JSON
and
AJAX

AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

AJAX
Asynchronous Javascript And Xml

Ajax

- Is a group of interrelated web development techniques used on the client-side to create asynchronous web applications.
- With Ajax:
  - web applications can send data to, and
  - retrieve data from, a server
  - asynchronously (in the background).
- Data can be retrieved using the **XMLHttpRequest** object.
- Despite the name, the use of XML is not required
  - **JSON is often used instead.**
- And the requests do not need to be asynchronous.

- *Ajax is not a single technology, but a group of technologies used in combination*.

Background

- In the 1990s, most web sites were based on complete HTML pages:
  - each user action required that the page be re-loaded from the server (or a new page loaded).
  - This process is inefficient: all page content disappears then reappears, etc.
  - Each time a page is reloaded due to a partial change, all of the content must be re-sent instead of only the changed information.
  - This can place additional load on the server and use excessive bandwidth

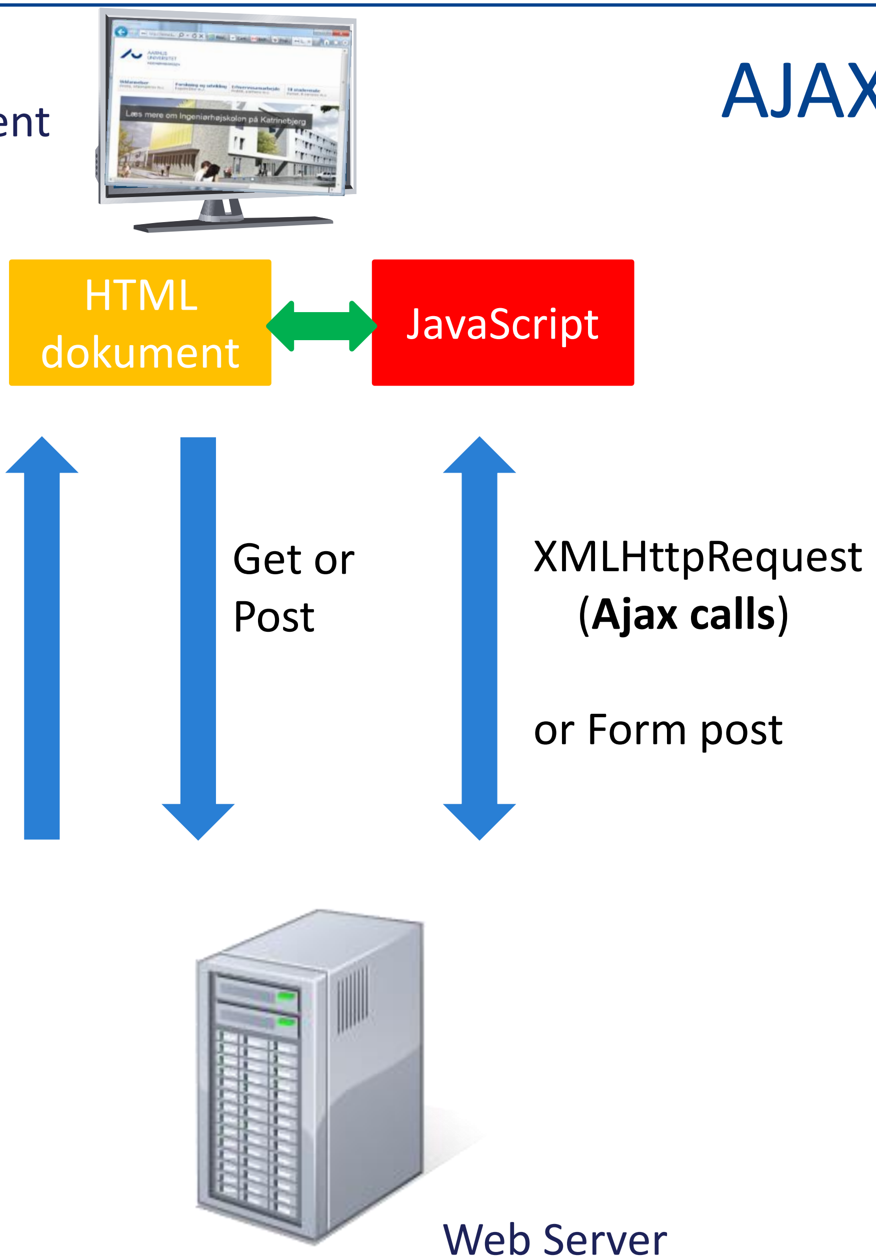
The Birth of AJAX

- In 1996, the iframe tag was introduced by Internet Explorer to load content asynchronously.
- In 1998, Microsoft Outlook Web Access team implemented the first component XMLHTTP by client script.
- In 1999, Microsoft utilized its iframe technology to dynamically update the news stories and stock quotes on the default page for Internet Explorer, and created the **XMLHTTP** ActiveX control in Internet Explorer 5.
- This technique was later adopted by Mozilla, Safari, Opera and other browsers as the **XMLHttpRequest** JavaScript object.
