

707.000
Web Science and Web Technology
„Web Technologies I“

Markus Strohmaier

Univ. Ass. / Assistant Professor
Knowledge Management Institute
Graz University of Technology, Austria

e-mail: markus.strohmaier@tugraz.at
web: <http://www.kmi.tugraz.at/staff/markus>

Overview

Agenda

- Intermediate Course Feedback
- REST
- JSON
- Home Assignment 6

Midterm Course Feedback

Verbesserungsvorschläge, Kritik, Meinungen, usw

- Inhalte
 - Vorkenntnisse
 - Umfang, Schwierigkeitsgrad
 - Geschwindigkeit
 - Relevanz / Interesse
- Pausen
- Vortragstil
 - Geschwindigkeit
 - Interaktion
- Unterrichtssprache
 - Deutsch / Englisch
- Gastvorträge
- Übungen
 - Umfang / Aufwand / Zeit
 - Schwierigkeitsgrad
 - Verständlichkeit
- Bakk.Arbeit / Projekt

Weitere Aspekte?

REST

Roy Fielding, Dissertation 2000

- Roy Fielding
 - Chief Scientist, [Day Software](#)
 - Co-founder and member, [The Apache Software Foundation](#)
 - Dissertation on Architectural Styles and the Design of Network-based Software Architectures at the [Information and Computer Science, UC Irvine](#)



In his dissertation, he „introduce[s] the Representational State Transfer (REST) architectural style and describe[s] how REST has been used to guide the design and development of the architecture for the modern Web“


Ressources:

- Chapter 5, <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- http://roy.gbiv.com/talks/200709_fielding_rest.pdf


Has played a role in authoring the Internet standards for the Hypertext Transfer Protocol (HTTP) and Uniform Resource Identifiers (URI)

REST

Roy Fielding, RailsConf Europe, September 2007



Absence of occupation is not REST,
A mind quite vacant is a mind distress'd. [William Cowper]

CO-PRESENTED BY O'REILLY 

The Problem (circa 1994)

Early architecture was based on solid principles

- URLs, separation of concerns, simplicity
- lacked architectural description and rationale

Protocols assumed a direct server connection

- no awareness of caching, proxies, or spiders
- many independent extensions

Public awareness of the Web was just beginning


- exponential growth threatened the Internet
- commercialization meant new requirements and new stakeholders

A modern Web architecture was clearly needed


- but how do we avoid breaking the Web in the process?

REST

Roy Fielding, RailsConf Europe, September 2007



Everywhere I have sought REST and not found it, except
sitting in a corner by myself with a little book. [Thomas Kempis]

CO-PRESENTED BY O'REILLY 


Software Architectures

A software architecture is an abstraction of the run-time elements of a software system during some phase of its operation. A system may be composed of many levels of abstraction and many phases of operation, each with its own software architecture.


- A software architecture is defined by a configuration of architectural elements—components, connectors, and data—constrained in their relationships in order to achieve a desired set of architectural properties.
- A configuration is the structure of architectural relationships among components, connectors, and data during a period of system run-time.

REST

Roy Fielding, RailsConf Europe, September 2007



Da requiem; requietus ager bene credita reddit. [Ovid]
(Take REST; a field that has RESTed gives a bountiful crop.)

CO-PRESENTED BY O'REILLY 

Architectural Styles

An architectural style is a coordinated set of architectural constraints that restricts the roles and features of architectural elements, and the allowed relationships among those elements, within any architecture that conforms to that style.


- A style can be applied to many architectures
- An architecture can consist of many styles

Design at the right level of abstraction


- Styles help architects communicate architecture
- Architecture determines potential system properties
- Implementation determines actual system properties
- Architectural patterns are styles with common recipes

REST

Roy Fielding, RailsConf Europe, September 2007



... And some seek fame, that hovers in the distance; ...

CO-PRESENTED BY O'REILLY 

Web Architecture

One abstraction level above the implementation

Components

- User agents, Intermediaries, Servers
- Browsers, Spiders, Proxies, Gateways, Origin Servers

Connectors

- HTTP: a standard transfer protocol to prefer over many

Data

- URI: one identifier standard for all resources
- HTML, XML, RDF, ...: common representation formats to describe and bind resources

REST

Roy Fielding, RailsConf Europe, September 2007

*... But all are seeking REST.
[Rev. Frederick Langbridge]*

CO-PRESENTED BY O'REILLY

Web Architectural Style

- One abstraction level above Architecture**
 - two abstraction levels above implementation
 - that's one too many for most folks
- An architectural style is a set of constraints**
 - unfortunately, constraints are hard to visualize
 - kind of like gravity or electromagnetism
 - observed only by their effect on others
- Constraints induce architectural properties**
 - both desirable and undesirable properties
 - a.k.a., software qualities
 - a.k.a., design trade-offs

Software Architecture and Software Qualities: Example

Gross, D. and Yu, E. (2001). "From Non-Functional Requirements to Design through Patterns." Requirements Engineering 6: 18-36.

• The Problem

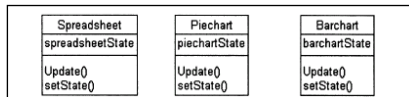
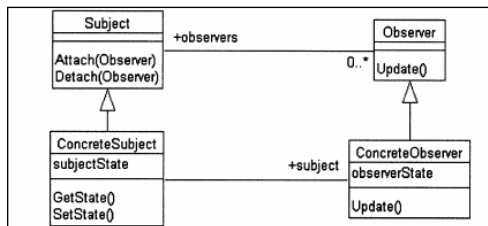


Fig. 3. Current design state: how to make those objects work together?

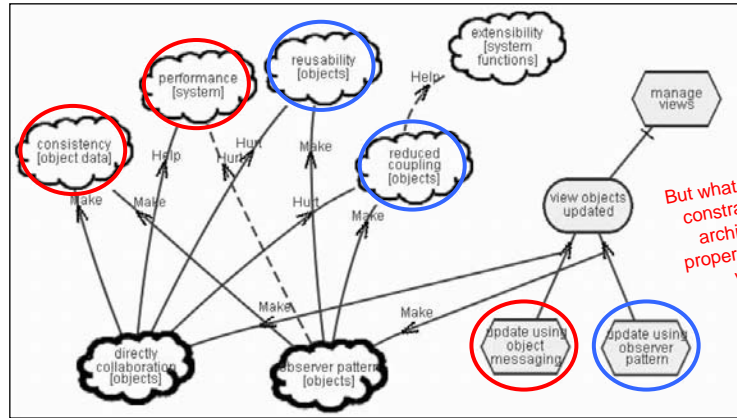
• One (!) Solution: Observer Pattern



Software Architecture and Software Qualities: Example

Gross, D. and Yu, E. (2001). "From Non-Functional Requirements to Design through Patterns." Requirements Engineering 6: 18-36.

- Conflicts, Tradeoffs, Implications



But what are these constraints and architectural properties on the web?

REST

Roy Fielding, RailsConf Europe, September 2007

How beautiful it is to do nothing, and then REST afterward. [Spanish Proverb]

CO-PRESENTED BY O'REILLY

Style = nil

Starting from a condition of no constraints...

REST

Roy Fielding, RailsConf Europe, September 2007

REST is not idleness, ...

CO-PRESENTED BY O'REILLY

Style += Client/Server

Apply separation of concerns: Client-Server

Representation Data

improves UI portability simplifies server

enables multiple organizational domains

Separating UI concerns from data storage concerns

REST

Roy Fielding, RailsConf Europe, September 2007

... and to lie sometimes on the grass ...

CO-PRESENTED BY O'REILLY

Style += Stateless

Constrain interaction to be stateless...

each request from client to server must contain all of the information necessary to understand the request

degrades efficiency
Repetitive data

simplifies server

improves scalability

improves reliability

REST

Roy Fielding, RailsConf Europe, September 2007

... under the trees on a summer's day, ...

CO-PRESENTED BY O'REILLY

RAILSCONF EUROPE

Style += Caching

Add optional non-shared caching

the potential to partially or completely eliminate some interactions

degrades reliability

reduces average latency

improves efficiency

improves scalability

Cachable vs. non-cachable content

REST

Roy Fielding, RailsConf Europe, September 2007

... listening to the murmur of water, ...

CO-PRESENTED BY O'REILLY

RAILSCONF EUROPE

Style += Uniform Interface

Apply generality: uniform interface constraint

improves visibility

degrades efficiency

independent evolvability

decouples implementation

information is transferred in a standardized form

REST

Roy Fielding, RailsConf Europe, September 2007

RAILSCONF EUROPE ... or watching the clouds float across the sky, ... CO-PRESENTED BY O'REILLY

Style += Layered System

Apply info hiding: layered system constraints

Complexity reduction: each component cannot "see" beyond the immediate layer

Markus S

Tuesday, September 18, 2007 18 17

REST

Roy Fielding, RailsConf Europe, September 2007

RAILSCONF EUROPE ... is by no means a waste of time. [Sir John Lubbock] CO-PRESENTED BY O'REILLY

REST Style

Optional! Finally, allow code-on-demand (applets/js)

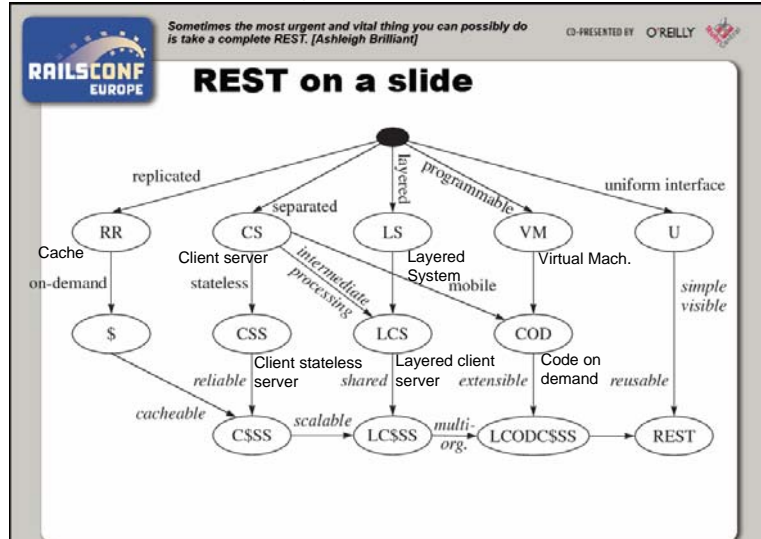
by downloading and executing code in the form of applets or scripts

Markus S

Tuesday, September 18, 2007 19 18

REST

Roy Fielding, RailsConf Europe, September 2007




Why bother?

Roy Fielding, Dissertation 2000

- “creating an **architectural model** for how the Web should work”
- “Using the new architectural style as a guide, we can **compare proposed extensions and modifications** to the Web architecture against the constraints within the style. **Conflicts** indicate that the proposal would violate one or more of the design principles behind the Web.
- For **severe conflicts**, such as a change in the interaction style, the same functionality would **either be replaced** with a design more conducive to the Web's style, or the proposer would be told to implement the functionality as a **separate architecture** running in parallel to the Web. “

REST

Roy Fielding, RailsConf Europe, September 2007




RAILSCONF
EUROPE

... And some seek fame, that hovers in the distance; ...

Web Architecture

CO-PRESENTED BY O'REILLY



One abstraction level above the implementation

Components

- User agents, Intermediaries, Servers
- Browsers, Spiders, Proxies, Gateways, Origin Servers

Connectors

- HTTP: a standard transfer protocol to prefer over many

Data

- URI: one identifier standard for all resources
- HTML, XML, RDF, ...: common representation formats to describe and bind resources

Representations

Representations

Roy Fielding, Dissertation 2000

- *“REST components perform **actions** on a **resource** by using a **representation** to capture the **current** or **intended** state of that resource and transferring that representation between **components**.”*
- *“less precise names for a representation include: document, file, and HTTP message entity, instance, or variant.”*
- *Depending on the message control data, a given representation may indicate the current state of the requested resource, the desired state for the requested resource, or the value of some other resource [...].*

REST

Chapter "Representational State Transfer (REST)" in "Pro PHP XML and Web Services", R. Richards 633–672 (2006)

Table 17-1. *HTTP Methods for REST*

Method	CRUD Operation	Description
GET	Retrieve	Retrieves the representation of a resource.
HEAD		Retrieves metadata for the representation and resource.
POST	Create	In the strict sense, POST creates a resource. In the real world, however, POST is typically used to create, update, and even delete a resource. It is normal to use REST services that support only GET and POST.
PUT	Update	Updates a resource. More often than not, you will not see this method used in the real world but instead will see POST used to perform the actions.
DELETE	Delete	Deletes a resource. Just like PUT, in the real world this is rarely used, and instead POST is used in its place.

Theory vs. Practice

Chapter "Representational State Transfer (REST)" in "Pro PHP XML and Web Services", R. Richards 633–672 (2006) and

- How has this theory influenced current practice?

REST applied to HTTP

- The REST service is expressed **as a URL** and is accessed with **basic HTTP requests**, such as <http://bea.com/content/getArticles?author=joe>.
- The **HTTP verb** is important: a GET is a read operation, POST is a creation, and PUT make updates to the service.
- The return payload is usually **XML** or **JSON**.

<http://dev2dev.bea.com/pub/a/2007/05/google-mashups.html>

JSON

Slides by Sang Shin, Java Technology Architect, Sun Microsystems, Inc, and
<http://www.json.org/fatfree.html>

JSON (JavaScript Object Notation) is a programming language model data interchange format. It is minimal, textual, and a subset of JavaScript.

What is JSON?

- Lightweight data-interchange format
 - > Compared to XML
- Simple format
 - > Easy for humans to read and write
 - > Easy for machines to parse and generate
- JSON is a text format
 - > Programming language independent
 - > Uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python

A Fat-Free Alternative to XML?

JSON

Slides by Sang Shin, Java Technology Architect, Sun Microsystems, Inc

JSON 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
- These are universal data structures supported by most modern programming languages

JSON

Slides by Sang Shin, Java Technology Architect, Sun Microsystems, Inc



JSON Object Notation

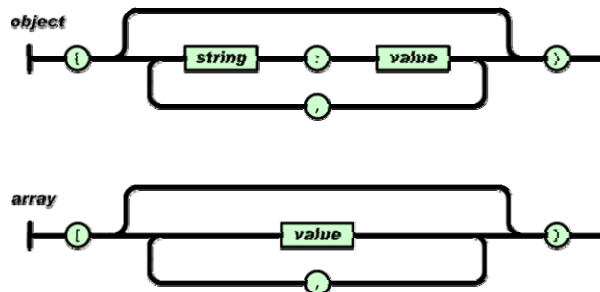
- A JSON object is an unordered set of name/value pairs
- A JSON object begins with { (left brace) and ends with } (right brace)
- Each name is followed by : (colon) and the name/value pairs are separated by , (comma)

10

JSON

www.json.org

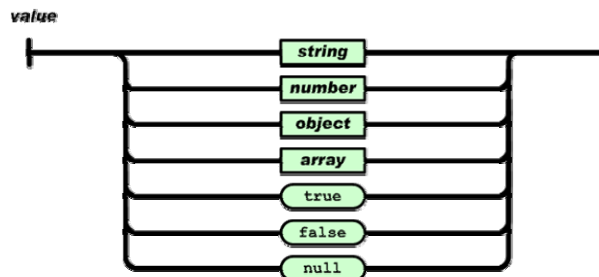
- The types represented in JSON are strings, numbers, booleans, object, arrays, and null.



JSON

www.json.org

- The types represented in JSON are strings, numbers, booleans, object, arrays, and null.



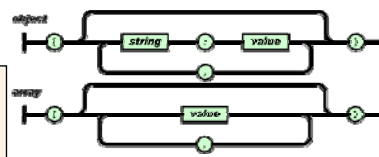
JSON

www.json.org

- JSON vs. XML

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        { "value": "New", "onclick": "CreateNewDoc()" },

```



```
contacts.htm - WordPad
File Edit View Insert Format Help
[Icons]
{"contacts":{"total":32,"values":[{"id":"2r1at7fefxxz9","relationship":{"description":"friend","friend":true},"contactRelationship":"friend","profileUrl":"http://browse.ning.com/any/2r1at7fefxxz9","updateDate":"2007-10-10T13:47:46.731Z","name":"Patrick Hoefler","screenName":"2r1at7fefxxz9","owner":null,"photoUrl":"http://api.ning.com/files/EX3L6wZzmjJFJ4SDGrv659CLS0x*s6fg6yEjt4w7f84_?crop=143&1","createDate":"2007-10-10T13:11:31.541Z"}]}
```

JSON

<http://www.json.org/fatfree.html>

- JSON has **no version number**. No revisions to the JSON grammar are anticipated. If something has a 1.0, it will inevitably get a 1.1 and a 2.0, and everything is crap until it is 3.0. JSON is very stable.
- JSON **doesn't have namespaces**. Every object is a namespace: its set of keys is independent of all other objects, even exclusive of nesting. JSON uses context to avoid ambiguity, just as programming languages do.
- JSON **is not extensible**. It does not need to be. It can represent any non-recurrent data structure as is. JSON is flexible. New fields can be added to existing structures without obsoleting existing programs.

JSON

Slides taken from Sang Shin, Java Technology Architect, Sun Microsystems, Inc

Text to Object Conversion in JavaScript code

