RESTful HTTP API - Containers

- GET Retrieve the content of the resource
- POST Create new resources within a LDP container
- PUT Create a resource with a specified path, or replace the triples associated with a resource with the triples provided in the request body.
- DELETE Delete a resource
- HEAD Retrieve the resource headers
- OPTIONS Outputs information about the supported HTTP methods, etc.
- MOVE Move a resource (and its subtree) to a new location
- COPY Copy a resource (and its subtree) to a new location

Request URI: /path/to/some/resource

Methods: GET, POST, PUT, PATCH, HEAD, OPTIONS, DELETE, MOVE, COPY

### GET
Retrieves the content of the resource

Request Headers:

- **RANGE** Byte range of content to retrieve, of the form: "Range: bytes=500-999"
- **ACCEPT** RDF sources support content negotiation with these formats: application/ld+json, application/n-triples, application/rdf+xml, application/x-turtle, application/xhtml+xml, application/xml, text/html, text/n3, text/plain, text/rdf+n3, text/turtle
- **LIMIT** Number of child resources to list. If paired with Accept: (X)HTML, the default is 100. Otherwise, the default is -1 (all children are listed)
- **IF-NONE-MATCH**
- **IF-MODIFIED-SINCE**
- **PREFER** return=minimal
- **PREFER** return=representation; include="URIs"
- **PREFER** return=representation; omit="URIs"

where minimal indicates that only triples directly related to a resource should be returned, and representation indicates that links to other resources and their properties should also be included. **URIs** is a space-separate list of LDP defined preferences, and also:

<table>
<thead>
<tr>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://fedora.info/definitions/v4/repository#EmbedResources">http://fedora.info/definitions/v4/repository#EmbedResources</a></td>
<td>Embed &quot;child&quot; resources in the returned representation</td>
</tr>
<tr>
<td><a href="http://fedora.info/definitions/v4/repository#InboundReferences">http://fedora.info/definitions/v4/repository#InboundReferences</a></td>
<td>Include assertions from other Fedora resources to this node (excluded from representation by default)</td>
</tr>
<tr>
<td><a href="http://fedora.info/definitions/v4/repository#ServerManaged">http://fedora.info/definitions/v4/repository#ServerManaged</a></td>
<td>Embed server managed properties in the representation (enabled by default)</td>
</tr>
</tbody>
</table>

**Example (1): Turtle**

```
curl -H "Accept: text/turtle" "http://localhost:8080/rest/path/to/resource"
```

**Response:**
Example (2): RDF/XML

curl -H "Accept: application/rdf+xml" "http://localhost:8080/rest/path/to/resource"

Response:

Status: 200 OK

Headers:
ETag: "0ed38fe211c2663ace5322e970e7b7d606196e00"
Last-Modified: Mon, 19 May 2014 19:44:59 GMT
Preference-Applied: return=representation; include="http://www.w3.org/ns/ldp#PreferMembership http://www.w3.org/ns/ldp#PreferContainment"
Vary: Prefer
Accept-Patch: application/sparql-update
Allow: MOVE,COPY,DELETE,POST,HEAD,GET,PUT,PATCH,OPTIONS
Accept-Post: text/turtle,text/rdf+n3,text/n3,application/rdf+xml,application/n-triples,multipart/form-data,application/sparql-update
Link: <http://www.w3.org/ns/ldp#Resource>;rel="type"
Link: <http://www.w3.org/ns/ldp#DirectContainer>;rel="type"
Content-Type: text/turtle

Body:

  <http://fedora.info/definitions/v4/repository#hasParent> <http://localhost:8080/rest/path/to> ;
  a <http://www.w3.org/ns/ldp#Container> , <http://www.w3.org/ns/ldp#DirectContainer> ;
  <http://www.w3.org/ns/ldp#hasMemberRelation> <http://fedora.info/definitions/v4/repository#hasChild> ;
  a <http://www.w3.org/ns/ldp#Container> , <http://www.w3.org/ns/ldp#DirectContainer> ;
  <http://www.w3.org/ns/ldp#hasMembershipResource> <http://localhost:8080/rest/path/to/resource> ;
  <http://www.w3.org/ns/ldp#hasMemberRelation> <http://fedora.info/definitions/v4/repository#hasChild> ;
Vary: Prefer

Accept-Patch: application/sparql-update
Allow: MOVE, COPY, DELETE, POST, HEAD, GET, PUT, PATCH, OPTIONS

Accept-Post: text/turtle, text/rdf+n3, application/n3, application/rdf+xml, application/n-triples, multipart/form-data, application/sparql-update

Link: <http://www.w3.org/ns/ldp#Resource>;rel="type"
Link: <http://www.w3.org/ns/ldp#DirectContainer>;rel="type"

