Layer 2: Linked Data

Background reading: http://linkeddatabook.com

Layer 1

Tuesday

Creating and Managing URIs

Layer 2

Wednesday

Content Negotiation & Linked Data

Layer 1

Tuesday

Creating and Managing URIs

Layer 3

Thursday!

Building Services - Data

Layer 2

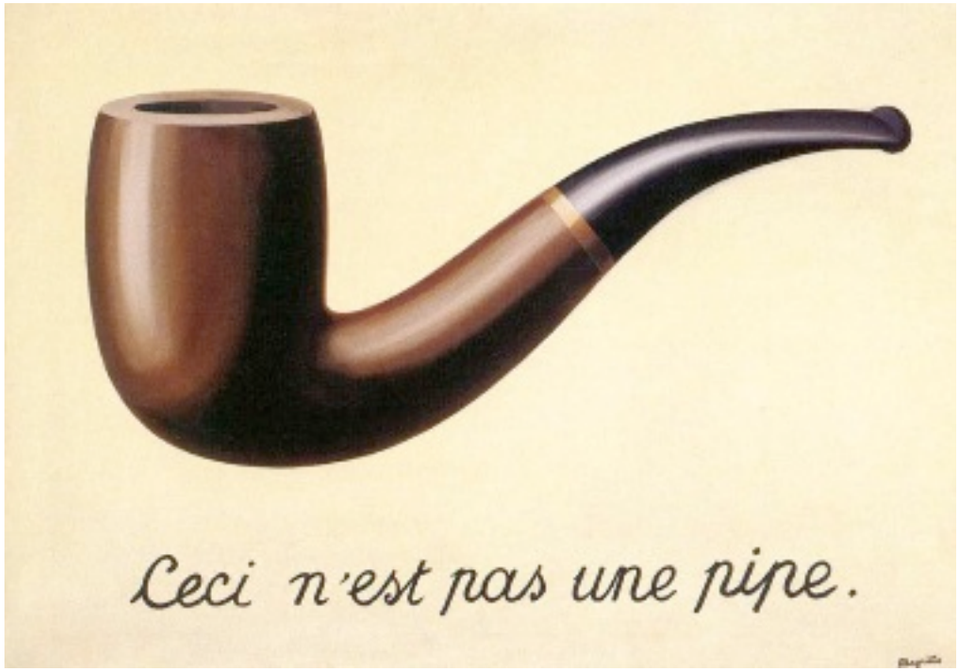
Wednesday

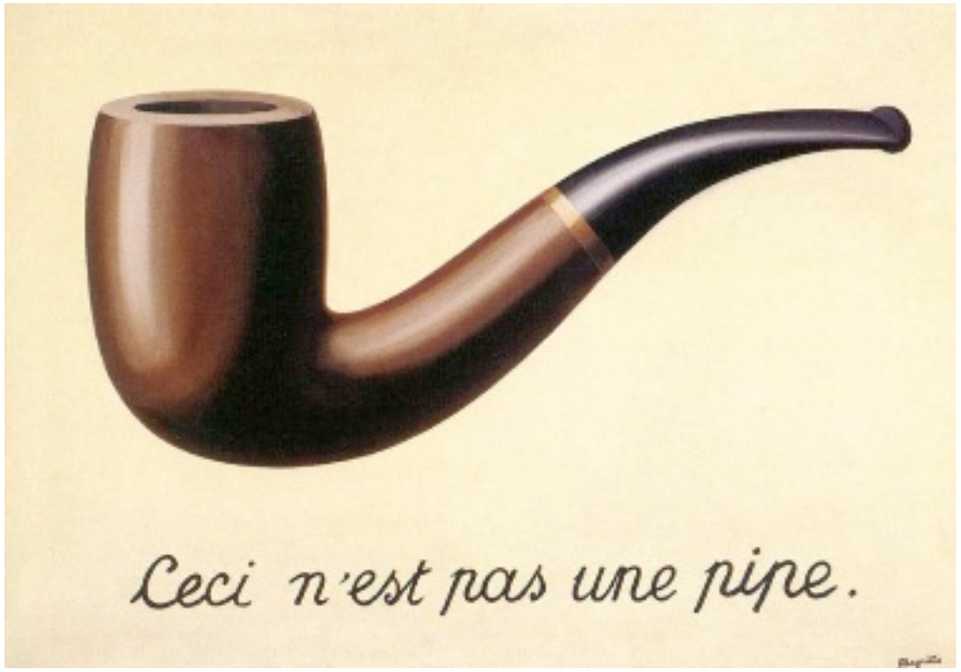
Content Negotiation & Linked Data

Layer 1

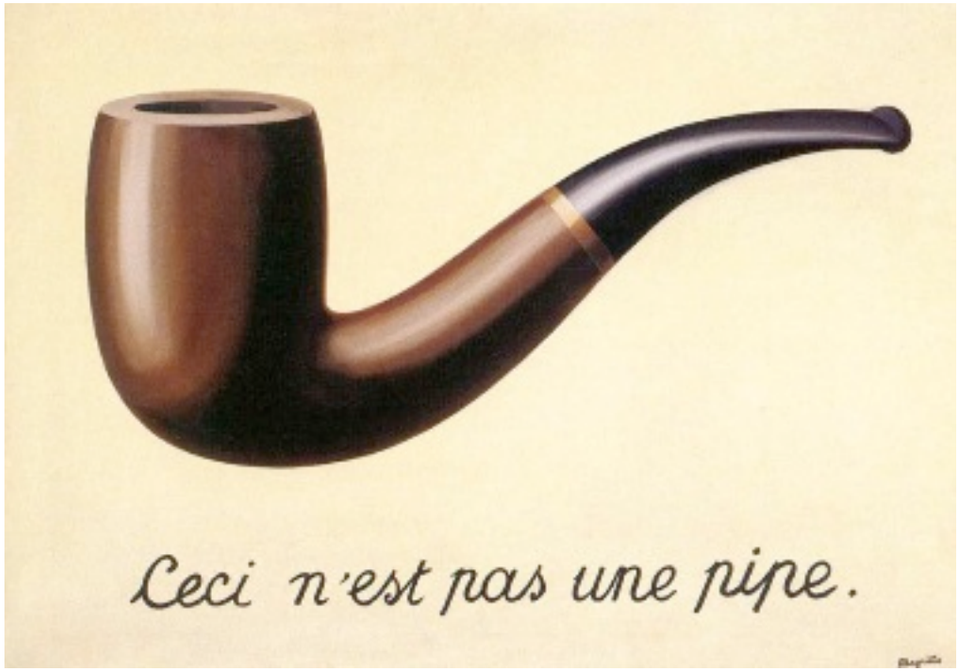
Tuesday

Creating and Managing URIs


Ceci n'est pas une pipe.

