Services in PHP

Web Services

- Web based applications
- HTTP is the interface
- Data is transmitted in XML or JSON
- Generally intended for machine to machine communication
- Required for mashups (generally)

HTTP

- GET
  - Requests data from a server
- POST
  - Sends data to a server
- PUT
  - Pushes a file onto a server
- DELETE
  - Asks server to delete a resource
- HEAD
  - Gets http header back from server, not whole resource

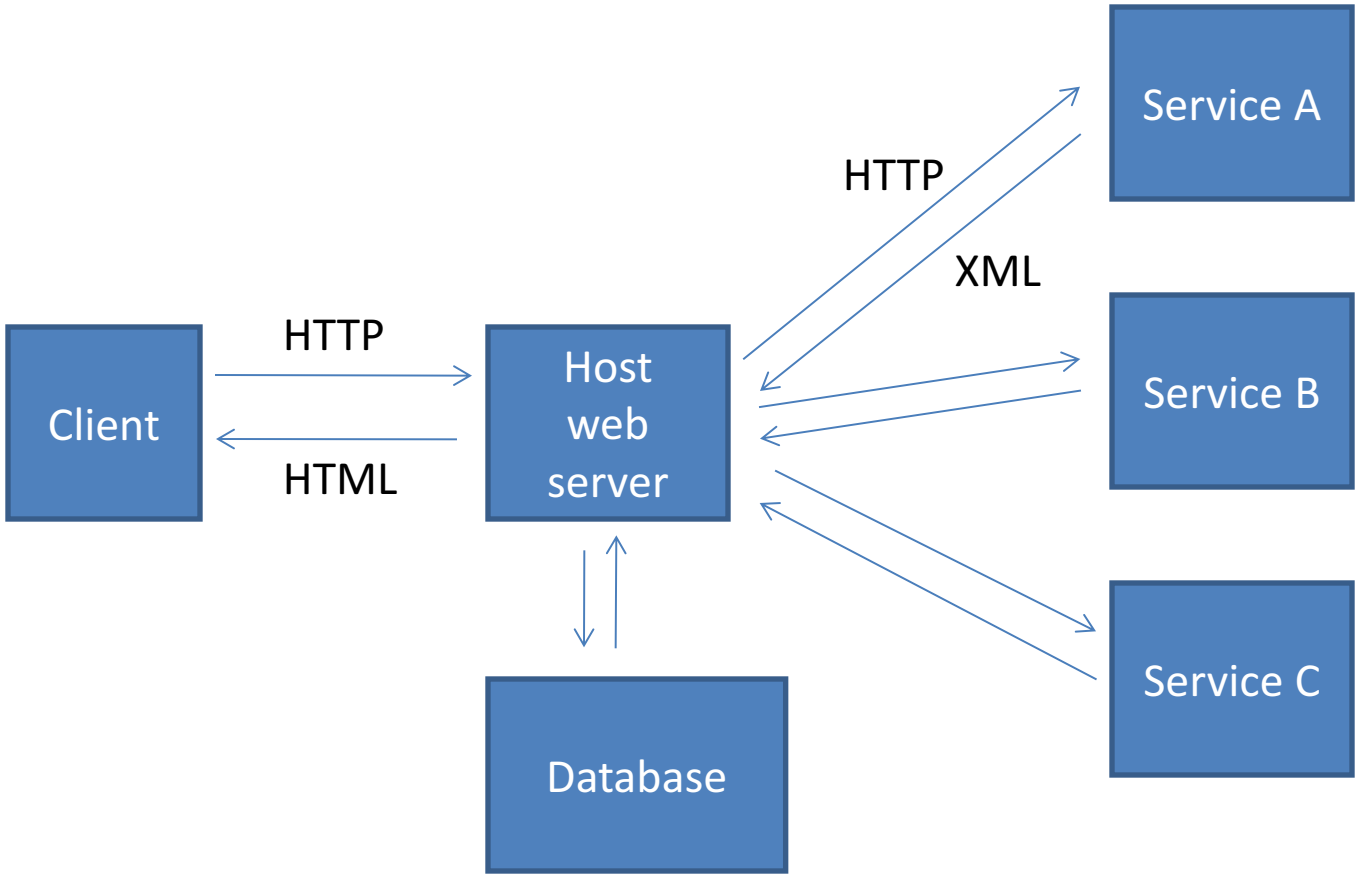
HTTP Responses

- 2xx = things are good
  - 200: OK, 201: created, 204: returning no content
- 3xx = redirect
  - Head(‘location … causes 300 class HTTP response from your server

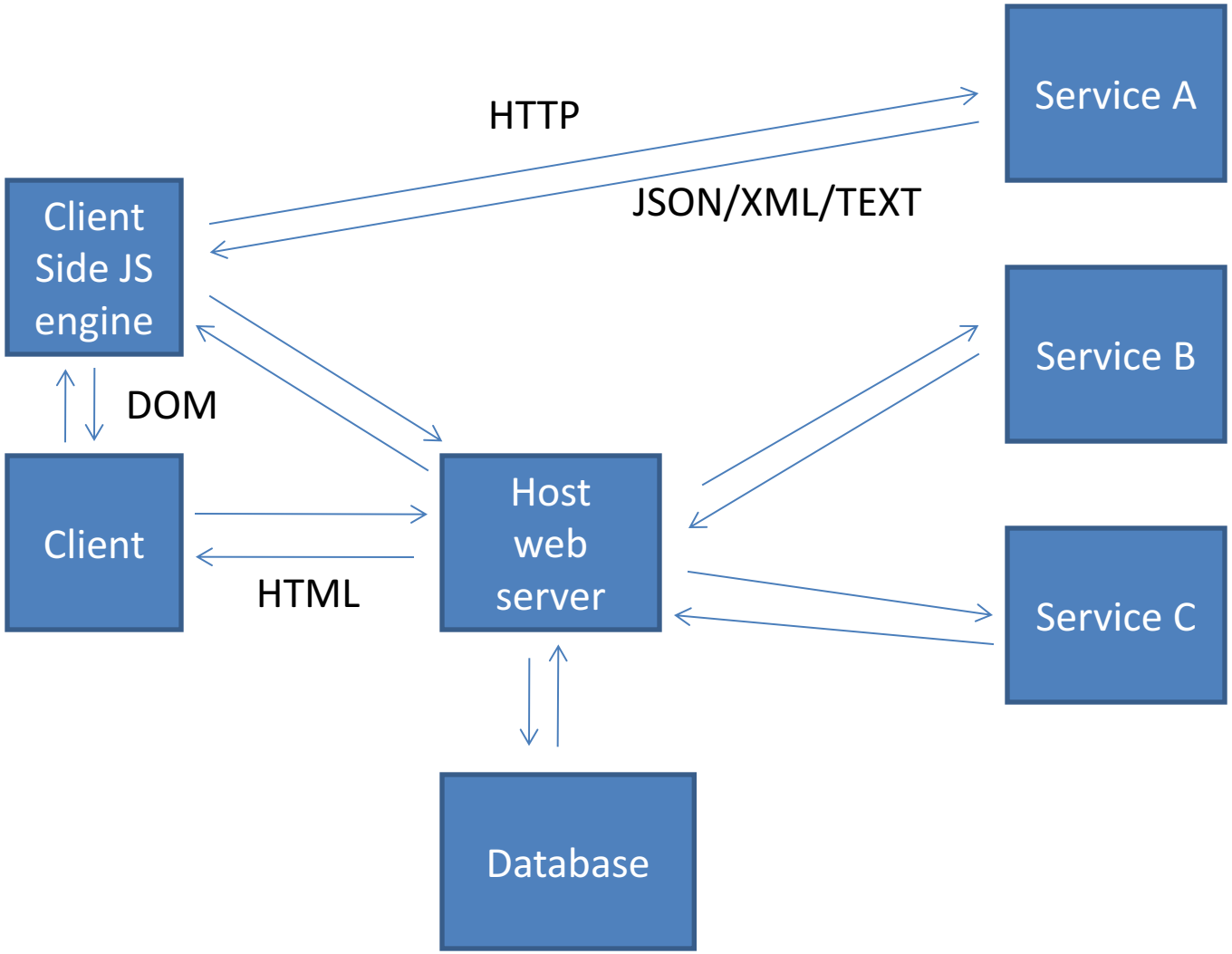
HTTP Responses

- 4xx = client error
  - 401: unauthorized, 403: forbidden, 404: not found, 411: length required
- 5xx = server error
  - 500: error, 501: not implemented, 503: Service Unavailable

Web Services…



AJAX Web Services…



An example

- Load up this URL in your browser (add an address)
- You could load this with simplexml_load_file()

http://maps.google.com/maps/geo?q=<<your address here>>&output=xml&sensor=false

Web Service Flavors

- SOAP
  - XML wrapper for HTTP data transfer
  - Can define any method (beyond Get, Post &c)
  - Lots of overhead
- REST
  - Lighter than SOAP
  - Only supports HTTP methods

REST

- Client gets a URL like

  example.com/products

- Server sends back XML

```
<products>
    <product id='1234'>A Chair</product>
    <product id='4958'>A Table</product>
</products>
```

REST

- Client gets a URL like

  example.com/products/1234

- Server sends back XML

  ….

```
<product id='1234'>
  <cost units='us dollars'>45.00</cost>
  <color>green</color>
</product>
```

REST

- Imagine the whole data structure is in XML
  - (but really, we are just getting data out of any database)
- Subfolders in the URL move in the complete pretend XML model like XPATH
- Snippets of XML are returned
- Examples:

http://www.flickr.com/photos/blahblahblah/4355665120/

http://www.flickr.com/photos/96592303@N00/4387898659/sizes/l/

GET / POST Examples

- Generate your own HTTP requests from PHP to other servers/services:
- Easy to use Basic HTTP auth between services

http://php.net/manual/en/function.fsockopen.php

Listen for GET or POST

- Apache/PHP Already does that!
- Requests don’t have to come from web browsers
- Get data via $_GET[‘varname’] and $_POST[‘varname’]
- Output XML or JSON (or anything…but structure makes it easiest for the people using your web service)

JSON

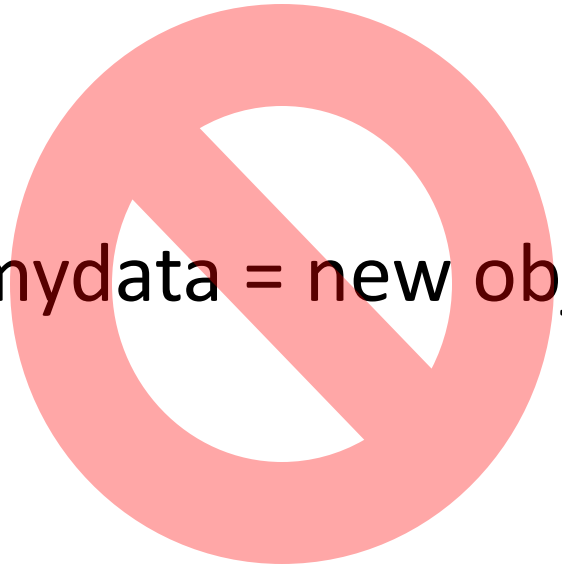
Java Script Object Notation

JSON is

- A lightweight data interchange format
- A replacement for XML
- Human readable
- Supports both hierarchical and unordered data
- Supported by most programming languages
- Supported by many modern web services
  - Flickr, Blogger, &c

Declaring a JSON object

```
var mydata = {};
```



```
var mydata = new object{}
```

Name Value Pairs (unorderd)

```
var js =
   {“runs” : “client”,
   “released” : 1995”,
   “type” : “scripting”};
```

Adding Depth

```
var lang = {
   js : {“runs” : “client”,
         “released” : 1995”,
         “type” : “scripting”},
   php: {“runs” : “server”,
         “released” : 1994”,
         “type” : “scripting”}
};
```

Arrays (ordered data)

```
var employers =
        {
        colleges : [“Sage”, “HVCC”, “RPI”]
        };
```

Nesting

- Can nest
  - arrays in arrays
  - arrays in the value part of name/value pairs
  - blocks of name/value pairs inside the value part of name/value pairs
  - blocks of name/value pairs inside arrays

Access JSON Data

- As an object

```
var blah = lang.js.released;
// blah = 1995
var blah = employers.colleges[1];
// blah = hvcc
```

Access JSON Data

- As an associative array

```
var blah = lang[“js”][“released”];
// blah = 1995
var blah = employers[“colleges”][1];
// blah = hvcc
```

Getting JSON Data

- XmlHttpRequest GET
  - For JSON resources on your service
- Call it from a script tag (within the src)
  - Used to get data from 3rd party services
  - Rewrite/reload script tag to get new data
  - Call backs……

Callback Functions

- Returns a json object as the first argument of a function call
- You write a function on your page to process json data
- You request a json object and specify a callback function name
- Service returns: callback_name(json_object)
- Once script loads, function automatically runs

Callback Functions

src = http://json.site.dom?callback=go

Response would be:

go({“blah” : [“blah”, blahblah, “blahblahblah”]});

PHP

json_encode
json_decode