REST

lesson #advanced03

James L. Parry
B.C. Institute of Technology

REST

REST := REpresentational State Transfer

The current "darling" of inter-application "glue".

Agenda

1. The Background
2. REST Philosophy
3. RESTful "Standards"
4. REST in CodeIgniter
5. REST Issues

THE BACKGROUND

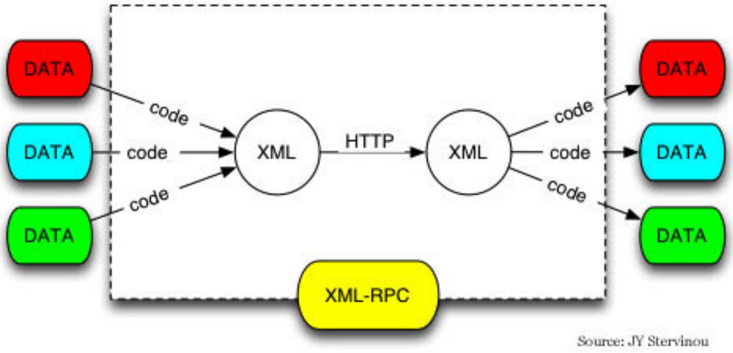
You have already seen "standard glue":

Transport: HTTP, HTTPS, AJAX

Representation: XML, JSON, RSS

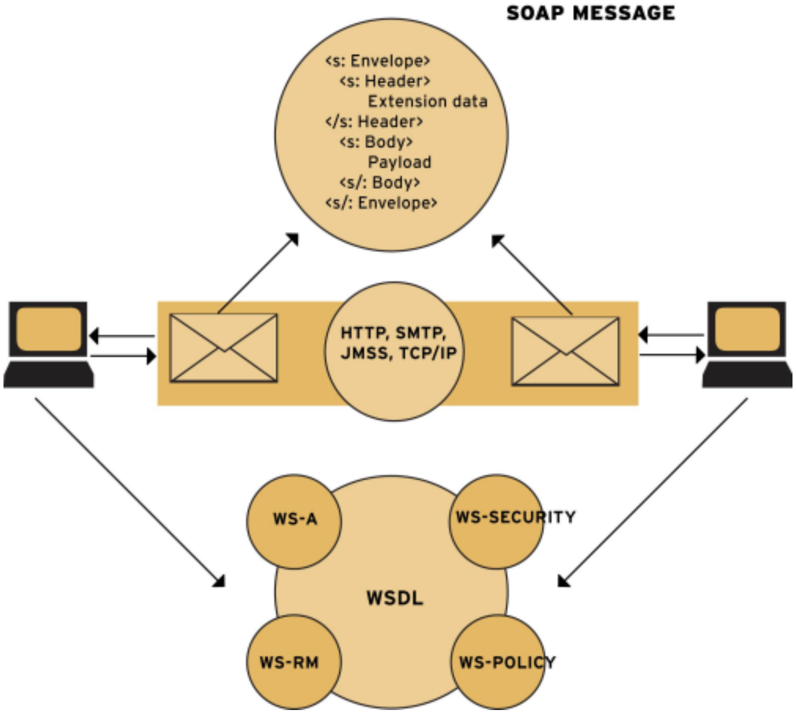
Application: RPC, SOA, **REST**

Reminder: RPC

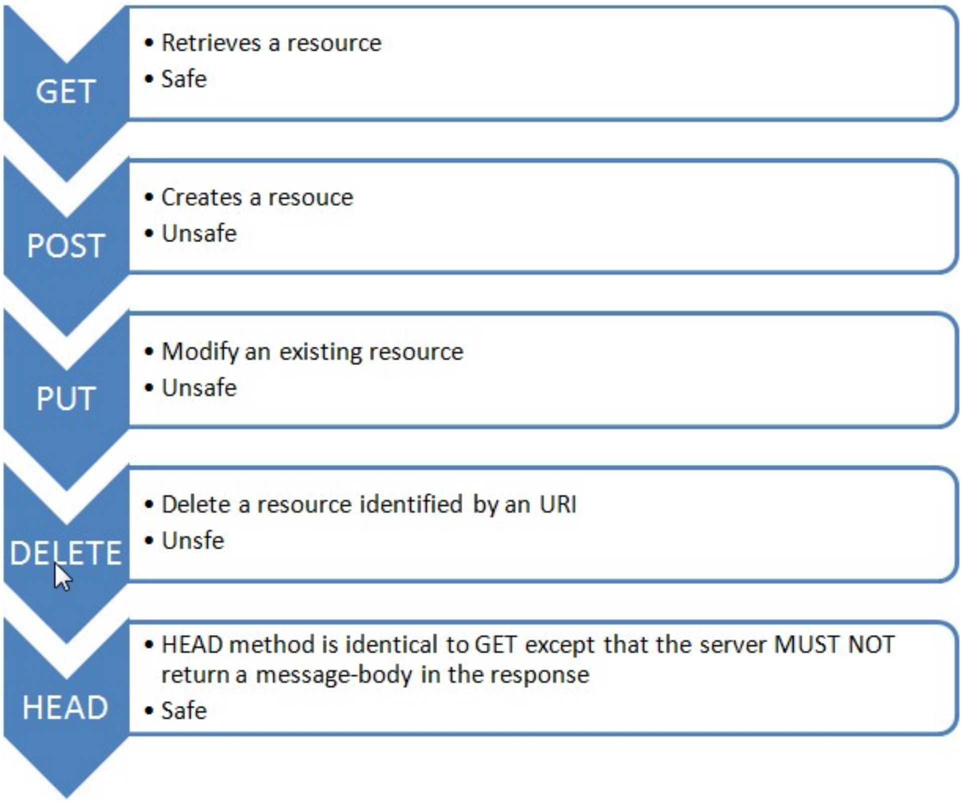


Source: JY Stervinou

Twist: Service Oriented Architecture (SOA)



Now for REST!



GET
- Retrieves a resource
- Safe

POST
- Creates a resouce
- Unsafe

PUT
- Modify an existing resource
- Unsafe

DELETE
- Delete a resource identified by an URI
- Unsfe

HEAD
- HEAD method is identical to GET except that the server MUST NOT return a message-body in the response
- Safe

THE REST PHILOSOPHY

REST is an architectural style or philosophy, not a standard.

- Map resources (not usecases) to URLs
- Map CRUD usecases to HTTP types
- All needed data passed (rep state)
- Accept or deliver multiple formats (XML, JSON, JSONP, CSV, XHTML, text)

REST URLs

- Http-get: `/product` returns a list of products
- Http-get: `/product/123` returns product 123
- Http-post: `/product/123` creates product 123
- Http-put: `/product/123` updates product 123
- Http-delete: `/product/123` deletes product 123

"RESTful" STANDARDS

Mostly REST, with interpretation

Single vs multiple records? collection treatment?

- (get) /parts {maybe have parm for pagination}
- (...) /part/abc

JSON or XML only - others unsafe

REST Data Formats

Specify in HTTP header

- Accept: application/xml
- Content-Type: application/xml
- Valid types: application/json, application/xml, application/xhtml+xml

Imply in URL

- /product/123.json

REST IN CODEIGNITER

- Stateless :)
- URL mapping native :)
- URL mapping supposed to be controllers for usecases :(
- HTTP payload not easily accessed :(
- HTTP type not originally supported :-/
- Emitting HTTP-POST not part of CI or PHP style :((

CodeIgniter Solution #1

Add-Ons...

- RESTclient base library as abstraction layer
