JSON

lesson #advanced04

James L. Parry
B.C. Institute of Technology

JSON &.AJAX

JSON := JavaScript Object Notation

The current "darling" of representational "glue".

AJAX := Asynchronous Javascript And Xml

Fancy name given to client-side service calling.

Agenda

JSON - WHATZIT?

We are looking at "standard glue":

Transport: HTTP, HTTPS, **AJAX**

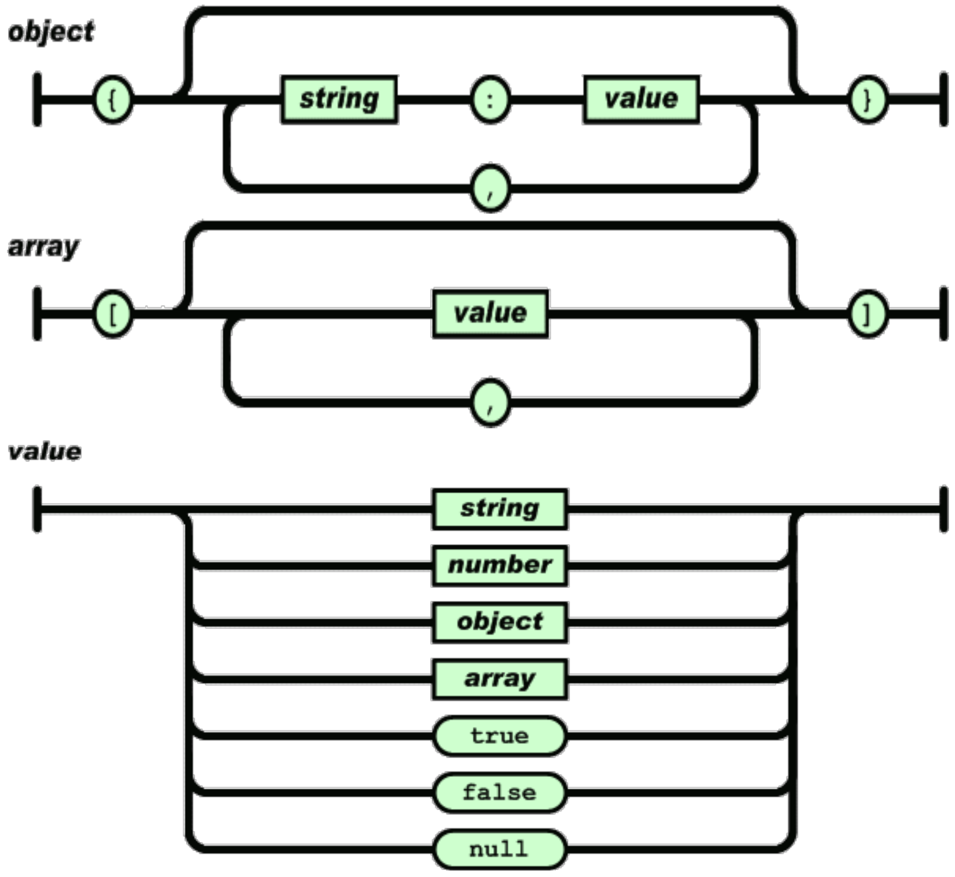
Representation: XML, **JSON**, RSS

Application: RPC, SOA, REST

JSON...

- JSON := JavaScript Object Notation
- Open standard for human-readable data exchange
- Used for simple data structures and associative arrays (called objects)
- *JSON Schema exists, with additional data types, but not W3C/IETF blessed (only a draft)*

