

AJAX & Web Services

① what is Ajax?

② other important concepts

- loose coupling aka weak coupling
- REST architectural model
- same origin policy / effect on proxies
- Javascript Sandbox
- SOAP from Web Services Lecture

③ Client-side

④ Server-side

1) What is Ajax

- term originated in 2005 (see Readings) but technology had been developed and implemented (by Microsoft) since circa 1998
- acronym-like term for Aynchronous JavaScript + XML
- what does it do?

Allows the web client (browser) to request & load one part of a web page at a time, instead of all at once

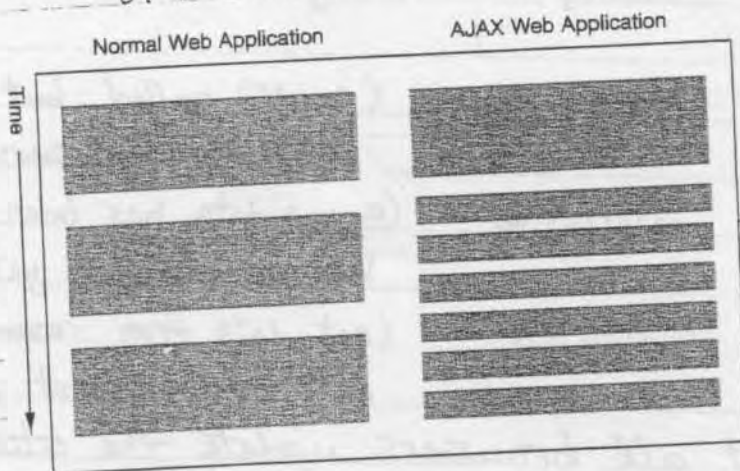


FIGURE 1-3 Normal Web application requests versus AJAX requests

Fig. from
 ? ← Eichorn (2007)
 reference at end

Ajax (cont.)

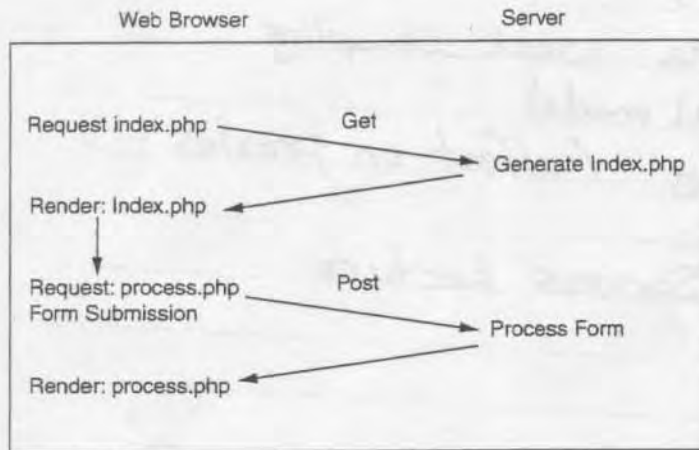


FIGURE 1-1
Web application request flow

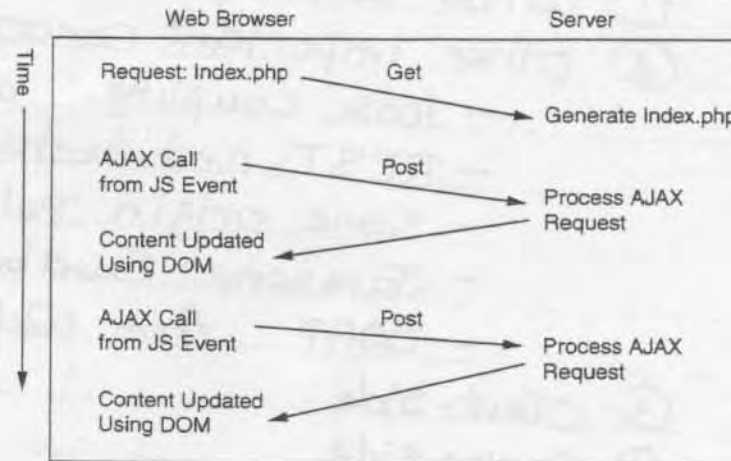


FIGURE 1-2
AJAX application request flow

Figs. from Eichorn (2007)

How does it do that?

- Ajax is a technique enabled by XMLHttpRequest
- See code by Pruett (2006) for example

→ 5 stages

- 0 uninitialized (before the open() call)
- 1 loading (object created but not ~~send()~~ send()-e)
- 2 loaded (send() called, but nothing retrieved from server, yet)
- 3 interactive (some data has been received but not finished yet)
- 4 completed (get data from response Body and responseText properties)

note not all browsers update the state the same way, but all are ready at stage 4

Concepts. (cont.)

loose (or weak) coupling

- closely related to encapsulation
- built into layer model

REST architectural model *

contrasted
with

- Representational State Transfer

- Coined by Roy Fielding in 2000

- ~~juxtaposed~~ against the RPC model we saw in the Web Services lecture

protocol

- based on client/server model
- stateless
- cachable
- layered

to communicate between resources { application state
functionality

using well-defined operations and a limited number of content-types

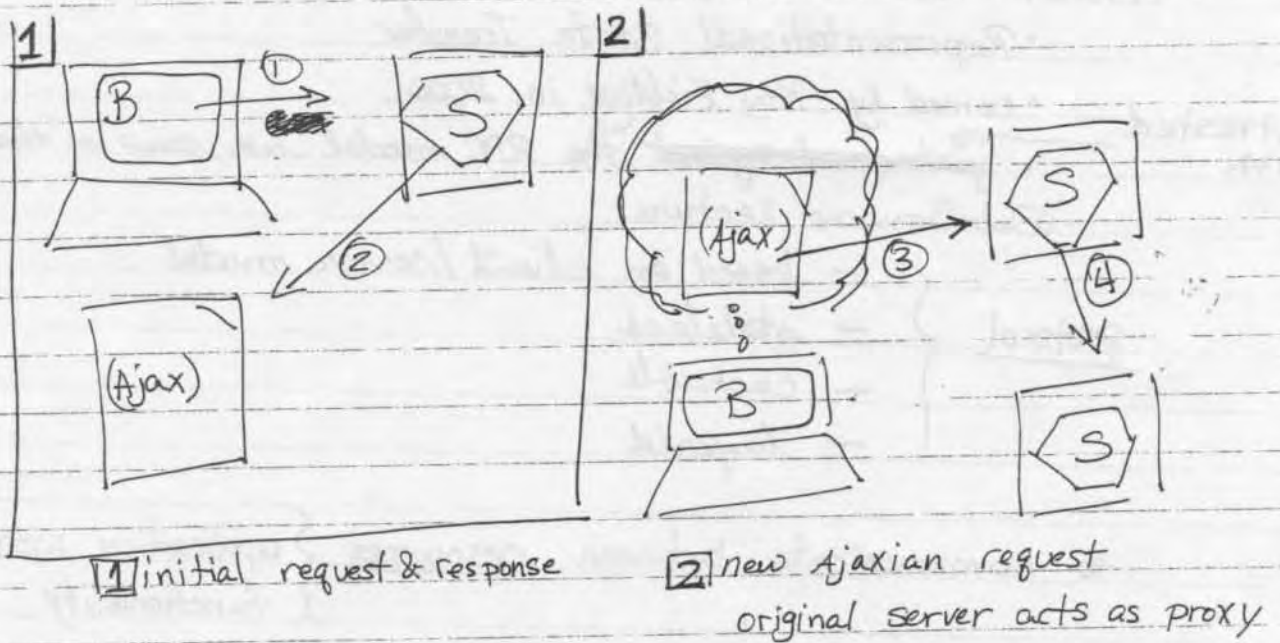
Allows for heterogeneity and scalability

Meant for large mega-scale hypermedia not WWW in general. Implemented in WebDAV (modulo the use of POST) and Ruby on Rails v.2.1.

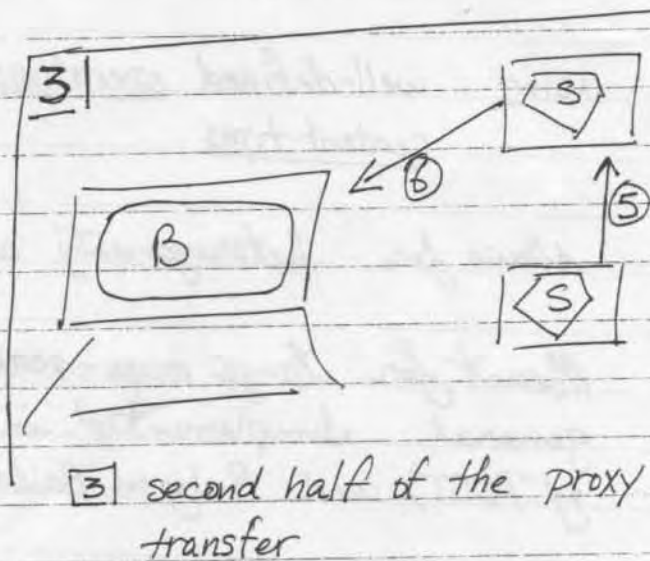
Concepts (cont.)

Same Origin Policy (for Proxies with Ajax)

Web browsers do not allow XMLHttpRequest connections to servers that did not deliver the original web page from which the request originates (Pruett (2006), p.24)



Q: why does this work?



Figures based on Fig. 4 from Pruett (2006)

CS4173

⑤

21 March 2007

References

Understanding AJAX

by Joshua Eichorn

(c) 2007 by Pearson Education

Published by Prentice-Hall

ISBN 0-13-221635-3

Ajax and Web Services

by Mark Pruett

(c) 2006 by O'Reilly Media, Inc.

O'Reilly ebook with ISBN 0-59-652853-1

In Appendix to these notes: pp. 6, 7, 16-23.

See Also

Tutorials & Readings in the Resources part of
the website

<http://www.cs.dal.ca/~jamie/CS4173/Res/#w3:Ajax>