Web Services, Web Apps, and REST

Stefan Tilkov, @stilkov | GOTO Copenhagen 2011



(Personal)

History

Sockets



























TWS -



http://www.loudthinking.com/arc/000585.html

"Add the missing parts to the Web"







RESTful HTTP



URIs Identifies Resources

http://example.com/orders?year=2008
http://example.com/customers/1234
http://example.com/orders/2007/10/776654
http://example.com/products/4554
http://example.com/processes/sal-increase-234



Resources are Linked



Representations in different Formats

XML JSON YAML HTML XHTML **PDF** Binary **Plain Text** inno









(a.k.a. The SOAP Way)













Enterprise

Web

"Here are the missing parts of the Web!"



"Thank you for nothing."



The Human Web











Microsoft[®]

Resources and URIs Pretty URLs Bookmarks Collaboration Caching Links/Hypermedia **Notification via Feeds**



Web Sites vs. Web Applications



Using the Web vs. Abusing the Web



There should be no applications



OrderEntryForm Product CustomerView Order Delivery

CustomerList

Customer

ProductMaintenanceForm

Product

ProductCatalog







What's in an app?










OrderManagementService

+ getOrders()

- + submitOrder()
- + getOrderDetails()
- + getOrdersForCustomers()
- + updateOrder()
- + addOrderItem()
- + cancelOrder()

CustomerManagementService

- + getCustomers()
- + addCustomer()
- + getCustomerDetails()
- + updateCustomer()
- + deleteCustomer()









Web Frameworks of 2001

Hide HTTP, URIs, HTML

Maintain Desktop-app state on server

Tunnel Stateful Apps through the Web



JSF Life Cycle





© 2011 innoQ Deutschland GmbF



How much HTTP would you like with that?



© 2011 innoQ Deutschland GmbH

Component-based

Wicket

JSF

Tapestry

ASP.NET

GWT



© 2011 innoQ Deutschland GmbH

Request/Response-oriented





© 2011 innoQ Deutschland GmbH

- Linkable and "pretty" URIs
- GET/POST differentiation
- Support for caching
- Full control over HTML
 - 1 Resource, n Views

Request/Response Frameworks



"But what about stateful Web applications?"



Turn session state ...





... into client or resource state









JavaScript



JavaScript Perception

"Toy language"

Incompatible

Inherent design problems

Low Performance





http://commons.wikimedia.org/wiki/File:Audi_S5_V8_FSI_engine.jpg

http://commons.wikimedia.org/wiki/File:Ateles_paniscus_-Brazil-8.jpg

The JavaScript Arms Race

Ext JS

CommonJS

http://oreilly.com/catalog/9780596517748

JavaScript Today

Popular & widely used

Often mandatory

Fast

Compatible

Best practices

5.1.7 Code-On-Demand

The final addition to our constraint set for REST comes from the code-on-demand style of <u>Section 3.5.3</u> (Figure 5-8). REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts. This simplifies clients by reducing the number of features required to be preimplemented. Allowing features to be downloaded after deployment improves system extensibility. However, it also reduces visibility, and thus is only an optional constraint within REST.

http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm#sec_5_1_7

Unobtrusive JavaScript

 Some Link

 Some Link

 Some Link

 Some Link

Caching

"I do think the REST-afarians are missing an opportunity by not driving home the secret sauce that is HTTP GET. [...] **GET is one of the most optimized pieces of distributed systems plumbing in the world.**

Don Box, Co-inventor of SOAP

Cacheable Elements



Obvious

Static text

Images CSS

JavaScript

Non-obvious

Menus Hierarchies Processing Results Immutable stuff

inno Q

The problem: Personalized dynamic elements







Logout



Login



Edge Side Includes (ESI)

```
<esi:include
src="http://example.com/1.html"
alt="http://bak.example.com/2.html"
onerror="continue"/>
```

<esi:include

src="http://example.com/search?query=\$(QUERY_STRING{query})"/>





Ajax







http://www.jenitennison.com/blog/node/154









"When a URI reference is used to perform a retrieval action on the identified resource, the optional fragment identifier, separated from the URI by a crosshatch ("#") character, consists of additional reference information **to be interpreted by the user agent after the retrieval action has been successfully completed**. As such, it is not part of a URI, but is often used in conjunction with a URI."

URI Specification, August 1998





The fragment identifier component of a URI **allows indirect identification of a secondary resource by reference to a primary resource and additional identifying information**. The identified secondary resource may be some portion or subset of the primary resource, some view on representations of the primary resource, or some other resource defined or described by those representations.

URI Specification, January 2005







http://example.com/someapp#!state=234

http://example.com/someapp?_escaped_fragment=234





http://googlewebmastercentral.blogspot.com/2009/10/proposal-for-making-ajax-crawlable.html



and #! Downsides

Complex

Inaccessible

Brittle

Invisible to search engines (#)



HTML 5

pushState() replaceState()



Summary



Embrace the Web, Don't fight it



Web UIs first, Client apps second



Don't repeat server mistakes on the client



If you can't link to it, does it exist?



If you can't build real Web apps with it, is it worth your time?



Thank you!







innoQ Deutschland GmbH Halskestr. 17 D-40880 Ratingen Phone: +49 21 02 77 172-100 innoQ Schweiz GmbH Gewerbestr. 11 CH-6630 Cham Phone: +41 41 02 743 01 11

www.innoq.com info@innoq.com

Stefan Tilkov stefan.tilkov@innoq.com http://www.innoq.com/blog/st/ @stilkov Phone: +49 170 471 2625