

# User Interaction: Making AJAX

Assoc. Professor Donald J. Patterson  
INF 133 Fall 2014



```
<html>
  <head>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
    <script src="Step01.js"></script>
  </head>

  <body>
    <div class="dataHere">Replace this text!</div>
  </body>
</html>
```

QUERY

ASYNCHRONOUS

CALLBACK

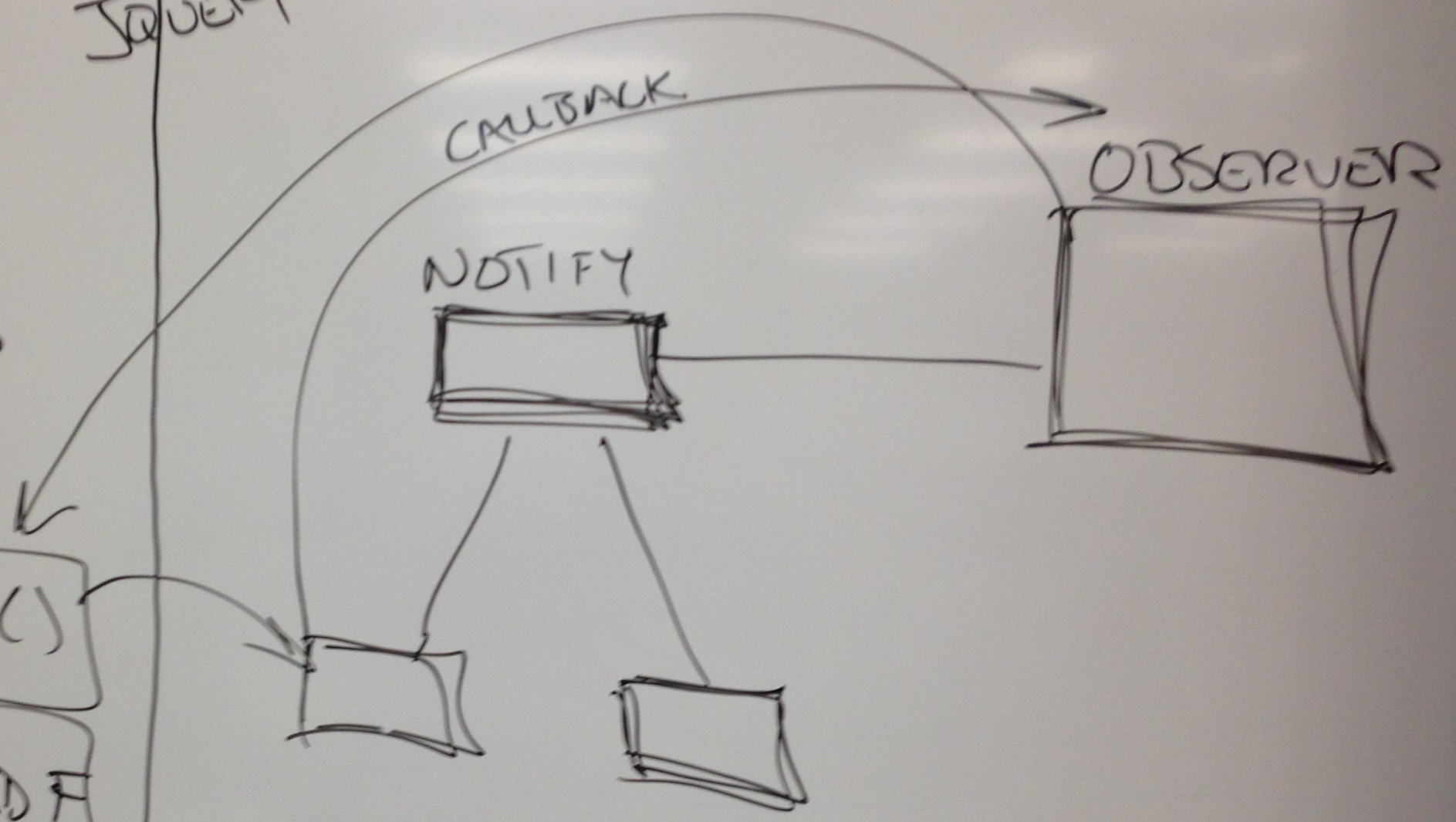
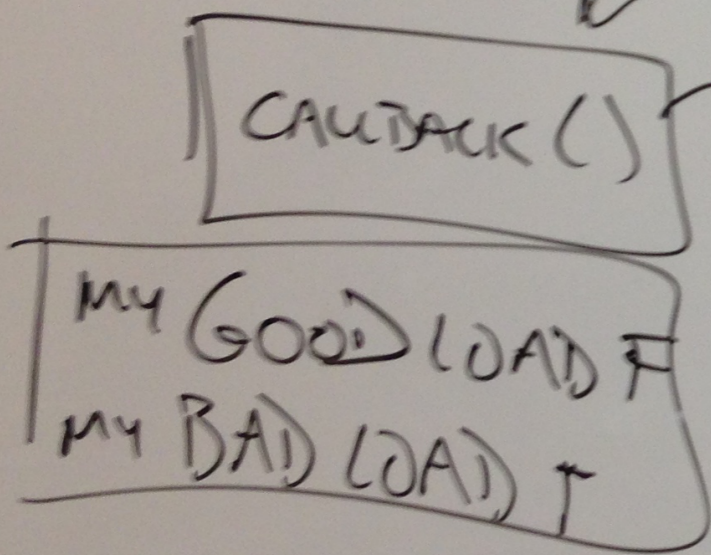
NOTIFY

OBSERVER

CALLBACK()

MY GOOD LOAD F

MY BAD LOAD T



# jQuery

```
function myGoodLoadFunction(data) {
    $("div.dataHere").text(data.HELLO);
}

function myBadLoadFunction(XMLHttpRequest, errorMessage, errorThrown) {
    alert("Load failed:"+errorMessage+": "+errorThrown);
}

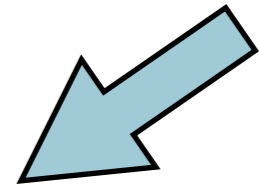
function myReadyFunction(){
    $.ajax({
        url: "/~djp3/classes/2014_09_INF133/Lectures/Lecture_09/test.json",
        dataType: "json",
        success: myGoodLoadFunction,
        error: myBadLoadFunction,
    });
}

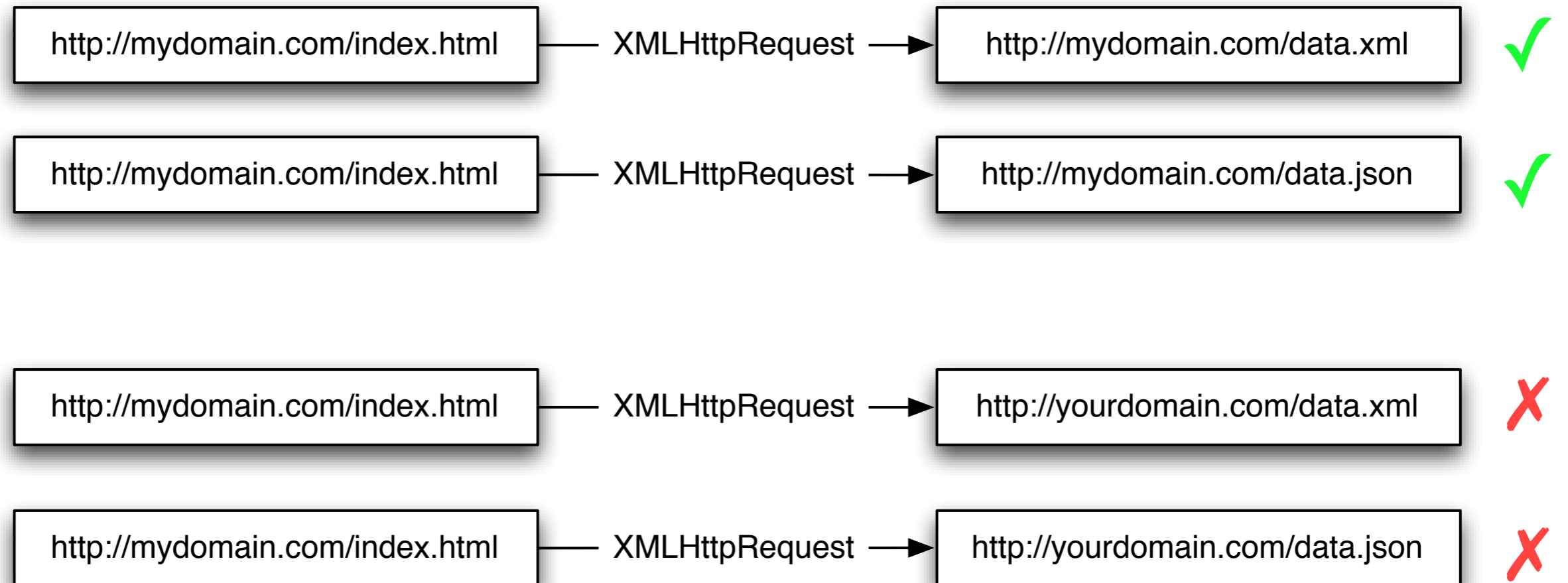
$(document).ready(
    function(){
        myReadyFunction();
    }
);
```

```
{
    "HELLO": "WORLD",
    "This": "is",
    "my": "data"
}
```

- Two problems:
  - Javascript does not allow you to make an **XMLHttpRequest** from an external domain
    - XMLHttpRequest is the javascript function which retrieves data asynchronously
      - Not just XML, but any data
      - For security reasons it was restricted
      - It is used by the jQuery **.ajax()** function
  - An http URL cannot request from an https URL and vice versa

What is this?





# Real-Time Web Assignment

- Solutions to the cross-domain security restriction
  - Run a “proxy”
    - a php proxy
    - You request
      - <http://mydomain.com/proxy.php?http://yourdomain.com/data.xml>
      - Javascript thinks its coming from local server
        - proxy.php contains something like this code:
  - php doesn't run on ics.uci.edu
    - You need to use “students.ics.uci.edu”
    - <http://www.ics.uci.edu/computing/web/faqs.php#students>

# Real-Time Web Assignment

- This is the contents of a file on your server
- It is running a language called php
- the language is interpreted by your web server code (dynamic web page)
- server-side scripting

```
<?php
//Get the requested url from our proxy url
$url = $_SERVER['QUERY_STRING'];

// initialize curl with given url
$ch = curl_init($url);

//Tell curl to write the response to a variable
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

//Tell curl to follow any redirects
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);

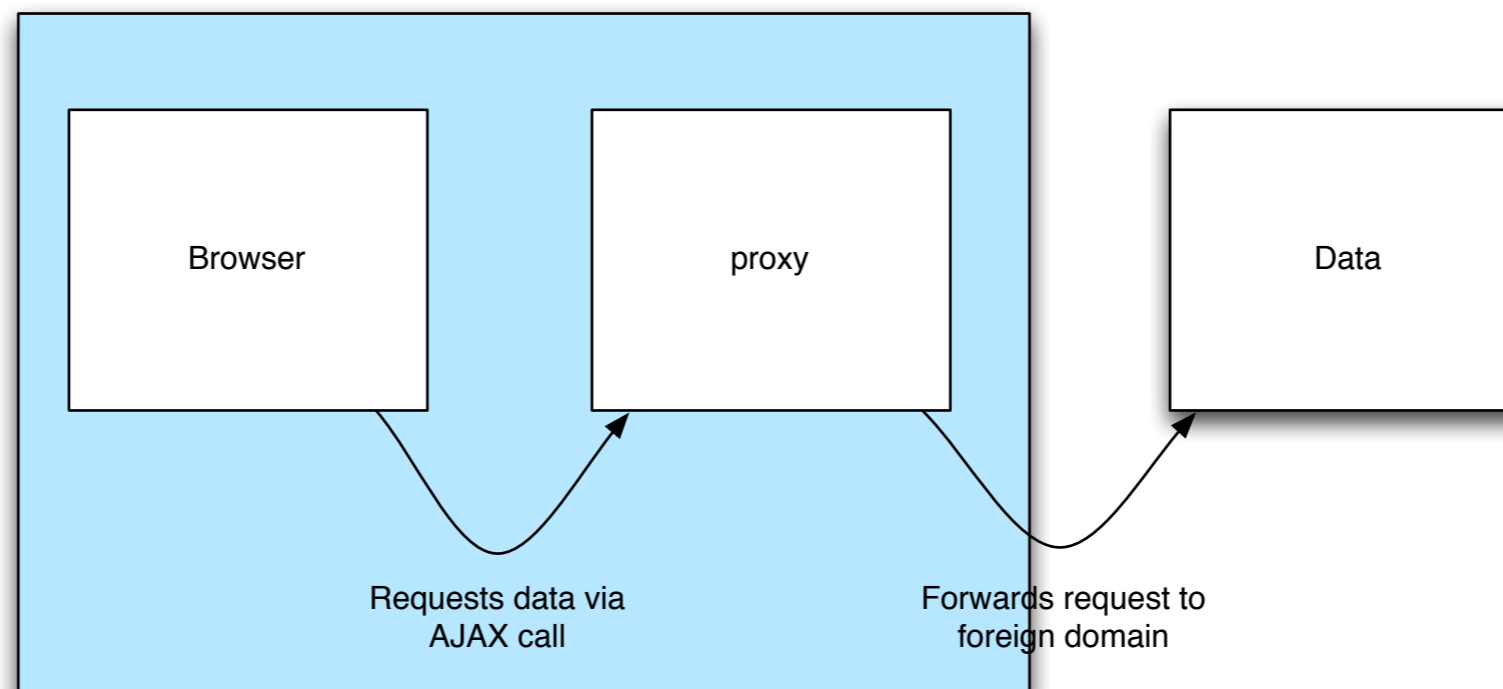
$response = curl_exec($ch);

$cType = curl_getinfo($ch,CURLINFO_CONTENT_TYPE);

//Output the right mime/type
header('Content-type: '.$cType);

//Pass through the data
print $response;

curl_close($ch);
?>
```





