Engraving of mammoth and horses superimposed, Combarelles, Dordogne.

REST RULES

Handbook

- Herve Caumont, Terradue herve.caumont@terradue.com
- Pedro Gonçalves, Terradue pedro.goncalves@terradue.com
- Volker Mische, Couchbase volker@couchbase.com
- Jeff Harrison, The Carbon Project jharrison@thecarbonproject.com
- Peter Vretanos, CubeWerx, pvmobile@cubewerx.com

Table of Contents

    - Frontend Caching
- Hypermedia As The Engine Of State
    - Embedded links in resources
    - Defines links relations
- Security
    - defines security protocol
- Cross Origin Resource Sharing
    - supports cross-origin resource sharing

So that can start interfacing with the web service

__rationale__ We want to make sure that the web service exists and is available

Acceptance Criteria

Given a supplied url
When I make a HEAD request
Then it is valid and available

So that I can find various document links

rationale *This allows the application to mine to API. It requires the page to be properly formatted using HTML5.*

Acceptance Criteria

Given a published web service endpoint
When I access the landing page
Then I can find an HTML5 namespace

So that I can understand API

__rationale__ This allows an external application to get knowledge information regarding the service

Acceptance Criteria

Given the available web service endpoint

When I access the page
Then I can find a link to the API discovery document
And I can find a link to the OpenSearch document
And I can find a link to the API Explorer document
And I can find a link to the Documentation root
And I can find a link to an atom feed

So that I can provide in-browser search to my users

> ***rationale*** *This allows a browser to provide in-line custom search*

Acceptance Criteria

Given link to OpenSearch document
When I follow the link
Then OpenSearch Document is accessible

So that I can can provide in-browser search to my users

rationale *This allows a browser to provide in-line custom search*

Acceptance Criteria

Given link to OpenSearch document
When I follow the link and specify HTML output
Then I should get results in html

So that I can provide in-browser search to my users

__rationale__ This allows a browser to provide in-line custom search

Acceptance Criteria

Given link to OpenSearch document
When I follow the link and specify ATOM output
Then I should get results in Atom format

So that I can can get knowledge of API

rationale *This allows an application to get context information to access API*

Acceptance Criteria

Given link to API Discovery document is accessible
When I follow the link
Then API Discovery Document is accessible

Acceptance Criteria

Given link to API Discovery document is accessible
When I follow the link
Then document should contain service metadata
Then document should contain resource collections
Then document should contain schemas
Then document should contain security protocol

So that I can can get knowledge of API

rationale *This allows an application to get context information to access API*

Acceptance Criteria

Given link to Catalog document is accessible
When I follow the link
Then Catalog Document is accessible
And Catalog Document should be accessible with proper headers as well

Acceptance Criteria

Given link to API Discovery document is accessible
When I follow the link
Then document should contain service metadata
Then document should contain service collections
Then document should contain folders

Acceptance Criteria

Given link to API Discovery document is accessible
When I follow the link
Then document should point to service catalogs
Then services should have their own description documents

Acceptance Criteria

Given link to Service document is accessible
When I follow the link
Then service description is available
Then service description document should contain description
Then service description document should contain list of published resources

When I follow the link

Then service description is available

Then folders should have their own description documents

So that I can can get knowledge of API

rationale *Follow the AtomPub Discovery Service - TO BE IMPLEMENTED*

So that I can can get knowledge of API

rationale *Not offering a discovery document is not very helpful! Sorry!*

So that I can can reduce interface complexity

rationale *This allows an application to access resources in a limited and consistent manner*

Acceptance Criteria

Given link to API Discovery document
When I exercise API
Then it uses limited http verbs

So that I can can reduce interface complexity

***rationale This allows an application to access resources in a limited and consistent manner*

Acceptance Criteria

Given link to API Discovery document
When I exercise API
Then it follows idempotence rules

So that I can can get a representation that I can process

rationale *This allows an external application to get resources using a specific representation that may be easier to process.*

Acceptance Criteria

Given list a resource urls and for each url
When I set the header: Accept=text/html
Then I obtain a resource with Content-Type=text/html