- RESTserver as base controller (tho awkward parms)
- Curl library for low-level HTTP types, in & out
- Format library for multiple format support

Only one choice ... there are multiple conflicting conventions or interpretations

CodeIgniter RESTful Client

Adapted from Phil Sturgeon's REST client, all format conversion is automatically handled by the REST layer, though you can tailor it.

```
// Load the library
$this->load->library('rest');

// Run some setup
$this->rest->initialize(array('server' => OUR_SERVER_NAME));

// use get/post/... as appropriate
$answer = $this->rest->get(methodURI,parameters_array);

// need to convert & pick apart the response
```

CodeIgniter RESTful Server

Adapted from Phil Sturgeon's REST server, all format conversion is automatically handled by the REST layer, and incoming requests are automatically mapped to the appropriate method.

```
class Service extends Rest_controller {

function index_get() {
    ...
    $this->response($assoc_array,$http_code);
}

function index_post() { ... }
function index_put() { ... }
function index_delete() { ... }

}
```

CodeIgniter Solution #2

Map requests by HTTP type

- $route['product']['get'] = 'product/get';
- $route['product']['put'] = 'product/put';
- $route['product']['post'] = 'product/post';
- etc ...

This does not address format conversions, nor does it clarify other RFEST issues :(

REST ISSUES

- Anyone/everyone can claim to be RESTful!
- No guidance on large data / pagination
- No guidance on filtering parameters handling
- “Extensions” to formats (eg “PHP”)
- Some formats have issues (JSONP, CSV)
- Seeing through the hype (Web 3.0000...)

REST - DO YOU CARE?

“REST” displacing XML-RPC and SOAP

Social networking APIs becoming reliant on it (or company interpretations)

So, yeah

Congratulations!

You have completed lesson #advanced03: REST

If you would take a minute to provide some feedback, we would appreciate it!

The next activity in sequence is: advanced04 JSON

You can use your browser's back button to return to the page you were on before starting this activity, or you can jump directly to the course homepage, organizer, or reference page.