

Introducing OData

Understanding the Open Data Protocol



Overview

- Web Services designs
- The value of constraints
- Understanding OData

Service designs

SOAP

- Simple Object Access Protocol

POX

- Plain Old Xml

REST

- REpresentational State Transfer

SOAP web services

- XML messages with embedded protocol
- WS - * (suite of protocols) – including metadata (WSDL, MEX)
- Operation (verb) focused
- HTTP Post only

Protocol



```
<s:Envelope xmlns:s="...">
  <s:Headers>
    . . .
  </s:Headers>
  <s:Body>
    <DoSomethingResponse xmlns="http://tempuri.org/">
      <DoSomethingResult xmlns:b="...">
        <b:Something>something</b:Something>
      </DoSomethingResult>
    </DoSomething>
  </s:Body>
</s:Envelope>
```

Verb



Data



POX services

- HTTP + XML
- Low barrier to implementation
- Few if any constraints (Schema sometimes used)

Protocol

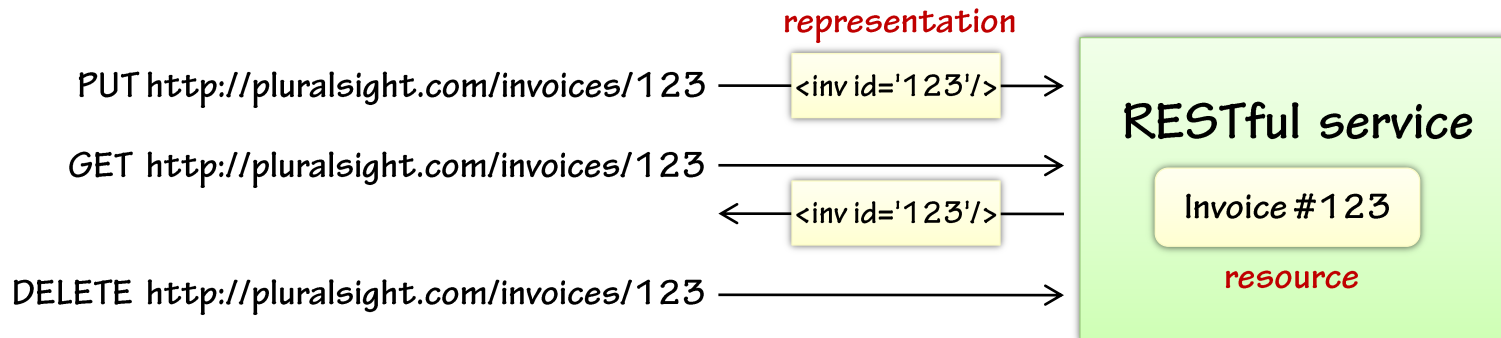
```
GET /movies/StarWars HTTP/1.1
Accept: application/json, application/atom+xml
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0;)
Accept-Encoding: gzip, deflate
```

Data

```
<Movie>
  <Title>Star Wars</Title>
  <ReleaseYear>1977</ReleaseYear>
</Movie>
```

REST services

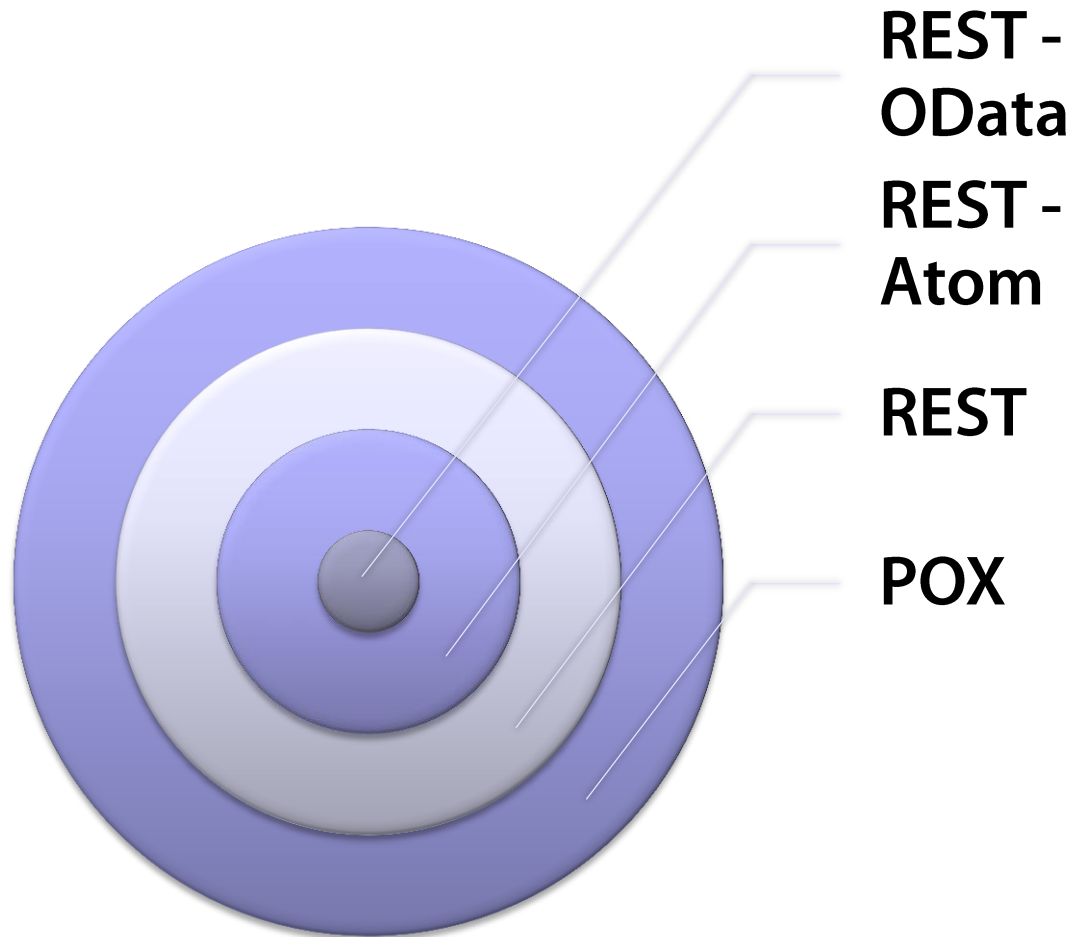
- **REST services embrace HTTP**
 - Services are modeled as "resources" with unique identifiers (URI's)
 - HTTP defines a **uniform** service contract: GET, POST, PUT, DELETE (**CRUD**)
- **REST constrains POX**
 - Constraints bring consistency
 - Consistency enables tooling / easier programming



Web data formats

- **XML**
 - Common data format used to represent data
 - Serialization of objects to an open wire format
- **Javascript Object Notation (JSON)**
 - AJAX optimized
 - Compact format
- **Atom**
 - Common syndication format (XML dialect)
 - Supports embedded data “microformats”
- **Atom Publishing Protocol (AtomPub)**
 - Protocol for retrieving and editing web resources

Degrees of constraint



What is OData

- **Open**

- Based on web standards – HTTP, AtomPub, JSON
- Fully embraces the web programming model

- **Data**

- Designed for querying and updating data
- Flexible enough to handle different data sources (db, cms, files, etc.)

- **Protocol**

- Conventions for representing data entities in AtomPub
- Query string conventions for addressing and filtering data
- Constrained HTTP operations for submitting requests

What makes OData different?

- **Built on standards**
 - HTTP and AtomPub form the base, making it easy to consume
- **Adds constraints within the existing standards**
 - Metadata support
 - Common data format for payload
 - Well defined query parameters (filtering, paging, functions)

Summary

- Web standards are being embraced to build services
- OData extends AtomPub to provide RESTful data services
- Constraints simplify tooling and client development
- OData is about exchanging data over the web

References