpson",...



# TXT • ASCII-TEXT

id [31] name [31] surname [31] parent.id [31] parent.desc [31] mother.desc [30]  
lisa [31] Lisa [31] Simpson [31] homer [31] Homer Simpson [31] Marge Simpson [30]  
bart [31] Bart [31] Simpson [31] homer [31] Homer Simpson [31] Marge Simpson [30]

## ASCII DELIMITED TEXT

#	Description
28	FS: File Separator
29	GS: Group Separator
30	RS: Record (row) Separator
31	US: Unit (field) Separator

# HTML • HYPERTEXT MARKUP LANGUAGE

GET /characters/lisa

```
<form name="character" method="GET" action="/characters/lisa;edit">
    Identifier: lisa
    Name: Lisa
    Surname: Simpson
    Nickname: Nerd
    Parent: <a href="/characters/homer">Homer Simpson</a>
    Mother: <a href="/characters/marge">Marge Simpson</a>
    Relations: <a href="/characters/lisa/brothers">Brothers</a>,
               <a href="/characters/lisa/cousins">Cousins</a>
    <input type="submit" value="Edit">
</form>
```

# HTML

GET /characters/**lisa**;edit

```
<form name="character" method="POST" action="/characters/lisa">
    Identifier: <input type="text" name="id" value="lisa" />
    Name: <input type="text" name="name" value="Lisa" />
    Surname: <input type="text" name="surname" value="Simpson" />
    Nickname: <input type="text" name="nickname" value="nerd" />
    Parent: <input type="text" name="parent.id" value="homer" />
    Mother: <input type="text" name="mother.id" value="marge" />

    <input type="submit" value="Save">
</form>
```

IHTML Foems not allows PUT (neither DELETE)

# HTML

GET /characters/**new**

```
<form name="character" method="POST" action="/characters">
    Identifier: <input type="text" name="id" value="" />
    Name: <input type="text" name="name" value="" />
    Surname: <input type="text" name="surname" value="" />
    Nickname: <input type="text" name="nick" value="" />
    Parent: <input type="text" name="parent.id" value="" />
    Mother: <input type="text" name="mother.id" value="" />

    <input type="submit" value="Create">
</form>
```

# MATHML • MATHEMATICAL MARKUP LANGUAGE

```
<math xmlns="http://www.w3.org/1998/Math/MathML">  
  <mi>E</mi>  
  <mo>=</mo>  
  <mi>m</mi>  
  <msup>  
    <mi>c</mi>  
    <mn>2</mn>  
  </msup>  
</math>
```



# MULTIPART

HEADERS	BODY
<b>Content-Type:</b> multipart/mixed boundary=----- <b>287032381131322</b> <b>Content-Length:</b> 5540	next rows
----- <b>287032381131322</b> <b>Content-Disposition:</b> name="homer"; filename="homer.gif" <b>Content-Type:</b> image/gif	
----- <b>287032381131322</b> <b>Content-Disposition:</b> name="bart"; filename="bart.svg" <b>Content-Type:</b> image/svg+xml	
----- <b>287032381131322</b> <b>Content-Disposition:</b> name="maggie"; filename="maggie.ico" <b>Content-Type:</b> image/x-icon ----- <b>287032381131322--</b>	



# FOUR RESPONSE CODES

## RESPONDE APROPIADAMENTE

Siempre hay que ser educado

# RESPONSE CODES

Code	Type
1XX	Informational
2XX	Success
3XX	Redirection
4XX	Client Error
5XX	Server Error

# SUCESS

#	Response	Description	GET	POST	PUT	DEL
200	Ok	Successful (*)	X	X	X	X
201	Created	Successful creation (*)		X	X	
202	Accepted	Request received		X	X	X
204	No Content	Successful without content	X		X	X

(\*) Return body contains created Resource

# REDIRECTION

#	Response	Description	GET
301/308	Moved Permanently	Resource has moved (*)	X
302/307	Temporary Redirect	Moved Temporary (*)	X
303	See Other	Redirection (*)	X
304	Not Modified	Resource not modified	X

(\*) Return Location Header to other Resource

# ERROR

#	Response	Description	GET	POST	PUT	DEL
<b>400</b>	Bad Request	Invalid request	X	X	X	X
401	Unauthorized	Authentication needed	X	X	X	X
403	Forbidden	Authentication failure	X	X	X	X
<b>404</b>	Not Found	Resource not found	X	X	X	X
<b>405</b>	Not Allowed	Method not allowed (*)	X	X	X	X
406	Not Acceptable	Representation not available	X			
408	Timeout	Request timeout	X	X		
<b>409</b>	Conflict	State not permit request		X	X	X
415	Unsupported Type	Unsupported representation		X	X	

(\*) Method not supported for concrete resource

# SERVER ERROR

#	Response	Description	GET	POST	PUT	DEL
500	Server Error	Internal server error	X	X	X	X
501	Not Implemented	Operation not supported (*)		X	X	X
503	Service Unavailable	Temporarily unavailable	X	X	X	X

(\*) Method not supported for any resource

A cartoon illustration of Marge Simpson from The Simpsons. She has her signature large blue afro hairstyle and is wearing a yellow tank top with a red beaded necklace. She is looking towards the right side of the frame with a neutral expression.