Content-Type: application/rdf+xml

Body:

```xml
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="http://localhost:8080/rest/path/to/resource">
    <rdf:type rdf:resource="http://www.w3.org/ns/ldp#DirectContainer"/>
    <hasMemberRelation xmlns="http://www.w3.org/ns/ldp#" rdf:resource="http://fedora.info/definitions/v4/repository#hasChild"/>
    <rdf:type rdf:resource="http://www.jcp.org/jcr/nt/1.0folder"/>
    <rdf:type rdf:resource="http://www.jcp.org/jcr/nt/1.0hierarchyNode"/>
    <primaryType xmlns="http://fedora.info/definitions/v4/repository#" rdf:resource="nt:folder"/>
    <lastModifiedBy xmlns="http://fedora.info/definitions/v4/repository#" rdf:resource="bypassAdmin"/>
    <uuid xmlns="http://fedora.info/definitions/v4/repository#" rdf:datatype="http://www.w3.org/2001/XMLSchema#string">2b655d37-e88a-44c5-86c1-3b3935cdea49</uuid>
    <createdBy xmlns="http://fedora.info/definitions/v4/repository#" rdf:resource="bypassAdmin"/>
  </rdf:Description>
</rdf:RDF>
```
Example (3): Prefer headers

```
curl -H "Prefer: return=representation; include="http://fedora.info/definitions/v4/repository#InboundReferences"; omit="http://www.w3.org/ns/ldp#PreferMembership http://www.w3.org/ns/ldp#PreferContainment"" http://localhost:8080/rest/parent
```

Response:
Example (4): Non-RDF Source

curl "http://localhost:8080/rest/path/to/binary/resource"

Response:
Create new resources within a LDP container

Request Headers:

**CONTENT-DISPOSITION** (Optional) The filename provided in the content disposition header will be stored in an ebucore:filename property. (See 'Example (6): Uploaded file with filename')

**CONTENT-TYPE** (Optional) MIME type of the uploaded binary or RDF content, including: message/external-body (See 'Example (2): Create a new child binary resource with empty content')

⚠️ If the MIME type corresponds to a supported RDF format or SPARQL-Update, the uploaded content will be parsed as RDF and used to populate the child node properties. RDF will be interpreted using the current resource as the base URI (e.g. <> will be expanded to the current URI). Namespaces must be declared in full.

For other MIME types, the uploaded content will be used to create a binary resource.

**SLUG** (Optional) A suggested name for the new child resource, which the repository may ignore.

⚠️ The slug you provide does not guarantee the location of the created resource. Clients must check the Location header for the path to the created resource.

**DIGEST** (Optional) A way of providing one or more checksums (SHA-1, SHA-256, MD5) which will be checked against the uploaded content to ensure error-free transfer. The header is interpreted according to RFC-3230 (See 'Example (4): Uploaded file with checksum' and 'Example (5): Uploaded file with checksum mismatch') If the provided checksum does not match the calculated value for the uploaded file a 409 Conflict will be returned. See PATC H example for documentation on how to change the default algorithm used during on-demand fixity checking.

Example (1): Create a new child node

curl -X POST "http://localhost:8080/rest/*"
Example (2): Create a new child binary resource with empty content

```bash
curl -i -X POST -H "Content-Type:text/plain" "http://localhost:8080/rest"
```

Response:

```
Status: 201 Created
ETag: "3ac31b09de1536a332d9c6ee8453669947721840"
Location: http://localhost:8080/rest/some/path/to/a/new/resource
Body: http://localhost:8080/rest/some/path/to/a/new/resource
```

Example (3): Create a new container with RDF properties

```bash
curl -i -X POST -H "Content-Type: text/turtle" --data-binary @body.rdf "http://localhost:8080/rest/
```

Request Body:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
<> dc:title "some-resource-title"
```

Response:

```
Status: 201 Created
ETag: "a66937ca13bc29c63f7b1a55d447a76cd8f1d573"
Location: http://localhost:8080/rest/new/node/uri
Body: http://localhost:8080/rest/new/node/uri
```

Example (4): Uploaded file with checksum

