Introduction to AJAX & jQuery

Jon Tedesco

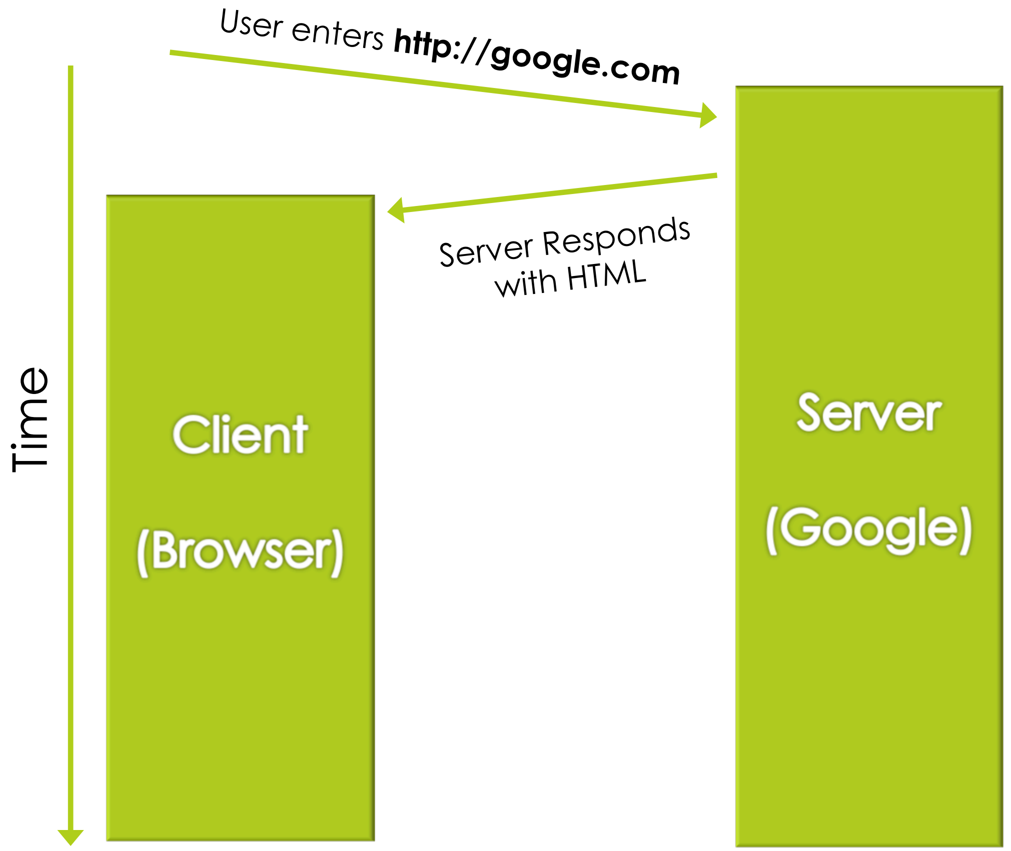
Outline

- Web Requests
  - GET &POST
- AJAX Basics
- JavaScript
  - Overview
  - Callbacks
  - AJAX
- jQuery
  - Features
- Demo
- Assignment Usage

Web Requests

GET & POST, Client & Server Perspectives

A Basic Web Request



- Client (Browser)
  - Submit single web request
  - Receive & render HTML
- Server
  - Receive request
  - Return HTML
- Static
  - One request, no further interaction