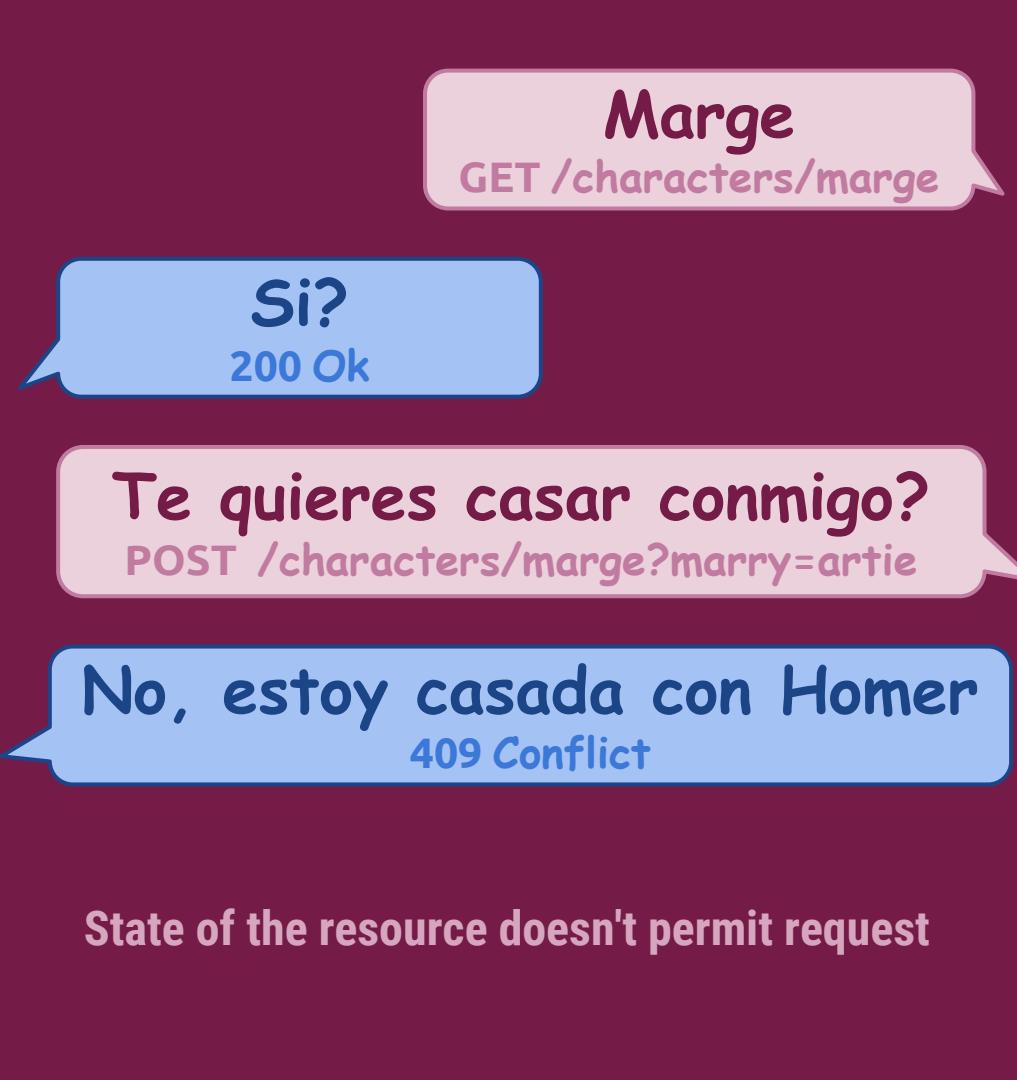
Marge

GET /characters/marge

A cartoon illustration of Artie Hargrove from The Simpsons. He has dark hair and glasses, and is wearing a pink shirt under a blue jacket. He is looking towards the left with a slight smile.

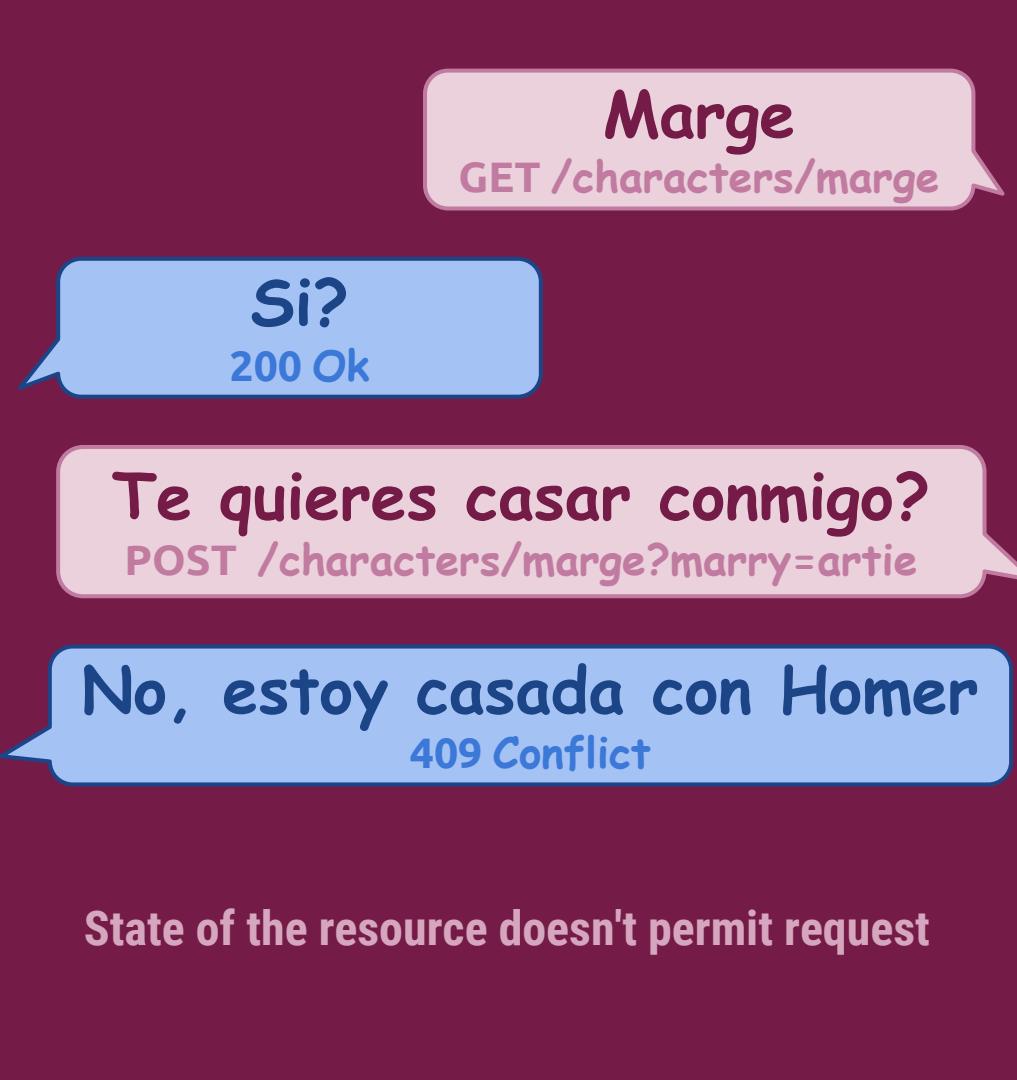
Si?