- Solutions to the cross-domain security restriction
  - Use JSONP
    - Requires server support
      - Yahoo APIs support this
  - Leverages Javascript loophole
    - XMLHttpRequests are restricted
    - Remote Javascript is not



## Real-Time Web Assignment

- JSONP adds a `<script src=>` element to your web page with the external URL as the src target
- Getting JSON looks like this:
  - Request: `http://yourdomain.com/data.json`
  - Return: `{"hello":"world"}`
- Getting JSONP looks like this:
  - Request `http://yourdomain.com/data.jsonp?callback=myCallback`
  - Return: `myCallback("{\"hello\":\"world\"}')`
- Your webpage writes a function called `myCallback` to deal with the data

- Security issue
  - You are running server generated code on your machine
  - !



```
<html>
  <head>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
    <script src="Step02.js"></script>
  </head>

  <body>
    <div class="dataHere">Replace this text!</div>
    <div class="textHere">Don't replace this text!</div>
    <div class="textAlsoHere">Don't replace this text either</div>
  </body>
</html>
```

# jQuery

```
function escapeText(t){
    return document.createTextNode(t).textContent;
}

function jsonFlickrFeed(data) {
    myGoodLoadFunction(data);
}

function myGoodLoadFunction(data) {
    var newHTML= "<div style=\"padding:20px;border:solid 1px black\">"+escapeText(data.title)+"<br/>";
    newHTML += escapeText(data.link)+"<br/>";
    newHTML += "<ul>"
    newHTML += "<li>"+escapeText(data.items[0].link)+"</li>";
    newHTML += "<li>"+escapeText(data.items[1].link)+"</li>";
    newHTML += "</ul></div>";
    $("div.dataHere").html(newHTML);
}

function myBadLoadFunction(XMLHttpRequest,errorMessage,errorThrown) {
    alert("Load failed:"+errorMessage+": "+errorThrown);
}

function myReadyFunction(){
    $.ajax({
        url: "http://api.flickr.com/services/feeds/geo/QDd_2PObCZ4ZsRM6Sw&format=json",
        dataType: "jsonp",
        jsonp: false,
        jsonpCallback: "jsonFlickrFeed",
        error: myBadLoadFunction,
    });
}

$(document).ready(
    function(){
        myReadyFunction();
    }
);
```

- What if you have something other than JSON to load?
  - XML
  - jQuery.parseXML()
- Where can I find info about jQuery actions?
  - jQuery API Reference
  - for example, `html()` vs `text()` vs `append()`?

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>jQuery.parseXML demo</title>
6   <script src="//code.jquery.com/jquery-1.10.2.js"></script>
7 </head>
8 <body>
9
10 <p id="someElement"></p>
11 <p id="anotherElement"></p>
12
13 <script>
14 var xml = "<rss version='2.0'><channel><title>RSS Title</title></channel>";
15   xmlDoc = $.parseXML( xml ),
16   $xml = $( xmlDoc ),
17   $title = $xml.find( "title" );
18
19   // Append "RSS Title" to #someElement
20   $( "#someElement" ).append( $title.text() );
21
22   // Change the title to "XML Title"
23   $title.text( "XML Title" );
24
25   // Append "XML Title" to #anotherElement
26   $( "#anotherElement" ).append( $title.text() );
27 </script>
28
29 </body>
30 </html>
```

# jQuery - Final example Step03

```
<html>
  <head>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
    <script src="Step03.js"></script>
  </head>

  <body>
    <div class="dataXML">
      <br/>
      Replace this with XML
    </div>
    <hr/>
    <div class="dataJSON">
      <br/>
      Replace this with JSON
    </div>
    <hr/>
    <div class="dataJSONP">
      <br/>
      Replace this with JSONP
    </div>
  </body>
</html>
```



```
function myBadLoadFunction(myXMLHttpRequest,myErrorMessage,myErrorThrown) {
    alert('status: ' + myErrorMessage + '\n' + myXMLHttpRequest.responseText);
}

function myReadyFunction(){
    $.ajax({
        url: "https://students.ics.uci.edu/~djp3/myProxy.php?http://api.flickr.com/service",
        dataType: "xml",
        success: myGoodLoadXML,
        error: myBadLoadFunction
    });
    $("div.dataXML").html("AJAX XML call initiated:<br/>");

    $.ajax({
        url: "https://students.ics.uci.edu/~djp3/myProxy.php?https://graph.facebook.com/c",
        dataType: "json",
        success: myGoodLoadJSON,
        error: myBadLoadFunction
    });
    $("div.dataJSON").html("AJAX JSON call initiated:<br/>");

    $.ajax({
        url: "https://gdata.youtube.com/feeds/base/users/djp3/uploads?alt=json-in-script&",
        dataType: "jsonp",
        success: myGoodLoadJSONP,
        error: myBadLoadFunction
    });
    $("div.dataJSONP").html("AJAX JSONP call initiated:<br/>");
}

$(document).ready(
    myReadyFunction
);
```

# jQuery - XML

```
function escapeText(t){
    return document.createTextNode(t).textContent;
}

function myGoodLoadXMLHelper()
{
    var newHTML= "<div style=\"margin-left:10px;padding:5px;border:solid 1px red\">";
    newHTML += escapeText($(this).text());
    newHTML += "</div>";
    $("div.dataXML").append(newHTML);
}

function myGoodLoadXML(data) {
    $("div.dataXML").html("<h1>AJAX XML call returned:</h1>");
    $(data).find("item title").each(myGoodLoadXMLHelper);
}
```

# jQuery - JSON

```
function myGoodLoadJSON(data) {
    $("div.dataJSON").html("<h1>AJAX JSON call returned:</h1>");

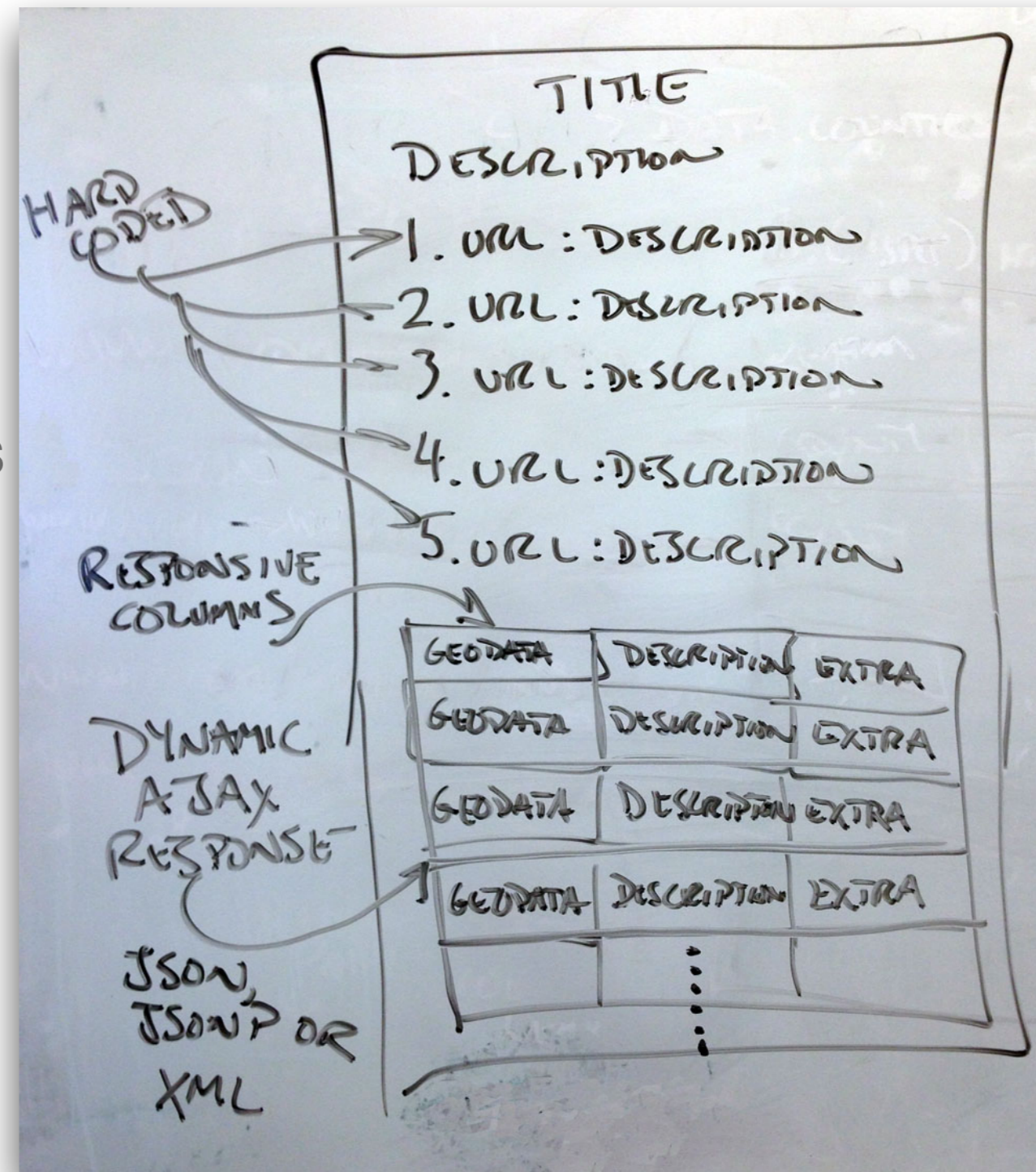
    var newHTML= "<div style=\"margin-left:10px;padding:5px;border:solid 1px green\">";
    newHTML += escapeText(data.name);
    newHTML += "</div>";
    $("div.dataJSON").append(newHTML);
}
```

# jQuery - JSONP

```
function myGoodLoadJSONP(data) {
    $("div.dataJSONP").html("<h1>AJAX JSONP call returned:</h1>");

    for(var i = 0; i < data.feed.entry.length; i++){
        var newHTML= "<div style=\"margin-left:10px;padding:5px;border:solid 1px blue\">";
        newHTML += escapeText(data.feed.entry[i].title.$t);
        newHTML += "</div>";
        $("div.dataJSONP").append(newHTML);
    }
}
```

- Real-time Web Assignment
  - Present the data as responsive bootstrap rows
    - <http://getbootstrap.com/css/#grid>



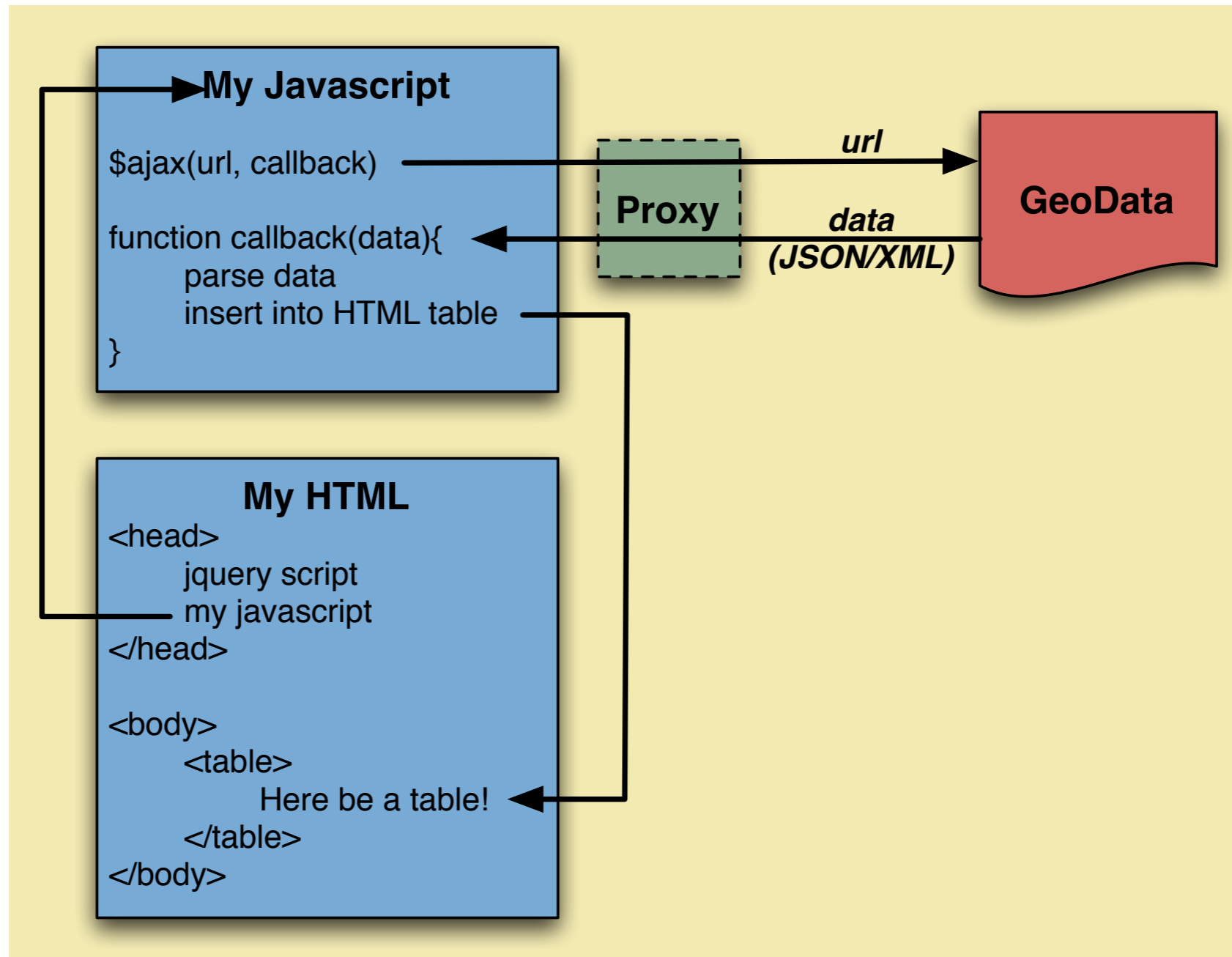
- Real-time web
  - Geocoded Feeds
    - [http://api.flickr.com/services/feeds/geo/QDd\\_2PObCZ4ZsRM6Sw&format=json](http://api.flickr.com/services/feeds/geo/QDd_2PObCZ4ZsRM6Sw&format=json)
    - Use dynamic data, not static data
  - jQuery
  - Bootstrap responsive columns
  - AJAX request

- The goal is a web page
  - which has issued an AJAX request
  - parsed the data
  - and displayed it in a new responsive way

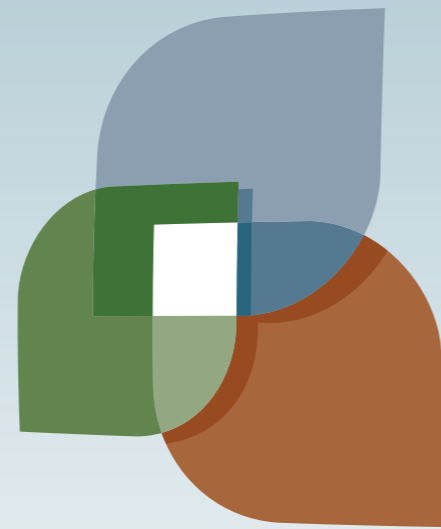
- Find some geo-data in XML or JSON format
  - (Hint: RSS is a specific type of XML)
- What is geo-data?
  - Anything that relates data to a spot on the earth
  - Data with a latitude and longitude
  - Data with an address
  - Data with a zip code
  - Data with a county
- For example:
  - Photos taken in Los Angeles
  - <http://www.flickr.com/places/United+States/California/Los+Angeles>



# Real-time web strategy



- At a high-level requesting data asynchronously requires:
  - The location of where you want to get the data from
    - (for us these are the 5 geo-feeds)
  - What function to call when the data is ready
    - Because we aren't waiting
    - aka, "the call back function"
  - The native function call to do this is called
    - "XMLHttpRequest"



L U C I

