Reflections on the REST Architectural Style and "Principled Design of the Modern Web Architecture"

Roy T. Fielding Richard N. Taylor Justin R. Erenkrantz Michael M. Gorlick Jim Whitehead Rohit Khare Peyman Oreizy Adobe University of California, Irvine Bloomberg University of California, Irvine University of California, Santa Cruz Google Dynamic Variable LLC



















Outline

- 1. The Story of REST
- Early history of the Web
- What REST is (and is not)
- Contemporary influences
- 2. Work inspired by REST
 - Decentralization
 - Generalization
 - Secure computation
- 3. Reflections on REST
 - Investing in entrepreneurial students
 - Role of Software Engineering research

















PACKE









Sufyan bin Uzayr

[PACKT] ODen source

Original proposal for the World Wide Web



[Berners-Lee, 1989]



A bit of context



5

A bit of context



Three (very different) perspectives of the Web



HTML 4.01 Specification

W3C Recommendation 24 December 1999

http://www.w3.org/TR/1999/REC-html401-19991224 [405Kb], gzip'ed Postscript file [746Kb, 389 pages], gzip'ed PDF file [963Kb]) Latest version of HTML 4.01: (plain text [794Kb], gzip'ed tar archive of HTML files [371Kb], a .zip archive of HTML files [405Kb], gzip'ed Postscript file [746Kb, 389 pages], gzip'ed PDF file [963Kb])

Network Working Group

STD: 66

Abstract

Updates: 1738

Request for Comments: 3986

Obsoletes: 2732, 2396, 1808

Category: Standards Track

http://www.w3.ora/

Latest version of HTML: //www.w3

Previous version of HTML 4.01:

- Previous HTML 4 Recommendation:
- http://www.w3.org/TR/1998/REC-html40

Dave Raggett <dsr@w3.org> Arnaud Le Hors, W3C lan Jacobs, W3C

Copyright ©1997-1999 W3C[®] (MIT, INRIA, Keio), All Rights Reserved. W

This specification defines the HyperText Markup Language (HTMI World Wide Web. This specification defines HTML 4.01, which is a Status of this Memo to the text, multimedia, and hyperlink features of the previous vers and HTML 2.0 [RFC1866]), HTML 4 supports more multimedia opt This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state sheets, better printing facilities, and documents that are more acce HTML 4 also takes great strides towards the internationalization of the Web truly World Wide. and status of this protocol. Distribution of this memo is unlimited.

HTML 4 is an SGML application conforming to International Standa Copyright Notice Generalized Markup Language [ISO8879].

Status of this document

Network Working Group Request for Comments: 2068 Category: Standards Track

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests ssion and suggestions for improvements. Please refer to the current edition of the "Internet Official

> T. Berners-Lee W3C/MIT R. Fielding Day Software L. Masinter Adobe Systems January 2005



Protocols

Uniform Resource Identifier (URI):

Generic Syntax

A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource. This specification defines the generic URI syntax and a process for resolving URI references that might be in relative form, along with guidelines

and security considerations for the use of URIs on the lattice form, along with guidemites and security considerations for the use of URIs on the Internet. The URI syntax defines a grammar that is a superset of all valid URIs, allowing an implementation to parse the common components of a URI reference without knowing the scheme-specific requirements of every possible identifier. This specification does not define a generative grammar for URIs; that task is performed by the individual specifications of each URI cohome.

Copyright © The Internet Society (2005). All Rights Reserved.

Web Implementation (user view)



Web Implementation (origin view)





Web Architecture

Architecture is a vertical abstraction on implementation



Web Architecture

Web protocols define that vertical abstraction on implementation



So, which one is the Web?



11

HTML 4.01 Specification

W3C Recommendation 24 December 1999

This version:

http://www.w3.org/TR/1999/REC-html401-19991224 [405Kb], gzip'ed Postscript file [746Kb, 389 pages], gzip'ed PDF file [963Kb]) Latest version of HTML 4.01: (plain text [794Kb], gzip'ed tar archive of HTML files [371Kb], a .zip archive of HTML files [405Kb], gzip'ed Postscript file [746Kb, 389 pages], gzip'ed PDF file [963Kb])

Network Working Group

STD: 66

Abstract

Updates: 1738

Request for Comments: 3986

Obsoletes: 2732, 2396, 1808

Category: Standards Track

Latest version of HTML 4: http://www.w3.org/T

Latest version of HTML:

Previous version of HTML 4.01:

Previous HTML 4 Recommendation:

http://www.w3.org/TR/1998/REC-html40

Editors: Dave Raggett <dsr@w3.org> Arnaud Le Hors, W3C

Copyright ©1997-1999 W3C[®] (MIT, INRIA, Keic), All Rights Reserved. W and software licensing rules apply.