- The term **Ajax** was coined on 18 February 2005 by Jesse James Garrett in an article entitled "**Ajax: A New Approach to Web Applications**", based on techniques used on Google pages.
- On 5 April 2006 the World Wide Web Consortium (W3C) released the first draft specification for the XMLHttpRequest object in an attempt to create an official web standard.

AJAX Technologies

- The term Ajax has come to represent a broad group of web technologies that can be used to implement a web application that communicates with a server in the background, without interfering with the current state of the page.
- According to Jesse James Garrett:
  - **HTML** and **CSS** for presentation
  - The Document Object Model (**DOM**) for dynamic display of data.
  - **XML** for the interchange of data, and XSLT for its manipulation.
  - The **XMLHttpRequest** object for asynchronous communication.
  - **JavaScript** to bring these technologies together.
- Since then, there have been a change in data formats:
  - XML is not required.
  - **JSON** is often used.
  - Other formats such as preformatted HTML or plain text can also be used.

AJAX Communication



Web Client

HTML dokument

JavaScript

Get or Post

XMLHttpRequest (**Ajax calls**)

or Form post

Web Server

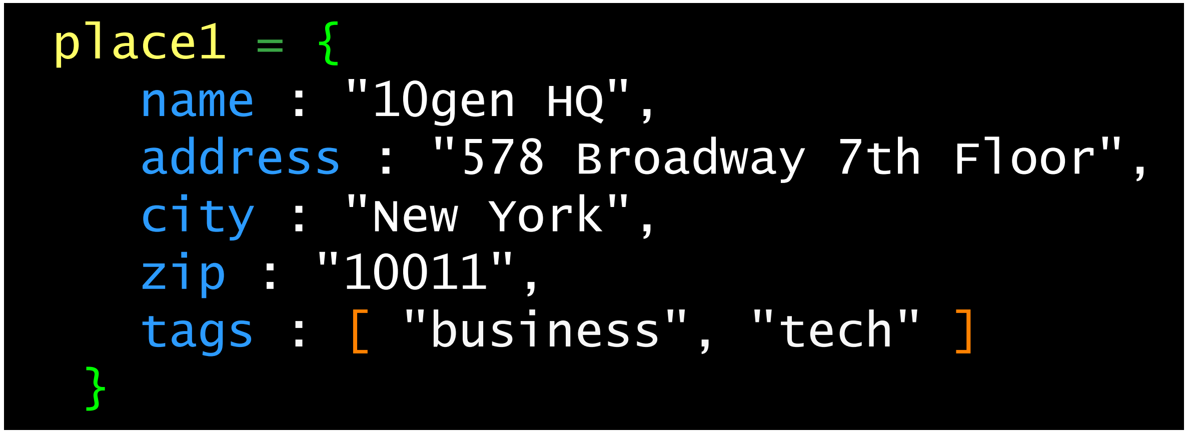
JSON
JavaScript Object Notation

JSON

- Is a lightweight data-interchange format.
- It is easy for humans to read and write.
- It is easy for machines to parse and generate.
- It is based on a subset of JavaScript.
- Is a text format that is completely language independent
  - but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others.
- These properties make JSON an ideal data-interchange language.
  - And JSON is widely used in data communication.
  - And several NoSQL databases uses JSON (or BSON) as data storage format.
    - BSON is a binary representation of JSON.

JSON Structure

- JSON is built on two structures:
  - A collection of name/value pairs.
    - In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
  - An ordered list of values.
    - In most languages, this is realized as an array, vector, list, or sequence.



```
place1 = {
   name : "10gen HQ",
   address : "578 Broadway 7th Floor",
   city : "New York",
   zip : "10011",
   tags : [ "business", "tech" ]
 }
```

History

- Douglas Crockford was the first to specify and popularize the JSON format.
- JSON was used at State Software, a company co-founded by Crockford, starting around 2001.
- The JSON.org website was launched in 2002.
- In 2005 Yahoo began offering some of its web services in JSON.
- In 2006 Google started offering JSON feeds for its GData web protocol.
- Is described in RFC 4627.

CLINT SIDE OF AJAX

HTTP request

- An object used to make an HTTP request can, on most browsers, be created by doing new XMLHttpRequest().

```
var request = new XMLHttpRequest();}
request.open("GET", "files/fruit.txt", false);
request.send(null);
print(request.responseText);
```

- The open method is used to configure the request.
- The send method perform the actual request to the server.
  - When the request is a POST request, the data to be sent to the server (as a string) can be passed to this method.
  - For GET requests, one should just pass null.
- The responseText property contains the content of the retrieved document (after the request has been made).
- The headers that the server sent back can be inspected with the getResponseHeader and getAllResponseHeaders functions.

Asynchronous HTTP Request

- When the server is slow, or the file is big, doing a request might take quite a while.
- When the third argument to open is true, the request is set to be '**asynchronous**'.
  - This means that send will return right away, while the request happens in the background.

```
var request = new XMLHttpRequest();}
request.open("GET", "files/fruit.txt", true);
request.send(null);
request.onreadystatechange = function() {
  if (request.readyState == 4)
    show(request.responseText.length);
};
```

Asynchronous HTTP Request

- A request object has a **readyState** property, indicating the state it is in.
- This will become 4 when the document has been fully loaded, and have a smaller value before that.
- To react to changes in this status, you can set the onreadystatechange property of the object to a function.
- This function will be called every time the state changes.

```
var request = new XMLHttpRequest();}
request.open("GET", "files/fruit.txt", true);
request.send(null);
request.onreadystatechange = function() {
  if (request.readyState == 4)
    show(request.responseText.length);
};
```

Working With XML

- When the file retrieved by the request object is an XML document, the request's responseXML property will hold a representation of this document.
- Such XML documents can be used to exchange structured information with the server.
- Their form — tags contained inside other tags — is often very suitable to store things that would be tricky to represent as simple flat text.
- The DOM interface is rather clumsy for extracting information though, and XML documents are notoriously wordy:.

```
var catalog =
request.responseXML.documentElement;
show(catalog.childNodes.length);
```

Working With JSON

- Use the basic notation of JavaScript values to represent 'hierarchical' information in a more minimalist way.
- A JSON document is a file containing a single JavaScript object or array,
  - which in turn contains any number of other objects, arrays, strings, numbers, booleans, or null values.

```
request.open("GET", "files/fruit.json", true);
request.send(null);
request.onreadystatechange = function() {
  if (request.readyState == 4)
    print(request.responseText);
};
```

Working With JSON

- Such a piece of text can be converted to a normal JavaScript value by using the eval function.
- Parentheses should be added around it before calling eval,
  - because otherwise JavaScript might interpret an object (enclosed by braces) as a block of code, and produce an error.

```
function evalJSON(json) {
  return eval("(" + json + ")");
}
var fruit = evalJSON(request.responseText);
show(fruit);
```

- When running eval on a piece of text, that this means you let the piece of text run whichever code it wants.
- Since JavaScript only allows us to make requests to our own domain, you will usually know exactly what kind of text you are getting, and this is not a problem.
- In other situations, **it might be unsafe**.

Native JSON

- A new function, JSON.parse(), was developed as a safer alternative to eval.
- It is specifically intended to process JSON data and not JavaScript.
- It was added to the Fifth Edition of the ECMAScript standard.
- Native JSON is generally faster compared to the JavaScript libraries commonly used before.
- All modern browsers have native JSON support, via:
  - JSON.parse() and
  - JSON.stringify().
- For older browsers, a compatible JavaScript library is available at JSON.org.

SERVER SIDE OF AJAX

Microsoft Frameworks

- ASP.NET AJAX
  - A variation of the Controllers in the MVC framework.
- WCF REST (Windows Communication Foundation)
  - lets developers build contract-first services that leverage transport protocols such as TCP, HTTP and MSMQ. Originally built for SOAP-based services that want WS-* capabilities, WCF eventually added a handful of REST-friendly capabilities.
- ASP.NET Web API
  - differentiates itself from the previous Microsoft in-box HTTP service solutions in that it was built from the ground up around the HTTP protocol and its messaging semantics.

REST
REPRESENTATIONAL STATE TRANSFER

REST didn't attract much attention when it was first introduced in 2000 by Roy Fielding at the University of California, Irvine, in his academic dissertation, "Architectural Styles and the Design of Network-based Software Architectures," which analyzes a set of software architecture principles that use the Web as a platform for distributed computing.

Representational State Transfer (REST) has gained widespread acceptance across the Web as a simpler alternative to SOAP- and Web Services Description Language (WSDL)-based Web services.

RESTful Web Services: the Basics

- REST defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages.

- A concrete implementation of a REST Web service follows four basic design principles:
  - **Use HTTP methods explicitly.**
  - **Be stateless.**
  - **Expose directory structure-like URIs.**
  - **Transfer XML, JSON, or both**.

Use HTTP methods explicitly

- One of the key characteristics of a RESTful Web service is the explicit use of HTTP methods in a way that follows the protocol as defined by RFC 2616.
- This basic REST design principle establishes a one-to-one mapping between create, read, update, and delete (CRUD) operations and HTTP methods.
- According to this mapping:
  - **To create a resource on the server, use POST.**
  - **To retrieve a resource, use GET.**
  - **To change the state of a resource or to update it, use PUT.**
  - **To remove or delete a resource, use DELETE**.

Use HTTP methods explicitly

API Design Principles:

- Use nouns in URIs instead of verbs.
  - In a RESTful Web service, the verbs—POST, GET, PUT, and DELETE—are already defined by the protocol.
- And the Web service should not define more verbs or remote procedures, such as /adduser or /updateuser.
- The body of an HTTP request should be used to transfer resource state
  - not to carry the name of a remote method to be invoked.

Use HTTP Methods Explicitly

- A non-RESTful API:

```
GET /adduser?name=Robert HTTP/1.1
```

- A RESTful API:

```
POST /users HTTP/1.1
Host: myserver
Content-Type: application/json
{name : Robert}
```

References & Links

- http://json.org/
- An Introduction to JavaScript Object Notation (JSON) in JavaScript and .NET
  http://msdn.microsoft.com/en-us/library/bb299886.aspx
- ASP.NET Web API
  http://www.asp.net/web-api
- ASP.NET Ajax
  http://www.asp.net/ajax
- ASP.NET SignalR
  http://signalr.net/
- REST and RESTful
  http://www.ibm.com/developerworks/webservices/library/ws-restful/
  http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm