REST

What is REST?

- Representational State Transfer
- Style of Software Architecture for WWW
- Based on Client and Server
- Request and Response are built on transfer of “representation” or “resources”
- Uses HTTP protocol

Sites Using REST

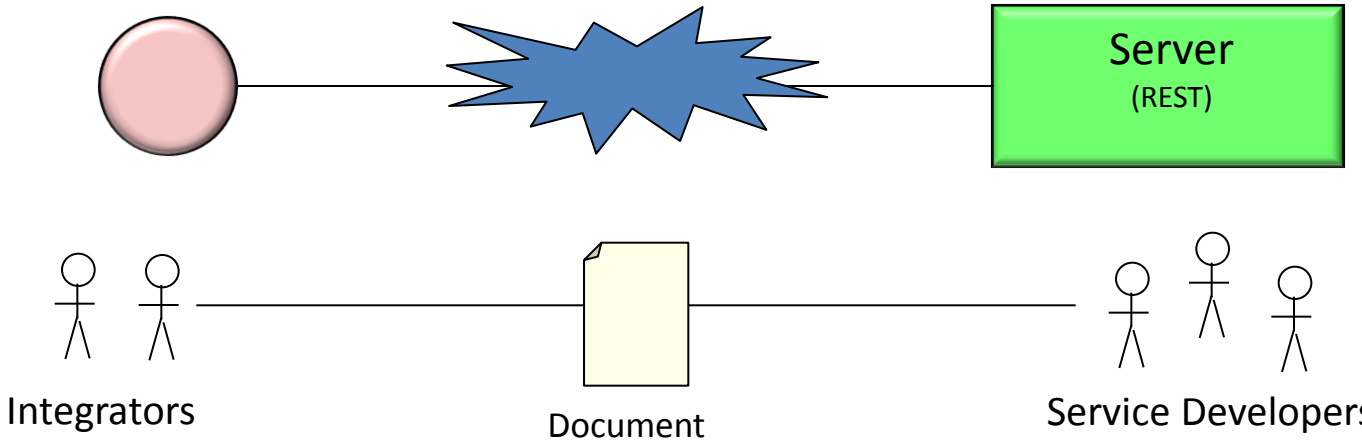
- Amazon
- Yahoo,
- Google (search, OpenSocial)
- Flickr
- FaceBook
- MySpace
- LinkedIn
- IBM
- Microsoft
- Digg
- eBay
- Etc…

Resources

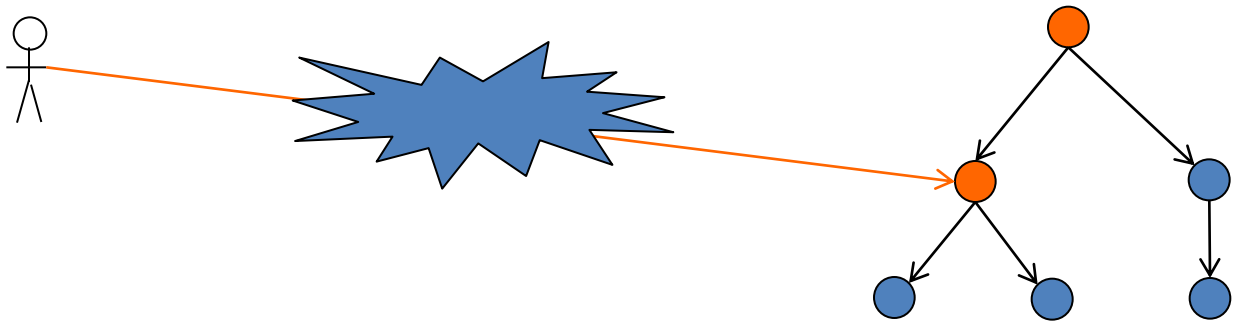
- Are just Concepts
- Located by URIs
- URIs tell client that there's a concept somewhere
- Client then asks for specific representation of the concept from the representations the server makes available
- E.g Webpage is a representation of a resource

contd..

and how REST is viewed



Path (URL) traversal



http://www.server.com/data/feature2

RESTful Web Services

- Representational State Transfer (REST)
  - Roy Fielding, 2000 (doctoral dissertation)
    - Examination of the Internet as a stateless service of near-limitless expansion model with a simple but effective information delivery system

- Key concepts

- Resources - source of information
- Consistent access to all resources
  - As in interface and communication – Not content or function
- Stateless protocol
- Hypermedia – links in the information to other data (connectedness)

REST

- Concepts/Components
  - Representational
    - Information returned from a resource (represented using URLs)
    - Uniquely identifiable information
  - State
    - The accessing of information by the client (e.g., browser) is a state change
    - Viewing http://www.mysite.com changes what information is being accessed
  - Transfer
    - The act of changing state (URL) transfers information to the client

Why is it called REST?

- Web is comprised of resources (item of interest)
- E.g. Library System may define a book, Harry Potter under author JK Rowling like
- http://www.library.sjsu.edu/JKRowling/HarryPotter
- This will return eg JK_HarryPotter.html
- This places the client in a state.

Definition

- REST is intended to evoke an image of how a well-designed Web application behaves: a network of web pages (a virtual state-machine), where the user progresses through an application by selecting links (state transitions), resulting in the next page (representing the next state of the application) being transferred to the user and rendered for their use.

  - Roy Fielding in his Ph.D. dissertation in the year 2000

Defining RESTful web services

- Standards
  - HTTP
  - URL
  - MIME Types
    - E.g., text/html, text/xml, image/png
  - XML
- As a style
  - REST uses informal guidelines
  - A large portion of the REST approach is driven by “what everyone else is doing”
  - This approach can simplify communication for projects or introduce significant integration problems

Resources

- Are just concepts
- Located by URIs
- URIs tell client that there's a concept somewhere
- Client then asks for specific representation of the concept from the representations the server makes available
- E.g Webpage is a representation of a resource

What is the REST style?

- REST is often described as an architecture style
  - Set of formal and informal guides to creating architectures – “constraints”
    - Client-server
    - Stateless
    - Cacheable
    - Layered system
    - Uniform interface
    - Code on demand (optional)

- An Application can interact with resource by knowing only two things :
  - Identifier of the resource
  - Action to be performed on the resource

Example

- Consider a Book store which has enabled a Web Service for book ordering.
- It should enable customers to
  - Get a list of books
  - Get detailed information on each book
  - Submit a order to purchase it

contd..

Get books list

- Client uses following URL to get books list
  http://www.bookstore.com/books

- The server returns

```
<?xml version="1.0"?>
<b:Books xmlns:p="http://www.Books-store.com"
xmlns:xlink="http://www.w3.org/1999/xlink">
 <Book id="00345" xlink:href="http://www.Books-store.com/Books/00345"/>
<Book id="00346" xlink:href="http://www.Books-store.com/Books/00346"/>
<Book id="00347" xlink:href="http://www.Books-store.com/Books/00347"/>
<Book id="00348" xlink:href="http://www.Books-store.com/Books/00348"/>
</p:Books>
```

contd..

Get detailed information on each book

- The client then uses one of the book id and traverses to that resource.
- The client uses

http://www.bookstore.com/books/00346

The Server responds with

```
<?xml version="1.0"?>
<p:Book xmlns:b="http://www.Books-store.com"
xmlns:xlink="http://www.w3.org/1999/xlink">
      <Book-ID>00345</Book-ID>
    <Name>Harry Potter</Name>
    <Description>This boos is written by JK Rowling</Description>
  <Specification xlink:href="http://www.Books-store.com/books/00346/readings"/>
     <UnitCost currency="USD">20.0</UnitCost>
      <Quantity>10</Quantity>
</b:Book>
```

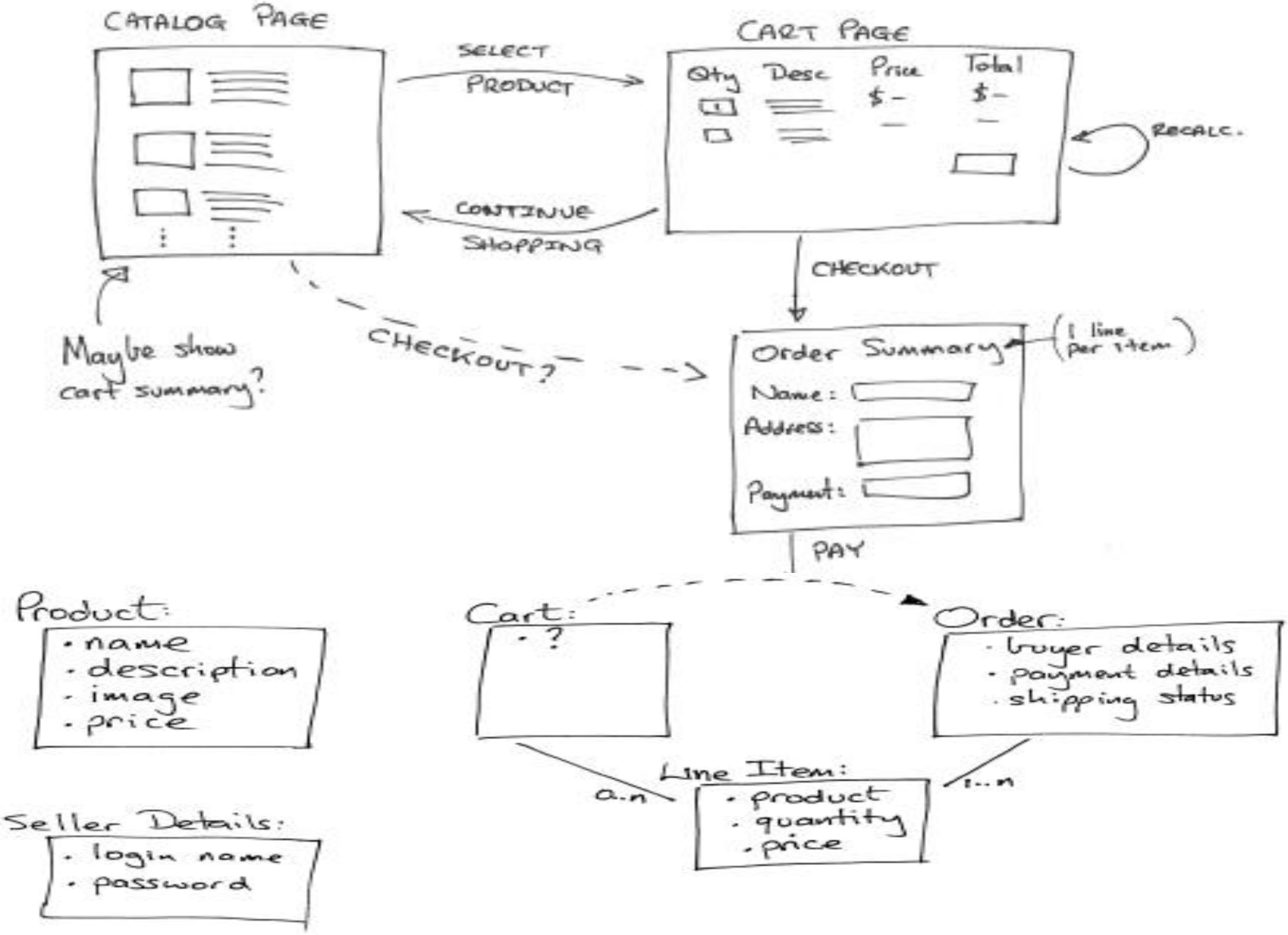
contd..

Creating a RESTful service

Steps

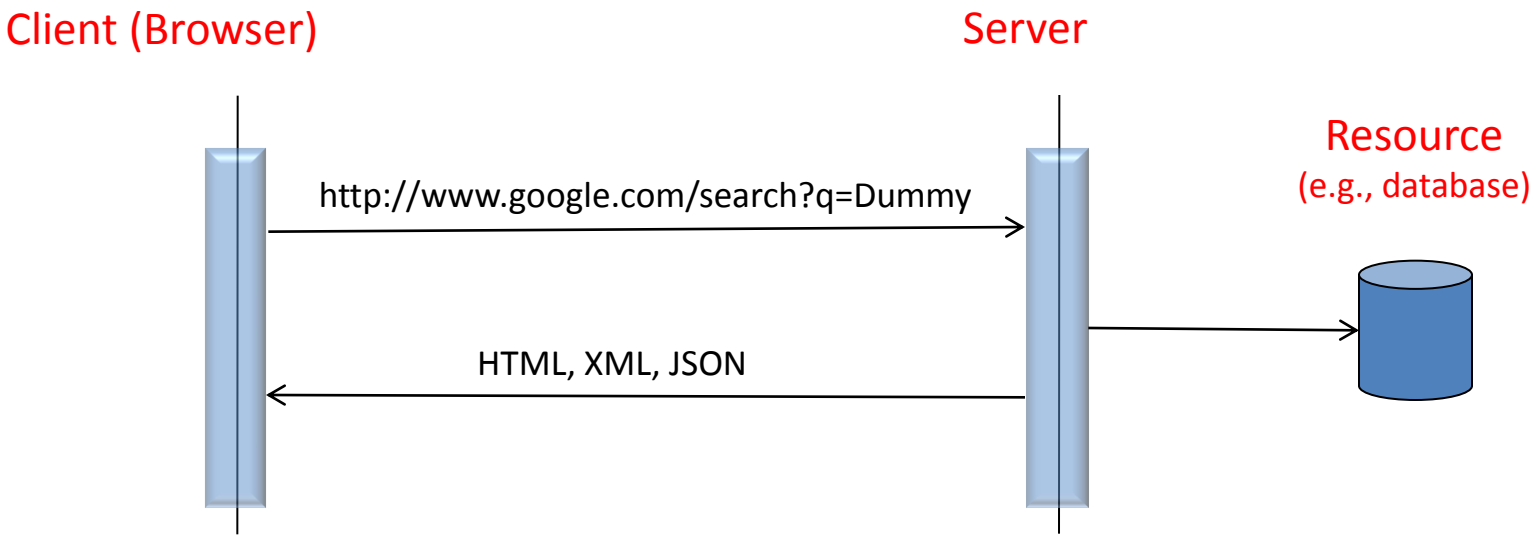
1. Define the domain and data
2. Organize the data in to groups
3. Create URI to resource mapping
4. Define the representations to the client
   (XML, HTML, CSS, …)
5. Link data across resources (connectedness or hypermedia)
6. Create use cases to map events/usage
7. Plan for things going wrong