This specification defines the HyperText Markup Language (HTMI World Wide Web. This specification defines HTML 4.01, which is a Status of this Memo to the text, multimedia, and hyperlink features of the previous vers and HTML 2.0 [RFC1866]), HTML 4 supports more multimedia opt This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state sheets, better printing facilities, and documents that are more acce HTML 4 also takes great strides towards the internationalization of the Web truly World Wide. and status of this protocol. Distribution of this memo is unlimited.

HTML 4 is an SGML application conforming to International Standa **Copyright Notice** Generalized Markup Language [ISO8879].

Status of this document

Network Working Group Request for Comments: 2068 Category: Standards Track

Hypertext Transfer Protocol -- HTTP/1.1

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests ission and suggestions for improvements. Please refer to the current edition of the "Internet Official

> T. Berners-Lee W3C/MIT R. Fielding Day Software L. Masinter Adobe Systems January 2005



Protocols

Uniform Resource Identifier (URI):

Generic Syntax

A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource. This specification defines the generic URI syntax and a process for resolving URI references that might be in relative form, along with guidelines

and security considerations for the use of VRIs on the Internet. The URI syntax defines a grammar that is a superset of all valid URIs, allowing an implementation to parse the common components of a URI reference without knowing the scheme-specific requirements of every possible identifier. This specification does not define a generative grammar for URIs; that task is performed by the individual specifications of each URI scheme.

Copyright © The Internet Society (2005). All Rights Reserved.

So, which one is the Web?

All of them!

- The Web is a World-Wide System:
 - constantly running,
 - always changing,
 - anarchically accessed, and
 - independently deployed.

Outline

- 1. The Story of REST
- Early history of the Web
- What REST is (and is not)
- Contemporary influences
- 2. Work inspired by REST
 - Decentralization
 - Generalization
 - Secure computation
- 3. Reflections on REST
 - Investing in entrepreneurial students
 - Role of Software Engineering research



Roy T. Fielding. 2000. Architectural Styles and the Design of Network-based Software Architectures. Ph.D. Dissertation. University of California, Irvine, California, USA. http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

Roy T. Fielding and Richard N. Taylor. 2000. Principled Design of the Modern Web Architecture. In Proceedings of the 22nd Int'l Conference on Software Engineering. IEEE, Limerick, Ireland, 407-416.

Roy T. Fielding and Richard N. Taylor. 2002. Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology 2, 2 (May 2002), 115–150.



Why talk about my definition of REST?

Because HHE S has become a BUZZWORD

There's nothing particularly wrong with that... unless you happen to be me... or working with me



 Roy Fielding looked at the Web and saw that it was good

What is REST Anyway? _ zap**think**

Representational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web

Copyright © 2012, ZapThink, a Dovèl Technologies Company



Architectural Styles

• A horizontal abstraction across multiple architectures (vertical abstractions)

- names a repeated architectural pattern
- defined by its design constraints
- chosen for the properties they induce

REST is an architectural style

- for network-based applications
- to induce a specific set of architectural properties
- that were desired for the World Wide Web







REST is an accumulation of design constraints



REST is an accumulation of design constraints

Constraint



REST is an accumulation of design constraints







[photo by dhester: http://mrg.bz/xVLmr1]

[photo by EmmiP: http://mrg.bz/P7BJRi]

upertjefferies: http://mrg.bz/Y9XThf] [photo by

REST's Five Uniform Interface Constraints

- 1. All important resources are identified by one resource identifier mechanism • induces simple, visible, reusable, stateless communication

2. Access methods have the same semantics for all resources

• induces visible, scalable, available through layered system, cacheable, and shared caches

3. Resources are manipulated through the exchange of representations

induces simple, visible, reusable, cacheable, and evolvable (information hiding)

4. Representations are exchanged via self-descriptive messages

- induces visible, scalable, available through layered system, cacheable, and shared caches Induces evolvable via extensible communication

5. Hypertext as the engine of application state

- induces simple, visible, reusable, and cacheable through data-oriented integration
- induces evolvable (loose coupling) via late binding of application transitions

Deep dive into the hypermedia constraint

Hypertext as the Engine of Application State

each state can be dynamic each transition can be redirected

ESEC/FSE'17, September 8, 2017, Paderborn, Germany

Outline

- 1. The Story of REST
- Early history of the Web
- What REST is (and is not)
- Contemporary influences
- 2. Work inspired by REST
 - Decentralization
 - Generalization
 - Secure computation
- 3. Reflections on REST
 - Investing in entrepreneurial students
 - Role of Software Engineering research

WebDAV

- Distributed Authoring and Versioning
 - Returning to the Web's roots
 - Resources vs Representations vs Metadata
 - Stateless Interaction vs Session Locks
- Overwhelmed by commercial demands
 - XML, Locks, remote data model
 - Moved away from REST constraints, but helped enlighten and refine them

E. James Whitehead, Jr. "World Wide Web Distributed Authoring and Versioning (WebDAV): An Introduction." StandardView, Vol. 5, No. 1, March 1997, pages 3-8.

Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen, HTTP Extensions for Distributed Authoring - WEBDAV. Microsoft, U.C. Irvine, Netscape, Novell, Internet Proposed Standard RFC 2518. February, 1999.

E. James Whitehead, Jr. and Yaron Goland. 2004. The WebDAV Property Design. Software, Practice and Experience 34 (2004), 135–161.

(single containment, single membership, ordered, inclusion, delete removes all contained items)

stores

Dynamic Software Architectures

- How do you make adaptability easier?
 - Independent post-deployment evolvability
- Expose the application's architecture
 - Allow third-parties to evolve application by changing architecture
- Verify changes against semantic annotations on the system model
 - with assistance of external analysis modules
 - if change is okay, apply it to the implementation; else, take appropriate action; notify, prevent, ...

Peyman Oreizy, Nenad Medvidovic, and Richard N. Taylor. 2008 Runtime Software Adaptation: Framework, Approaches, and Styles. In *Companion of 30th International Conference on Software Engineering (ICSE Companion 2008).* ACM, 899-910. (Most Influential Paper Award for ICSE 1998)

Outline

- 1. The Story of REST
 - Early history of the Web
 - What REST is (and is not)
 - Contemporary influences

2. Work inspired by REST

- Decentralization
- Generalization
- Secure computation
- 3. Reflections on REST
 - Investing in entrepreneurial students
 - Role of Software Engineering research

New challenges for real-time network-based applications

By 2000, there was a boom in real-time applications:

 Push, Peer-to-Peer, Publish/ Subscribe, Instant Messaging, and Internet-Scale Event Notification...

- REST had gaps for real-time:
 - One-shot: no retry if response lost
 - One-to-one: no concurrent groups
 - One-way: no asynchronous links

Latency & Agency concerns

Rohit Khare and Richard N.Taylor. 2004. Extending the REpresentational State Transfer Architectural Style for Decentralized Systems. In Proceedings of the 26th International Conference on Software Engineering (ICSE'04). IEEE Computer Society, Edinburgh, Scotland, UK, 428–437. (Distinguished Paper award for ICSE 2004) http://www.ics.uci.edu/~rohit/ARRESTED-ICSE.pdf

Computation exchange: CREST

Technical triad of the Web:

- URLs name information resources
- Metadata used for distinguishing representations 2.
- HTTP defines exchanges between clients and servers З.
- What happens when you generalize?
 - A. CURLs name *computation resources*
 - B. *Metaprogamming* is used for examining and describing computations
 - C. Asynchronous protocol for *peer-to-peer* exchanges

• Analysis of the essential architectural decisions of the WWW, followed by generalization, opened up an entirely new space of decentralized, Internet-based applications based on computations as the fundamental concept.

Justin R. Erenkrantz, Michael M. Gorlick, Girish Suryanarayana, and Richard N. Taylor. 2007. From Representations to Computations: The Evolution of Web Architectures. In ACM SIGSOFT Symposium on The Foundations of Software Engineering (FSE'07). 255–264.

CREST learned about deferring code from 'living labs' like Subversion

CREST with security: COAST

- Capability based security model with computation exchange
- Exchange active computations among peers: Code + run-time state (reified as closures and continuations)
- Novel security mechanism: Capability URL (CURL)
 - Dictates where computations may go
 - Bounds what visiting computations can do
 - Limits resource consumption of computations
- Architectural style: Computational State Transfer (COAST)
 - Build capability security into the architectural style
 - Functional capability: What can a visiting computation do?
 - Communication capability: With whom, when, and how often may that computation communicate?

Michael Martin Gorlick. 2016. Computational State Transfer: An Architectural Style for Decentralized Systems. Ph.D. Dissertation. University of California, Irvine.

Michael M. Gorlick, Kyle Strasser, and Richard N. Taylor. 2012. COAST: An Architectural Style for Decentralized On-Demand Tailored Services. In Proceedings of 2012 Joint Working Conference on Software Architecture & 6th European Conference on Software Architecture (WICSA/ECSA'12). 71–80.

COAST: Computational State Transfer

• An Architectural Style for the Idiom of Computation Exchange

- 1. Service: All services are computations whose sole means of interaction is the asynchronous messaging of values, closures, continuations and binding environments
- 2. Execution: Each computation executes within the confines of some execution site $\langle E, B \rangle$ where E is an execution engine and B is a binding environment
- 3. Messaging: Computation x may deliver a message to computation y only if x holds a CURL u and y has the authority to read a message via u
- A Capability URL (CURL) conveys the authority to communicate and is a tamper-proof cryptographic structure that can not be forged or guessed.
- 4. Interpretation: The interpretation of a message delivered to computation y via CURL u of y is u-dependent
- 5. The Service and Messaging rules confer communication by introduction:
- Computation x can message computation y only if x has been introduced to y beforehand 6. Ab initio, endowment, Messaging, or Execution

Outline

- 1. The Story of REST
 - Early history of the Web
 - What REST is (and is not)
 - Contemporary influences
- 2. Work inspired by REST
 - Decentralization
 - Generalization
 - Secure computation

3. Reflections on REST

- Investing in entrepreneurial students
- Role of Software Engineering research

Investing in entrepreneurial students, over long periods...

The Role of Software Engineering research

This is Software Engineering research

- It is about Design
- It is fundamentally software architecture
- It is based on reflection
- This is how styles....good styles... get developed: Experience, evaluation, reflection. Wash Rinse Repeat.
- The research environment was "unusual"
 - DARPA funding with a long leash (Thanks Bill and John!)
 - Lots of travel. Lots
 - A (mostly) accommodating university process
 - •9 years to Ph.D after B.S.
- Contrast with today's environment...

• A highly interactive research community: UCI's PhD students, W3C, IETF, and close industry links

The Paper

- Web Architecture was submitted to FSE99 a year earlier.
- old technolgoy [sic] now. lots of jargon make the paper difficult to this paper."
- The ICSE 2000 paper had:

• no surveys, no statistical analyses, and essentially no evaluation section. It merely stated:

architecture."

That work, in its multiple forms, has now been cited over 8000 times...

• The first version of what eventually became Principled Design of the Modern

It was rejected, with reviewer comments including "Over all, the originality of the paper is quite low. There is only little to learn from it." and "- the web is understand. ... - I can't find a novel lessons [sic] for software engineers in

• "The REST architectural style has been validated through six years of development of the HTTP/1.0 and HTTP/1.1 standards, elaboration of the URI and relative URL standards, and successful deployment of several dozen independently developed, commercial-grade software systems within the modern Web

Acknowledgments

Funding

• DARPA: Bill Scherlis and John Salasin; National Science Foundation; ISR's corporate sponsors

The Web

- Tim Berners-Lee, Henrik Frystyk Nielsen, Dan Connolly, Dave Raggett, and Larry Masinter
- REST advocates

Colleagues

Jason Robbins, David Rosenblum, and Girish Suryanarayana.

Ihanks —

• To the SIGSOFT community for the honor of this Impact Award.

• Mark Baker, Paul Prescod, Mike Amundsen, Leonard Richardson, Sam Ruby, and Aaron Swartz

• Mark Ackerman, Ken Anderson, Greg Bolcer, Eric Dashofy, Nenad Medvidovic, Kari Nies, Jie Ren,

Questions?

6

2

PENGUINS

