Data Culture
with
Culture Data

Dr. Barry Norton,
Development Manager, ResearchSpace*

* Funded by the Andrew W. Mellon Foundation
* Hosted by the Curatorial Directorate, British Museum
Culture Data

collection.britishmuseum.org
>2M objects described
RDF & public SPARQL

researchspace.org

musicbrainz.org
>1M albums described
RDF & public SPARQL

linkedbrainz.org
Not the Starting Point

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs so that they can discover more things.

Tim Berners-Lee
Date: 2006-07-27
1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs so that they can discover more things.

Why?

Tim Berners-Lee
Date: 2006-07-27
Why not CSVs?

- There are many of them
- They’re easy to import into a spreadsheet (if they ever left)
- They’re line-by-line processable
  - resurgence of GNU tools
  - powerful scripting languages
  - chunkable for Hadoop
- W3C CSV on the Web
<table>
<thead>
<tr>
<th>Artist</th>
<th>Album</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Beatles</td>
<td>Rubber Soul</td>
</tr>
<tr>
<td>The Beatles</td>
<td>Revolver</td>
</tr>
<tr>
<td>The Beatles</td>
<td>Sgt. Pepper’s Lonely Hearts Club Band</td>
</tr>
<tr>
<td>The Rolling Stones</td>
<td>Their Satanic Majesties Request</td>
</tr>
<tr>
<td>The Rolling Stones</td>
<td>Beggars Banquet</td>
</tr>
<tr>
<td>The Rolling Stones</td>
<td>Let It Bleed</td>
</tr>
</tbody>
</table>
CSV Simple Denormalisation

<table>
<thead>
<tr>
<th>Artist</th>
<th>Album</th>
<th>Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Beatles</td>
<td>Rubber Soul</td>
<td>Drive My Car</td>
</tr>
<tr>
<td>The Beatles</td>
<td>Rubber Soul</td>
<td>Norwegian Wood</td>
</tr>
<tr>
<td>The Beatles</td>
<td>Rubber Soul</td>
<td>...</td>
</tr>
<tr>
<td>The Beatles</td>
<td>Revolver</td>
<td>...</td>
</tr>
<tr>
<td>The Beatles</td>
<td>Sgt. Pepper’s …</td>
<td>...</td>
</tr>
<tr>
<td>The Rolling Stones</td>
<td>Their Satanic Majesties …</td>
<td>...</td>
</tr>
<tr>
<td>The Rolling Stones</td>
<td>Beggars Banquet</td>
<td>...</td>
</tr>
</tbody>
</table>

- Redundant but feasible because this is a hierarchy
### CSV Denormalisation

<table>
<thead>
<tr>
<th>Artist</th>
<th>Member</th>
<th>Album</th>
<th>Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Beatles</td>
<td>John</td>
<td>Rubber Soul</td>
<td>?</td>
</tr>
<tr>
<td>The Beatles</td>
<td>Paul</td>
<td>Rubber Soul</td>
<td>?</td>
</tr>
<tr>
<td>The Beatles</td>
<td>George</td>
<td>Rubber Soul</td>
<td>?</td>
</tr>
<tr>
<td>The Beatles</td>
<td>Ringo</td>
<td>Rubber Soul</td>
<td>?</td>
</tr>
<tr>
<td>The Beatles</td>
<td></td>
<td>Sgt. Pepper’s …</td>
<td>?</td>
</tr>
<tr>
<td>The Rolling Stones</td>
<td></td>
<td>Their Satanic Majesties …</td>
<td>…</td>
</tr>
<tr>
<td>The Rolling Stones</td>
<td></td>
<td>Beggars Banquet</td>
<td>…</td>
</tr>
</tbody>
</table>

- Orthogonal (arguably) hierarchies – either deeply redundant or very sparse…
• But this is just a relational database without the rigour…
Why not RDBMS?

• Not as unlikely as you might think

MusicBrainz Database

Products > MusicBrainz Database

Schema

Schema details are in the process of being written.

Download

A complete data snapshot of the entire database is generated twice a week.
RDBMS on Web?

- Tricky to install and update, but feasible
  - MusicBrainz, for instance, distributes the postgresql index, but also a VM and postgresql replication
- Actually the basis of most CSV
  - project and denormalise
  - equally feasible with SPARQL (see later)
- Difficult to extend and integrate new data...
MusicBrainz

‘Advanced Relationships’

• `link_types` (rows) are extensible binary relationships

• Each pair of entities (e.g. `artist_artist`) are paired and then links typed
Why not XML/SOAP?

• XML is the ‘data equivalent’ to HTML Web documents
• XML is communicable via the Web protocol, HTTP, in SOAP
• So didn’t we solve all this in the 90s?
Ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha ha.

No.
HTML versus XML

**HTML:**

```html
<html>
  <body>
    <h1>title</h1>
    <p>Some text referring to:</p>
    <img src="...">
    <ul>
      <li>one</li>
      <li>two</li>
      <li>three</li>
    </ul>
  </body>
</html>
```

**XML (example):**

```xml
<artists>
  <artist name="The Beatles">
    <member>John</member>
    <member>Paul</member>
    <album name="Rubber Soul">
      <track>Drive My Car</track>
      <track>Norwegian Wood</track>
    </album>
    <album name="Revolver">
      ...<br>
    </album>
  </artist>
  <artist name="The Rolling Stones">
    ...<br>
  </artist>
</artists>
```
HTTP versus Services

HTTP:

GET /artist/b10bbbfccf9e-42e0-be17-e2c3e1d2600d HTTP/1.1
Host: musicbrainz.org

SOAP:

POST /ws
Host: www.example.com
Why not JSON/REST?

- JSON removes a lot of the ambiguous structure of XML
- JSON is a lot more flexible in terms of schema (expectations)
- REST argues that resources have URIs and all representations be resolved there
{
  "id": "fcbcdc39-8851-4efc-a02a-ab0e13be224f",
  "title": "LAST ANGEL",
  "disambiguation": "video",
  "artist-credit": [
    {
      "name": "倖田來未",
      "joinphrase": " feat. ",
      "artist": {
        "id": "455641ca-fff4-49f6-8fb4-49f961d8f1ac",
        "name": "倖田來未",
        "sort-name": "Koda, Kumi",
        "disambiguation": null
      }
    },
    {
      "name": "東方神起",
      "artist": {
        "id": "05c5af37-6dc2-4f71-a0cc-d63347d90c3",
        "name": "東方神起",
        "sort-name": "TVXQ",
        "disambiguation": null
      },
      "joinphrase": ""
    }
  ],
  "isrcs": [ "JPB600760301" ],
  "length": 229106,
  "releases": [
    {
      "id": "abcd76db-7d5f-3eb7-b386-051c97bfe2e4",
      "title": "Kingdom",
      ...
    }
  ]
}
Usage of Linked Data
Introduction and Application Scenarios

Presented by: Barry Norton
Agenda

1. Motivation Scenario
2. Linked Data Foundations
3. Introduction to Linked Data
4. Linked Data use case scenarios
MOTIVATION SCENARIO
• Provision of a music-based portal.

• Bring together a number of disparate components of data-oriented content:

1. **Musical content** (streaming data & downloads)
2. **Music and artist metadata**
3. **Review content**
4. **Visual content** (pictures of artists & albums)
Expected Results

• The developer will contribute back the aggregated and interlinked content to the Linked Open Data Cloud.

• Linking of artists will be improved.

• Metadata, visual content and reviews will be improved.

• Links to emerging Web technologies that inherit from semantics: Google RichSnippets, Facebook OpenGraph and schema.org annotation.
LINKED DATA FOUNDATIONS
• Extension of the technology of computer networks.

• The technology supporting the Internet includes the Internet Protocol (IP).

• Each computer on the Internet is assigned an IP number.

• Messages can be routed from one computer to another.
Internet

The growth of the Internet

Global Internet Users

Global Internet Traffic (in petabytes per month)

Source: http://www.evolutionoftheweb.com
The Web

• There is a wealth of information on the Web.

• It is aimed mostly towards consumption by humans as end-users:
  • Recognize the meaning behind content and draw conclusions,
  • Infer new knowledge using context and
  • Understand background information.
The Web

- Billions of diverse documents online, but it is not easily possible to automatically:
  - Retrieve relevant documents.
  - Extract information.
  - Combine information in a meaningful way.

- Idea:
  - Also publish machine processable data on the web.
  - Formulate questions in terms understandable for a machine.
  - Do this in a standardized way so machines can interoperate.

- The Web becomes a Web of Data
  - This provides a common framework to share knowledge on the Web across application boundaries.
The Web: Evolution

Web of Documents → Web of Data

Hyperlinks

"Documents"

Typed Links

"Things"
The Web: Evolution

Source: http://www.radarnetworks.com
Web Technology Basics

HTML – HyperText Markup Language
• Language for displaying web pages and other information in a web browser.
• HTML elements consist of tags (enclosed in angle brackets), attributes and content.

HTTP – Hypertext Transfer Protocol
• Foundation of data communication for the WWW.
• Client-server protocol.
• Every interaction is based on: request and response.
Uniform Resource Identifier (URI)

- Compact sequence of characters that identifies an abstract or physical resource.

- **Examples:**
  - ldap://[2001:db8::7]/c=GB?objectClass?one
  - mailto:John.Doe@example.com
  - news:comp.infosystems.www.servers.unix
  - tel:+1-816-555-1212
  - telnet://192.0.2.16:80/
  - http://dbpedia.org/resource/Karlsruhe
Vocabularies

• Collections of defined relationships and classes of resources.
  • Classes group together similar resources.

• Terms from well-known vocabularies should be reused wherever possible.

• New terms should be defined only if you cannot find required terms in existing vocabularies.
## Describing Data

### Vocabularies

A set of well-known vocabularies has evolved in the Semantic Web community. **Some** of them are:

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>Description</th>
<th>Classes and Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friend-of-a-Friend <em>(FOAF)</em></td>
<td>Vocabulary for describing people.</td>
<td>foaf:Person, foaf:Agent, foaf:name, foaf:knows, foaf:member</td>
</tr>
<tr>
<td>Dublin Core <em>(DC)</em></td>
<td>Defines general metadata attributes.</td>
<td>dc:FileFormat, dc:MediaType, dc:creator, dc:description</td>
</tr>
<tr>
<td>Semantically-Interlinked Online Communities <em>(SIOC)</em></td>
<td>Vocabulary for representing online communities.</td>
<td>sioc:Community, sioc:Forum, sioc:Post, sioc:follows, sioc:topic</td>
</tr>
<tr>
<td>Music Ontology <em>(MO)</em></td>
<td>Provides terms for describing artists, albums and tracks.</td>
<td>mo:MusicArtist, mo:MusicGroup, mo:Signal, mo:member, mo:record</td>
</tr>
<tr>
<td>Simple Knowledge Organization System <em>(SKOS)</em></td>
<td>Vocabulary for representing taxonomies and loosely structured knowledge.</td>
<td>skos:Concept, skos:inScheme, skos:defintion, skos:example</td>
</tr>
</tbody>
</table>
Vocabularies

More extensive lists of well-known vocabularies are maintained by:

• **W3C SWEO Linking Open Data community project**
  http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies

• **Mondeca: Linked Open Vocabularies**
  http://labs.mondeca.com/dataset/lov

• **Library Linked Data Incubator Group: Vocabularies in the library domain**
  http://www.w3.org/2005/Incubator/lld/XGR-lld-vocabdataset-20111025
Semantics on the Web

Semantic Web Stack
Berners-Lee (2006)
Semantics on the Web

- **Application specific declarative-knowledge**
- **Query language**
- **Expressive vocabulary (ontology) language**
- **Basic data model**
- **Syntactic basis**

- **Unifying Logic**
- **Query: SPARQL**
- **ontology: OWL**
- **Rules: RIF**
- **RDF-S**
- **Data interchange: RDF**
- **XML**
- **URI**
- **Unicode**
- **Trust**
- **Proof**
- **Crypto**
- **Digital signatures, recommendations**
- **Proof generation, exchange, validation**
- **Simple vocabulary (schema) language**
- **Semantic Web Stack**

Berners-Lee (2006)
Semantics on the Web

RDF – Resource Description Framework

Semantic Web Stack
Berners-Lee (2006)
RDF – Resource Description Framework

• RDF is the basis layer of the Semantic Web stack ‘layer cake’.

• Basic building block: RDF triple.
  • **Subject** – a resource, which may be identified with a URI.
  • **Predicate** – a URI-identified reused specification of the relationship.
  • **Object** – a resource or literal to which the subject is related.
Semantics on the Web

RDF – Resource Description Framework (Example)

<http://musicbrainz.org/artist/b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d#_>
<http://www.w3.org/2002/07/owl#sameAs>
<http://dbpedia.org/resource/The_Beatles>.

<http://musicbrainz.org/artist/b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d#_>
<http://xmlns.com/foaf/0.1/name>
"The Beatles". Literals are given in quotes in N-Triples.

URIs are given in angle brackets in N-Triples.
In N-Triples every statement is terminated with a full stop.
RDF Graphs

• Every set of RDF assertions can then be drawn and manipulated as a (labelled directed) graph:
  • **Resources** – the subjects and objects are nodes of the graph.
  • **Predicates** – each predicate use becomes a label for an arc, connecting the subject to the object.
Semantics on the Web

RDF Graphs (Example)

```
<http://musicbrainz.org/artist/b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d#_>
<http://www.w3.org/2002/07/owl#sameAs>
<http://dbpedia.org/resource/The_Beatles>
```

"Subject"

"Predicate"

"Object"

"The Beatles"
RDF Blank Nodes

- RDF graphs can also contain unidentified resources, called *blank nodes*:

  ```
  [ ]
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  <http://www.w3.org/2003/01/geo/wgs84_pos#Point>
  
  <http://www.w3.org/2003/01/geo/wgs84_pos#lat> 49.005
  <http://www.w3.org/2003/01/geo/wgs84_pos#long> 8.386
  ```

- Blank nodes can group related information, but their use in Linked Data is discouraged.
Semantics on the Web

RDF Turtle

• Turtle is a syntax for RDF more readable.

• Since many URIs share same basis we use prefixes:
  @prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
  @prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>.
  @prefix owl:<http://www.w3.org/2002/07/owl#>.
  @prefix mo:<http://purl.org/ontology/mo/>.
  @prefix dbpedia:<http://dbpedia.org/resource/>.

And (sometimes) a unique base:
@base <http://musicbrainz.org/>.
RDF Turtle

- Also has a simple *shorthand* for class membership:

  ```
  @base <http://musicbrainz.org/>.
  @prefix mo:<http://purl.org/ontology/mo/>.
  <artist/b10bbbfccf9e-42e0-be17-e2c3e1d2600d#_> a mo:MusicGroup.
  ```

*Is equivalent to:*

```
<http://musicbrainz.org/artist/b10bbbfccf9e-42e0-be17-e2c3e1d2600d#_>
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
```
RDF Turtle

• When multiple statements apply to **same subject** they can be abbreviated as follows:

```turtle
<artist/b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d#_>
    rdfs:label "The Beatles" ;
    owl:sameAs dbpedia:The_Beatles ,
<http://www.bbc.co.uk/music/artists/
b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d#artist> .
```
RDF Turtle

• Turtle also provides a simple syntax for datatypes and language tags for literals, respectively:

```turtle
<recording/5098d0a8-d3c3-424e-9367-1f2610724410#_> a mo:Signal;
   rdfs:label "All You Need Is Love";
   mo:duration "PT3M48S"^^xsd:duration .

dbpedia:The_Beatles dbpedia-owl:abstract
   "The Beatles were an English rock band formed (...) "@en,
   "The Beatles waren eine britische Rockband in den (...) "@de .
```
RDF/XML

• This is most useful for inter-machine communication.
• The primary (recurring) element in encoding assertions (thereby triples) is rdf:Description, e.g.:

```xml
<rdf:Description
    rdf:about="http://musicbrainz.org/artist/b10bbbcf-cf9e-42e0-be17-e2c3e1d2600d#_"
>
    <foaf:name>The Beatles</foaf:name>
    <owl:sameAs rdf:resource="http://dbpedia.org/resource/The_Beatles">
</rdf:Description>

<rdf:Description
    rdf:about="http://musicbrainz.org/artist/4d5447d7-c61c-4120-ba1b-d7f471d385b9#_"
>
    <foaf:name>John Lennon</foaf:name>
</rdf:Description>
```
Semantics on the Web

RDF-S – RDF Schema

Semantic Web Stack
Berners-Lee (2006)
RDF-S – RDF Schema

Language for two tasks w.r.t. the RDF data model:

- **Expectation** – nominate:
  - the ‘types’, i.e., *classes*, of things we might make assertions about, and
  - the *properties* we might apply, as predicates in these assertions, to capture their relationships.

- **Inference** – given a set of assertions, using these classes and properties, specify what should be inferred about assertions that are *implicitly* made.
RDF-S – RDF Schema

- **rdf:Property** - Class of RDF properties. Example:
  
  mo:member - Indicates a member of a musical group.

- **rdfs:domain** - States that any resource that has a given property is an instance of one or more classes.
  
  mo:member rdfs:domain mo:MusicGroup .

- **rdfs:range** - States that the values of a property are instances of one or more classes.
  
  mo:member rdfs:range foaf:Agent .
We expect to use this vocabulary to make assertions about music groups.

Having made such an assertion...

Inferences can be drawn that we did not explicitly make
<table>
<thead>
<tr>
<th>Resources and predicates with (limited) inferences:</th>
<th>Some predicates with NO inferences:</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:Resource</td>
<td>rdfs:comment</td>
</tr>
<tr>
<td>rdfs:Literal, rdfs:Datatype</td>
<td>rdfs:label</td>
</tr>
<tr>
<td>rdfs:Class, rdfs:subClassOf</td>
<td>rdfs:seeAlso</td>
</tr>
<tr>
<td>rdfs:subPropertyOf</td>
<td>rdfs:isDefinedBy</td>
</tr>
<tr>
<td>rdfs:range, rdfs:domain</td>
<td></td>
</tr>
<tr>
<td>rdf:Property (an instance of rdfs:Class)</td>
<td></td>
</tr>
</tbody>
</table>
Semantics on the Web

OWL – Web Ontology Language

Semantic Web Stack
Berners-Lee (2006)
OWL – Web Ontology Language

• RDFS provides a simplified ontological language for defining vocabularies about specific domains.

• Sometimes it is necessary to have access to a wider range of ontological constructs.

• **Web Ontology Language (OWL)** provides more ontological constructs and avoids some of the potential confusion in RDF-S.
Semantics on the Web

OWL 2.0 – Web Ontology Language 2.0
Extends the DL further, but has three more computable fragments (profiles).

**OWL 2 Full**
- Used informally to refer to RDF graphs considered as OWL 2 ontologies and interpreted using the RDF-Based Semantics.

**OWL 2 DL**
- Used informally to refer to OWL 2 DL ontologies interpreted using the Direct Semantics.

**OWL 2 EL**
- Limited to basic classification, but with polynomial-time reasoning.

**OWL 2 QL**
- Designed to be translatable to relational database querying.

**OWL 2 RL**
- Designed to be efficiently implementable in rule-based systems.
OWL – Web Ontology Language

OWL is made up of **terms** which provide for:

- **Class construction**: forming new classes from membership of existing ones (e.g., unionOf, intersectionOf, etc.).

- **Property construction**: distinction between OWL ObjectProperties (resources as values) and OWL DatatypeProperties (literals as values).

- **Class axioms**: sub-class, equivalence and disjointness relationships.

- **Property axioms**: sub-property relationship, equivalence and disjointness, and relationships between properties.

- **Individual axioms**: statements about individuals (sameIndividual, differentIndividuals).
Semantics on the Web

SPARQL – Protocol and RDF Query Language

Semantic Web Stack
Berners-Lee (2006)
SPARQL – * Protocol and RDF Query Language

• Query language designed to use a syntax similar to SQL for retrieving data from relational databases.

• Different query forms:
  
  • **SELECT** returns variables and their bindings directly.
  
  • **CONSTRUCT** returns a single RDF graph specified by a graph template.
  
  • **ASK** test whether or not a query pattern has a solution. Returns yes/no.
  
  • **DESCRIBE** returns a single RDF graph containing RDF data about resources.
Semantics on the Web

SPARQL – * Protocol and RDF Query Language

• The syntax of a SELECT query is as follows:
  – SELECT nominates which components of the matches made against the data should be returned.
  – FROM (optional) indicates the sources for the data against which to find matches.
  – WHERE defines patterns to match against the data.
  – ORDER BY defines a means to order the selected matches.
SPARQL – * Protocol and RDF Query Language

Retrieve the names of the albums and tracks recorded by The Beatles.

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX music-ont: <http://purl.org/ontology/mo/>

SELECT ?album_name ?track_title
WHERE {
  <http://musicbrainz.org/artist/b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d#> foaf:made ?album .
  ?album dc:title ?album_name ;
}
**Semantics on the Web**

**SPARQL – * Protocol and RDF Query Language**

<table>
<thead>
<tr>
<th>SQL</th>
<th>SPARQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Based on relations (tables).</td>
<td>Based on labelled directed graphs.</td>
</tr>
<tr>
<td>The relations (tables) to be matched over should be indicated.</td>
<td>Assumes a default graph. (The FROM clause populates this with specific identified subgraphs).</td>
</tr>
<tr>
<td>(Retrieval) queries produce a relation from a relation.</td>
<td>SPARQL SELECT queries produce a relation from a graph. CONSTRUCT queries (considered later) produce a graph from a graph.</td>
</tr>
</tbody>
</table>
Semantics on the Web

SPARQL – * Protocol and RDF Query Language

• SPARQL 1.1 provides graph update operations:
  
  • **INSERT DATA**: adds explicit triples, given inline.
  
  • **DELETE DATA**: removes explicit triples, given inline.
  
  • **DELETE/INSERT WHERE**: updates based on triples calculated from WHERE clause (as in SELECT and CONSTRUCT).
  
  • **LOAD**: reads the content of a document into a graph.
  
  • **COPY/MOVE/APPEND**: manipulates at named graph level.
  
  • **CLEAR/DROP**: removes all triples in one or more graph.
SPARQL – * Protocol and RDF Query Language

*Insert* the following albums recorded by *The Beatles* into the graph *http://myFavGroups/The_Beatles*

```sparql
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

INSERT DATA { GRAPH <http://myFavGroups/The_Beatles> {
<http://musicbrainz.org/artist/b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d#> foaf:made <http://musicbrainz.org/release/3a685770-7326-34fc-9f18-e5f5626f3dc5#> ,
<http://musicbrainz.org/release/cb6f8798-d51e-4fa5-a4d1-2c0602bfe1b6#> .

<http://musicbrainz.org/release/3a685770-7326-34fc-9f18-e5f5626f3dc5#> dc:title "Please Please Me".

<http://musicbrainz.org/release/cb6f8798-d51e-4fa5-a4d1-2c0602bfe1b6#> dc:title "Something New". } }
Delete all the information about the album Casualities of The Beatles.

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

DELETE { ?album ?predicate ?object . }
WHERE {
  <http://musicbrainz.org/artist/b10bbbcf-cf9e-42e0-be17-e2c3e1d2600d> foaf:made ?album .
  ?album dc:title "Casualities";
  ?predicate ?object .}
INTRODUCTION TO: LINKED DATA
Linked Data

• Set of best practices for **publishing** data on the Web.

• Data from different knowledge domains, self-described, linked and accessible.

• Follows 4 simple principles...
Linked Data Principles

1. Use URIs as names for things.
2. Use HTTP URIs so that users can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
4. Include links to other URIs, so that users can discover more things.
Linked Data Principles

1. Use URIs as **names** for things.

- A foundational issue in Linked Data was the distinction of URIs for **object documents** that might describe them.
Linked Data Principles

2. Use HTTP URIs so user can look up those names.

- HTTP allows a second way to distinguish real-world objects from documents.
- Best practice says HTTP 303 and Location header should be used.
3. When someone looks up a URI, **provide useful information**, using the standards (RDF*, SPARQL, Turtle\(^1\)).

- While RDF/XML should be the default for look-up.
  - RDFa annotations in HTML are now also standard.

- SPARQL endpoint for queries are encouraged, or a dump of the whole dataset.

---

\(^1\) To become a standard.
3. When someone looks up a URI, **provide useful information**, using the standards (RDF*, SPARQL, Turtle¹).

What to return for a URI?

- **Immediate description**: triples where the URI is the subject.
- **Backlinks**: triples where the URI is the object.
- **Related descriptions**: information of interest in typical usage scenarios.
- **Metadata**: information as author and licensing information.
- **Syntax**: RDF descriptions as RDF/XML and human-readable formats.

Source: *How to Publish Linked Data on The Web* - Chris Bizer, Richard Cyganiak, Tom Heath.
Linked Data Principles

4. Include links to other URIs, so that users can **discover** more things.

There are several ways to reuse URIs:

- direct **reuse**
- (OWL) **sameAs**
- (RDFS) **seeAlso**

- direct **reuse** of class/property
- (RDFS) **sub**-class/-property
- (OWL) **equivalent** class/property
- SKOS **broad match**

Instance Level

Schema Level
Linked Data 5 Star

🌟 Data is available on the Web.
🌟🌟 Data is available as machine-readable structured data.
🌟🌟🌟 Non-propietary formats are used.
🌟🌟🌟🌟 Individual data identified with open standards.
🌟🌟🌟🌟🌟 Data is linked to other data provider.
Linked Data 5 Star

Example:

My Data

"John Lennon"
"Paul McCartney"
"George Harrison"
"Ringo Starr"
Linked Data 5 Star

Data is available on the Web

My Data

"John Lennon"

"Paul McCartney"

"George Harrison"

"Ringo Starr"

It can be retrieved using HTTP.
Linked Data 5 Star

Data is available as machine-readable structured data

My Data

"The Beatles"  http://upload.wikimedia.org/wikipedia/commons/thumb/d/df/The_Fabs.JPG/600px-The_Fabs.JPG
"John Lennon"
"Paul McCartney"
Please Please Me – 1963
A Hard Day’s Night – 1964
Help! – 1965
Revolver – 1966
... 
"George Harrison"
"Ringo Starr"

Machine-readable data:

Images
Scanned Information
Plain text or ...
(to continue on the next slide)
Non-propietary formats are used

My Data

```xml
<schema "http://www.example.com/2012/XMLMyMusic"
version="1.0" >
<band>
  <name>The Beatles</name>
  <member>John Lennon</member>
  <member>Paul McCartney</member>
  <member>George Harrison</member>
  <member>Ringo Starr</member>
  <picture>http://upload.wikimedia.org/wikipedia/commons/thumb/d/df/The_Fabs.JPG/600px-The_Fabs.JPG</picture>
  <album year=1963>Please Please Me</album>
  <album year=1964>A Hard Day’s Night</album>
  <album year=1965>Help!</album>
  <album year=1966>Revolver</album>
  ...
</band>
```
Individual data identified with open standards

My Data

```xml
<schema "http://www.example.com/2012/XMLMyMusic"
version="1.0" >
<band>
  <name>http://musicbrainz.org/artist/b10bbbf-cf9e-42e0-be17-e2c3e1d2600d</name>
  <member>http://musicbrainz.org/artist/4d5447d7-c61c-4120-ba1b-d7f471d385b9</member>
  ...
  <album year=1963>http://musicbrainz.org/release/5f3ba07b-4a24-4cd5-b8ad-95ba0f3bec1</album>
  ...
</band>
```

URI: Uniform Resource Identifier

- Data is uniquely **identified**
  - The Beatles
  - John Lennon
  - Revolver

- **Dissambiguation**
  In this context, "Revolver" is an album! Not a gun.
Data is linked to other data provider

```xml
<schema
"http://www.example.com/2012/XMLMyMusic"
version="1.0" >
  <band>
    <name>http://musicbrainz.org/artist/b10bbbfc-cf9e-42e0-be17-e2c3e1d2600d</name>
    <member>http://musicbrainz.org/artist/4d5447d7-c61c-4120-ba1b-d7f471d385b9</member>
    ...
    <album
      year=1963>http://musicbrainz.org/release/5f3ba07b-4a24-4cd5-b8ad-95ba0fcebec1</album>
    ...
    <seeAlso>http://dbpedia.org/resource/The_Beatles</seeAlso>
  </band>
</schema>
```

http://dbpedia.org/resource/The_Beatles
Linked Data Cloud

2007
Linked Data Cloud

2008

As of September 2008
Linked Data Cloud

2009

As of July 2009
Linked Data Cloud

2010

EUCLID
Educational Curriculum for the usage of Linked Data

As of September 2019

67
Linked Data Cloud

2011
State of the LOD Cloud

• Total Datasets: 295

• Total Triples: 31,634,213,770

Distribution of triples by domain

1 Version 0.3, 09/19/2011
http://www4.wiwiss.fu-berlin.de/lodcloud/state
State of the LOD Cloud

• Total (Out-)Links: 503,998,829

Distribution of links by domain

1 Version 0.3, 09/19/2011
http://www4.wiwiss.fu-berlin.de/lodcloud/state
Exploring the Web of Data

- Linked Data browsers
- Linked Data mashups
- Search engines
Linked Data Browsers

Marbles

http://marbles.sourceforge.net
Broken Flowers

Links
Homepage: http://www.brokenflowersmovie.com/
See Also: http://en.wikipedia.org/wiki/Broken_Flowers

Tags
billy-murray film jessica-langse jia-jarmusch juliet-deely movie sharon-stone

Reviews (1)

★★★★★★ by tom on 30 Jan 2007
Broken Flowers provides a fantastic vehicle for a classic deadpan Bill Murray performance. The film centers around his character Don, who one day receives a letter from an ex-girlfriend, telling him he has a teenage son. The letter is unsigned, so (with encouragement from his neighbour) he sets off round the country, visiting each of the exes who could be the mother of his son. Predictably they're all different in personality and life situation, giving plenty of raw material for awkward silences and dubious encounters. This is great viewing for any Bill Murray fans, or anyone who likes their humour intelligent and a little bit quirky. The soundtrack is also excellent, and deserves a separate review.

What do you think of Broken Flowers? Write Your Own Review...

http://revyu.com
Linked Data Mashup

DBPedia Mobile

http://wiki.dbpedia.org/DBPediaMobile

Pictures from revyu.com
Tim Berners-Lee

picture: [Image]

given name: Tim [1,11,12,14,15,17,18,19,20]
family name: Berners-Lee [1,11,12,14,15,17,18,19,20]

comment: Sir Timothy John "Tim" Berners-Lee, OM, KBE, FRSE, FREng, FRSA (born 8 June 1955, also known as "TimBL"), is a British engineer and computer scientist and MIT professor credited with inventing the World Wide Web, making the first proposal for it in March 1989. On 29 December 1990, with the help of Robert Cailliau and a young student at CERN, he implemented the first successful computer-to-computer transmission of an HTTP client and server via the Internet.

is creator of: Tabulator [0,10,11,12,13,14,15,17,18,19,20]
author name: vicente181096 [5]
author url: http://www.slideshare.net/vicente181096 [5]
admins: 1124313582, 5000004654, 220400, 512158401, 808870558, 160220752, 729508126 [5]
birth year: 1955-06-08 [9]

http://sig.ma
Linked Data Search Engines

NYTimes

Linked Open Data

Alumni In The News

San Francisco State University

Enter a school name below and see our coverage of that school's alumni.

San Francisco State University

George Miller
Attorney
Born: May 17, 1945

Congress Considers Concussion Protections - September 24, 2010
EDITORIAL; Fairness for Older Workers - September 14, 2010
EDITORIAL; Saving the Teachers - May 06, 2010
House Bill Would Assure Workers Paid Sick Days - November 04, 2009
EDITORIAL; Preventing Age Discrimination - October 13, 2009
OP-ED COLUMNIST; Someday, a Bill Will Pass - September 17, 2009
Obama Plan to End Role of Banks in Federal Student Loans Wins Support - July 11, 2009
House Unveils Health Bill, Minus Key Details - June 20, 2009
Democrats Nearing Consensus on Health - June 10, 2009
U.S. Charges 7 Accused of Ties To Bonannos - August 29, 2008

Please note that portions of this application rely on user generated data from external sources. It is hoped but not guaranteed that this data is accurate.

http://data.nytimes.com/schools/schools/schools.html
Some Application Scenarios

BBC
### Some Application Scenarios

**LinkedGeoData.org**

This federated Linked Geo Data browser is based on data obtained from the OpenStreetMap project (licensed under CC-BY-SA) and was developed by deKKN-research group.

#### Search results

<table>
<thead>
<tr>
<th>Rank</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Berlin</td>
</tr>
<tr>
<td>2</td>
<td>Berlin, Deutschland, Europe</td>
</tr>
<tr>
<td>3</td>
<td>Berlin, Coos, New Hampshire, United States of America</td>
</tr>
<tr>
<td>4</td>
<td>Berlin, Stadt, Mitte, Berlin, Deutschland, Europe</td>
</tr>
<tr>
<td>5</td>
<td>Berlin, Worcester County, Maryland, United States of America</td>
</tr>
<tr>
<td>6</td>
<td>Berlin, Hartford, Connecticut, United States of America</td>
</tr>
<tr>
<td>7</td>
<td>Berlin, LaMoure, North Dakota, United States of America</td>
</tr>
<tr>
<td>8</td>
<td>Berlin, Coos, New Hampshire, United States of America</td>
</tr>
<tr>
<td>9</td>
<td>Berlin, Camden, New Jersey, United States of America</td>
</tr>
<tr>
<td>10</td>
<td>Berlin</td>
</tr>
</tbody>
</table>

**View**

[node:937335593]

**Edit on OpenStreetMap**

- **Name**: 
- **Description**: 
- **Image**: 
- **Source_ref**: 
- **natural**: stone
- **historic**: monument
Some Application Scenarios

Linked Government Data: USA
Some Application Scenarios

Linked Government Data: UK

Opening up government

Instructions for data publishers
Calling all data publishers - new guide to publishing to data.gov.uk

Over 6,900 datasets to view

Inside Government Data

Who's who in Government and where does the money go? Follow these links to find the data that opens it all up.

- Government spend over £26,000, by department
- Who does what in Whitehall - and how much are they paid?
- Hospitality, gifts and expenses

Share this

Facts, figures, apps and more

Find data of interest
Looking for something specific, or just want to know more about how Government spends your money? You'll find datasets here to help you get answers.

Apps
Want your phone to make you when you get to your train station? Discover over 100 apps harnessing public data to make your life easier.

Tags
Can't decide where to start? You can browse the data by clicking on popular topics. Try one of the tags here to find what you're most interested in.

health (2,326) care (1,846)
transparency (1,594) communities (1,318)
Summary

In this chapter we studied:

• **The Web** and its evolution.

• Web technology basics: **HTTP, HTML, URI**.

• **Vocabularies** to describe data.

• The **Semantic Web stack**: RDF, RDF-S, OWL, SPARQL.

• **Linked Data** concept and principles.

• Evolution of the **LOD cloud**.

• Browsers, mashups and search engines to **explore the Web of Data**.

• Some **application** scenarios.
For exercises, quiz and further material visit our website:

http://www.euclid-project.eu

Other channels:

@euclid_project
euclidproject
euclidproject
Acknowledgements

• Alexander Mikroyannidis
• Alice Carpentier
• Andreas Harth
• Andreas Wagner
• Andriy Nikolov
• Barry Norton
• Daniel M. Herzig
• Elena Simperl
• Günter Ladwig
• Inga Shamkhalov
• Jacek Kopecky
• John Domingue

• Juan Sequeda
• Kalina Bontcheva
• Maria Maleshkova
• Maria-Esther Vidal
• Maribel Acosta
• Michael Meier
• Ning Li
• Paul Mulholland
• Peter Haase
• Richard Power
• Steffen Stadtmüller