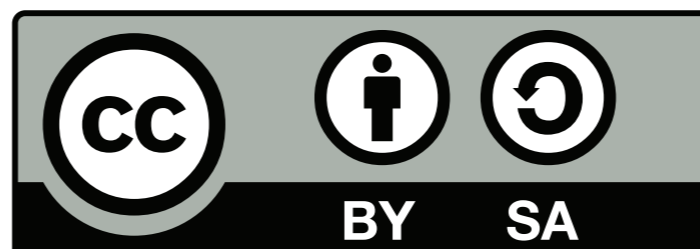


Tecnologia e Applicazioni Internet 2011/12

Lezione 8 - Ajax

Matteo Vaccari

<http://matteo.vaccari.name/>
matteo.vaccari@uninsubria.it



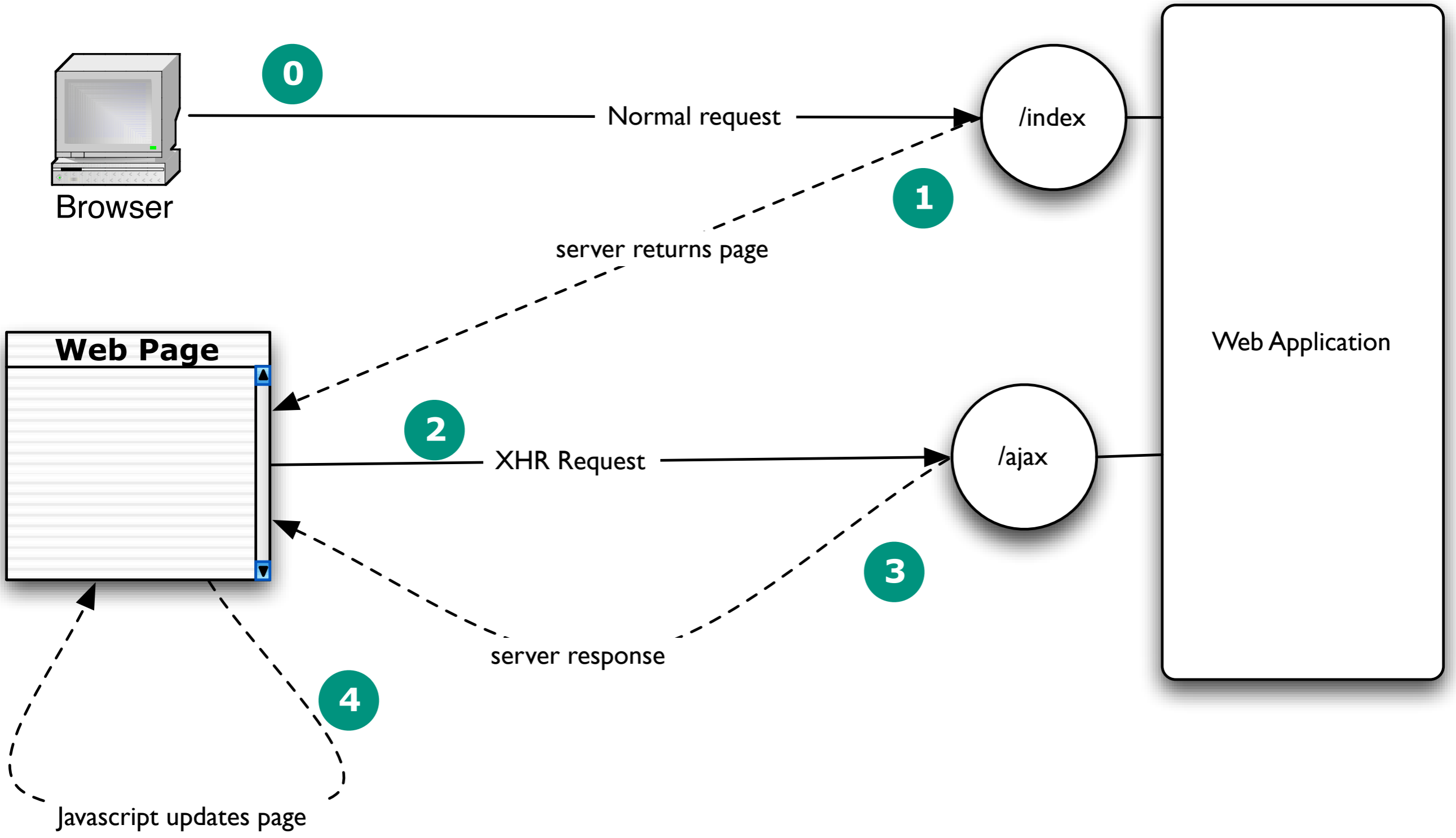
Un avvertimento

Tempo necessario a sviluppare una feature **con** Ajax

=

10 * tempo necessario per la stessa feature **senza** Ajax

XMLHttpRequest (XHR)



Ajax

- Load a page
- Make an XHR request
- When response arrives, update page
- Handle errors
- Handle timeouts

Ottenere l'oggetto *XMLHttpRequest*

```
// non-Microsoft browser, IE >= 7  
var request = new XMLHttpRequest();
```

```
// IE < 7
```

```
var request = new ActiveXObject("Msxml2.XMLHTTP");  
var request = new ActiveXObject("Microsoft.XMLHTTP");
```

```
// prova in diverse maniere fino a che una funziona
```

```
var request = Try.these(  
    function() { return new XMLHttpRequest(); },  
    function() { return new ActiveXObject("Msxml2.XMLHTTP"); },  
    function() { return new ActiveXObject("Microsoft.XMLHTTP"); }  
);
```

Chiamare il server

```
// settare metodo e url
request.open("GET", url);

// altri parametri opzionali
request.setRequestHeader("User-Agent", "XMLHttpRequest");
request.setRequestHeader("Accept-Language", "en");

// callback eseguita quando il server risponde
request.onreadystatechange = function() {...};

// eseguire la chiamata
request.send(null);
```

Rispondere a una chiamata *asincrona*

```
// readyState meaning
// 0    open() has not been called yet.
// 1    open() has been called, but send() has not been called.
// 2    send() has been called, but the server has not responded yet.
// 3    Data is being received from the server.
// 4    The server's response is complete.

request.onreadystatechange = function() {
    if (request.readyState == 4) { // If the request is finished
        if (request.status == 200) // If it was successful
            alert(request.responseText); // Display the server's response
    }
}
```


XHR Limitations

- *Same Origin Policy*
- Non più di due chiamate aperte contemporaneamente

Ajax con jQuery

- `$('#div#intro').load('/some/file.html');`

La funzione completa

- `$.ajax(settings)`

```
$.ajax({  
  url: "/some/url",  
  data: { "q", "facoceri" },  
  success: function(data) { $("#result").text(data) },  
});
```

Callback functions

```
function success(data, textStatus, jqXHR) {  
    // data: interpretato da jQuery  
    // textStatus: p es. "200"  
    // jqXHR: XMLHttpRequest object  
}
```

```
function error(jqXHR, textStatus, errorThrown) {  
    // textStatus: "timeout", "error", "abort", or "parsererror"  
    // errorThrown: p. es. "Not Found" or "Internal Server Error."  
}
```

Il parametro *dataType*

- **"xml"**: Returns a XML document that can be processed via jQuery.
- **"html"**: Returns HTML as plain text
- **"script"**: Evaluates the response as JavaScript
- **"json"**: Evaluates the response as JSON and returns a JavaScript object.
- **"text"**: A plain text string.