GET & POST: Summary

GET

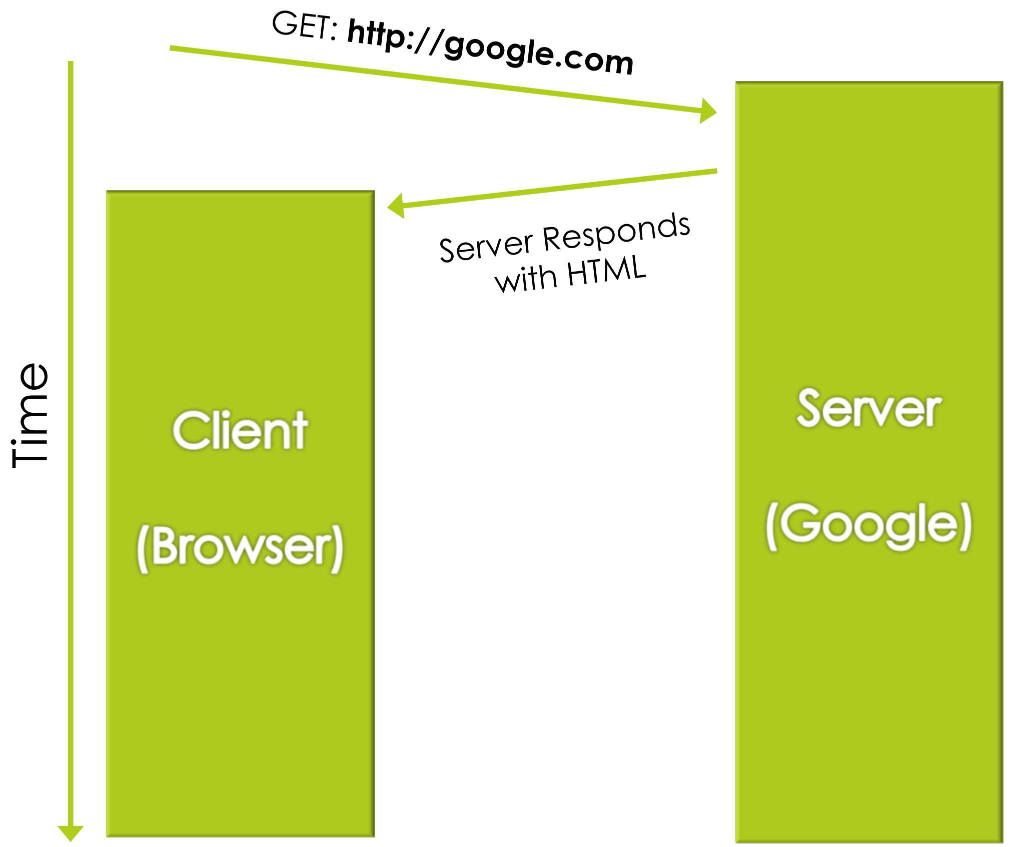
- Retrieve data
- Display data in URL
- Example
  - Fetching a web page

POST

- Modify or submit data
- Data hidden from URL
- Example
  - Submitting a form

- Either allow passing parameters
  - How does each behave? When should we use one over the other?

A Basic Web Request



- GET
  - Most basic request
  - Sent by browser for fetching a page
- POST
  - Often used for form data
- Why?
  - Simple
  - Fast
  - Idempotent

GET & POST: Characteristics

GET

- More usable
  - GET requests can be cached
  - GET requests can remain in the browser history
  - GET requests can be bookmarked
  - GET requests can be distributed & shared
- URL length
  - Prohibitive for large requests

POST

- More complex
  - Slower
- Secure (sort of)
  - Parameters are not shown in URL
- Better for large amounts of data
- Not sharable

GET & POST: Best Practices

GET

- Use for idempotent actions
- Use for AJAX
  - Fast
- Use for sharing links
- Example
  - YouTube
  - Google

POST

- Use for actions with side effects
  - Potential modify or delete
- Use for large data
- Use for sensitive data
- Example
  - Comments

GET & POST: From PHP

GET

myserver/endpoint.php?foo=1&bar=word

```
<?php
  $foo = $_GET['foo']; // 1
  $bar = $_GET['bar']; // 'word'
  ...
  echo "Response text...";
?>
```

POST

myserver/endpoint.php

```
<?php
  $foo = $_POST['foo']; // 1
  $bar = $_POST['bar']; // 'word'
  ...
  echo "Response text...";
?>
```

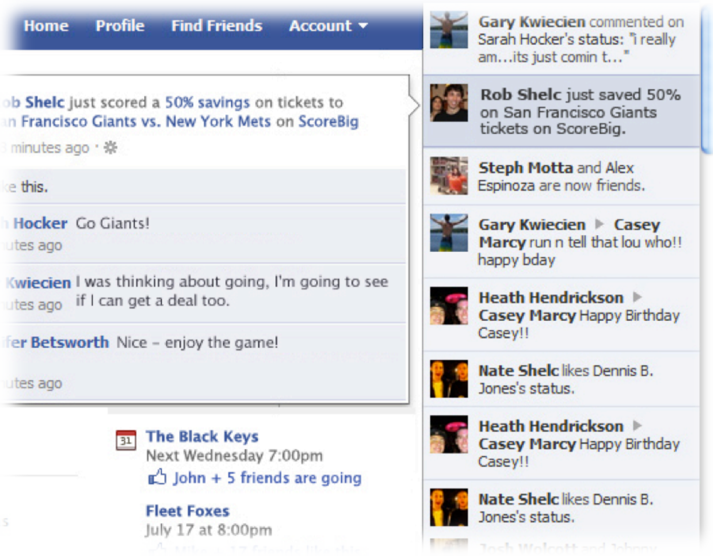
- Received as associative array in PHP
- Consider characteristics of each

AJAX

A high-level introduction

AJAX: Overview

- What is it?
  - Use JavaScript to send web requests after page load
  - **A**synchronous **J**avaScript **A**nd **X**ML
- Why use it?
  - Make site more interactive
  - Update content without reloading
- Example
  - Facebook Ticker



ob **Shelc** just scored a 50% savings on tickets to
n Francisco Giants vs. New York Mets on ScoreBig
minutes ago · ✻

e this.