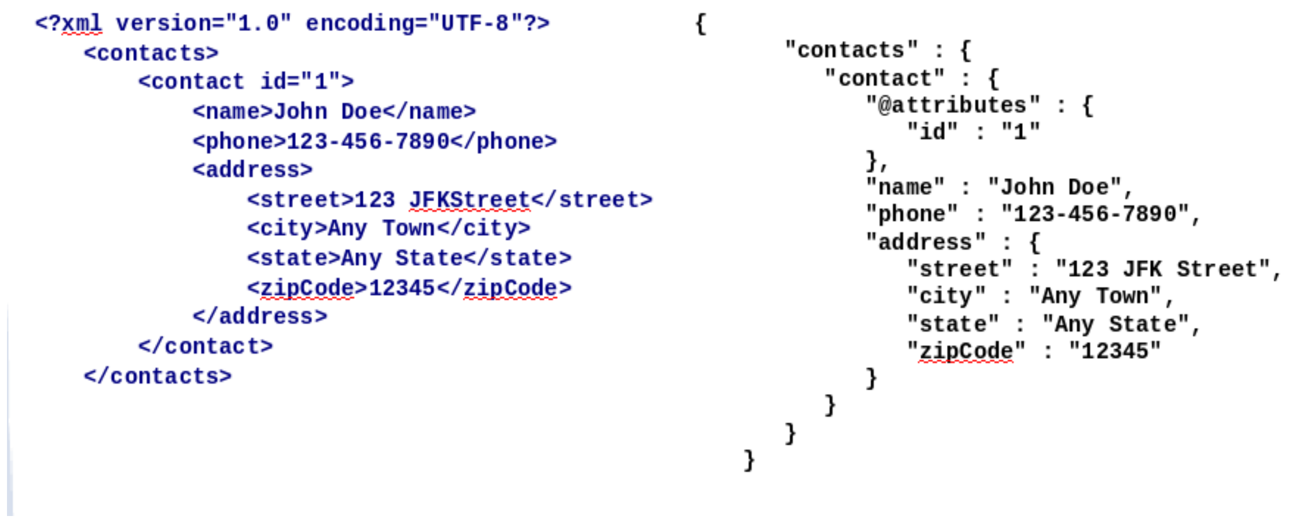
JSON Defined



JSON Example

```
{
"firstName": "John",
"lastName" : "Smith",
"age"      : 25,
"address"  :     {
         "streetAddress": "21 2nd Street",
         "city"         : "New York",
         "state"        : "NY",
         "postalCode"   : "10021"
     },
"phoneNumber":     [
         { "type"  : "home",
           "number": "212 555-1234"
         },
         { "type"  : "fax",
           "number": "646 555-4567"
         }
     ]
 }
```

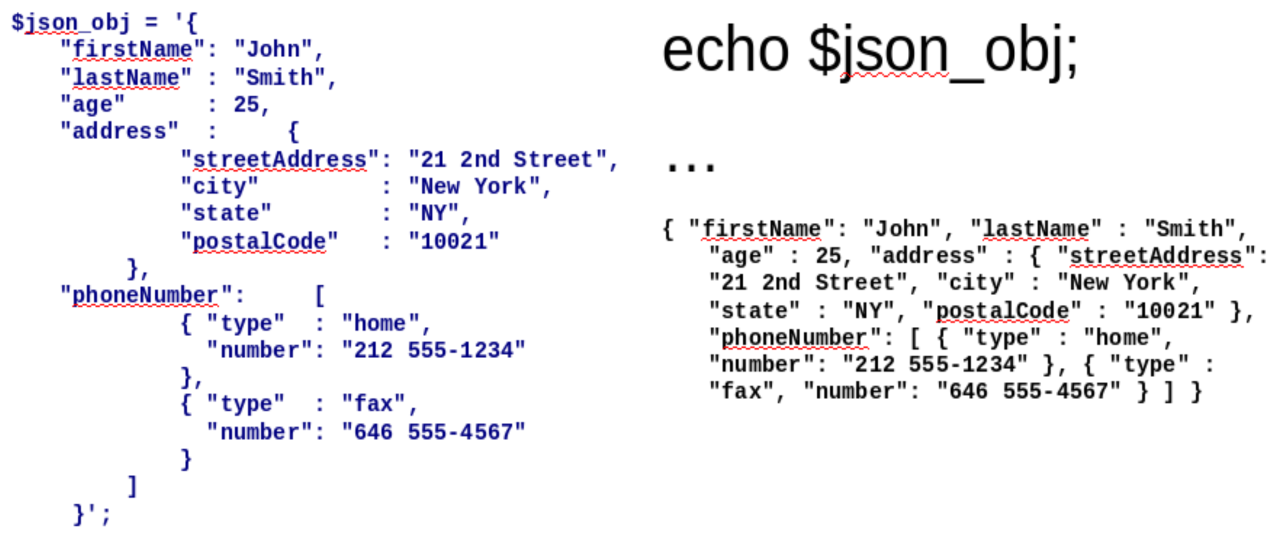
CONVERTING TO/FROM JSON



```
<?xml version="1.0" encoding="UTF-8"?>
  <contacts>
     <contact id="1">
        <name>John Doe</name>
        <phone>123-456-7890</phone>
        <address>
           <street>123 JFKStreet</street>
           <city>Any Town</city>
           <state>Any State</state>
           <zipCode>12345</zipCode>
        </address>
     </contact>
  </contacts>
```

```
{
      "contacts" : {
        "contact" : {
           "@attributes" : {
              "id" : "1"
           },
           "name" : "John Doe",
           "phone" : "123-456-7890",
           "address" : {
              "street" : "123 JFK Street",
              "city" : "Any Town",
              "state" : "Any State",
              "zipCode" : "12345"
           }
        }
      }
   }
```

JSON On Its Own



```
$json_obj = '{
   "firstName": "John",
   "lastName" : "Smith",
   "age"      : 25,
   "address"  :     {
            "streetAddress": "21 2nd Street",
            "city"         : "New York",
            "state"        : "NY",
            "postalCode"   : "10021"
        },
   "phoneNumber":     [
            { "type"  : "home",
              "number": "212 555-1234"
            },
            { "type"  : "fax",
              "number": "646 555-4567"
            }
        ]
    }';
```

```
echo $json_obj;

...

{ "firstName": "John", "lastName" : "Smith",
   "age" : 25, "address" : { "streetAddress":
   "21 2nd Street", "city" : "New York",
   "state" : "NY", "postalCode" : "10021" },
   "phoneNumber": [ { "type" : "home",
   "number": "212 555-1234" }, { "type" :
   "fax", "number": "646 555-4567" } ] }
```

JSON <-> Object

```
$json2 =
   json_decode($json_obj,false);

echo $json2;

object(stdClass)#1 (5)
   { ["firstName"]=>  string(4) "John"
   ["lastName"]=>  string(5) "Smith"
   ["age"]=>  int(25) ["address"]=>
   object(stdClass)#2 (4)
   { ["streetAddress"]=>  string(13)
   "21 2nd Street" ["city"]=>
   string(8) "New York" ["state"]=>
   string(2) "NY" ["postalCode"]=>
   string(5) "10021" }
   ["phoneNumber"]=>  array(2) { [0]=>
   object(stdClass)#3 (2) { ["type"]=>
   string(4) "home" ["number"]=>
   string(12) "212 555-1234" } [1]=>
   object(stdClass)#4 (2) { ["type"]=>
   string(3) "fax" ["number"]=>
   string(12) "646 555-4567" } } }
```

```
$json5 = json_encode($json2);

echo var_dump($json5);

string(239)
   "{"firstName":"John","lastName":"Smi
   th","age":25,"address":
   {"streetAddress":"21 2nd
   Street","city":"New
   York","state":"NY","postalCode":"100
   21"},"phoneNumber":
   [{"type":"home","number":"212
   555-1234"},
   {"type":"fax","number":"646
   555-4567"}]}"
```

JSON <-> Array

```
$json3 =
   json_decode($json_obj,true);

echo $json3;

array(5) { ["firstName"]=>  string(4)
   "John" ["lastName"]=>  string(5)
   "Smith" ["age"]=>  int(25)
   ["address"]=>  array(4)
   { ["streetAddress"]=>  string(13)
   "21 2nd Street" ["city"]=>
   string(8) "New York" ["state"]=>
   string(2) "NY" ["postalCode"]=>
   string(5) "10021" }
   ["phoneNumber"]=>  array(2) { [0]=>
   array(2) { ["type"]=>  string(4)
   "home" ["number"]=>  string(12) "212
   555-1234" } [1]=>  array(2)
   { ["type"]=>  string(3) "fax"
   ["number"]=>  string(12) "646
   555-4567" } } }
```

```
$json4 = json_encode($json3);

echo var_dump($json4);

string(239)
   "{"firstName":"John","lastName":"Smi
   th","age":25,"address":
   {"streetAddress":"21 2nd
   Street","city":"New
   York","state":"NY","postalCode":"100
   21"},"phoneNumber":
   [{"type":"home","number":"212
   555-1234"},
   {"type":"fax","number":"646
   555-4567"}]}"
```

XML -> JSON

```
$v1 = '<?xml version="1.0"
   encoding="UTF-8"?>
   <contacts>
       <contact id="1">
           <name>John Doe</name>
           <phone>123-456-7890</phone>
           <address>
               <street>123
   JFKStreet</street>
               <city>Any Town</city>
               <state>Any State</state>
               <zipCode>12345</zipCode>
           </address>
       </contact>
   </contacts>
   ';
```

```
$xml = simplexml_load_string($v1);

echo var_dump(json_encode($xml));

string(172) "{"contact":{"@attributes":
   {"id":"1"},"name":"John
   Doe","phone":"123-456-7890","address
   ":{"street":"123
   JFKStreet","city":"Any
   Town","state":"Any
   State","zipCode":"12345"}}}"
```

JSON -> XML

```
$v1 = '{"contact":{"@attributes":
   {"id":"1"},"name":"John
   Doe","phone":"123-456-7890","address
   ":{"street":"123
   JFKStreet","city":"Any
   Town","state":"Any
   State","zipCode":"12345"}}}';
```

```
$v2 = json_decode($v1,true);

// convert array to XML (coming up)
```

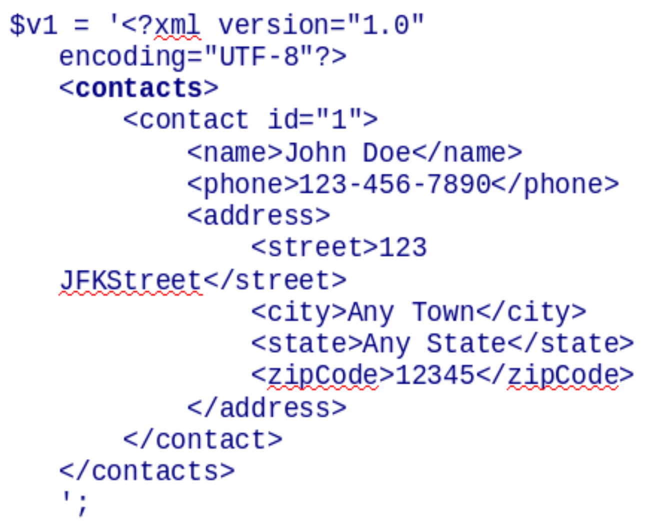
XML -> Object?

```
$v1 = '<?xml version="1.0"
   encoding="UTF-8"?>
   <contacts>
       <contact id="1">
           <name>John Doe</name>
           <phone>123-456-7890</phone>
           <address>
               <street>123
   JFKStreet</street>
               <city>Any Town</city>
               <state>Any State</state>
               <zipCode>12345</zipCode>
           </address>
       </contact>
   </contacts>
   ';
```

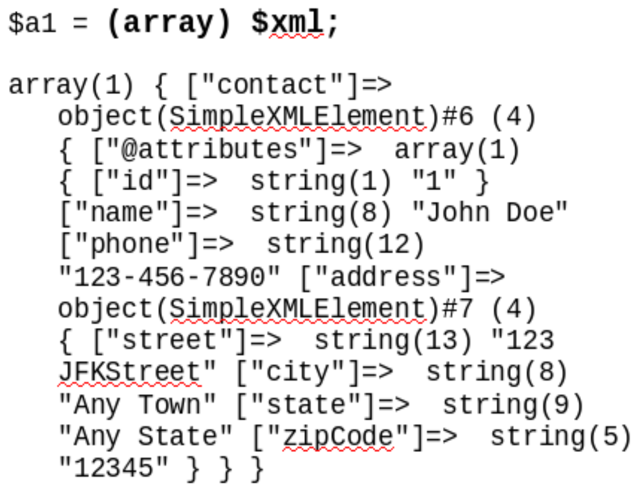
```
$a1 = (object) $v1;

object(SimpleXMLElement)#5 (1)
   { ["contact"]=>
   object(SimpleXMLElement)#10 (4)
   { ["@attributes"]=>  array(1)
   { ["id"]=>  string(1) "1" }
   ["name"]=>  string(8) "John Doe"
   ["phone"]=>  string(12)
   "123-456-7890" ["address"]=>
   object(SimpleXMLElement)#11 (4)
   { ["street"]=>  string(13) "123
   JFKStreet" ["city"]=>  string(8)
   "Any Town" ["state"]=>  string(9)
   "Any State" ["zipCode"]=>  string(5)
   "12345" } } }
```

XML -> Array



```
$v1 = '<?xml version="1.0"
   encoding="UTF-8"?>
   <contacts>
       <contact id="1">
           <name>John Doe</name>
           <phone>123-456-7890</phone>
           <address>
               <street>123
   JFKStreet</street>
               <city>Any Town</city>
               <state>Any State</state>
               <zipCode>12345</zipCode>
           </address>
       </contact>
   </contacts>
   ';
```



```
$a1 = (array) $xml;

array(1) { ["contact"]=>
   object(SimpleXMLElement)#6 (4)
   { ["@attributes"]=>  array(1)
   { ["id"]=>  string(1) "1" }
   ["name"]=>  string(8) "John Doe"
   ["phone"]=>  string(12)
   "123-456-7890" ["address"]=>
   object(SimpleXMLElement)#7 (4)
   { ["street"]=>  string(13) "123
   JFKStreet" ["city"]=>  string(8)
   "Any Town" ["state"]=>  string(9)
   "Any State" ["zipCode"]=>  string(5)
   "12345" } } }
```

XML <-> Array

No easy way, especially with attributes.

Suggestion: Separate class or library

Handles attributes :)

Works with PHP-DOM

May not be compatible with other converters :(

```
$array = XML2Array::createArray($xml);

$xmldoc =
Array2XML::createXML('root_node_name',$php_array
```

JSONP Caution

JSONP (JSON with padding) is the convention of wrapping a JSON payload inside a function call

JSON: `{"Name": "Foo", "Id": 1234, "Rank": 7}`

JSONP: `parseResponse({"Name": "Foo", "Id": 1234, "Rank": 7});`

Typical usage (script injection):

```
<script type="text/javascript"

   src="http://example.com/Users/1234?jsonp=parseResponse">

</script>
```

Considered dangerous because of XSS

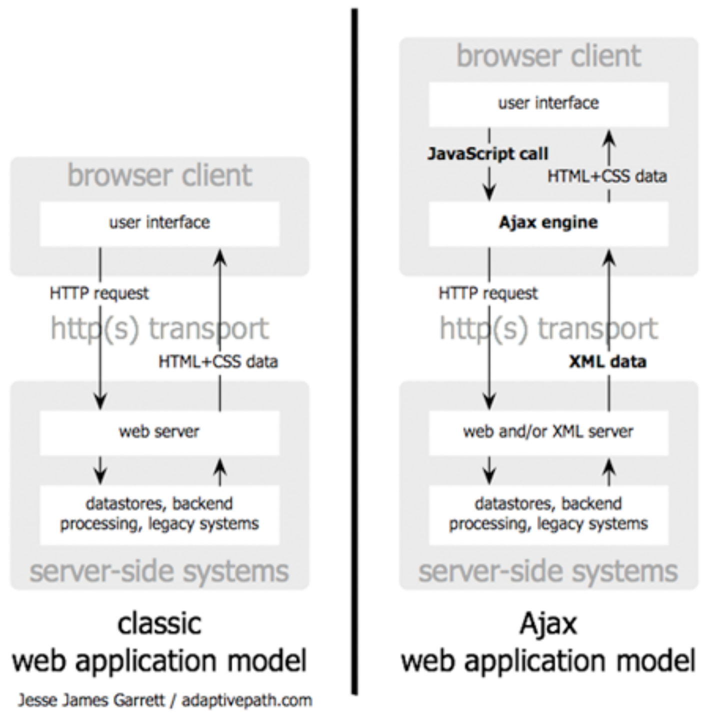
AJAX - WHATZIT?

- Asynchronous (content loading)
- Javascript (logic & control)
- And
- XML (request handling)

AJAX - Really?

- Set of cross-platform conventions and techniques to load data browser-side, without interfering with the display and behaviour of the existing page
- Relies on Javascript & the DOM, assumes XHTML & CSS used
- XMLHttpRequest object in browser

HTTP Transport with AJAX



XmlHttpRequest

- Browser object to retrieve content from server page was dished from
- Need a new such object for each request
- Uses a callback mechanism when the HTTP request state changes

AJAX ISSUES

- Browser integration: Back-button, Bookmark ease, Javascript enabled
- Reponse time concerns (separate HTTP requests, frequency)
- Reliance on Javascript & DOM (has to be enabled, must be well-formed)
- Web analytics (based on page requests)

Bottom line: use a Javascript framework with good support for it!

AJAX WITH CODEIGNTIER

Hard to distinguish an incoming AJAX request

Follow conventions, return "AJAX" object, not HTML page - designate a controller or sub-controller as one intended to handle an AJAX request.

Publish your own "protocol", eg. method X will return XML/JSON/HTML fragment

Congratulations!

You have completed lesson #advanced04: JSON

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

The next activity in sequence is: There is nothing further in this course

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.