RESTful web service interfaces

don dempsey

Center for Clinical and Translational Science (CCTS)

RESTful web services are easy.

Examples of RESTful web services

1) RxNorm RESTful web services (National Library of Medicine)

2) Amazon Relational Database Service

Example 1: RxNorm RESTful web services

(National Library of Medicine)

"RxNorm provides normalized names for clinical drugs and links its names to many of the drug vocabularies commonly used in pharmacy management and drug interaction software."

Action: Resource "/drugs?name=value"

Sample Request:

```
http://rxnav.nlm.nih.gov/REST/drugs?name=cymbalta
```

Sample Response:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<rxnormdata>
  <drugGroup>
    <name>cymbalta</name>
    <conceptGroup>
      <tty>SBD</tty>
      <conceptProperties>
        <rxcui>596932</rxcui>
        <name>duloxetine 30 MG Enteric Coated Capsule [Cymbalta]</name>
        <synonym>Cymbalta 30 MG Enteric Coated Capsule</synonym>
        <tty>SBD</tty>
        <language>ENG</language>
        <suppress>N</suppress>
        <umlscui>C1614249</umlscui>
      </conceptProperties>
      <conceptProperties>
        <rxcui>615186</rxcui>
        <name>duloxetine 60 MG Enteric Coated Capsule [Cymbalta]</name>
        <synonym>Cymbalta 60 MG Enteric Coated Capsule</synonym>
        <tty>SBD</tty>
        <language>ENG</language>
        <suppress>N</suppress>
        <umlscui>C1624617</umlscui>
      </conceptProperties>
      <conceptProperties>
        <rxcui>596928</rxcui>
        <name>duloxetine 20 MG Enteric Coated Capsule [Cymbalta]</name>
        <synonym>Cymbalta 20 MG Enteric Coated Capsule</synonym>
        <tty>SBD</tty>
        <language>ENG</language>
        <suppress>N</suppress>
        <umlscui>C1656295</umlscui>
      </conceptProperties>
    </conceptGroup>
    <conceptGroup>
      <tty>BPCK</tty>
    </conceptGroup>
  </drugGroup>
</rxnormdata>
```

Example 2: Amazon Relational Database Service

"Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud."

Action: CreateDBInstance

Sample Request:

```
https://rds.amazonaws.com/
    ?Action=CreateDBInstance
    &DBInstanceIdentifier=SimCoProd01
    &Engine=mysql
    &MasterUserPassword=Password01
    &AllocatedStorage=10
    &MasterUsername=master
    &Version=2011-04-01
    &DBInstanceClass=db.m1.large
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2011-05-23T05%3A54%3A53.578Z
    &AWSAccessKeyId=<AWS Access Key ID>
    &Signature=<Signature>
```

Sample Response:

```xml
<CreateDBInstanceResponse xmlns="http://rds.amazonaws.com/doc/2011-04-01/">
  <CreateDBInstanceResult>
    <DBInstance>
      <ReadReplicaDBInstanceIdentifiers/>
      <Engine>mysql</Engine>
      <PendingModifiedValues>
        <MasterUserPassword>****</MasterUserPassword>
      </PendingModifiedValues>
      <BackupRetentionPeriod>1</BackupRetentionPeriod>
      <MultiAZ>false</MultiAZ>
      <LicenseModel>general-public-license</LicenseModel>
      <DBInstanceStatus>creating</DBInstanceStatus>
      <EngineVersion>5.1.50</EngineVersion>
      <DBInstanceIdentifier>simcoprod01</DBInstanceIdentifier>
      <DBParameterGroups>
        <DBParameterGroup>
          <ParameterApplyStatus>in-sync</ParameterApplyStatus>
          <DBParameterGroupName>default.mysql5.1</DBParameterGroupName>
        </DBParameterGroup>
      </DBParameterGroups>
      <DBSecurityGroups>
        <DBSecurityGroup>
          <Status>active</Status>
          <DBSecurityGroupName>default</DBSecurityGroupName>
        </DBSecurityGroup>
      </DBSecurityGroups>
      <PreferredBackupWindow>00:00-00:30</PreferredBackupWindow>
      <AutoMinorVersionUpgrade>true</AutoMinorVersionUpgrade>
      <PreferredMaintenanceWindow>sat:07:30-sat:08:00</PreferredMaintenanceWindow>
        <AllocatedStorage>10</AllocatedStorage>
        <DBInstanceClass>db.m1.large</DBInstanceClass>
        <MasterUsername>master</MasterUsername>
    </DBInstance>
    (more)
```
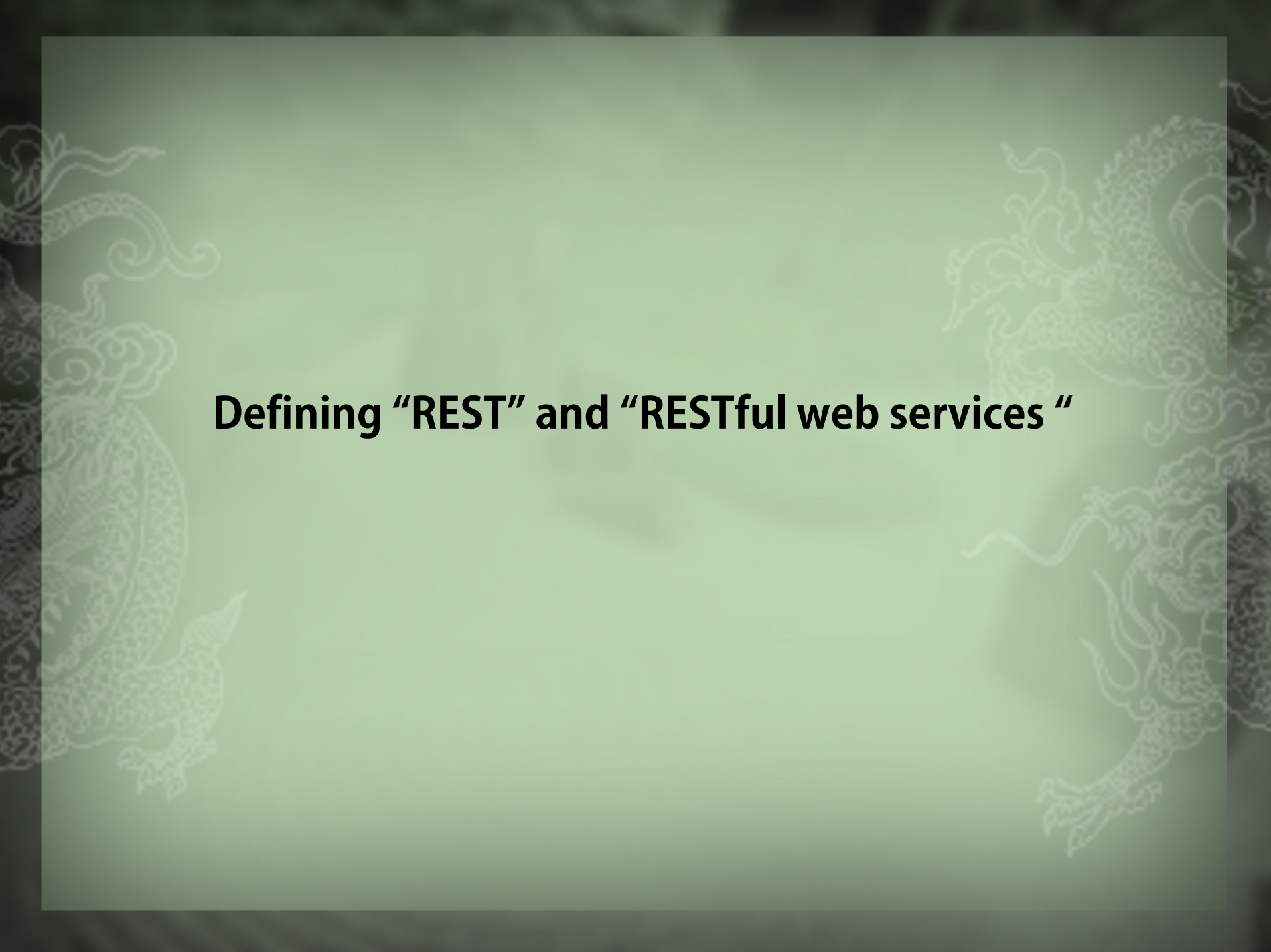
Who else is using it?

- Amazon.com offers a "REST" version of its main 'E-Commerce' developer interface;
- eBay offers a "REST" developer interface;
- Facebook offers a "REST" developer interface (NOTE: Facebook is in the process of deprecating its REST API).
- Yahoo! offers a number of "REST" developer interfaces.
- YouTube offers a number of "REST" developer interfaces.
- Truveo video search engine offers "REST" developer interfaces.
- NewsCloud offers a number of "REST" interfaces.
- movideo offers a number of "REST" interfaces.
- Microsoft SharePoint version 2010
- **UAB Center for Clinical and Translational Science (CCTS)**

Used extensively by the CCTS

- MOUSER (UAB Mouse Registry)
- Aggregate Cohort Estimator (ACE)
- BMI Project Manager

Defining "REST" and "RESTful web services "

REST = "representational state transfer"

"The name 'Representational State Transfer' is intended to evoke an image of how a well-designed Web application behaves: a network of Web pages forms a virtual state machine, allowing a user to progress through the application by selecting a link or submitting a short data-entry form, with each action resulting in a transition to the next state of the application by transferring a representation of that state to the user."

Roy T. Fielding

A brief history...

Remote procedure calls

- Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA)

- Microsoft's Distributed Component Object Model (DCOM)

- Sun Microsystems's Java/Remote Method Invocation (RMI)

Service-oriented architecture

- "message-oriented" services (the basic unit of communication is a message rather than an operation)

- SOA aims to allow users to string together fairly large chunks of functionality to form ad hoc applications that are built almost entirely from existing software services.

(http://en.wikipedia.org/wiki/Service-oriented_architecture)

Representational state transfer (REST)

Here, the focus is on interacting with stateful resources, rather than messages or operations.

(http://en.wikipedia.org/wiki/Web_services)

REST is a style, not a standard

"REST defines a set of architectural principles by which you can design web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages."

(http://www.ibm.com/developerworks/webservices/library/ws-restful/)

REST is stateless

"A RESTful HTTP interaction has to be stateless. This means each request contains all information which is required to process the request."

(http://www.infoq.com/articles/designing-restful-http-apps-roth)

Two styles/philosophies/interpretations of RESTful web services

1) "Low" REST

2) "High" REST

"Low" REST

- "Plain Old XML over HTTP" (POX)

- HTTP GET is used for all requests

- Critics call it "Accidentally RESTful"

"High REST"

- Uses the 4 main HTTP verbs (GET, POST, PUT, and DELETE) to manipulate resource representations.
- These align nicely with the CRUD acronym from the database world

HTTP method	CRUD action	Description
POST	CREATE	Create a new resource
GET	RETRIEVE	Retrieve a representation of a resource
PUT	UPDATE	Update a resource
DELETE	DELETE	Delete a resource

REST has mostly displaced SOAP-and-WSDL based interface design because:

- it's a considerably simpler style to use

- it's less dependent on proprietary middleware

- Ajax and REST work well together

Summary

"Exposing a system's resources through a RESTful API is a flexible way to provide different kinds of applications with data formatted in a standard way. It helps to meet integration requirements that are critical to building systems where data can be easily combined (mashups) and to extend or build on a set of base, RESTful services into something much bigger."

(http://www.ibm.com/developerworks/webservices/library/ws-restful/)

Future implementation plans for UAB IT (for RESTful web services)

Resource management
Web APIs for virtual machine (VM) register/start/stop via OpenNebula

Project management
Web APIs for group management, mailing lists, Wordpress, Trac, MediaWiki (and new code to tie them together)

Job Management
Web APIs from Galaxy to control workflows

Thanks!