200 Ok

A cartoon illustration of Homer Simpson from The Simpsons. He has his characteristic mustache and is wearing a blue shirt. He is looking towards the left with a neutral expression.

Tequieres casar conmigo?

POST /characters/marge?marry=artie

A cartoon illustration of Homer Simpson from The Simpsons. He has his characteristic mustache and is wearing a blue shirt. He is looking towards the left with a neutral expression.

No, estoy casada con Homer

409 Conflict

State of the resource doesn't permit request

Esta "Al"?

GET /places/moes-bar?name=Al

Quien?

404 Not found

"Al" de apellido "Reves"

GET /places/moes-bar?name=Al&surname=Revés

Esta "Al Reves"?

Ja, Ja, Ja, Ja

#☠️@⚡☢️!\*

400 Bad Request



Dios creó a Adán y Eva

PUT /characters/adan

PUT /characters/eva

201 Created

Después nacieron  
Caín y Abel

PUT /characters/cain

PUT /characters/abel

201 Created

Caín mató a Abel

DELETE /characters/abel

200 Ok

Con estos hechos,  
Por que los ateos  
niegan a Dios?

DELETE /God

405 Method  
Not Allowed





Por que yo Señor?  
En que me he equivocado?  
No bebo, no fumo, ...

Puedes devolverme a mi mujer?

GET /characters/maude



...

404 Not Found



Duffman, necesito un pedido  
de 10 barriles de cerveza  
POST /requests/duff?quantity=10



Los tendrás esta  
misma tarde

202 Accepted  
(but not processed)

Pending Request: could be rejected while it's  
being processed...



Tienes más Donuts?

GET /apu-store/donut?quantity=10



En la sección de bollería

302/307 Temporary redirect



Y armas de fuego?

GET /apu-store/guns?quantity=1



No, la policía las  
confiscó...

301/308 Moved permanently

# FIVE

## USE THE HEADERS

HTTP HEADERS  
HTML CLIENTS



# REQUEST HEADERS

Header	Description
Accept	Content-Types acceptable for the response
Accept-Charset	Character sets that are acceptable
Accept-Encoding	List of acceptable encodings: gzip, deflate
Accept-Language	List of human languages for response
Authorization	Authentication credentials
Cookie	HTTP cookie previously sent by the server
Content-Type	Mime type of the request body (POST, PUT)



# RESPONSE HEADERS

Header	Description
<b>Content-Encoding</b>	Type of encoding used: gzip, deflate
<b>Content-Language</b>	The human language of content
<b>Content-Type</b>	The MIME type of this content
<b>Location</b>	In redirection, or new resource created
<b>Set-Cookie</b>	HTTP cookie
<b>WWW-Authenticate</b>	Authentication scheme to access
<b>Cache-Control</b>	Control cache mechanism