Hocker Go Giants!
utes ago

Kwiecien I was thinking about going, I'm going to see if I can get a deal too.
utes ago

fer Betsworth Nice – enjoy the game!
utes ago

The Black Keys
Next Wednesday 7:00pm
John + 5 friends are going

Fleet Foxes
July 17 at 8:00pm

Gary Kwiecien commented on Sarah Hocker's status: "i really am...its just comin t..."

Rob Shelc just saved 50% on San Francisco Giants tickets on ScoreBig.

Steph Motta and Alex Espinoza are now friends.

Gary Kwiecien ▸ **Casey Marcy** run n tell that lou who!! happy bday

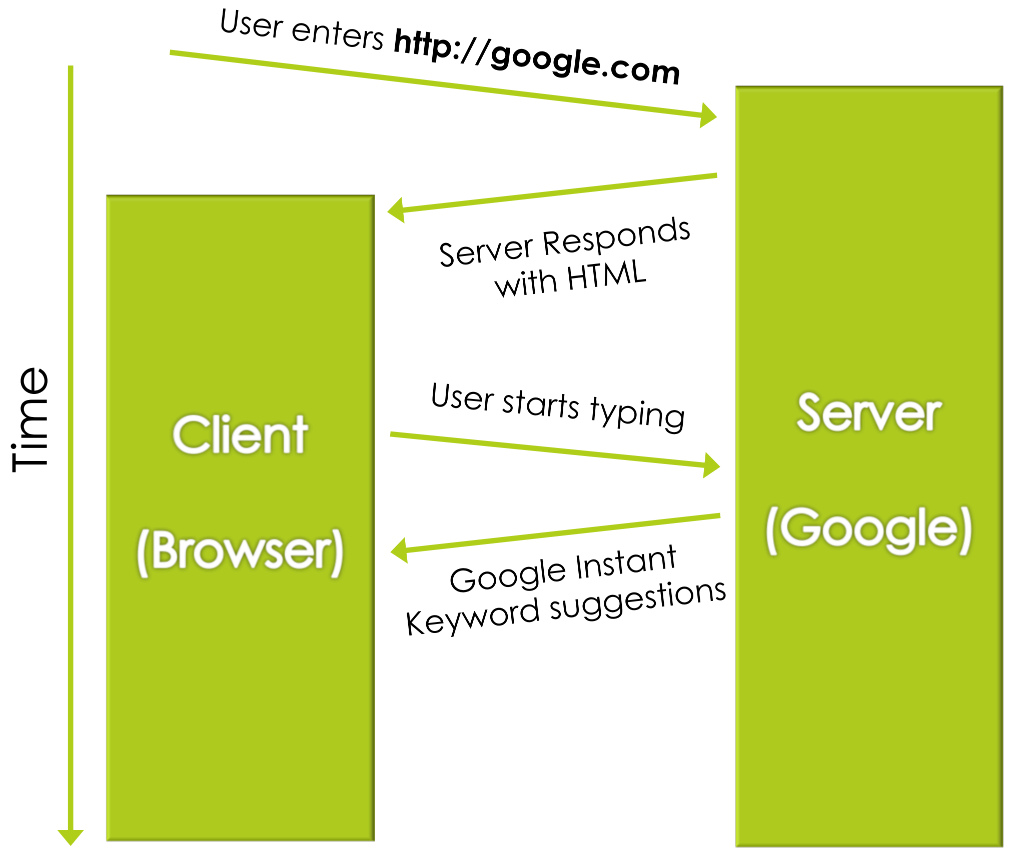
Heath Hendrickson ▸ **Casey Marcy** Happy Birthday Casey!!

Nate Shelc likes Dennis B. Jones's status.

Heath Hendrickson ▸ **Casey Marcy** Happy Birthday Casey!!

Nate Shelc likes Dennis B. Jones's status.

AJAX: Diagram


User enters **http://google.com**
Server Responds with HTML
User starts typing
Google Instant Keyword suggestions
Time
Client
(Browser)
Server
(Google)

- Client (Browser)
  - Submit request
  - No page reload
  - Handle response
- Server (Google)
  - Receive request
  - Respond with some data
    - HTML, JSON, XML, etc.
- Simply GET or POST requests
  - Parameter data
  - Receive & handle response from server
- How do we do this?
  - JavaScript

JavaScript

Key features & use with AJAX

JavaScript: Introduction

- Runs on user’s browser
  - Place in *<script>* … *</script>*
  - Run on page load
- Basic language properties
  - Interpreted
  - Single-threaded

JavaScript: Basics

- Language
  - Basic, readable syntax
  - Automatic type casting
  - Arrays, dictionaries
- Development
  - Console Demo

```
// Declare a few vars
var foo = 5;
var bar = 'abcde';

// Automatic type casting
var foobar = foo + bar;

// Arrays & dictionaries
var myArray = [1,2,3];
var myDict = {
   'key1': 1,
   'key2': 2,
   'key3': [1,2]
};
```

JavaScript: Functions

- Functions
  - First-class citizens
  - Associated with scope
  - Callbacks
- JavaScript is built for asynchronous programming!

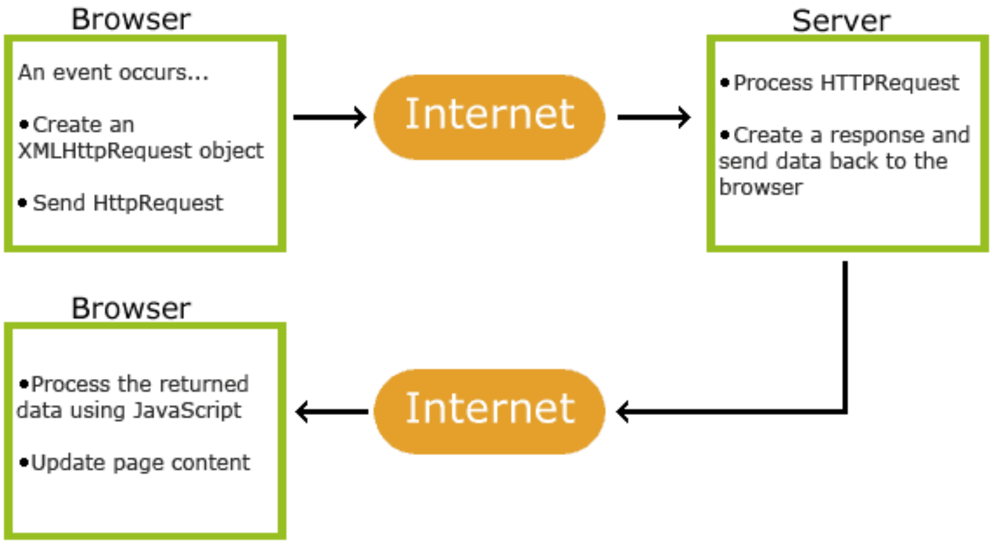
```
// Function variables
var equals =  function(a, b) {
  return a == b;
};

var areEqual = equals(1, 2);

// Anonymous function & callback
var a = 1;
setTimeout(function() {
  a++;
}, 5000);
```

JavaScript: Browser Example

```
<html>
  <head>
     ...
    <script type="text/javascript" src="dependency.js"></script>
  </head>
  <body>
    ... some content ...
    <script>
      // Browser runs this code
      var foo = 5;
      var bar = functionInDependency(foo);
    </script>
  </body>
</html>
```

JavaScript: AJAX Example

- Trigger JavaScript
  - On page load
  - On user event
  - On a timer
  - …
- Submit HTTP request
- Handle server response
  - Callbacks



Browser

An event occurs...

- Create an XMLHttpRequest object
- Send HttpRequest

Server

- Process HTTPRequest
- Create a response and send data back to the browser

Browser

- Process the returned data using JavaScript
- Update page content

JavaScript: AJAX Example

JavaScript (triggered by event)

```javascript
// Create request
var req = new XMLHttpRequest();

// Set handler for server response
req.onreadystatechange = function() {
  console.log(req.responseText);
};

// Set URL for request (including data)
req.open("GET","endpoint.php?n=7",true);

// Send the request
req.send();
```

PHP (*endpoint.php*)

```php
<?php

  // Get parameter data
  $n = int($_GET['n']);

  // Do some work
  $ans = $n + $n^2;

  // Send response
  echo $ans;
?>
```

jQuery

Key features & use with AJAX

jQuery: Motivation

- JavaScript can be challenging
  - Too much code for simple tasks
  - Browser inconsistency
- jQuery simplifies common tasks
  - DOM traversal & manipulation
  - Event handling
  - Animations
  - AJAX

jQuery: Motivation

- jQuery is …
  - Fast
  - Concise
  - Lightweight
  - Multi-browser compatible
  - Widely used
    - Over 55% of 10,000 most-visited sites
    - Over half of all websites

jQuery: Basic Selectors

Without jQuery

```
// Different across browsers...
document.getElementById('myId');
document.getElementsByName('myName');
document.getElementsByTagName('div');
document.getElementsByClass('myClass');
```

- DOM selection differs across browsers
- Using jQuery, you can easily handle element selections
  - Hide/show
  - Animate
  - Modify content
  - …

With jQuery

```
// Multi-browser support
$('#myId');
$('[name=myName]');
$('div');
$('.myClass');
```

```
// Do something...
$('#myId').hide();
$('#myId').addClass('testClass');
```

```
<style type='text/css'>
  .testClass {
    font-weight: bold;
  }
</style>
```

jQuery: DOM Manipulation & Animation

Without jQuery

```
// Doesn't work on all browsers
var div = document.getElementById('target');
div.innerHTML = "<p>Some HTML content!</p>";

div.style.display = "none";

// Hand-written code for animations ...
```

With jQuery

```
var div = $('#target')
div.html("<p>Some HTML content!</p>");

div.hide();

div.fadeIn();
```

jQuery: AJAX

Without jQuery

```
// Create request (fails on IE)
var req = new XMLHttpRequest();

// Create callback
req.onreadystatechange = function() {
  var r = document.getElementById('res');
  r.innerHTML = req.responseText;
}

// Specify endpoint & data
req.open('GET','getContent.php?n=7',true);

// Send request
req.send();
```

With jQuery

```
// All in one line ...
$.get(
  'getContent.php',
  {n: 7},
  function(responseText) {
    $('#res').html(responseText);
  }
);
```

AJAX Demo

- Let’s write the code…
  - PHP endpoint
  - Empty web page
- Tools
  - jQuery
  - PHP
  - Google Chrome console

AJAX Demo – Code

```
<html>
  <head>
    <title>AJAX Demo Page</title>
    <script src="http://.../jquery.min.js"></script>
  </head>
  <body>
    <center>
      <br/><h1>AJAX Demo Page</h1><br/>
      <p id="myId">
      </p>
    </center>
  </body>
</html>
```

index.html

```
<?php
  $n = $_GET['n'];
  $m = $_GET['m'];
  echo $n * $m;
?>
```

endpoint.php

```
$.get(
  'endpoint.php', // Your endpoint
  {n:5, m:6},
  function(response) {
    $('#myId').text(response);
  }
);
```

Console / JavaScript

AJAX Demo – Code With Animation

```html
<html>
  <head>
    <title>AJAX Demo Page</title>
    <script src="http://.../jquery.min.js"></script>
  </head>
  <body>
    <center>
      <br/><h1>AJAX Demo Page</h1><br/>
      <p id="myId">
      </p>
    </center>
  </body>
</html>
```

index.html

```php
<?php
  $n = $_GET['n'];
  $m = $_GET['m'];
  echo $n * $m;
?>
```

endpoint.php

```javascript
$.get(
  'endpoint.php', // Your endpoint
  {n:5, m:6},
  function(response) {
    var myDiv = $('#myId');
    myDiv.hide();
    myDiv.text(response);
    myDiv.fadeIn();
  }
);
```

Console / JavaScript

Assignment 3.2

- Final project proposals
- Makes use of
  - jQuery
  - AJAX
- Use jQuery to
  - Post comments via AJAX
    - Is POST or GET better for this?
  - Load file contents via AJAX
    - Use jQuery selectors
    - Create your own PHP endpoint to handle requests

Summary

- Web Requests
  - GET &POST
- AJAX Basics
- JavaScript
  - Overview
  - Callbacks
  - AJAX
- jQuery
  - Features
- Demo
- Assignment Usage

Questions?

Jon Tedesco

References

- GET v. POST
  - http://blog.teamtreehouse.com/the-definitive-guide-to-get-vs-post#disqus_thread
  - http://stackoverflow.com/questions/46585/when-do-you-use-post-and-when-do-you-use-get
  - http://www.jquery4u.com/ajax/key-differences-post/#.UIoD0GI26QY
- AJAX
  - http://scorebig.files.wordpress.com/2011/09/feed1.jpg
  - http://www.w3schools.com/ajax/ajax_intro.asp
- JavaScript
  - http://en.wikipedia.org/wiki/JavaScript
- jQuery
  - http://en.wikipedia.org/wiki/JQuery
  - http://www.sitepoint.com/jquery-used-on-50-percent-of-websites/
  - http://www.slideshare.net/arwid/jquery-4470993