Acceptance Criteria

Given list a resource urls and for each url
When I set the header: Accept=application/json
Then I obtain a resource with Content-Type=application/json

Acceptance Criteria

Given list a resource urls and for each url
When I set the header: Accept=application/rip_me (or any invalid mime type)
Then I get a 406 error

So that I can can get a representation that I can process without using Accept Headers

rationale *This allows an external user to get specific resource representation directly from the browser without using Accept headers*

Acceptance Criteria

Given list a resource urls and for each url
When I add .html extension to the URL
Then I obtain a resource with Content-Type=text/html

Acceptance Criteria

Given list a resource urls and for each url
When I add .json extension to the URL
Then I obtain a resource with Content-Type=application/json

Acceptance Criteria

Given list a resource urls and for each url
When I set .xyz or invalid extension to the URL
Then I get a 406 error

So that I can can signify my acceptance of the interface and get proper resource representation

Acceptance Criteria

Given desire to use Content Negotiation to constrain the interface
When I use Mime-types in Accept headers to determine the proper resource representation to retrieve
Then I use one of the existing IANA-approved mime-types that service may support

Acceptance Criteria

Given desire to use custom media type for Content Negotiation to constrain the interface
When I use custom Mime-type in Accept headers to determine the proper resource representation to output
Then I use existing IANA-approved mime-types
And I use a profile information attribute
And I use a version information attribute
And I use a describedby information attribute to point to an existing schema or profile document

> ***rationale*** *Content negotiation is important in some cases to specify a particular resource representation. The danger is to expand the concept to dor*
> *content. Some organizations have been pushing for custom media types of the form: application/vnd.myapp+json. This form extends the defined app*
> *mime-type and specifies the vendor-domain specific content vnd.myapp. This is problematic and turns out to hurt more than help content negotiatio*
> *developers to change their browser settings to accept the new type and may require IT administrators to change firewall/proxy settings (which my ca*
> *highy secured environments). It will not scale easily. An alternate form may be used to alleviate those problems but does not seem to be of much valu*
> *application provides a schema or profile (which it should). This alternate form is, using a json example: application/json; profile='vnd.myapp'; version=*
> *describedby='http://myapp.com/schema.rnc'. It is highly likely that the schema and/or profile will contain version information and all domain specific*
> *schema and/or profile will have been accessed in the discovery phase making that over-specification unnecessary and overly complex with no value a*

So that I can subscribe to and visualize the data feed in browser

> ***rationale*** *This allows a user to subscribe to particular data feed using a familiar NewsFeed Reader. If the feed is published to an aggregator, the user notifications on changes.*

Acceptance Criteria

Given list of resources, for each resource end point
When I use GET with an application/atom+xml Accept Header
Then server returns application/atom+xml content
And it returns a valid feed
And I can see that the feed is published to an aggregator

rationale *This allows an external application to mine the entries of a feed without having to parse the HTML content of the entry. ATOM Extensions c*
GData 2.0 or 1.0 protocol, or OData 1.0.

So that I can publish my own resources in a meaningful way

rationale *This allows an external Application to have access to the resources in a meaningful way without having to parse the HTML content*

Acceptance Criteria

Given list of resources, for each resource end point
When I use GET with an application/atom+xml Accept Header
Then feed uses etag attribute
And feed uses category tag and proper accessible schema
And feed uses OpenSearch namespace and attributes
And entries use etag attribute
And entries use category tag and proper accessible format

So that I can avoid sending superfluous data

rationale *This reduces overall bandwidth.*

So that I can avoid sending superfluous data

rationale *This reduces overall bandwidth.*

Acceptance Criteria

Given http transaction
When using GData 2.0 protocol
Then I should use GData-Version 2.0 in headers
Then I should use ETag in headers
Then I should use Last-Modified in headers
Then I should return a 304 Not Modified if using If-None-Match
Then I should return a 304 Not Modified if using Last-Modified
Then I should support Conditional Replace with If-Match in headers
Then I should support Conditional Replace with If-Match in headers
Then I should support Override Replace with If-Match: * in headers
Then I should support Conditional Delete with If-Match in headers
Then I should support Delete with If-Match: * in headers