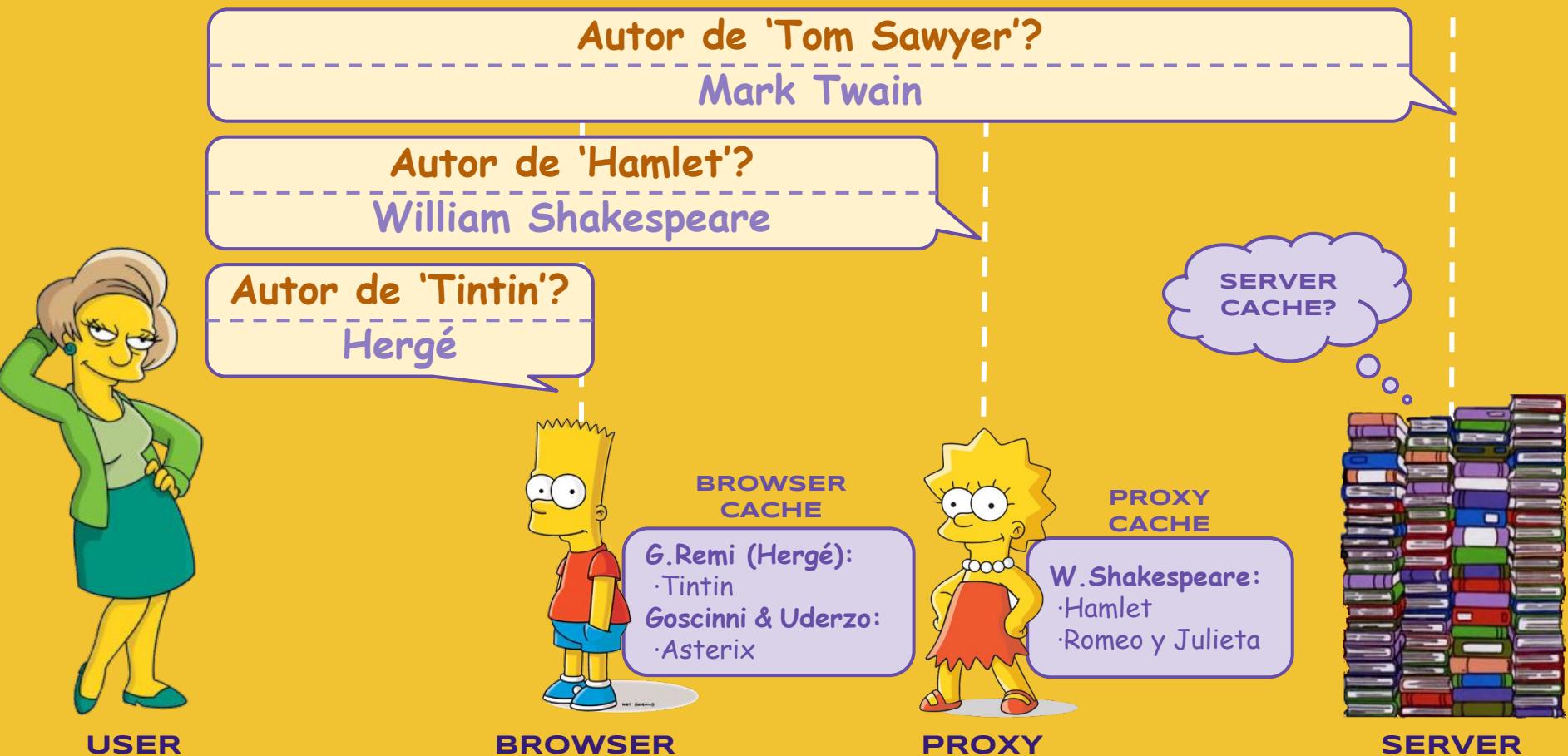
# CACHE-CONTROL HEADER

Cache-Control	Action
<b>private</b>	Enable (End)Client cache
<b>public</b>	Enable Proxy/Client cache
<b>no-cache, no-store</b>	Disable cache
<b>public, max-age=X (*)</b>	Enable cache (X <= 31536000sg/1yr)

(\*) Enable cache until <date expiration> or <version update> with headers:  
· **Expires:** Mon 4 May 2015 21:31:12 GMT (until date expiration)  
· **ETag:** 737060cd8c284d8af7ad3082f209582d (until version update)



# CACHE MECHANISMS



# LANGUAGE SUPPORT



# REDIRECT TO LOCATION



Tienes más Donuts?

GET /apu-store/donut?quantity=10

En la sección de bollería

302/307 Temporary redirect

Location: /apu-store/buns-section

Y armas de fuego?

GET /apu-store/guns?quantity=1

No, la policía las confiscó...

Pregunta en la armería

301/308 Moved permanently

Location: /weapons-store/guns



# CREATION WITHOUT ID

**POST /funzos?color=brown**  
201 Created  
Location: /funzos/1024

**POST /funzos?color=blue**  
201 Created  
Location: /funzos/1022

**POST /funzos?**

**-color=purple-**

201 Created

Location: /funzos/1020



**POST /funzos?color=green**  
201 Created  
Location: /funzos/1023

**POST /funzos?color=red**  
201 Created  
Location: /funzos/1021

# CREATE WITH IDENTIFIER



# BASIC SECURITY



Toc, toc...

<No header Authorization>

Quien es?

401 Unauthorized

WWW-Authenticate: Basic

policia:abra

Authorization: Basic cG9saWNpYTphYnJh

Lo siento, sólo socios

403 Forbidden



# COMPRESSION



Si fuera pequeño las  
rosquillas durarían más  
GET /characters/homer  
Accept-Encoding: gzip

Yuuuuuhuuuuuuuu!! Tengo  
rosquillas para 2 semanas  
Content-Encoding: gzip



# CONDITIONAL REQUESTS

Header	Description
ETag	<u>Version</u> of a resource, often a message digest
If-Match (PUT)	Optimistic Locking: Only update a resource if 'etag' has not been modified else return 412 (Precondition Failed)
If-None-Match (GET)	Allows a 304 (Not Modified) if 'etag' is unchanged
Last-Modified	The last modified <u>date</u> for the requested object
If-Unmodified-Since (PUT)	Optimistic Locking: Only update a resource if 'date' has not been modified else return 412 (Precondition Failed)
If-Modified-Since (GET)	Allows a 304 (Not Modified) if 'date' is unchanged

ETag: 737060cd8c284d8af7ad3082f209582d

Last-Modified: Tue, 12 May 2015 12:45:26 GMT



# CONDITIONAL GET

Tienes un dolar?  
GET /milhouse/money

Si  
200 Ok  
ETag: 258df1f625

Last-Modified: Tue, 12 May 2015 12:45:26 GMT

Que si tienes un dolar!

GET /milhouse/money  
If-None-Match: 258df1f625

If-Modified-Since: Tue, 12 May 2015 12:45:26 GMT

Ya te he dicho que Si  
304 Not Modified (No body)

Grrr.. Dame todo tu dinero!

DELETE /milhouse/money  
If-None-Match: 258df1f625

If-Modified-Since: Tue, 12 May 2015 12:45:26 GMT



# CONDITIONAL PUT

Tienes novio?

GET /lisa

No tengo

200 Ok

ETag: 12df1f5

Last-Modified: Tue, 12  
May 2015 12:45:26 GMT

Quieres ser mi novia?

PUT /lisa?boyfriend=milhouse

If-Match: 12df1f5

If-Unmodified-Since: Tue,  
12 May 2015 12:45:26 GMT

Vale

200 Ok

Quieres ser mi novia?

PUT /lisa?boyfriend=nelson

If-Match: 12df1f5

If-Unmodified-Since: Tue,  
12 May 2015 12:45:26 GMT

Ya tengo novio

412 Precondition Failed



ONE

## ASIGNA URLs A TUS RECURSOS

<http://simpsons/characters/bart/donations>



TWO

## INTERACCIONA CON MÉTODOS HTTP

POST <http://simpsons/characters/bart/donations>



THREE

## PERMITE MÚLTIPLES REPRESENTACIONES

```
xml: <money currency="USD">0</money>
json: {"money": {"amount": 1, "currency": "EUR"}}
form: }
```

[money=5&currency=GBP](#)



FOUR

## RESPONDE APROPIADAMENTE

Tacaño: 400 Bad Request (money is 0)  
Ladrón: 405 Method Not Allowed (DELETE)  
Gracias: 200 OK



FIVE

## USA LAS CABECERAS

Seguridad, Multidioma, Redirección  
Cache, Conditional Requests,

Formatos



*Un pesimista ve dificultad  
en cada oportunidad.*

*Un optimista ve  
oportunidad  
en cada dificultad.*

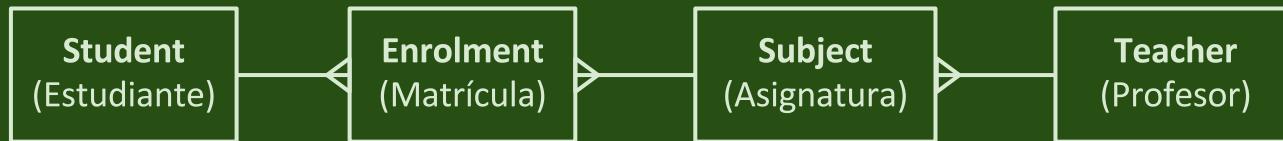
*Winston Churchill...*





/countries/**ES**/regions/madrid/cities/getafe?region.country.language=es  
/cities/**getafe**/region/country?regions...cities...areas...





Wich are the subjects and teacher of Lisa enrollments on 2015-2016?

```
/students/lisa/enrollments/2015-2016/subjects;teacher  
/subjects;teacher?enrollments.id=2015-2016&enrollments.student.id=lisa
```

Wich are the students of teacher Elsa for subject english-a1 on 2015-2016?

```
/teachers/elso/subjects/english-a1/enrollments/2015-2016/students  
/students?enrollments.id=2015-2016&enrollments.subject.id=english-a1&enrollments.subject.teacher.id=elso
```

Wich are the enrollments on 2015-2016 sorter by subject.name and student.surname (desc)?

```
/enrollments?enrollments.course=2015-2016&order=subject.name&order=-student.surname
```



Wich are the subjects and teacher of Lisa enrollments on 2015-2016?

```
/customers/john/sales/20150229000001/products  
/subjects;teacher?enrollments.id=2015-2016&enrollments.student.id=lisa
```

Wich are the students of teacher Elsa for subject english-a1 on 2015-2016?

```
/teachers/elsa/subjects/english-a1/enrollments/2015-2016/students  
/students?enrollments.id=2015-2016&enrollments.subject.id=english-a1&enrollments.subject.teacher.id=elsa
```

Wich are the enrollments on 2015-2016 sorter by subject.name and student.surname (desc)?

```
/enrollments?enrollments.course=2015-2016&order=subject.name&order=-student.surname
```

# HTTP +20 YEARS



Spec	Version	Year
CERN	0.*	1991
RFC 1945	1.0	1996
RFC 2616	1.1	1999
RFC 2774	1.2	2000
DRAFT	2.0	¿2015?

*La primera señal de que estás innovando es que la gente no te entiende...*

*Ferran Adrià*



# ALIAS vs. IDENTIFIERS

## ALIAS:VALUE

/robots/bender

TABLE



/robots/unit:22

TABLE



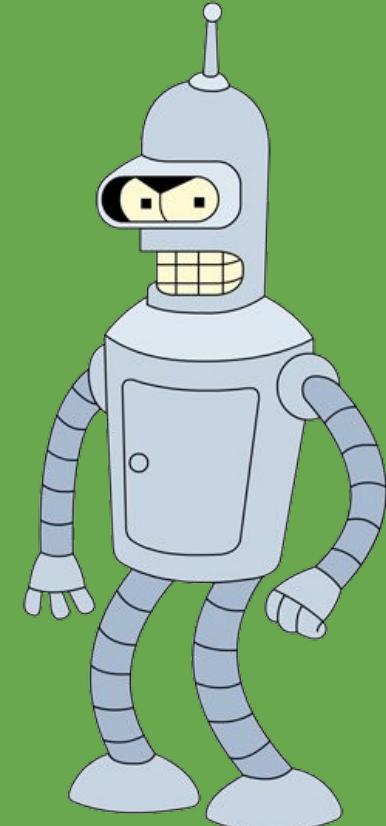
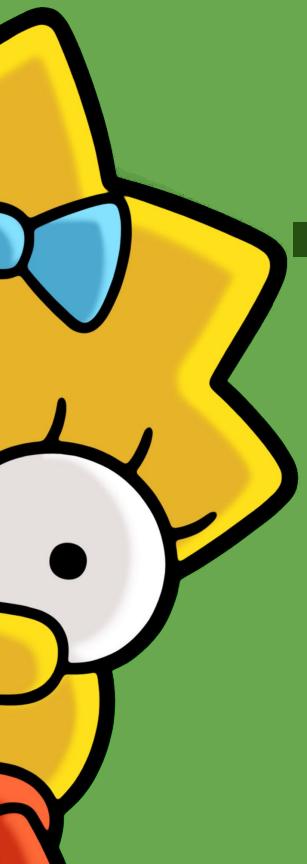
/robots/serial:2716057

TABLE



PK: Primary Key

UI: Unique Index



<http://www.simpsons.com/zeroes/>

(superheroes - superpowers = zeroes)

/burnsman



/everyman



/bartman



/falloutboy





edwina



louis



abraham



mona



jackeline



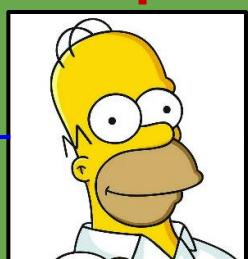
clancy



abbie



herb



homer



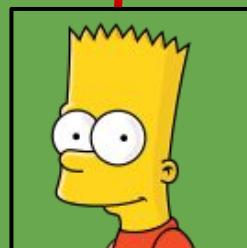
marge



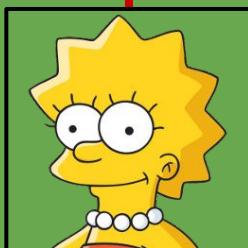
patty



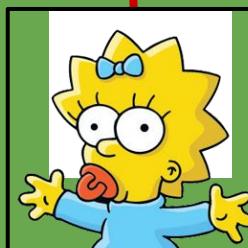
selma



bart



lisa



maggie



ling

<http://simpsons/characters/marge/sisters>



/homer/sisters-in-law



/bart/aunts



/lisa/mother/sisters



/maggie/father/sisters-in-law



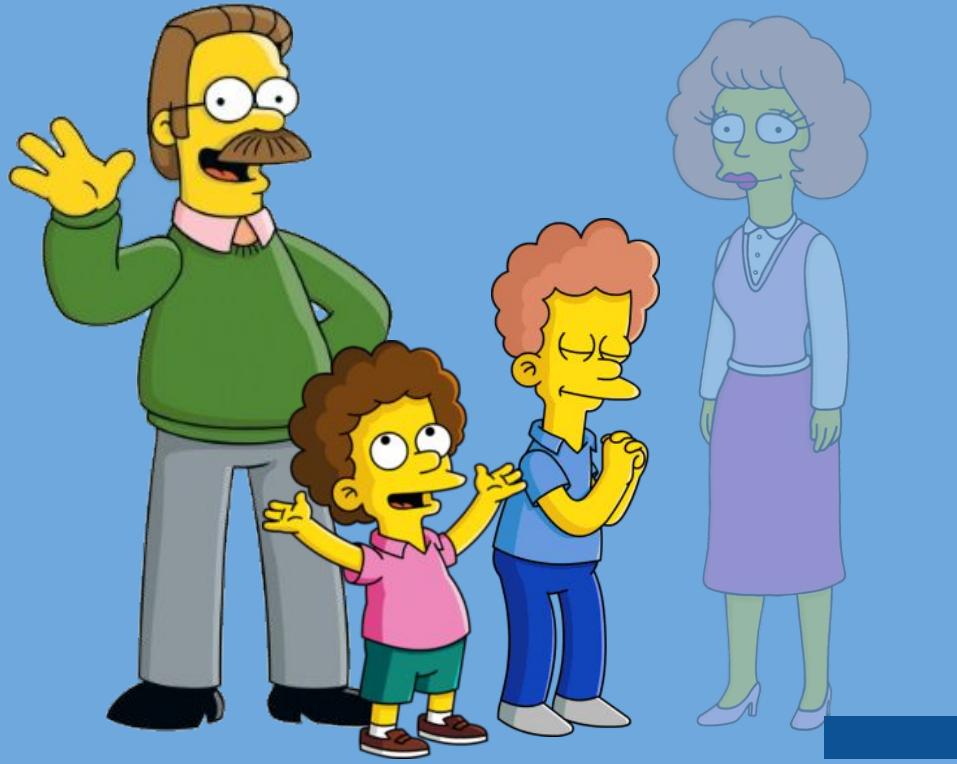
/clancy/children?surname=Bouvier



/jackeline/children?status=single



# DELETE /characters/maude



GET /simpsons/characters?surname=Simpson



# DELETE /characters/bart



Un claro caso  
de accidente  
doméstico  
(mira que caerse  
sobre el cuchillo)



# DELETER LICENSE

## SPRINGFIELD

EXP. 007



NAME: JAMES BOND

CLASS: SECRET AGENT

EXPIRES: 2017.07.07

SIGNATURE: JAMES BOND

# COMPRESSION

Vamos a China por nuestro aniversario?

Cuanto vale el billete para China? Y cada maleta?

GET /air-china/prices  
Accept-Encoding: gzip

AIR CHINA

Pasajeros: \$1200  
Maletas: \$15

Content-Encoding: gzip

Vale



# HTML CLIENTS

**form.action:** only allows GET, POST (ni PUT, ni DELETE)

**links:** can't send headers request

- mime-types (.extension)

**creation form:** GET /resources/new

**edit forms:** GET /resources/<id>;edit (;view)

**Post-Redirect-Get:** avoid re-post on reloads

**disable buttons on submit:** avoid re-submit on double click

**url aliases:** transform url

- subresources as resources & path to query params

**polimorphic urls:**

- /orders/periodic-transfers?date=today
- /orders/transfers?type=periodic&date=today



# SUMMARY

## 1. Asigna URLs a tus recursos

`http://simpsons/characters/bart`

## 2. Interacciona con métodos HTTP

`POST http://simpsons/characters/bart/donations`

## 3. Permite múltiples Representaciones

```
<money currency='USD'>0</money>
{'money':{'amount':1,'currency':'EUR'}}
money=5&currency=GBP
```

## 4. Responde apropiadamente

`400 Bad Request (money is 0)`

`405 Method Not Allowed (GET)`

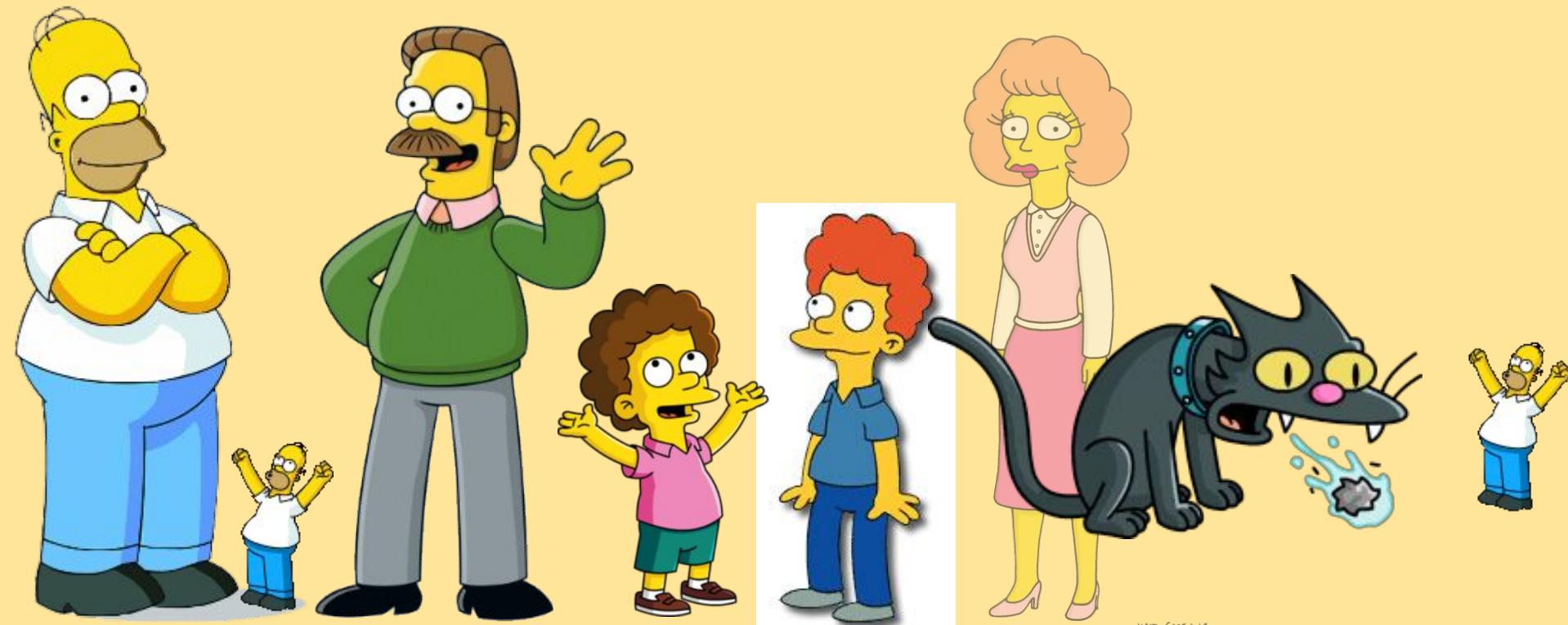
`200 OK`

## 5. “Abraza” el protocolo HTTP

Usa cabeceras HTTP: Seguridad, Multidioma, Redirección  
Cache, Conditional Requests, Formatos



GET /simpsons?surname=Flanders



MATT GROENING

**DELETE** ~~http://simpsons/rasca~~

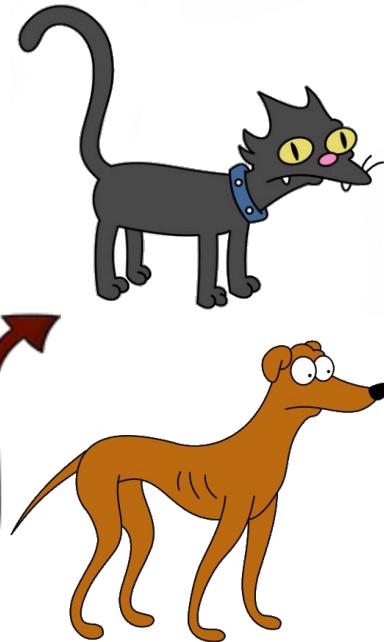




Edmund



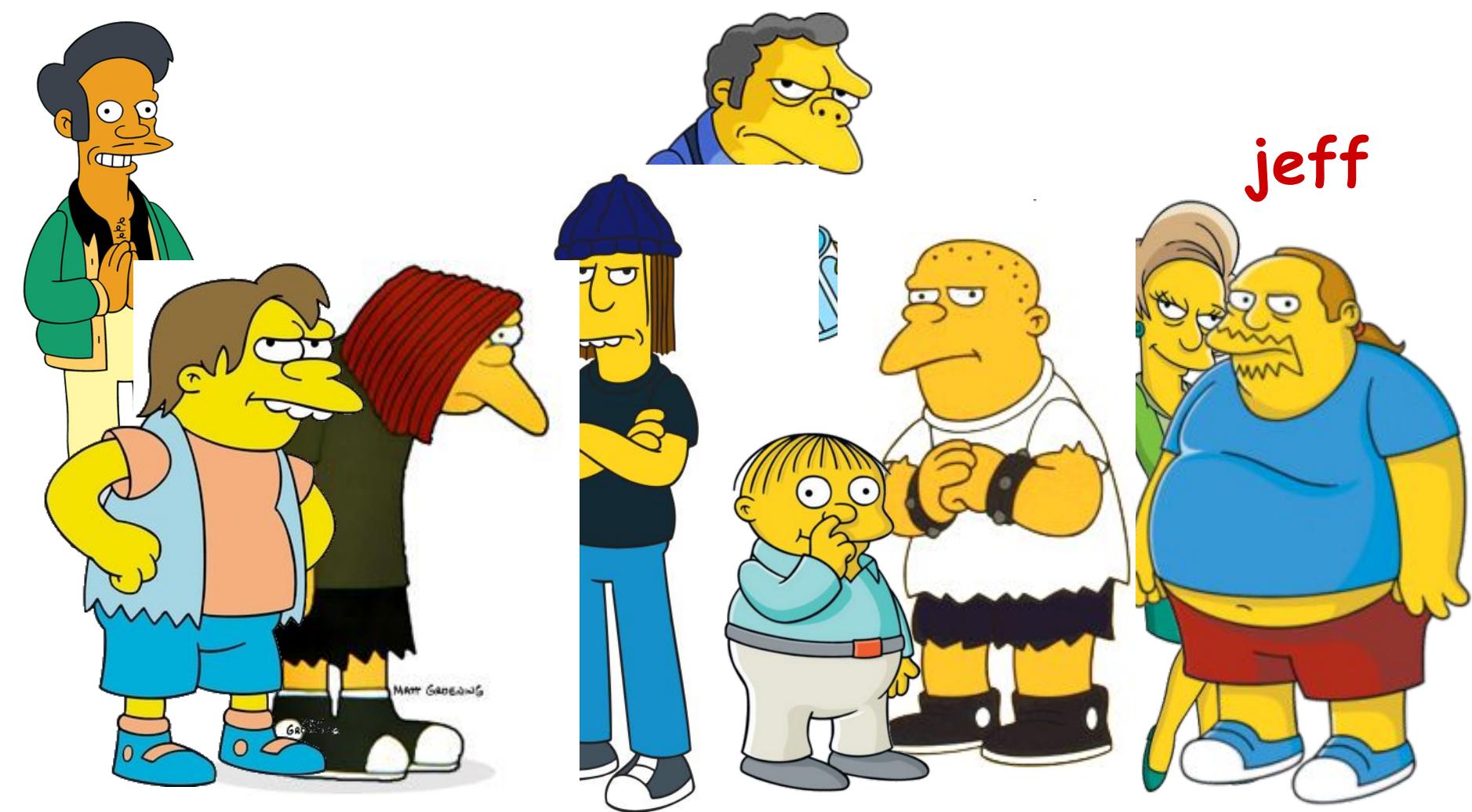
Thelonious





<http://simpsons>





# GET <http://simpsons/places/moe's>





clobber girl



the-collector



blue-falcon



radiactive-man



streich-dude



cupcake-kid



pieman  
cupcakekid

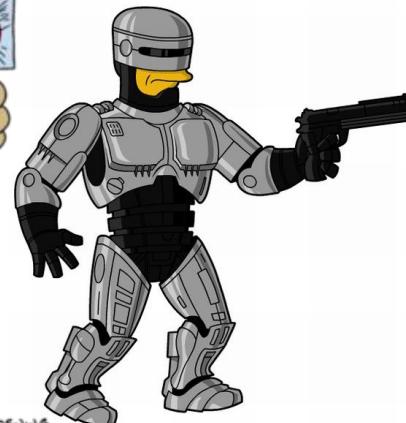


everyman

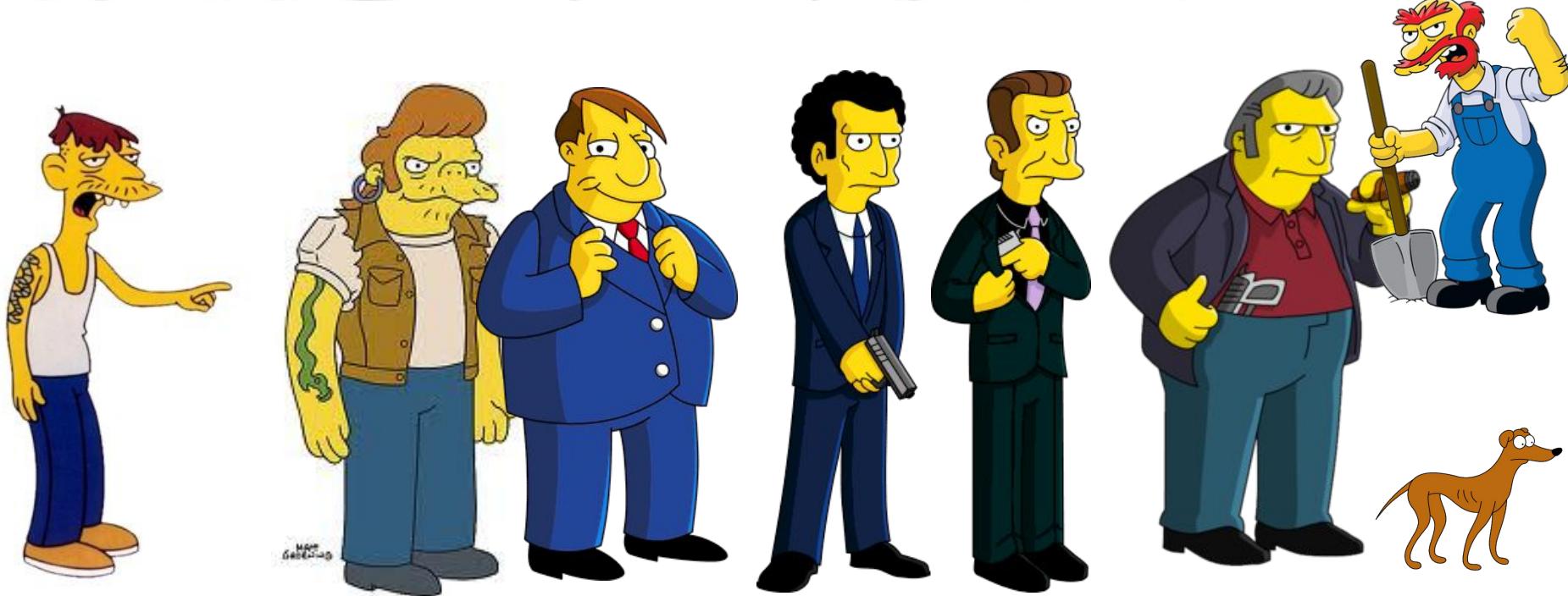


I'm the  
Goddamn  
Bartman





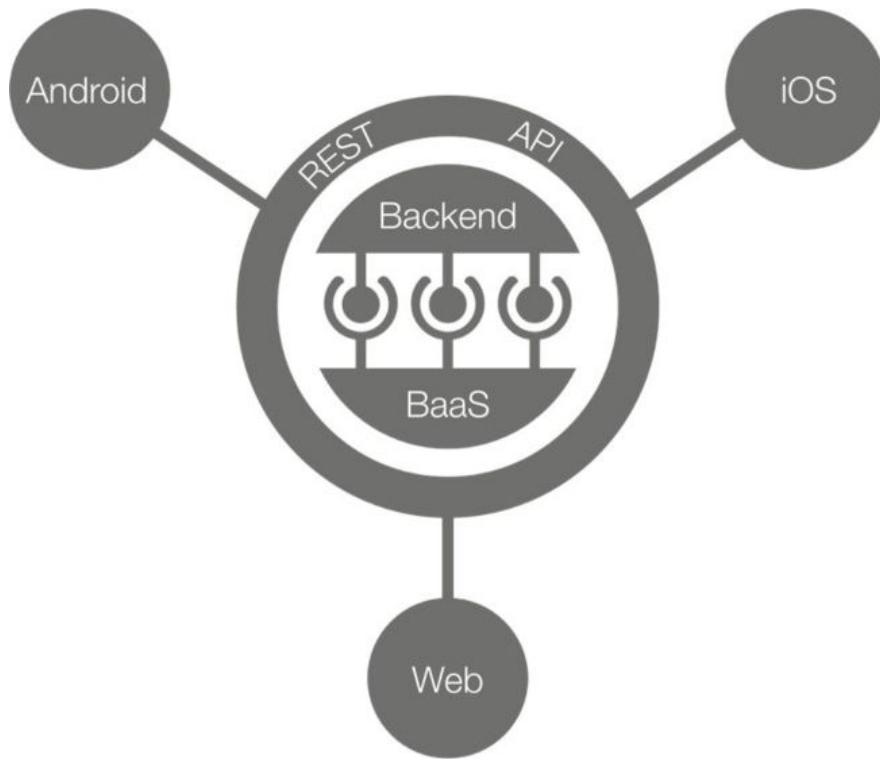
MATT GROENING



# THE SIMPSONS







[http://es.wikipedia.org/wiki/Backend\\_as\\_a\\_service](http://es.wikipedia.org/wiki/Backend_as_a_service)