

Web Services

REST

Alfredo Casado Bernardez
casado.alfredo@gmail.com

¿De que vamos ha hablar?

¿REST?, ¿no tenemos bastante con
SOAP?

A algunos *principios de diseño* REST

¿Que me ofrece java para construir
servicios REST?

Aplicación web vista como un
cliente REST.

¿Por qué REST?

Una palabra: simplicidad

"The purpose of software engineering is to control complexity, not to create it."

Pamela Zave

“No matter how hard I try, I still think the WS stack is bloated, opaque, and insanely complex. I think it's going to be hard to understand, hard to implement, hard to interoperate, and hard to secure.”*

Tim Bray

uno de los padres de XML

Director de tecnologías web en Sun

“Show me the interoperable, full and free implementations of WS in Python, Perl, Ruby and PHP. You won’t see them, because there’s no intrinsic value in WS* unless you’re trying to suck money out of your customers. Its complexity serves as a barrier to entry at the same time that it creates “value” that can be sold”*

Mark Nottingham,

líder técnico en el área de Web Services en Yahoo!

*Ha participado en la definición de muchos de los estándares WS-**

De momento dejemos de criticar
SOAP...

veamos algunas buenas practicas
para diseñar servicios REST

Cam b i o e n l a f o r m a d e p e n s a r

S O A P : *“que servicios ofrezco”*

R E S T : *“que recursos expongo”*

Definir URI's para mis recursos

Ejemplos

<http://www.javahispano.org/noticia/1234>

<http://www.javahispano.org/noticias/18/06/2009>

<http://www.javahispano.org/foros/thread/25/mensaje/14>

Usar métodos estándar HTTP

GET obtener la representación del
recurso

POST para crear un nuevo recurso

PUT para modificar un recurso

DELETE para borrar un recurso

P e t i c i ó n H T T P R E S T

```
DELETE /noticia/1234 HTTP/1.1
```

```
Host: www.javahispano.org
```

```
User-Agent: .....
```

P e t i c i ó n X M L ~ R P C

```
POST /endpoint HTTP/1.1
```

```
Host: www.javahispano.org
```

```
<methodCall>
```

```
  <methodName>borrarNoticia</methodName>
```

```
  <params>
```

```
    <param><value>1234</value></param>
```

```
  </params>
```

```
</methodCall>
```

Enlazar los recursos

```
<noticia>
  <Titulo>Sun communities forum</Titulo>
  <Resumen>...</Resumen>
  <Contenido>...</Contenido>
  <NoticiasRelacionadas>
    <noticia rel="http://javahispano.org/noticia/2545"/>
    <noticia rel="http://javahispano.org/noticia/5484"/>
    . . . .
  </NoticiasRelacionadas>
</methodCall>
```

Códigos de respuesta HTTP

Todos conocemos:

200 (OK)
404 (Not Found)
500 (Internal Server Error)

Hay muchos más:

201 (Created)
409 (Conflict)
412 (Precondition failed)

Con SOAP, siempre se devuelve 200
(ok)

Multiple representaciones

No sólo de XML vive el hombre

Por ejemplo JSON, más fácil de
tratar en JavaScript

HTTP content negotiation

Servicios sin estado

Lo ideal es que los servicios REST no tengan estado

Simplifica el desarrollo

Fundamental en sistemas altamente escalables

¿pero alguien usa REST?



REST en java

JSR 311 - jersey

Spring

Restlet

Un servlet, ¿y por qué no?

C l i e n t e s R E S T

S ó l o n e c e s i t o d o s c o s a s :

A l g u n a l i b r e r í a q u e e n t i e n d a H T T P

U n p a r s e r d e X M L

R E S T e s i n t e r o p e r a b l e , p e r o d e v e r d a d

A p l i c a c i ó n A j a x c l i e n t e

R E S T

E l n a v e g a d o r m e d a l o q u e n e c e s i t o :

O b j e t o X M L H t t p R e q u e s t e n t i e n d e
H T T P

C u a l q u i e r b r o w e r d i s p o n e d e u n
p a r s e r X M L q u e p u e d o u s a r d e s d e
J a v a S c r i p t

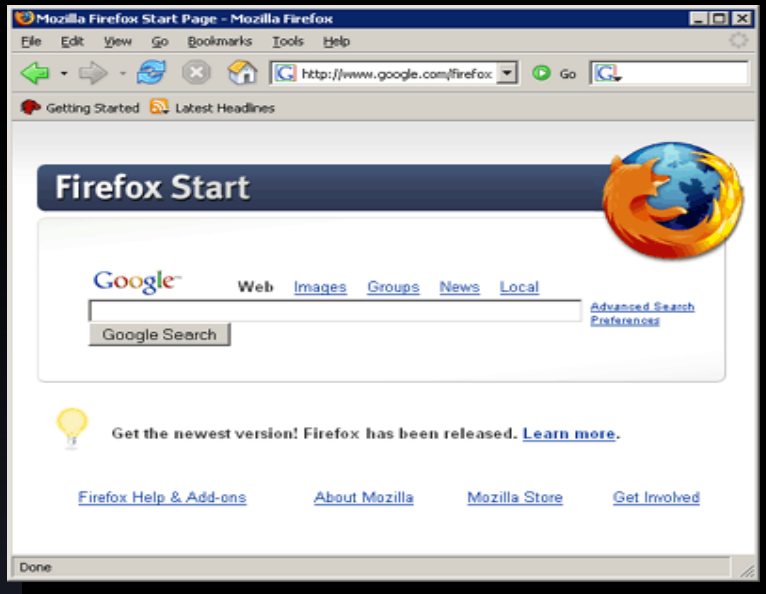
Muchas librerías facilitan la tarea de acceder a servicios REST, un ejemplo con JQuery:

```
$.ajax({type: "DELETE",  
        url: "noticia/1234",  
        complete: function(xhr){  
            var statusCode = xhr.status;  
            // hacer lo que toque en función del  
            // código de respuesta recibido  
        }  
    });
```

Es igual de fácil con Prototype,
dojo, YUI, GWT...

Single Page Interface

Servidor REST



HTTP

XML / JSON / ...



Otras aplicaciones pueden hacer uso de los servicios

En resumen

Todavía estamos a tiempo de
RESTificar la forma en la que
hacemos y diseñamos Web Services.

Nota: la organización prohíbe lanzar tomates u otras hortalizas a los ponentes, aunque se lo merezcan.

Gracias!

¿Preguntas?