JavaScript
Lecture 4c (Ajax)

Waterford Institute of Technology

June 18, 2016

John Fitzgerald

JavaScript Introduction
Topics discussed this presentation

- Asynchronous JavaScript & XML (Ajax).
- This presentation based on jQuery Ajax.
- Using a very small subset of available functionality.

Ajax
What is it?

- A technology to manage transmission of data.
- Between client and server.
- Generally text-based data.
- Binary data transmission also possible.
- In this course we focus on text-based.
- Originally data format Extensible Markup Language (XML).
- JSON now increasingly the format of choice.

Ajax
Why use it?

- Once Upon a Time in the Web ...
- data request caused whole-page refresh.
- Ajax requests server what it needs,
- when it needs it and,
- for exactly where on page it is needed...
- finding the target with perfect aim.
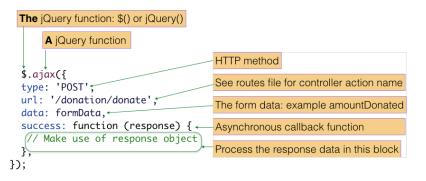- This avoids nuisance page flicker and,
- facilitates greater efficiency.

Ajax
Asynchronous communication

- Web page sends HTTP Ajax request.
- User free to continue other page activity.
- Request processed independently.
- Server transmits respose to web page.
- Synchronous communication also possible.

DON'T CALL US
WE'LL CALL YOU

jQuery

Donation ajax call

The jQuery function: $() or jQuery()

A jQuery function

HTTP method

```
$.ajax({
  type: 'POST',
  url: '/donation/donate',
  data: formData,
  success: function (response) {
    // Make use of response object
  },
});
```

See routes file for controller action name

The form data: example amountDonated

Asynchronous callback function

Process the response data in this block

jQuery
Donation ajax call

```html
<form class="ui form" action="/donation/donate" method="POST">
    ...
    ...
</form>
```

```html
<form class="ui form">
    ...
    ...
</form>
```

Ajax requires **form** change

jQuery
Donation ajax call

```html
<form class ="ui form segment">
    <input name="candidateEmail" type="hidden">
    <input name="amountDonated" type="hidden">
<div class="ui blue submit button">Donate <i class="add icon"></i></div>
```

HTML

```javascript
$('.ui.submit.button').click(function() {
    var formData = $('ui.form.segment input').serialize();
    $.ajax({
        type: 'POST',
        url: '/donation/donate',
        data: formData,
        success: function(response) {
            console.log("make donation page submitForm response: " + response.progress);
        }
    });
});
```

JavaScript

Provides routing to controller action donate
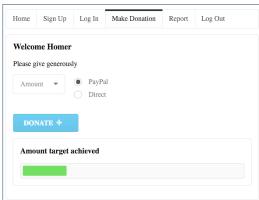
The form data: example amountDonated

Asynchronous call back delivers response

Controller

```
JSONObject obj = new JSONObject();
obj.put("progress", getProgress());
renderJSON(obj);
```

Composes and transmits JSON object response
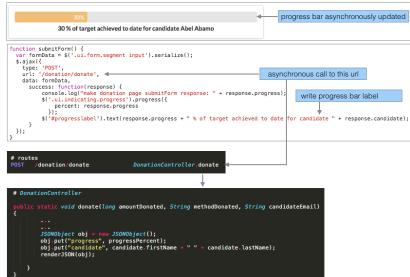
jQuery
Asynchronous Java and XML (Ajax)

Not using Ajax

- Press Donate button
- Entire page refreshes
- Noticeable flicker

Using Ajax

- Press Donate button
- Only the progress bar affected
- No noticeable flicker

Home	Sign Up	Log In	**Make Donation**	Report	Log Out

Welcome Homer

Please give generously

Amount ▾   ● PayPal
            ○ Direct

DONATE +

Amount target achieved

jQuery
Asynchronous Java and XML (Ajax)

> 30%
>
> 30 % of target achieved to date for candidate Abel Abamo

progress bar asynchronously updated

```
function submitForm() {
    var formData = $('.ui.form.segment input').serialize();
    $.ajax({
        type: 'POST',
        url: '/donation/donate',
        data: formData,
        success: function(response) {
            console.log("make donation page submitForm response: " + response.progress);
            $('.ui.indicating.progress').progress({
                percent: response.progress
            });
            $('#progresslabel').text(response.progress + " % of target achieved to date for candidate " + response.candidate);
        }
    });
}
```

asynchronous call to this url

write progress bar label

```
# routes
POST   /donation/donate              DonationController.donate
```

```
# DonationController

public static void donate(long amountDonated, String methodDonated, String candidateEmail)
{
        ...
        ...
        JSONObject obj = new JSONObject();
        obj.put("progress", progressPercent);
        obj.put("candidate", candidate.firstName + " " + candidate.lastName);
        renderJSON(obj);

    }
}
```

JavaScript
Presentation summary

- Ajax
    - Asynchronous JavaScript & XML (Ajax)
    - Now Asynchronous JavaScript & JSON (AJAN -:)
    - In widespread use.
    - Allows specific page element refresh.
    - Avoids flicker associated with whole-page refresh.

JavaScript
Some suggestions for consideration

- Write code complying with ECMAScript6 (ES6).
- Use quality IDE such as WebStorm.
- Apply styleguide, example Airbnb.
- Use strict mode.
- Avoid use of global variables:
    - Use global abatement or other techniques.
- Do not rely on semicolon insertion.
- Do not use:
    - == (use ===)
    - != (use !==)
- Avoid use of *continue* statement.
- Do not use block-less statements (e.g. following *for, while, if*).

References

1. jQuery

http://jquery.com/

[Accessed 2014-08-04]

2. w3schools: Learn JavaScript

http://www.w3schools.com/js/default.asp

[Accessed 2014-07-09]

3. HTML5 Element List

MDN: Mozilla Developer Network

https://developer.mozilla.org/en/docs/Web/Guide/HTML/
HTML5/HTML5_element_list

[Accessed 2014-08-09]

References

4. Haverbeke Marijn. 2007-2013. Eloquent JavaScript: A Modern Introduction to Programming

http://eloquentjavascript.net/

[Accessed 2014-07-09]

5. Mozilla Developer Network (MDN)

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array

[Accessed 2016:-03-22]

6. w3schools: Learn jQuery

http://www.w3schools.com/jquery/default.asp

[Accessed 2014:-07-09]

References

7. The Deep Roots of Javascript Fatigue

`https://segment.com/blog/the-deep-roots-of-js-fatigue/`

[Accessed 2016:-03-23]

8. Mozilla Developer Network (MDN)

`https://developer.mozilla.org/en-US/Add-ons/SDK/Guides/Contributor_s_Guide/Classes_and_Inheritance`

[Accessed 2016:-05-14]

References

9. Airbnb JavaScript Style Guide

http://airbnb.io/javascript/

[Accessed 2016:-005-15]

10. Code Conventions for JavaScript

http://javascript.crockford.com/code.html

[Accessed 2016:-006-16]

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit