中山大學
SUN YAT-SEN UNIVERSITY

Lecture 21
Web Serivices

SE-805 Web 2.0 Programming (supported by Google)

http://my.ss.sysu.edu.cn/courses/web2.0/

School of Software, Sun Yat-sen University

Outline

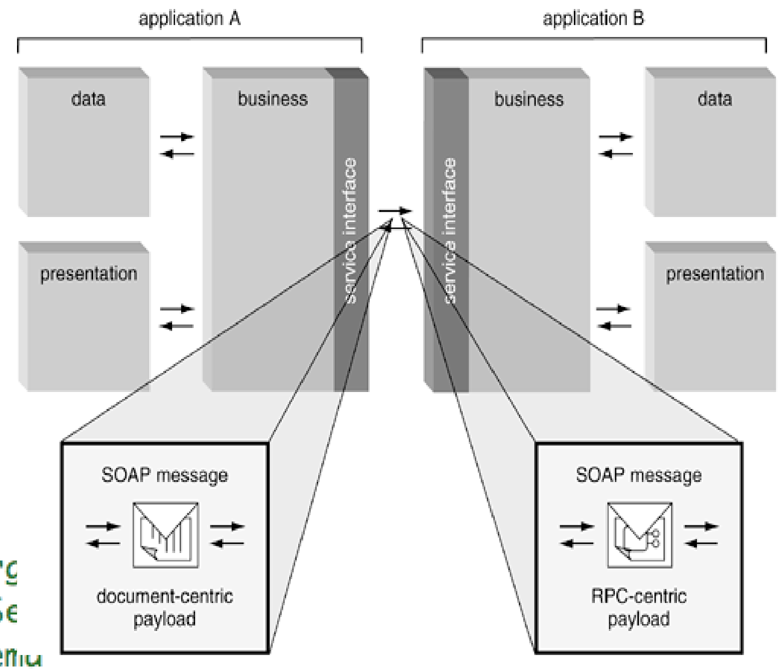
- **Web Services Essentials**
- REST in PHP

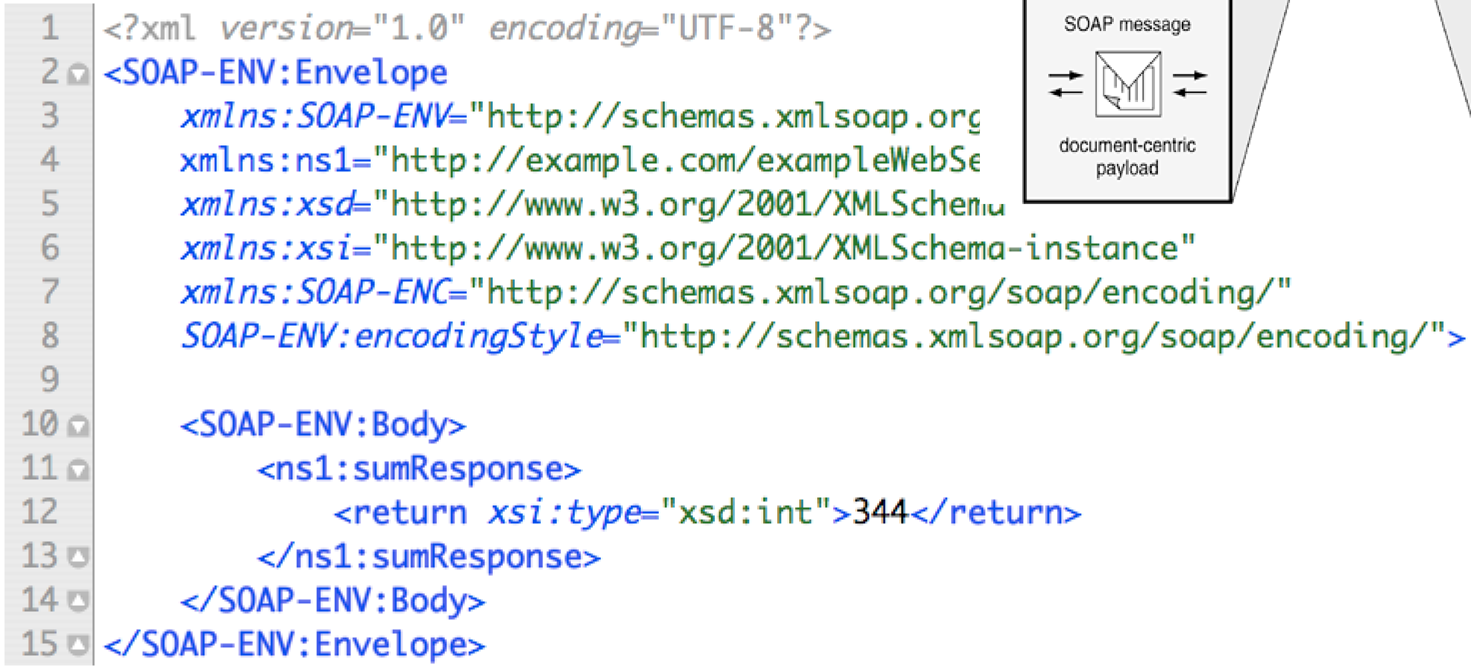
What is a Web Service?

- **Web Service**: software functionality that can be invoked through the internet using common protocols
- Like a remote function(s) you can call by contacting a program on a web server
- Many web services accept parameters and produce results
- Can be written in PHP and contacted by the browser in XHTML and/or Ajax code
- Service's output is often not HTML but rather text, XML, or other content types

Web Services - SOAP

- Simple Object Access Protocol
- Usually an HTTP POST request
- Call is encapsulated in XML
- Response is an XML document



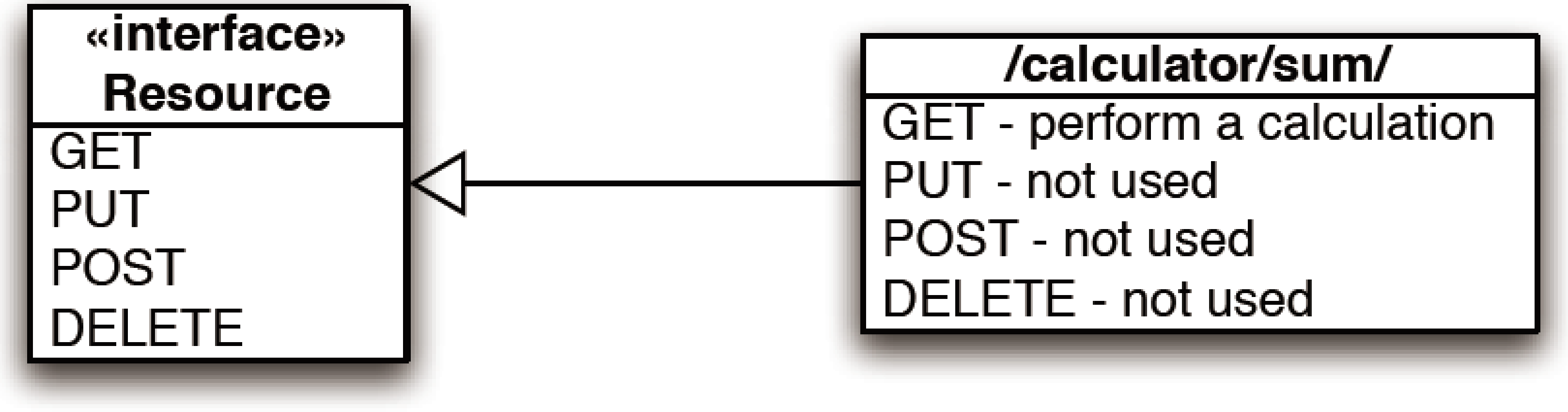


```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org
    xmlns:ns1="http://example.com/exampleWebSe
    xmlns:xsd="http://www.w3.org/2001/XMLSchem
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

    <SOAP-ENV:Body>
        <ns1:sumResponse>
            <return xsi:type="xsd:int">344</return>
        </ns1:sumResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Web Services - REST

- Representational State Transfer
- Use HTTP “GET”, “POST”, “PUT”, “DELETE” actions
- Response can be either XML, JSON, plain text, or even customized format
- We use REST in this course



http://example.com/calculator/sum/?x=121&y=233

Outline

- Web Services Essentials
- **REST in PHP**

Content ("MIME") types

MIME type	Related File Extension
text/plain	.txt
text/html	.html, .htm, ...
text/css	.css
text/javascript	.js
text/xml	.xml
image/gif	.gif
image/jpeg	.jpg, .jpeg
video/quicktime	.mov
application/octet-stream	.exe

- Lists of MIME types: by type, by extension

Setting Content Type with Header

```php
header("Content-type: type/subtype");
```
PHP

```php
header("Content-type: text/plain");
print("This output will appear as plain text now!\n");
```
PHP

- By default, a PHP script's output is assumed to be HTML
- Use the header function to specify non-HTML output
  - must appear before any other output generated by the script

Example: Exponent web service

- Write a web service that accepts a **base** and **exponent** and outputs **base** raised to the **exponent** power. For example, the following query should output **81** :

```
http://example.com/exponent.php?base=3&exponent=4
```

- Solution:

```
header("Content-type: text/plain");
$base = $_REQUEST["base"];
$exp = $_REQUEST["exponent"];
$result = pow($base, $exp);
print $result;
```
PHP

Recall: HTTP GET vs. POST

- HTTP: the set of commands understood by a web server and sent from a browser
- **GET** : asks a server for a page or data
  - if the request has parameters, they are sent in the URL as a query string
- **POST** : submits data to a web server and retrieves the server's response
  - if the request has parameters, they are embedded in the request's HTTP packet, not the URL
- For submitting data, a POST request is more appropriate than a GET
  - GET requests embed their parameters in their URLs
  - URLs are limited in length (~ 1024 characters)
  - URLs cannot contain special characters without encoding
  - private data in a URL can be seen or modified by users

The $_SERVER Superglobal Array

Index	Description	Example
$_SERVER["SERVER_NAME"]	name of this web server	“sysu.edu.cn"
$_SERVER["SERVER_ADDR"]	IP address of web server	"128.208.179.154"
$_SERVER["REMOTE_HOST"]	user's domain name	"hsd1.wa.comcast.net"
$_SERVER["REMOTE_ADDR"]	user's IP address	"57.170.55.93"
$_SERVER["HTTP_USER_AGENT"]	user's web browser	"Mozilla/5.0 (Windows; ..."
$_SERVER["HTTP_REFERER"]	where user was before this page	"http://www.google.com/"
$_SERVER["REQUEST_METHOD"]	HTTP method used to contact server	"GET" or "POST"

- Call phpinfo(); to see a complete list

GET or POST?

```
if ($_SERVER["REQUEST_METHOD"] == "GET") {
  # process a GET request
  ...
} elseif ($_SERVER["REQUEST_METHOD"] == "POST") {
  # process a POST request
  ...
}                                                        PHP
```

- Some PHP web services process both **GET** and **POST** requests
- Can find out which kind of request we are currently processing by looking at the "**REQUEST_METHOD**" key of the global **$_SERVER** array
- You can also access query parameters through **$_GET** and **$_POST** rather than **$_REQUEST**

Emitting Partial-page HTML data

```
# suppose my web service accepts a "type" query parameter
if ($_REQUEST["type"] == "html") {
  # client wants their output to be HTML format
  ?>
  <ul>
  <?php
  foreach ($students as $kid) {
    ?>
    <li> <?= $kid ?> </li>
    <?php
  }
  ?>
  </ul>
  <?php
}
```
PHP

- Some web services do output HTML, but not a complete page
- The partial-page HTML is meant to be fetched by Ajax and injected into an existing page

Emitting XML Data

```php
header("Content-type: text/xml");
print("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
print("<books>\n");
foreach ($books as $title) {
  print("<book title=\"$title\" />\n");
}
print("</books>\n");
```

PHP

- Specify a content type of **`text/xml`** or **`application/xml`**
- Print an XML prologue (the **`<?xml`** line) first
    - **important:** no whitespace output can precede the prologue
- Then print each line of XML data/tags as output
- Some PHP libraries automatically generate XML for you from other data (e.g. databases)

Reporting Errors

- How does a web service indicate an error to the client? error messages (`print`) are not ideal, because they could be confused for normal output
- Web service should return an HTTP "**error code**" to the browser, possibly followed by output these are the codes you see in Firebug's console and in your Ajax request's **`.status`** property

HTTP code	Description
200	OK
301-303	page has moved (permanently or temporarily)
400	illegal request
403	you are forbidden to access this page
404	page not found
500	internal server error
complete list	

User **headers** for HTTP Error Codes

```
header("HTTP/1.1 code description");                                  PHP
```

```
if ($_REQUEST["foo"] != "bar") {
  # I am not happy with the value of foo; this is an error
  header("HTTP/1.1 400 Invalid Request");
  die("An HTTP error 400 (invalid request) occurred.");
}                                                                      PHP
```

```
if (!file_exists($input_file_path)) {
  header("HTTP/1.1 404 File Not Found");
  die("HTTP error 404 occurred: File not found ($input_file_path)");
}                                                                      PHP
```

- **header** can also be used to send back HTTP error codes
  - `header("HTTP/1.1 403 Forbidden");`
  - `header("HTTP/1.1 404 File Not Found");`
  - `header("HTTP/1.1 500 Server Error");`

Summary

- Web Services Essentials
  - SOAP
  - REST
- REST in PHP
  - MIME
  - GET vs. POST
  - $_SERVER
  - HTML data, xml data
  - errors in HTTP error codes

Exercises

- Write a php REST web service which calculates the sum of two given numbers
  - What's action should be used, the "GET" or the "POST"?
  - Return your result data in different format (plain text, xml, html)
  - Use HTTP error code to report errors
  - Test your web service in a browser or in a telnet shell

Further Readings

- Introduction of Web Service http://en.wikipedia.org/wiki/Web_service
- Introduction of SOAP http://en.wikipedia.org/wiki/SOAP
- W3C SOAP spec. http://www.w3.org/TR/soap/
- Representation State Transfer http://en.wikipedia.org/wiki/REST
- Create a REST API with PHP http://www.gen-x-design.com/archives/create-a-rest-api-with-php/
- PHP Cookbook: http://commons.oreilly.com/wiki/index.php/PHP_Cookbook

Thank you!