RESTful Web Services, *Richardson & Ruby*


CATALOG PAGE
SELECT PRODUCT
CART PAGE
Qty
Desc
Price
Total
$-
$-
RECALC.
CONTINUE SHOPPING
CHECKOUT
Maybe show cart summary?
CHECKOUT?
Order Summary
(1 line per item)
Name:
Address:
Payment:
PAY
Product:
• name
• description
• image
• price
Cart:
• ?
Order:
• buyer details
• payment details
• shipping status
Line Item:
• product
• quantity
• price
0..n
1..n
Seller Details:
• login name
• password

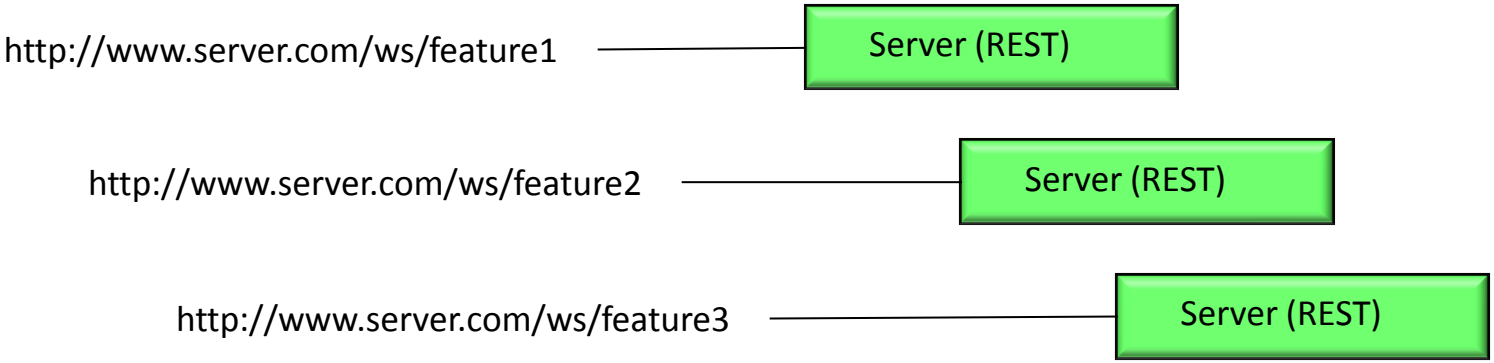
Lets look at an example of the RESTful approach

- If we view the web as repositories of information and the _____as the primary key (or index) to this information, we have the basic design of a REST-based system



Server Side

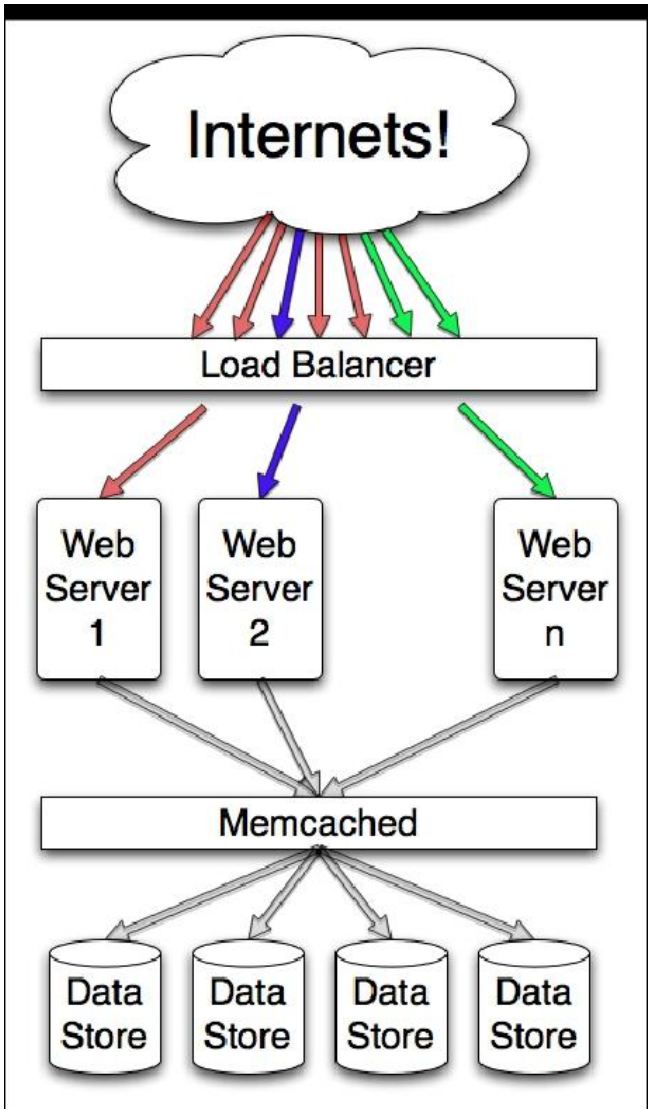
- Distribution of services are provided by ____

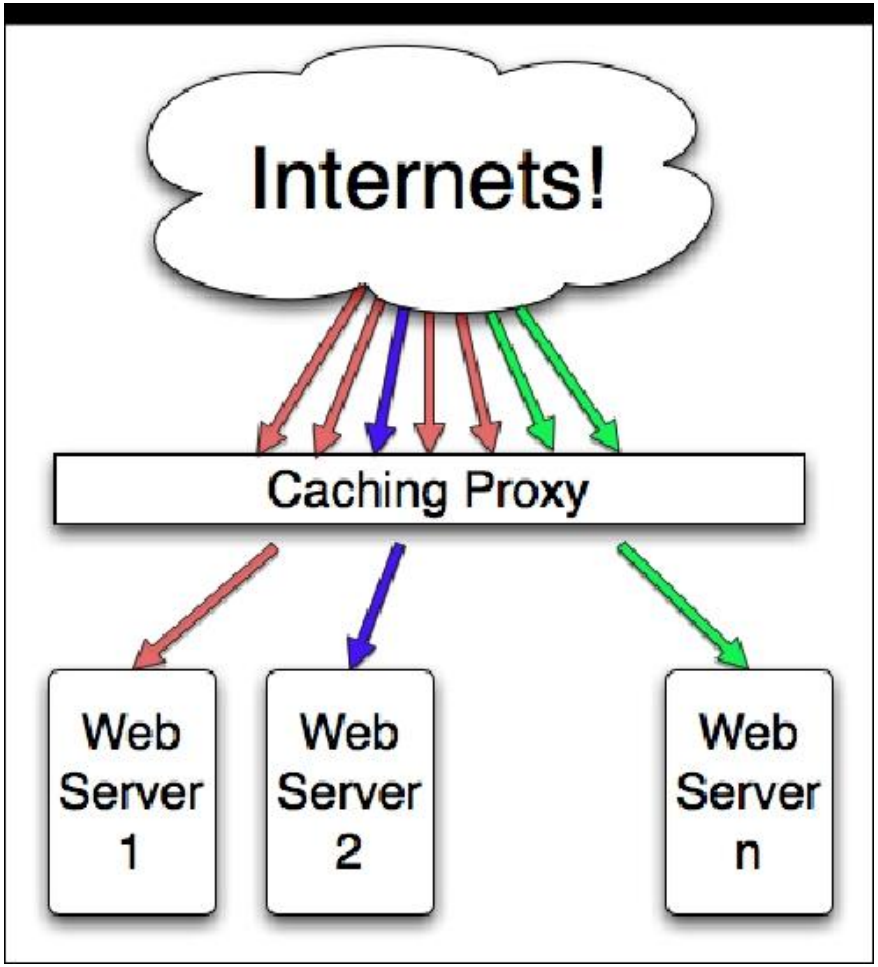


When is it appropriate?

- The web services are completely stateless
- Caching infrastructure can be leveraged for performance
- The service producer and consumer have mutual understanding of context and content being passed e.g NetFlix Api
- Bandwidth is a constraint

Multi-level Caching





References

- Roy Fielding, 2000 (doctoral dissertation)
  - http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
- http://java.sun.com/developer/technicalArticles/WebServices/restful/
- http://en.wikipedia.org/wiki/Representational_State_Transfer
- http://www.devx.com/DevX/Article/8155
- http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm