

WEB APPLICATION DEVELOPMENT

DR. ALISTAIR MORRISON

*Joao Almeida-Domingues**

University of Glasgow

January 15th, 2020 – March 25th, 2020

CONTENTS

1	AJAX	2
1.1	Asynchronicity	2
1.2	The XMLHttpRequest Object	3
1.3	Callbacks	3

These lecture notes were collated by me from a mixture of sources , the two main sources being the lecture notes provided by the lecturer and the content presented in-lecture. All other referenced material (if used) can be found in the *Bibliography* and *References* sections.

The primary goal of these notes is to function as a succinct but comprehensive revision aid, hence if you came by them via a search engine , please note that they're not intended to be a reflection of the quality of the materials referenced or the content lectured.

Lastly, with regards to formatting, the pdf doc was typeset in L^AT_EX, using a modified version of Stefano Maggiolo's [class](#)

*2334590D@student.gla.ac.uk

1 AJAX

AJAX stands for *Asynchronous Javascript and XML*, and represents a set of key technologies which when used together allow web applications to be updated without needing to reload each time. With AJAX, web applications can send and retrieve data from a server asynchronously effectively decoupling the data interchange layer from the presentation layer

1.1 remark. nowadays is far more common to use JSON instead of XML

1.1 Asynchronicity

The key element behind AJAX is the `XmlHttpRequest` Object, which is just a DOM object, with the special property of being able to exchange data with the server behind the scenes.

In contrast with classical webpages, the JS engine handles requests and is then able to update only the targetted parts of the web app which need updating. Unlike forms, or links the `XmlHttpRequest` Object does not block script execution the JS keeps running in the background, hence the asynchronous nature of the requests.

```
// https://www.w3schools.com/xml/ajax_intro.asp

function loadDoc() {
  var xmlhttp = new XMLHttpRequest();
  xmlhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText
    }
  };
  xmlhttp.open("GET", "ajax_info.txt", true);
  xmlhttp.send();
}
```

Summary

1. An event occurs
2. An `XmlHttpRequest` Object is created by JS
3. An HTTP request is sent to the server
4. The server processes the request
5. The server sends back the response
6. The JS engine processes the response
7. The DOM is manipulated appropriately by the JS

1.2 The XMLHttpRequest Object

https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp

1.3 Callbacks

When performing multiple tasks using AJAX, there should be a main function which sets the request object and then there should be different *callback* functions for each task.

1.2 remark. the main function call should contain the URL and which function to call when the response is ready

```
<html>
<head>
  <title>AJAX example</title>
  <script type="text/javascript">
    function loadDoc(url, cFunction) {
      var xhttp = new XMLHttpRequest(); // create request
      object
      xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status ==
          200) {
          cFunction(this);
        }
      };
      xhttp.open("GET", url, true); // send request
      xhttp.send();
    }
    function myFn1(xhttp) {
      document.getElementById("demo1").innerHTML = xhttp.
        responseText;
    }
    function myFn2(xhttp) {
      document.getElementById("demo2").innerHTML = xhttp.
        responseText;
    }
  </script>
</head>
<body>
  <div id="demo1">
    <h2>Let AJAX change this text</h2>
    <button type="button" onclick="loadDoc('ajax.txt?t
      =' + Math.random(), myFn1)">AJAX Info</button>
    <button type="button" onclick="loadDoc('wad2.txt?t
      =' + Math.random(), myFn2)">WAD2 Info
    </button>
  </div id="demo2"> </div>
```

REFERENCES

REFERENCES

Almeida-Domingues: Knowledge Base **knowBase**

Joao Almeida-Domingues. *Knowledge Base*. URL: <https://app.gitbook.com/@knowbase/s/workspace/git>.

Kozlovski: A Thorough Introduction to Distributed Systems **kozlovski'2019**

Stanislav Kozlovski. *A Thorough Introduction to Distributed Systems*. June 2019. URL: <https://www.freecodecamp.org/news/a-thorough-introduction-to-distributed-systems-3b91562c9b3c/>.

Ovens: The evolution of package managers **ovens**

Steve Ovens. *The evolution of package managers*. URL: <https://opensource.com/article/18/7/evolution-package-managers>.