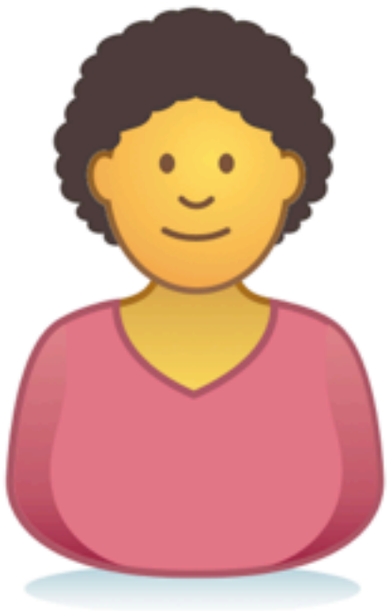


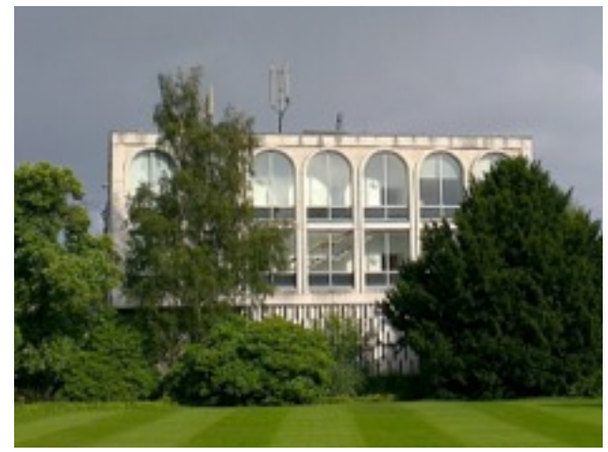
This is not a pipe

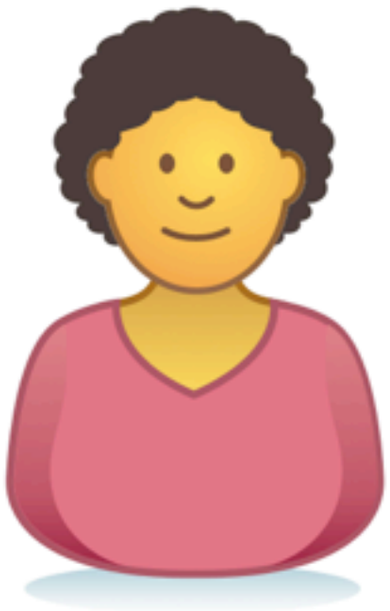


This is not a pipe

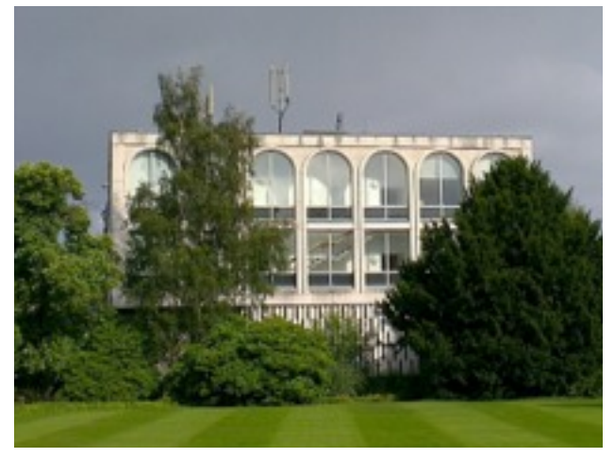
Neither is this a painting by Magritte

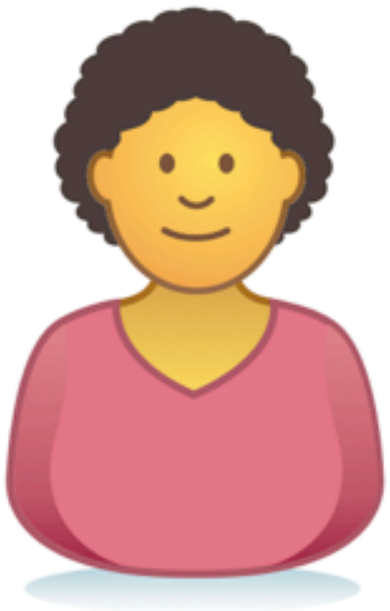




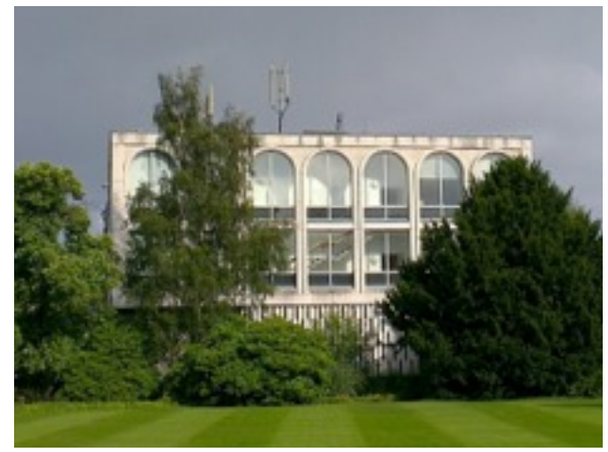


"Please may I see specimen XYZ."





"Please may I see specimen XYZ."



"Yes. Here it is."

What do we send her when she asks for a specimen?

Does she get the specimen?

No

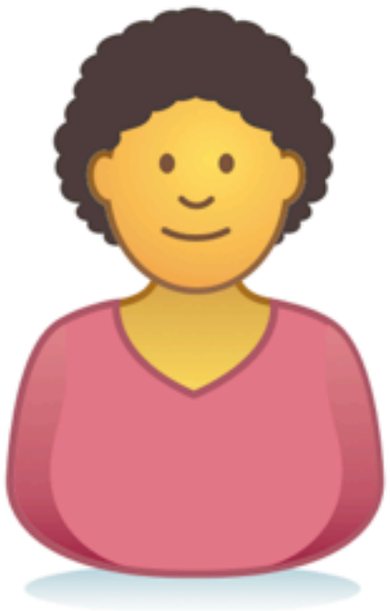
So what does she get?

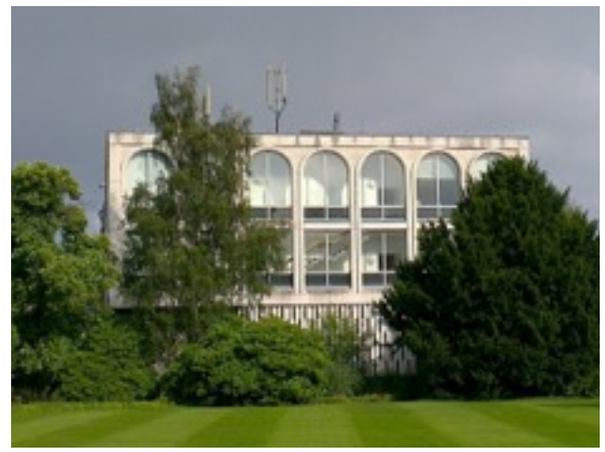
Nothing!

HTTP URIs - Linked Data Compliant

HTTP URIs - Linked Data Compliant

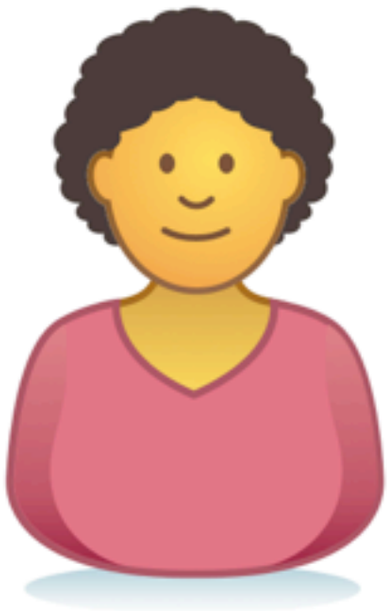
http://data.rbge.org.uk/herb/E00421509



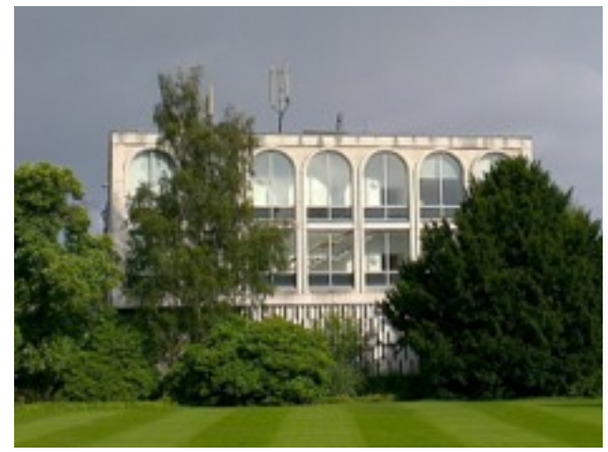


HTTP URIs - Linked Data Compliant

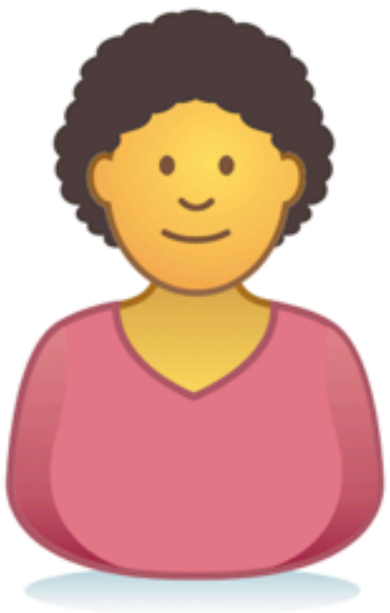
http://data.rbge.org.uk/herb/E00421509



See: http://elmer.rbge.org.uk/...



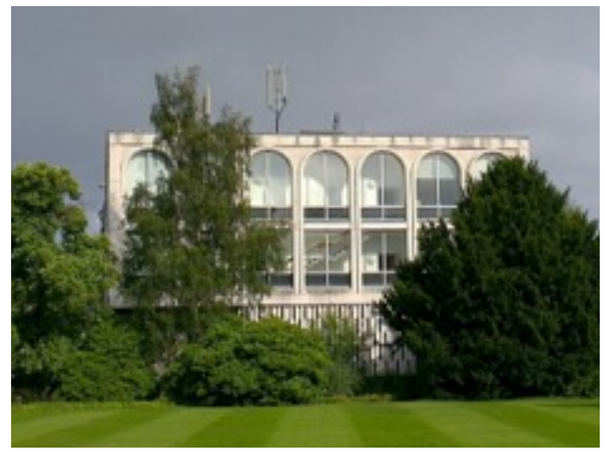
HTTP URIs - Linked Data Compliant



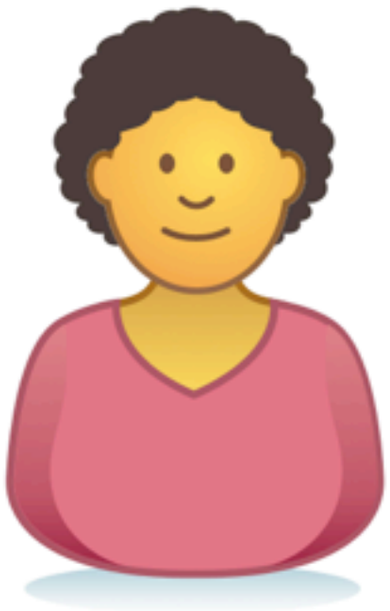
http://data.rbge.org.uk/herb/E00421509

See: http://elmer.rbge.org.uk/...

http://elmer.rbge.org.uk/...

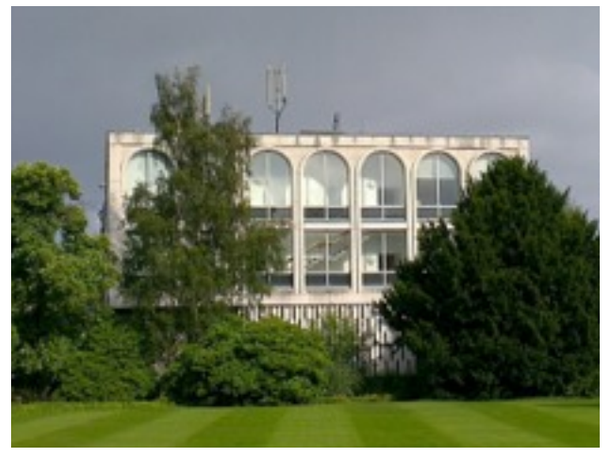


HTTP URIs - Linked Data Compliant



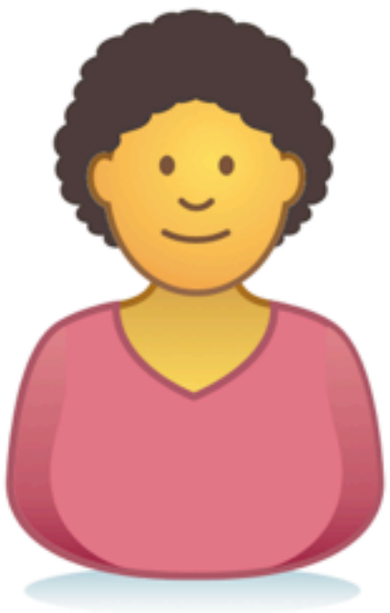
http://data.rbge.org.uk/herb/E00421509

See: http://elmer.rbge.org.uk/...



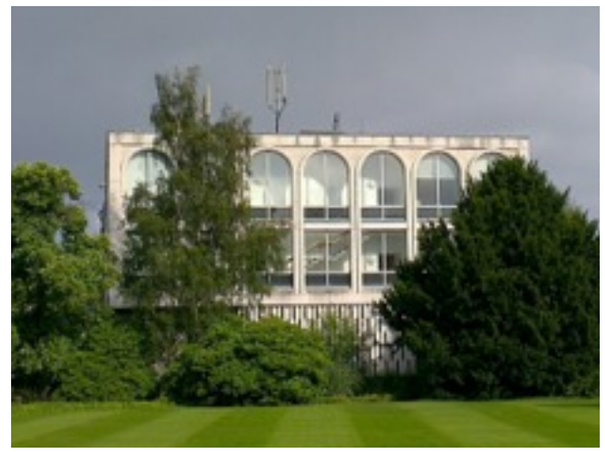
http://elmer.rbge.org.uk/...

HTTP URIs - Linked Data Compliant



http://data.rbge.org.uk/herb/E00421509

See: http://elmer.rbge.org.uk/...



http://elmer.rbge.org.uk/...

~ Just following web best practice ~

She gets a 303 redirect
to a resource
suitable for what is in
the header of her request

Why do we do this?

Data/Metadata Reason

Data/Metadata Reason

- Dublin Core possibly most widely used metadata standard

Data/Metadata Reason

- Dublin Core possibly most widely used metadata standard
- dc:created - good example

Data/Metadata Reason

- Dublin Core possibly most widely used metadata standard
- dc:created - good example
- If we return an HTML document with dc:created in it does it contain the creation date of the specimen or the HTML document?

Data/Metadata Reason

- Dublin Core possibly most widely used metadata standard
- dc:created - good example
- If we return an HTML document with dc:created in it does it contain the creation date of the specimen or the HTML document?
- App developers probably want to know both!

Data/Metadata Reason

- Dublin Core possibly most widely used metadata standard
- dc:created - good example
- If we return an HTML document with dc:created in it does it contain the creation date of the specimen or the HTML document?
- App developers probably want to know both!
- Therefore have two URIs for two sets of assertions

Human/Machine Reason

Human/Machine Reason

- Does a human user want the same response as a machine?

Human/Machine Reason

- Does a human user want the same response as a machine?
- Does a German user want the same response as a Chinese user?

Human/Machine Reason

- Does a human user want the same response as a machine?
- Does a German user want the same response as a Chinese user?
- We have multiple versions of metadata about the specimen.

Human/Machine Reason

- Does a human user want the same response as a machine?
- Does a German user want the same response as a Chinese user?
- We have multiple versions of metadata about the specimen.
- How do they dc:hasVersion to each other?

Human/Machine Reason

- Does a human user want the same response as a machine?
- Does a German user want the same response as a Chinese user?
- We have multiple versions of metadata about the specimen.
- How do they dc:hasVersion to each other?
- They all need their own URIs.

Is this the only way?

Is this the only way?

- We could just return a suitable document immediately and not do a 303 redirect

Is this the only way?

- We could just return a suitable document immediately and not do a 303 redirect
- But we would then have to make up some way of differentiating between the document and the specimen for our dc:created

Some People Don't Like 303 Redirects!

Some People Don't Like 303 Redirects!

- HTTP requests are resource intensive.

Some People Don't Like 303 Redirects!

- HTTP requests are resource intensive.
- Many (most) CMS environments can't handle the notion

Some People Don't Like 303 Redirects!

- HTTP requests are resource intensive.
- Many (most) CMS environments can't handle the notion
- Some people can't handle the notion

Some People Don't Like 303 Redirects!

- HTTP requests are resource intensive.
- Many (most) CMS environments can't handle the notion
- Some people can't handle the notion
- The debate has raged for a decade

Some People Don't Like 303 Redirects!

- HTTP requests are resource intensive.
- Many (most) CMS environments can't handle the notion
- Some people can't handle the notion
- The debate has raged for a decade
- If you don't like them stick with a Level 1 implementation - no problem

“Linked Data”

“Linked Data”

- 303 Redirects for non digital resources

“Linked Data”

- 303 Redirects for non digital resources
- Minimum RDF-XML for the data resource

“Linked Data”

- 303 Redirects for non digital resources
- Minimum RDF-XML for the data resource
- Could have HTML or other data resources

“Linked Data”

- 303 Redirects for non digital resources
- Minimum RDF-XML for the data resource
- Could have HTML or other data resources
- Do not get hung up on “Semantic Web”

“Linked Data”

- 303 Redirects for non digital resources
- Minimum RDF-XML for the data resource
- Could have HTML or other data resources
- Do not get hung up on “Semantic Web”
- Do not get hung up on “OWL” + Inference

“Linked Data”

- 303 Redirects for non digital resources
- Minimum RDF-XML for the data resource
- Could have HTML or other data resources
- Do not get hung up on “Semantic Web”
- Do not get hung up on “OWL” + Inference
- All that may never happen

Walk through of how we do it

(real, messy code)

Apache config or .htaccess File

```
RewriteCond %{REQUEST_URI} !^/service/.*$ [NC]
RewriteRule ^/(.+)$  /service/index.php?path=$1 [QSA,PT]
```

Switch to text Editor

Validator Example

http://validator.linkeddata.org/vapour

http://data.rbge.org.uk/herb/E00421509

Curl Example

```
curl -L http://data.rbge.org.uk/herb/E00421509
```

```
curl -I http://data.rbge.org.uk/herb/E00421509
```

```
curl -L -H "Accept: application/rdf+xml" http://data.rbge.org.uk/herb/E00421509
```

```
curl -L -H "Accept: application/xhtml+xml,text/html" http://data.rbge.org.uk/herb/E00421509
```