```
var myObject = eval('(' + myJSONtext + ');');
```

- To convert a JSON text into an JSON object, use the `eval()` function
 - > `eval()` invokes the JavaScript compiler
 - > Since JSON is a proper subset of JavaScript, the compiler will correctly parse the text and produce an object structure

JSON

Slides taken from Sang Shin, Java Technology Architect, Sun Microsystems, Inc

Security & JSON Parser

```
// Include http://www.json.org/json.js  
var myObject = myJSONtext.parseJSON();
```

- `eval()` can compile and execute any JavaScript program, so there can be security issues (cross-site scripting)
 - > Use `eval()` when the source can be trusted
- When security is a concern - the source cannot be trusted -, it is better to use a JSON parser
 - > A JSON parser will only recognize JSON text and so is much safer

14

JSON

Slides taken from Sang Shin, Java Technology Architect, Sun Microsystems, Inc

How to Receive JSON Data at the Client Side

- JSON data is received as a string
- Calling `eval()` will generate JSON object in JavaScript code
 - > `var JSONdata = eval(req.responseText);`
- Once you have JSON object, you can use `.` notation to access its properties
 - > `var name = JSONdata.name;`
 - > `var address = JSONdata.addresses[3];`
 - > `var streetname = JSONdata.addresses[3].street;`

21

JSON + REST

DEMO

<http://dev2dev.bea.com/pub/a/2007/05/google-mashups.html?page=last>

Remember Home Assignment 1?

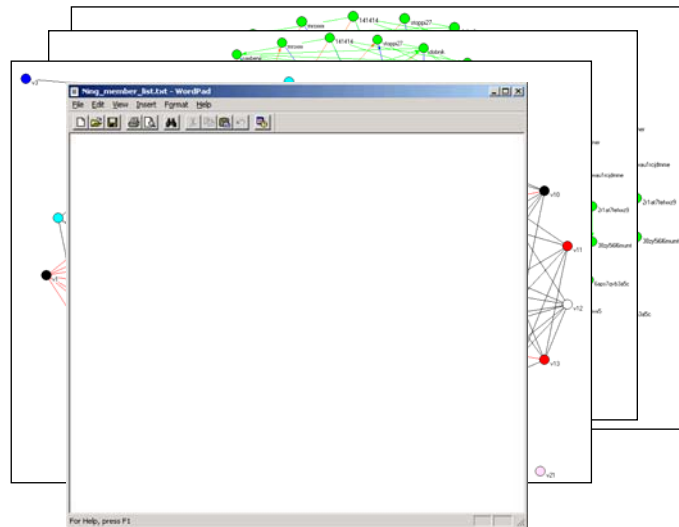


TASK OVERVIEW

Let's apply some of the tools, platforms, technologies, techniques and concepts we have discussed in the past:

- our experimental social network on ning.com
- Ego-centred Networks
- Average degree, degree distributions, diameter, clustering coefficient
- Page Rank
- Configuration model and random graphs
- Pajek
- REST
- JSON

Home Assignment 6



Home Assignment 6

Get familiar

with Ning's REST API

- <http://documentation.ning.com/sections/rest.php>

with Pajek and the Pajek .net Format

- <http://vlado.fmf.uni-lj.si/pub/networks/pajek/>
- <http://iv.slis.indiana.edu/lm/lm-pajek.html>

Useful Links

JSON Reader Tool

- <http://jsontools.berlios.de/>

Eclipse Plugin ANTLR (parser generator)

- <http://antreclipse.sourceforge.net/>

Other JSON Tools

- <http://www.json.org/java/>

Any questions?

See you next week!