

{ Servicios Web REST }

RESTFul

Formador: Ezequiel Llarena Borges

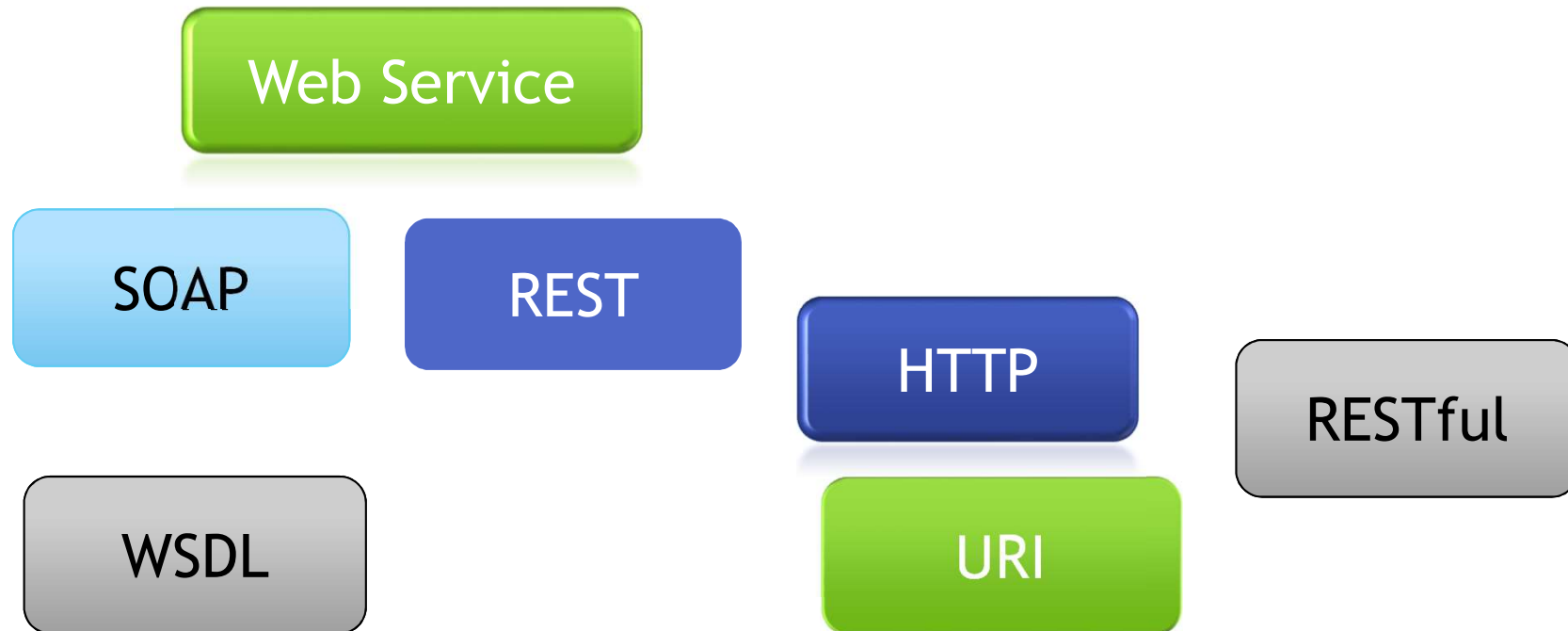
HOW TO INSULT A DEVELOPER

gork & pole



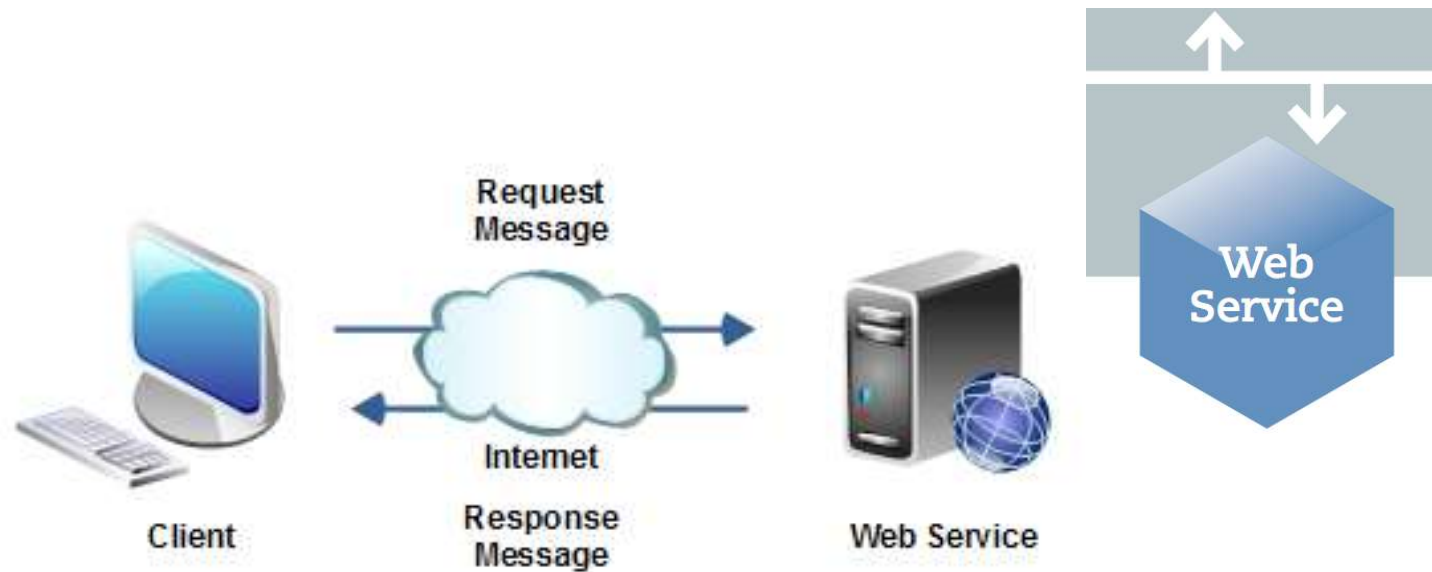
Desarrollo Servicios REST

Algunos términos...



Desarrollo Servicios REST

Servicio Web



Desarrollo Servicios REST

Web Service

- **API online** a la que se puede acceder programáticamente
- Código desplegado en diferentes máquinas
- Retorna datos en formato XML o JSON
- Facebook y Twitter publican web services consumidos por otros desarrolladores desde sus códigos
- Apps, Games, ...

Desarrollo Servicios REST

Web Service

<http://www.twitter.com>



HTML



<http://api.twitter.com>



XML / JSON



JSON

```
{
  "siblings": [
    { "firstName": "Anna", "lastName": "Clayton" },
    { "lastName": "Alex", "lastName": "Clayton" }
  ]
}
```

XML

```
<siblings>
<sibling>
<firstName>Anna</firstName>
<lastName>Clayton</lastName>
</sibling>
<sibling>
<firstName>Alex</firstName>
<lastName>Clayton</lastName>
</sibling>
</siblings>
```

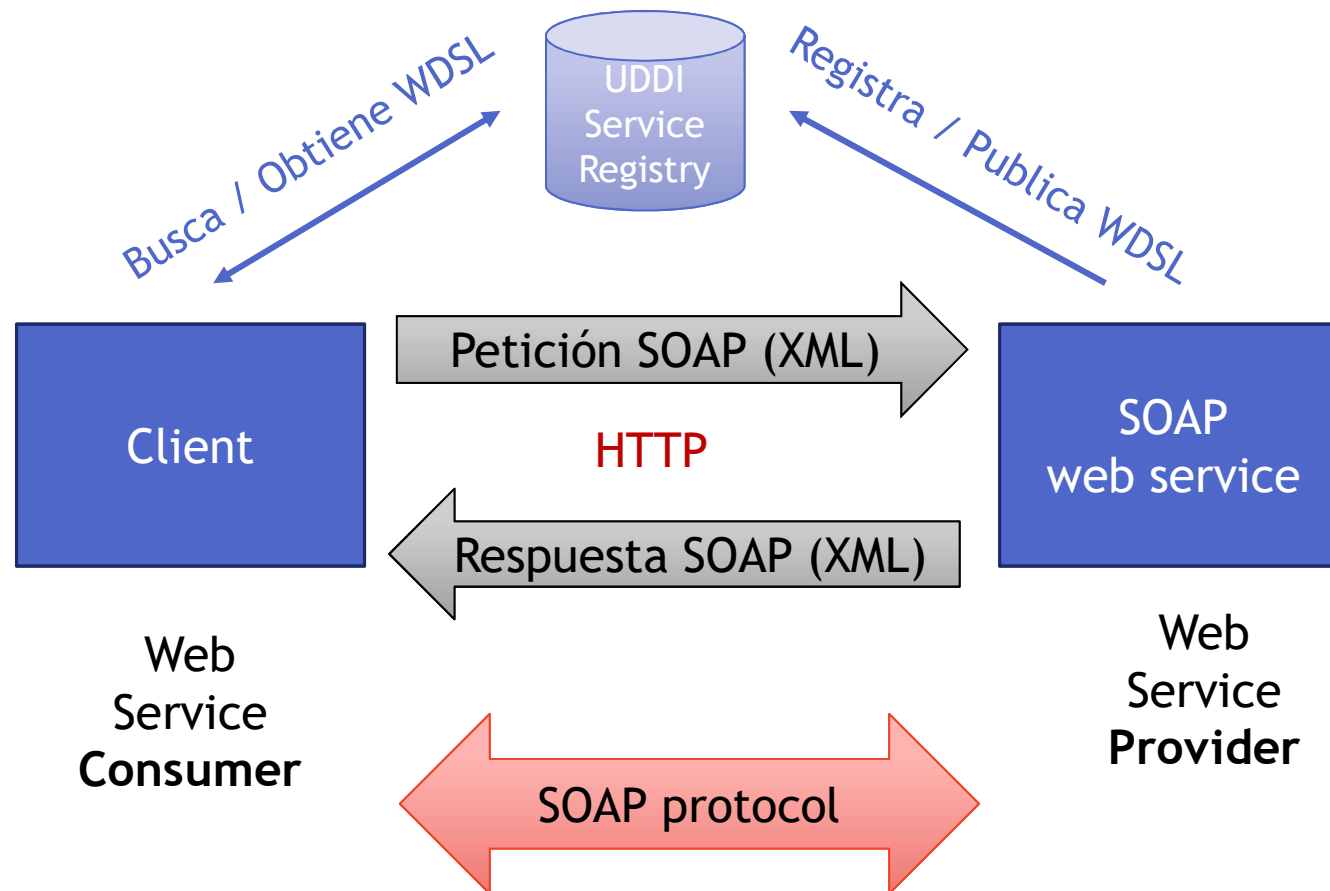
Desarrollo Servicios REST

Estilos de Servicios Web

- **RPC (Remote Procedure Calls)**
 - Llamadas a procedimientos y funciones distribuidas (operación WSDL)
- **SOA (Service-Oriented Architecture)**
 - Comunicación vía mensajes (servicios orientados a mensajes)
- **REST (Representational State Transfer)**
 - Interacción con recursos con estado

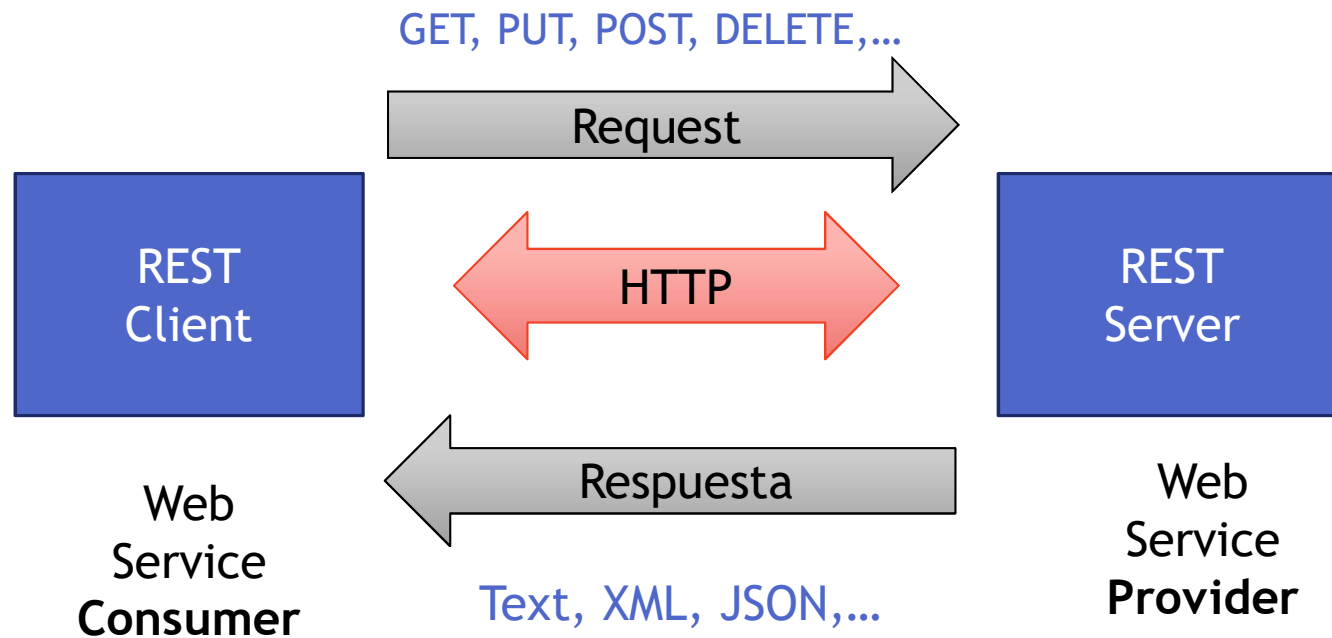
Desarrollo Servicios REST

SOAP Web Service



Desarrollo Servicios REST

REST Web Service



Desarrollo Servicios REST

SOAP vs REST

#	SOAP Simple Object Access Protocol	REST Representational State Transfer
1	Protocolo de mensajes XML	Protocolo de estilo arquitectural
2	Usa WSDL en la comunicación entre el consumidor y el proveedor	Usa XML o JSON para enviar y recibir datos
3	Invoca a los servicios mediante llamadas a métodos RPC	Llamada a un servicio vía URL
4	Información que devuelve no legible para el humano	Resultado es legible por el humano (XML, JSON...)
5	Transferencia sobre HTTP y otros protocolos (SMTP, FTP, etc...)	Transferencia es sólo sobre HTTP
6	JavaScript permite invocar SOAP (implementación compleja)	Fácil de invocar desde JavaScript
7	El rendimiento no es tan bueno comparado a REST	Rendimiento mucho mejor que SOAP - menor consumo CPU, código más pulido, etc.

Desarrollo Servicios REST

RESTful Web Services



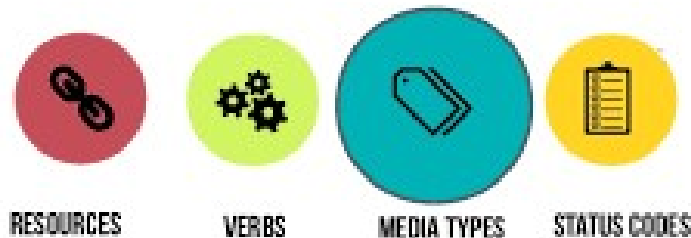
La idea es... “ponérselo fácil a los consumidores”

Desarrollo Servicios REST

REST (REpresentational State Transfer)

- Estilo de **arquitectura** de software para desarrollar servicios web
- Basado en **estándares web** y protocolo **HTTP**
- REST no es un estándar
- Todo es un **Recurso**
- Un servidor REST permite acceso a los recursos
- Recursos identificados mediante un ID global (**URIs**)
- Diferentes **representaciones** de los recursos (text, XML,JSON)

REST Style consists of ...



Desarrollo Servicios REST

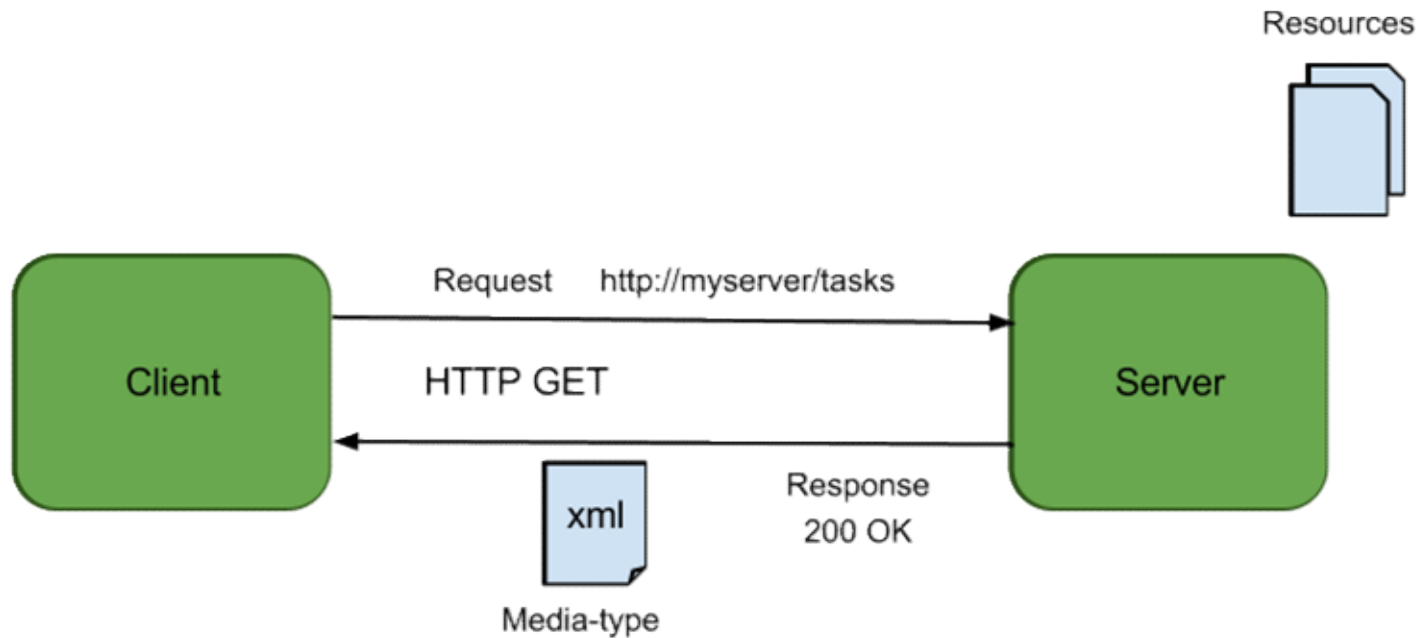
Introducción a Servicios REST

- REST y el renacimiento de HTTP
- Principios de arquitectura REST
 - ✓ El direccionamiento
 - ✓ Interfaces Constrained y Uniform*
 - ✓ HATEOAS

*REST usa un **conjunto reducido** y bien definido de **comandos** para gestionar los recursos.

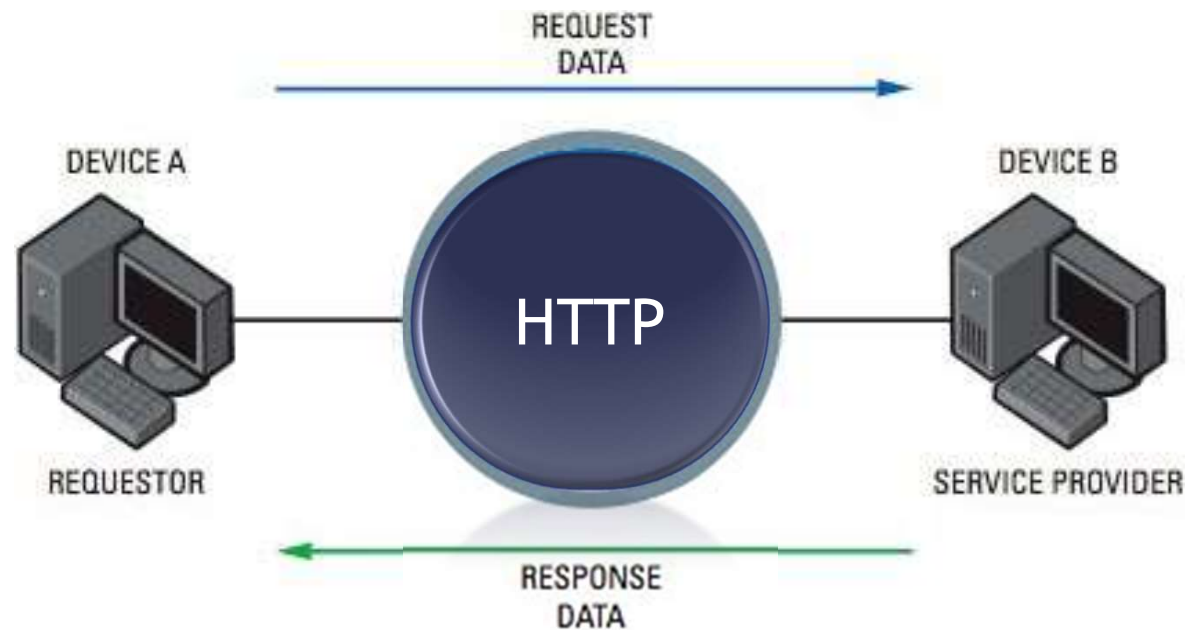
Desarrollo Servicios REST

Funcionamiento de un Web Service REST



Desarrollo Servicios REST

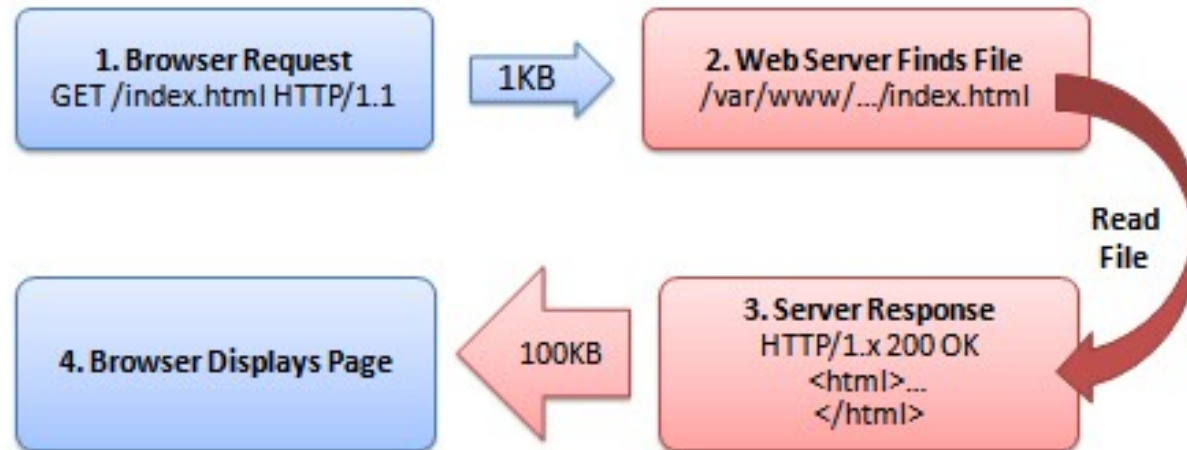
Arquitectura Cliente/Servidor



Desarrollo Servicios REST

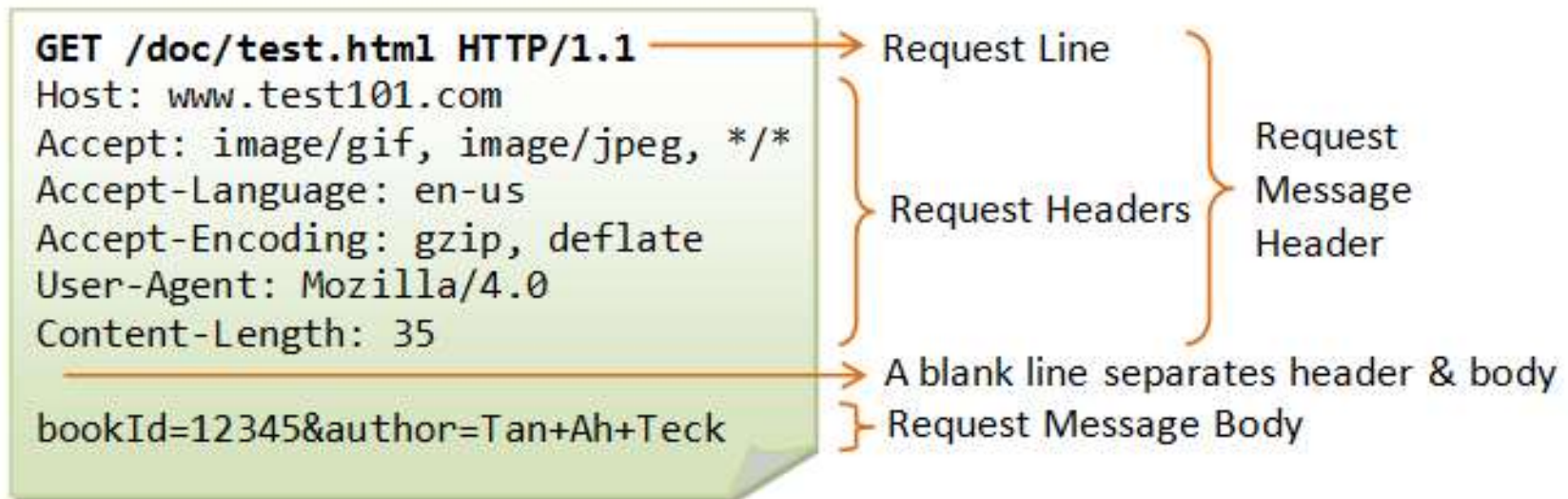
HTTP

HTTP Request and Response



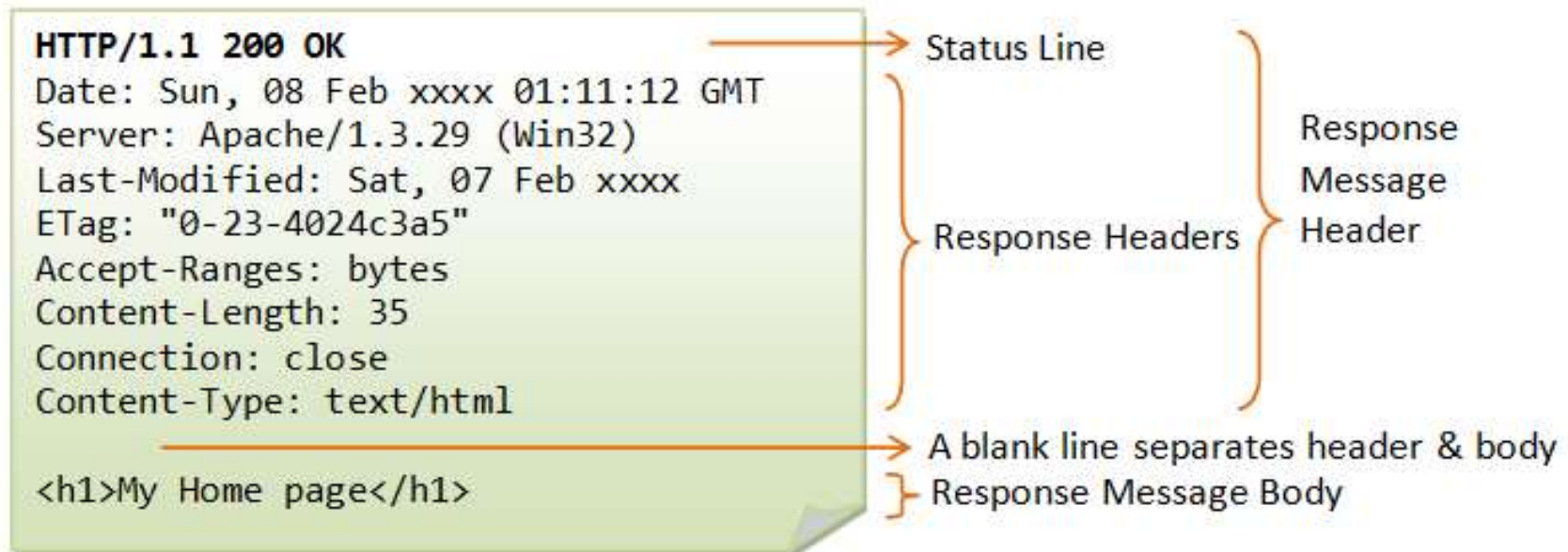
Desarrollo Servicios REST

HTTP Request



Desarrollo Servicios REST

HTTP Response



Desarrollo Servicios REST

REST y HTTP

- Resource based URIs
- HTTP operations
 - ✓ GET
 - ✓ POST
 - ✓ PUT
 - ✓ DELETE
- HTTP status codes (*Metadata*)
 - ✓ 200 - Success (GET)
 - ✓ 201 - Created (POST)
 - ✓ 500 - Server error
 - ✓ 404 - Not found (GET, PUT, DELETE)
- Message headers
 - ✓ Content types: `text/xml`, `application/json`

Desarrollo Servicios REST

RESTful

- Define URI base para los servicios
- Basado en métodos HTTP y concepto REST
 - ✓ GET
 - ✓ POST
 - ✓ PUT
 - ✓ DELETE
- Soporta MIME-types
 - ✓ XML
 - ✓ text
 - ✓ JSON
 - ✓ User-defined, ...

Desarrollo Servicios REST

Métodos HTTP

GET

- Obtiene un recurso
- Es seguro (no side-effects)
- Resultados “cacheables”
- Idempotente

POST

- Crea un recurso nuevo
- No es idempotente

PUT

- Actualiza un recurso existente
- Idempotente

DELETE

- Elimina un recurso
- Idempotente*

Desarrollo Servicios REST

Idempotencia

- El **cliente** puede invocar repetidamente un método generándose siempre el mismo resultado.
- Las operaciones idempotentes producen siempre el mismo resultado en el **servidor**.

Idempotente	No Idempotente
<ul style="list-style-type: none">• GET• PUT• DELETE• HEAD• OPTIONS	<ul style="list-style-type: none">• POST

Desarrollo Servicios REST

Resource locations

URI = Uniform Resource Identifier

Resource based URI

`weatherapp.com/zipcodes/12345`

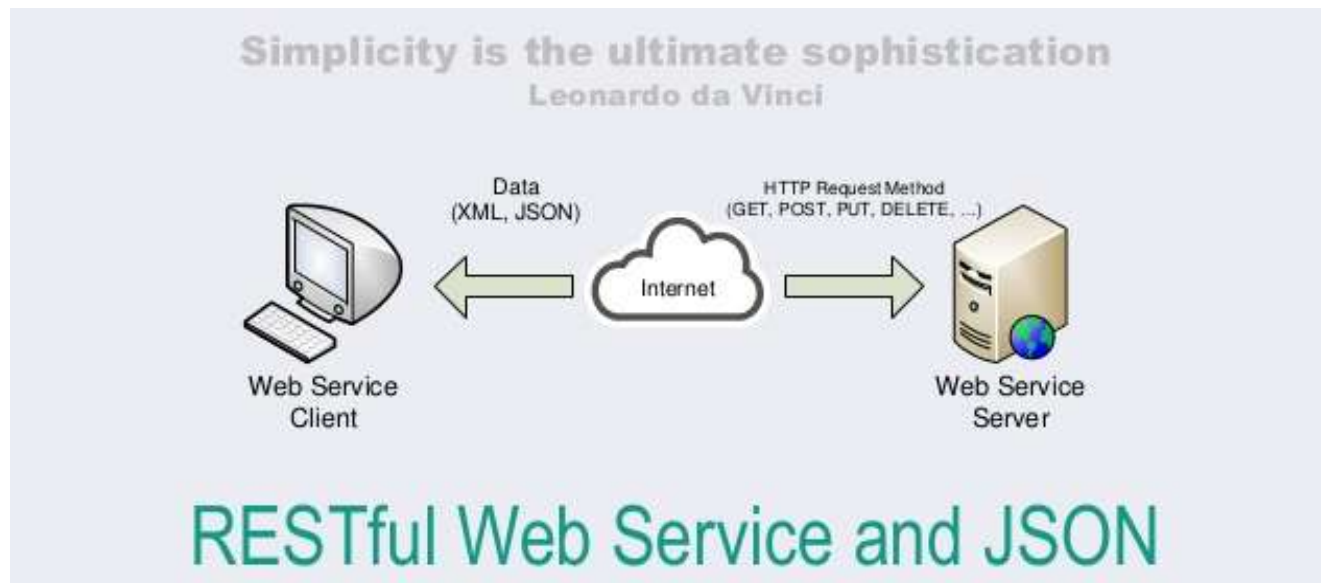
`weatherapp.com/zipcodes/56789`

`weatherapp.com/countries/brazil`

Desarrollo Servicios REST

RESTful

Arquitectura
REST + Web Service = **RESTful**
web service



Desarrollo Servicios REST

HATEOAS

- Hypermedia As the Engine Of Application State
- Permite incluir links a recursos en la respuesta del API, así que el cliente no necesita manejar construcción de la URI



Desarrollo Servicios REST

HATEOAS - the “rel” attribute

```
{
  firstname: "M.",
  lastName: "Ibrahim",
  age: 35,
  departement: "HR",
  - links: [
    - {
      rel: "self",
      href: "http://localhost:8080/person/2"
    },
    - {
      rel: "account",
      href: "http://localhost:8080/person/2/account"
    },
    - {
      rel: "profile",
      href: "http://localhost:8080/person/2/profile"
    }
  ]
}
```

Desarrollo Servicios REST

Richardson Maturity Model

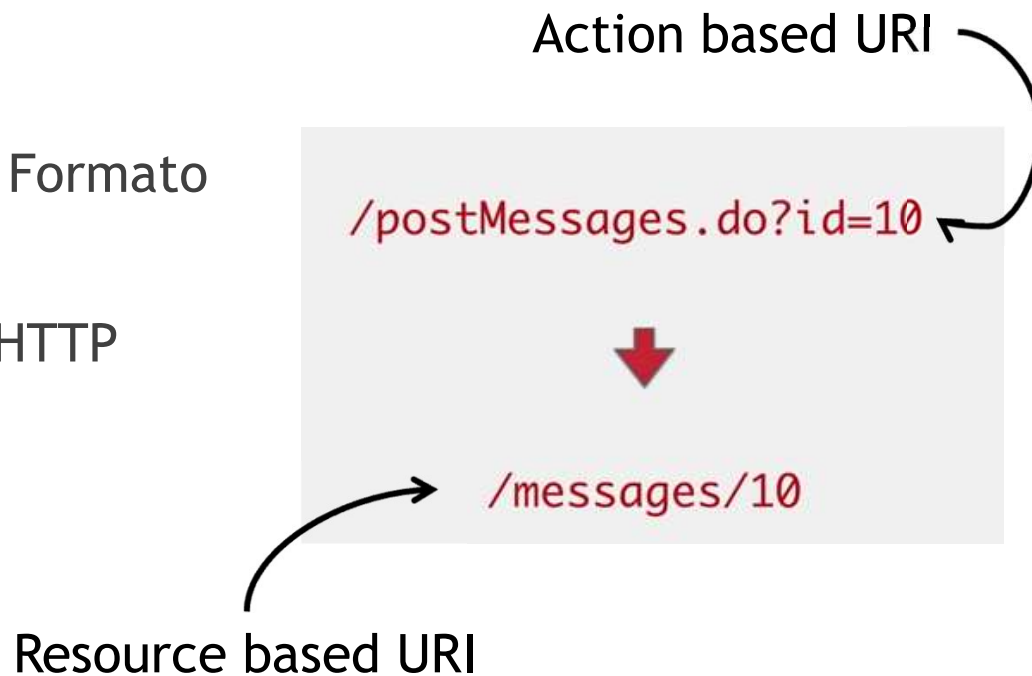
Is this API “fully RESTful”?

Level	Características	
0	<ul style="list-style-type: none">• One URI• Message request body contains all the details	Is not a RESTful API
1	<ul style="list-style-type: none">• Resource URI	Individual URIs for each resource
2	<ul style="list-style-type: none">• HTTP Methods	Use the right HTTP methods, status codes
3	<ul style="list-style-type: none">• HATEOAS	Responses have links that the clients can use

Desarrollo Servicios REST

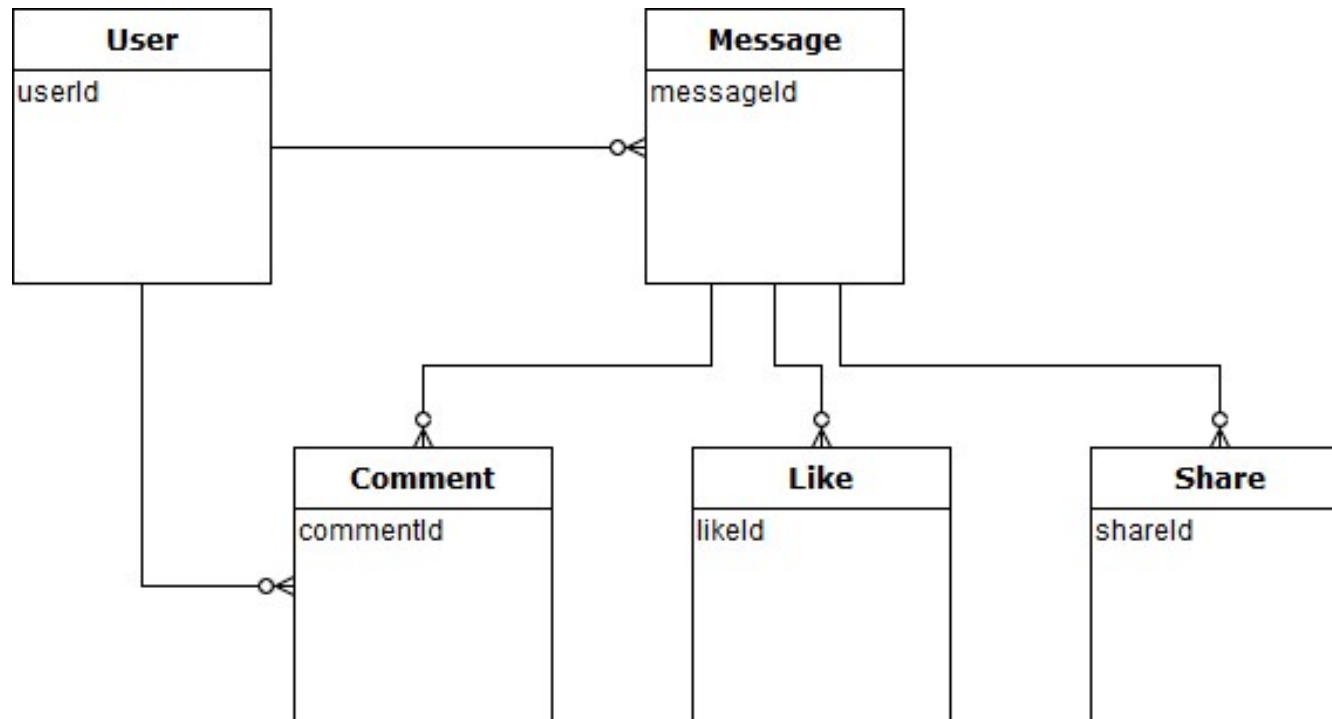
Diseño de Servicios REST

- El modelo de objetos
- Formato de datos
- Leer y actualización de Formato
- Crear Formato
- Asignación de Métodos HTTP



Desarrollo Servicios REST

Diseño de API REST (Ejemplo)



Desarrollo Servicios REST

Instance resource URI

instance / item



Resource based URI

/profiles/{profileName}

/messages/{messageId}

/messages/{messageId}/likes/{likeId}

/coordinates/lats/{latVal}

/coordinates/lons/{lonVal}

Desarrollo Servicios REST

Collection resource URI

Collection



Resource based URI

/profiles/

/messages/

/messages/{messageId}/likes/

/profiles/{profileName}/comments/

Desarrollo Servicios REST

Filtering results

Query Parameters



`/messages/?offset=5&limit=10`



starting point



page size

Custom Filters



`/messages/?year=2017`



Filtering Collection

Desarrollo Servicios REST

Formato de datos



```
<user>
  <id>1</id>
  <name>Me</name>
  <email>me@gmail.com</email>
</user>
```

content-type: **text/xml**



```
{
  "id" :1,
  "name" : "Me",
  "email" : "me@gmail.com"
}
```

content-type: **application/json**

Desarrollo Servicios REST

Operaciones HTTP

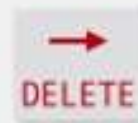
/getProducts.do?id=10 ❌



/products/10



/deleteOrder.do?id=10 ❌

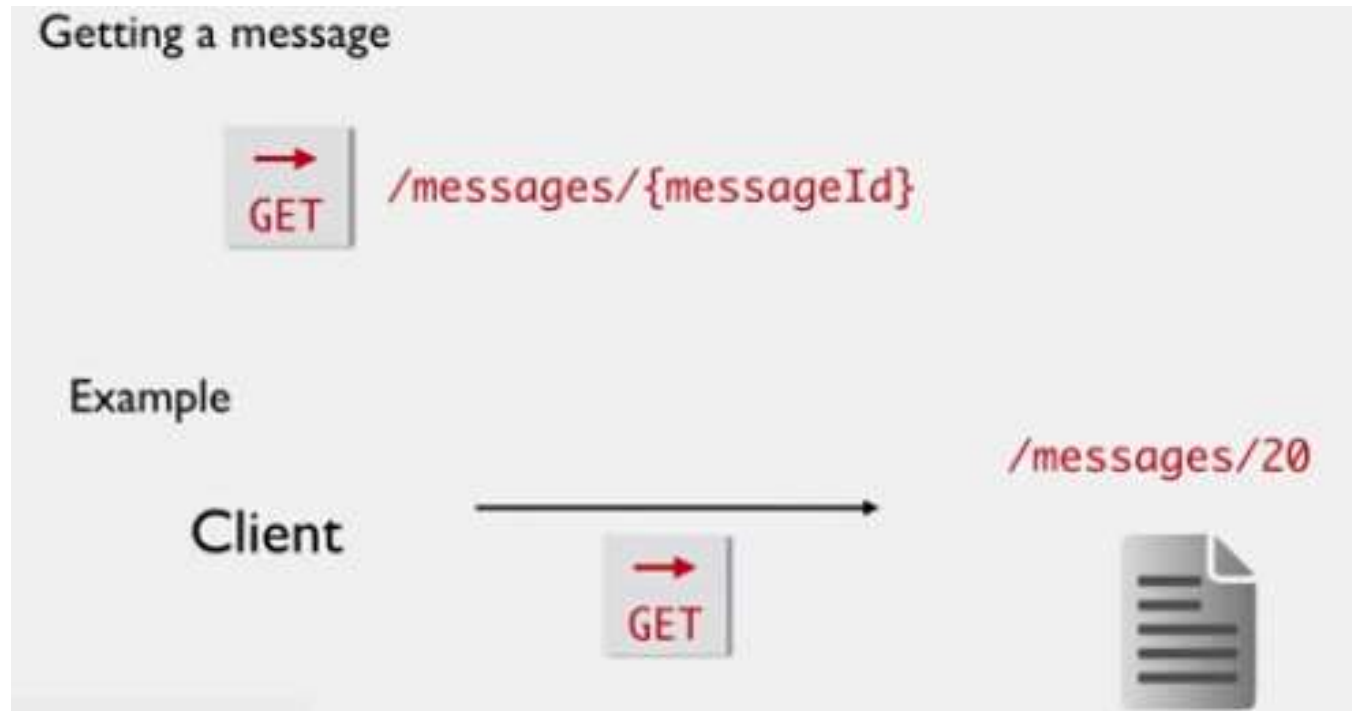


/orders/10



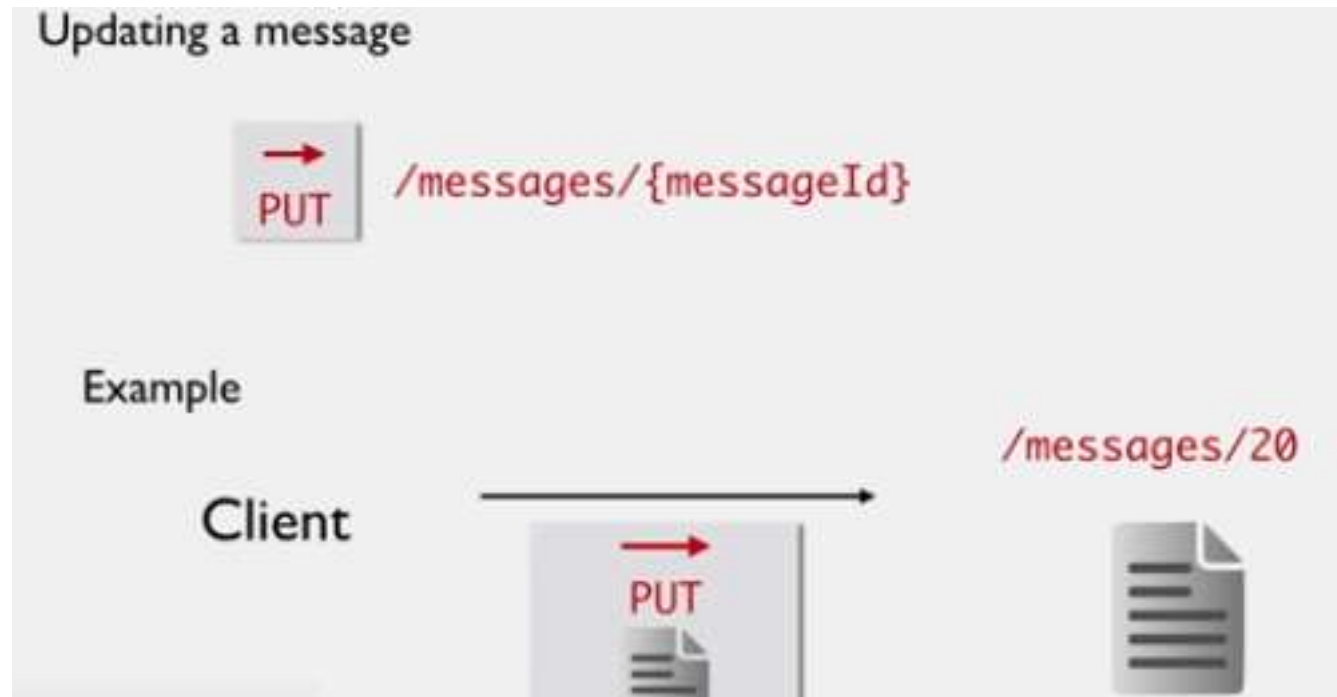
Desarrollo Servicios REST

Operaciones HTTP GET



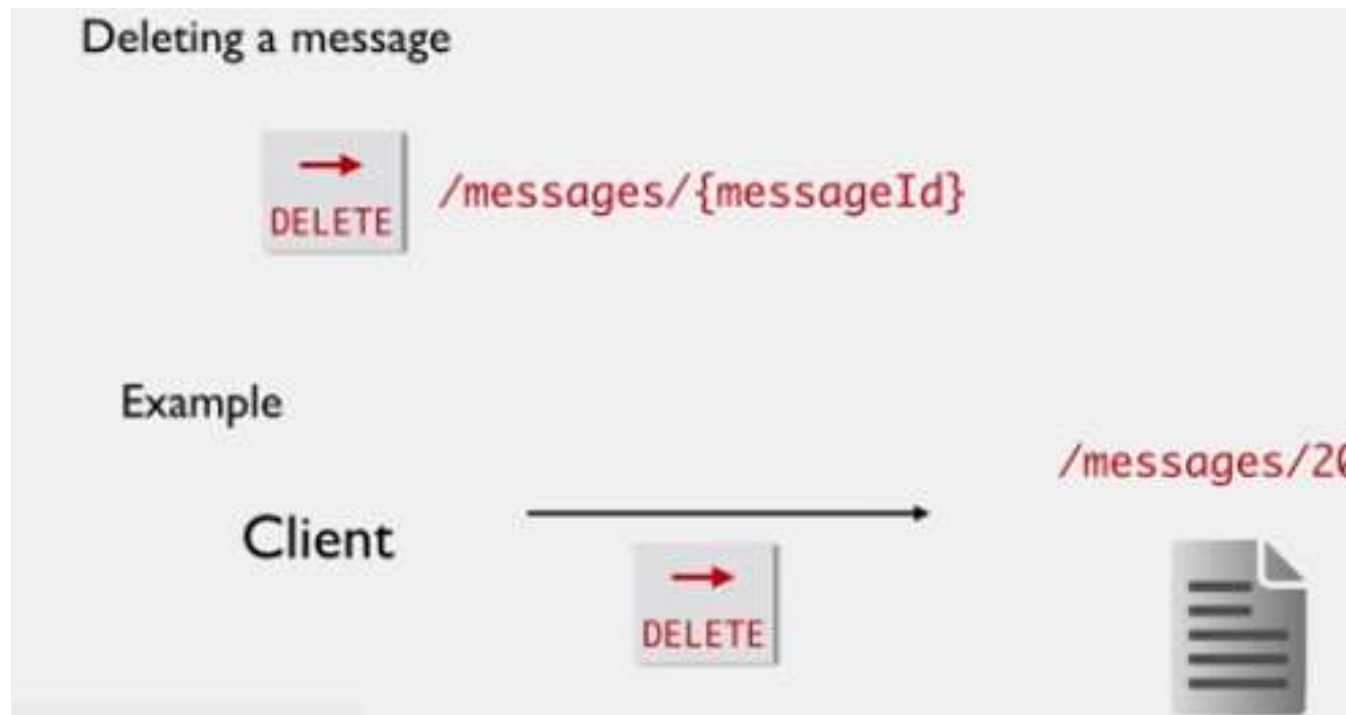
Desarrollo Servicios REST

Operaciones HTTP PUT



Desarrollo Servicios REST

Operaciones HTTP DELETE



Desarrollo Servicios REST

Operaciones HTTP POST

Creating a new message



/messages

Example

Client



/messages

Desarrollo Servicios REST

Operaciones HTTP

Collection URI scenarios



/messages/10/comments

creates a new comment for message 10



/messages/20/comments

replaces **all** comments for message 20 with a new list