Web and Mobile Application Development

Web Services

Material created by:
David Augenblick, Bill Mongan, Dan Ziegler, Samantha Bewley, and Matt Burlick

Obtaining Data

- When we looked at Ajax we saw how to get data from an external source
    - Pretty cool!
- Ultimately in the course we'll look at two ways to get data for our sites:
    1. Get the response from a *web service* which might give us nicely formatted data to use (JSON?)
    2. Get the data from a server and/or database (we'll do this later once we introduce servers and databases).

Web Services

- The example we just did with AJAX was an example of using a *web services*
- Web services provide a safe and easy way to request data from a different domain.
- Remember our Ajax call to get JSON from http://jsonplaceholder.typicode.com/posts/1
- Here’s the full JSON response

```
{
	"userId": 1,
	"id": 1,
	"title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
	"body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
}
```

Drexel University - CS275

Weather Web Service Example

- Let’s do a more interesting example of using a web service
  - We’re going to get weather information and use it to populate our website.

Weather Web Service Call

- Let’s use the Open Weather Map web service to obtain the current temperature.
- The API can be found at https://openweathermap.org/api
- You’ll first need to create an account
- Then you’ll need to “buy” a free key at
  https://openweathermap.org/price

Weather Web Service Call

- The “Documentation” link is on this page
  https://openweathermap.org/api
- The “current conditions” call is
  https://api.openweathermap.org/data/2.5/weather?zip={zip code}&appid={your key}
- Where
    - YOUR KEY is your wunderground key
    - ZIPCODE is the zipecode for which to get the current conditions

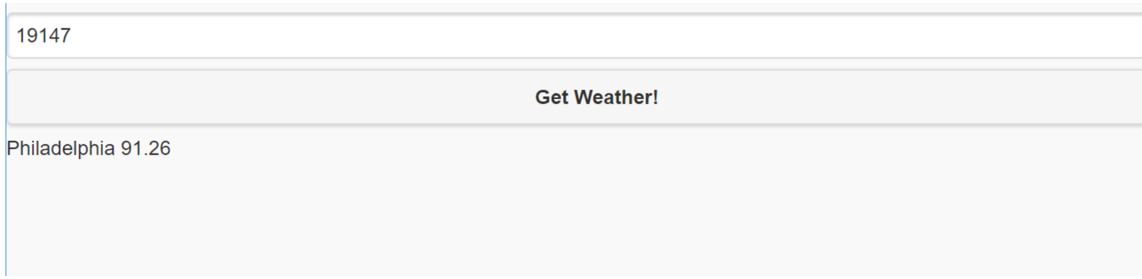
Example: Weather

- Following that URL, the API will give us a JSON response that looks something like below

```
{"coord":{"lon":-122.09,"lat":37.39},
"weather":[{"id":500,"main":"Rain","description":"light rain","icon":"10d"}],
"base":"stations",
"main":{"temp":280.44,"pressure":1017,"humidity":61,"temp_min":279.15,"temp_max":281.15},
"visibility":12874,
"wind":{"speed":8.2,"deg":340,"gust":11.3},
"clouds":{"all":1},
"dt":1519061700,
"sys":{"type":1,"id":392,"message":0.0027,"country":"US","sunrise":1519051894,"sunset":1519091585},
"id":0,
"name":"Mountain View",
"cod":200}
```

Weather Example

- Ahh now we can finally do some cool stuff!
- Let's make a webpage that has
    - A text box
    - A button
    - An empty `div`
- When the user clicks the button
    - Construct an HTTP request based on the content of the text box
    - Send that request asynchronously via an AJAX
    - When the request comes back, populate the `div` based on what came back!
- NOTE: Before sending the request we should probably to some regular expression checking to make sure there’s a valid ZIP in the textbox. But that’s not the focus so we’ll omit that.



```
Philadelphia 91.26
```

Weather Example

- First let’s create the HTML content

```
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/jquerymobile/1.4.5/jquery.mobile.min.js">
</script>
<link type="text/css" rel="stylesheet"
href="http://ajax.googleapis.com/ajax/libs/jquerymobile/1.4.5/jquery.mobile.min.css"/>
<meta name="viewport" content="width=device-width, initial-scale=1">

<script>
    function requestWeather(){
        //todo (next slide)
    }
</script>
</head>

<body>
<input type=text id="zipcode" placeholder="Enter desired zip code here"/>
<input type=button onclick="requestWeather()" value="Get Weather!"/><div id=current></div>
</body>
</html>
```

Weather Example

- Now let’s implement that `requestWeather` function
  - Note: It’s probably not best to hard-code in your API key (others can see/get it). But for security JS can’t read from a file. So we could either
    - Have the user enter theirs in an other input field (probably a pain)
    - Have our server do the processing and store our key there
      - We’ll do this once we get to server-side processing.
    - But for now we’ll hard-code it ☹

Weather Example

```
<script>
function requestWeather(){
	var zip = $("#zipcode");
	var code = 'YOURCODE';
	var URL = "https://api.openweathermap.org/data/2.5/weather?zip=" + zip.val() + "&appid=" +
			code+"&units=imperial";
	$.ajax({
		type: "GET",
		url : URL,
		dataType : "jsonp",
		success : function(msg){
			var json = msg;
			if(json.cod=200){
				var city = json.name;
				var temp = json.main.temp;
				document.getElementById("current").innerHTML=city+" " + temp + "F";
			}
			else
				document.getElementById("current").innerHTML="ERROR";
		},
		error: function(jgXHR, textStatus,errorThrown){
			alert("Error: " + textStatus + " " + errorThrown);
		}
	});
}
</script>
```