```bash
curl -i -X POST --data-binary @picture.jpg -H "digest: sha1=cb1a576f22e8e3e110611b616a3e2f5ce9bd941" "http://localhost:8080/rest/parent/container"
```

Response:
Example (4b): Uploaded file with multiple checksum

curl -i -X POST --data-binary @picture.jpg -H "digest: sha1=cb1a576f22e8e3e110611b616e3e2f5ce9dbdb941,
sha256=9533b60710d0400ee4913b5613c78f54" "http://localhost:8080/rest/parent/container"

Response:

Status: 201 Created
Headers:
ETag: "fe4e2bf389fe05dab6390564e40012d8d36eca19"
Location: http://localhost:8080/rest/parent/object/auto/generated/ds/id
Body:
http://localhost:8080/rest/parent/object/auto/generated/ds/id

Example (5): Uploaded file with checksum mismatch

curl -X POST --data-binary @picture.jpg -H "digest: sha1=checksumdoesntmatch" "http://localhost:8080/rest/parent/object"

Response:

Status: 409 Conflict
Body:
Checksum Mismatch of cb1a576f22e8e3e110611b616e3e2f5ce9dbdb941 and urn:sha1:checksumdoesntmatch

Example (6): Uploaded file with filename

curl -i -X POST --data-binary @picture.jpg -H "Content-Disposition: attachment; filename="picture.jpg"" "http://localhost:8080/rest/parent/container"

Response:

Status: 201 Created
Headers:
ETag: "fe4e2bf389fe05dab6390564e40012d8d36eca19"
Location: http://localhost:8080/rest/parent/object/auto/generated/ds/id
Body:
http://localhost:8080/rest/parent/object/auto/generated/ds/id
Status:

- **201**: Created
- **404**: Not Found (resource does not exist – use PUT to specify a new resource name)
- **409**: Conflict (checksum mismatch)

**PUT**
Create a resource with a specified path, or replace the triples associated with a resource with the triples provided in the request body.

- **Some resource properties are repository managed and cannot be removed or modified. Any attempt to do so may result in a 4xx error.**
- **RDF will be interpreted using the current resource as the base URI (e.g. <> will be expanded to the current URI)**

**Request Headers:**

- **CONTENT-TYPE**: text/turtle, text/rdf+n3, application/n3, text/n3, application/rdf+xml, application/n-triples, text/html, text/plain, application/ld+json, message/external-body
- **IF-MATCH**
- **IF-UNMODIFIED-SINCE**
- **DIGEST** *(Optional) A way of providing one or more checksums (SHA-1, SHA-256, MD5) which will be checked against the uploaded content to ensure error-free transfer. The header is interpreted according to RFC-3230 (See 'Example (3): Creating new binary resource at a specified path) If the provided checksum does not match the calculated value for the uploaded file a 409 Conflict will be returned. See Fixity Service for documentation on how to change the default algorithm used during on-demand fixity checking.*
- **PREFER** *(With the value handling=lenient; received="minimal" allows replacing the properties of a container without having to provide all of the server-managed triples.)*

**Example (1): Updating properties with RDF content**

1. **Perform a GET to the current container's RDF content**

   ```bash
   ```

   **Body:**
   ```turtle
   PREFIX dc: <http://purl.org/dc/elements/1.1/>
   ...
   PREFIX ldp: <http://www.w3.org/ns/ldp>
   fedora:primaryType "nt:folder"^^<http://www.w3.org/2001/XMLSchema#string> ;
   ...
   ```

1.2 **Add your updates (e.g. <> dc:title "some-resource-title" .)** under the full result of the GET response in 1.1

Request Body:

PREFIX dc: <http://purl.org/dc/elements/1.1/>


Example (2): Creating a new resource at a specified path

curl -i -X PUT "http://localhost:8080/rest/node/to/create"

Response:

Status: 201 Created

Response Headers:
ETag: "ef214795c3b9109389ffa542a6c081976e1e9587"
Last-Modified: Mon, 19 May 2014 20:52:57 GMT
Location: http://localhost:8080/rest/node/to/create

Body:
http://localhost:8080/rest/node/to/create

Example (3): Creating new binary resource at a specified path

curl -X PUT --upload-file image.jpg -H"Content-Type: image/jpeg" -H"digest: sha1=cba576f22e8e3e110611b616e3e2f5ce9b0b9b41" "http://localhost:8080/rest/new/image"

Response:
Note that once a resource is created as an RDF or non-RDF source, it must remain an RDF or non-RDF source. In order to change the type of resource, you must DELETE and then re-create the resource.

**Status:**

- 204 No Content
- 412 Precondition Failed

**Example (4): Creating a new binary resource at a specified path redirecting to external content**

```
curl -X PUT -H"Content-Type: message/external-body; access-type=URL; URL="http://www.example.com/file\"" "http://localhost:8080/rest/node/to/create"`
```

**Response:**

```
Status: 201 Created

Headers:
ETag: "ef214795c3b9109389ffa542a6c081976e1e9587"
Last-Modified: Mon, 19 May 2014 20:52:57 GMT
Link: <http://localhost:8080/rest/node/to/create/fcr:metadata>; rel="describedby"
Location: http://localhost:8080/rest/node/to/create

Body:
http://localhost:8080/rest/node/to/create
```

---

**PATCH**

Modify the triples associated with a resource with SPARQL-Update

**Request Headers:**

- IF-MATCH
- IF-UNMODIFIED-SINCE
- CONTENT-TYPE

**Example (1):**
curl -X PATCH -H "Content-Type: application/sparql-update" --data-binary @body.rdf "http://localhost:8080/rest/node/to/update"

Request Body:

```sparql
PREFIX dc: <http://purl.org/dc/elements/1.1/>
INSERT {
  <> dc:title "some-resource-title" .
}
WHERE { }
```

Response:

Status: 204 No Content

Example (2) Update default fixity algorithm:

```sparql
curl -X PATCH -H "Content-Type: application/sparql-update" --data-binary @body.rdf "http://localhost:8080/rest/binary/resource/to/update"

Request Body:

```sparql
PREFIX fedoraconfig: <http://fedora.info/definitions/v4/config#>
INSERT {
  <> fedoraconfig:defaultDigestAlgorithm "sha-256"
}
WHERE { }
```

Response:

Status: 204 No Content

Delete a resource

Example:

```
curl -X DELETE "http://localhost:8080/rest/resource/to/delete"
```

Response:

Status: 204 No Content

Delete requests create "tombstone" resources which need to be removed before you can reuse a URI.

Example:
curl -X GET "http://localhost:8080/rest/some/deleted/resource"

Status: 410 Gone
Link: <http://localhost:8080/rest/some/deleted/resource/fcr:tombstone>; rel="hasTombstone"

Body:
Discovered tombstone resource at /some/deleted/resource {jcr:primaryType=fedora:Tombstone, jcr:created=2014-11-20T16:04:21.131Z, jcr:createdBy=bypassAdmin}

curl -X DELETE "http://localhost:8080/rest/some/deleted/resource/fcr:tombstone"

Status: 204 No Content

Create a new node and reuse the deleted URL

curl -X PUT "http://localhost:8080/rest/some/deleted/resource"

Status: 201 Created

Status:

<table>
<thead>
<tr>
<th></th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>No Content</td>
</tr>
<tr>
<td>404</td>
<td>Not Found</td>
</tr>
</tbody>
</table>

HEAD Retrieve the resource headers

Example:

curl -i -X HEAD "http://localhost:8080/rest/resource/to/head"

Response:

Status: 200 OK
ETag: "bb92e395800153a686773f73b6ad80a51f47b"  
Last-Modified: Wed, 28 May 2014 18:31:36 GMT  
Link: <http://www.w3.org/ns/ldp#Resource>;rel="type"  
Link: <http://www.w3.org/ns/ldp#Container>;rel="type"  
Link: <http://www.w3.org/ns/ldp#BasicContainer>;rel="type"  
Accept-Patch: application/sparql-update  
Accept-Post: text/turtle,text/rdf+n3,text/n3,application/rdf+xml,application/n-triples,multipart/form-data,application/sparql-update  
Allow: MOVE,COPY,DELETE,POST,HEAD,GET,PUT,PATCH,OPTIONS

Status:

<table>
<thead>
<tr>
<th></th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
</tr>
<tr>
<td>404</td>
<td>Not Found</td>
</tr>
</tbody>
</table>
OPTIONS  Outputs information about the supported HTTP methods, etc.

Example:

curl -i -X OPTIONS "http://localhost:8080/rest/node/to/options"

Response:

<table>
<thead>
<tr>
<th>Status: 200 OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept-Patch: application/sparql-update</td>
</tr>
<tr>
<td>Allow: MOVE, COPY, DELETE, POST, HEAD, GET, PUT, PATCH, OPTIONS</td>
</tr>
<tr>
<td>Accept-Post: text/turtle, text/rdf+n3, application/n3, text/n3, application/rdf+xml, application/n-triples, multipart/form-data, application/sparql-update</td>
</tr>
</tbody>
</table>

Status:

| 200 OK |

MOVE  Move a resource (and its subtree) to a new location

Note: A federated resource can be moved within the federation, but not from the federation to the internal repository (or vice versa)

Example (1):


Response:

<table>
<thead>
<tr>
<th>Status: 201 Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Headers: Location: <a href="http://localhost:8080/rest/new/path/to/resource">http://localhost:8080/rest/new/path/to/resource</a></td>
</tr>
</tbody>
</table>

Example (2):

curl -X MOVE "http://localhost:8080/rest/path/to/resource"

Response:

| Status: 502 Bad Gateway |

Status:

| 201 Created |
| 409 Source path doesn't exist |
| 412 Destination path already exists |
| 502 Destination URI isn't a valid resource path |
**Copy a resource (and its subtree) to a new location**

**Example:**

```
```

**Response:**

<table>
<thead>
<tr>
<th>Status</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Created</td>
</tr>
<tr>
<td>409</td>
<td>Source path doesn't exist</td>
</tr>
<tr>
<td>412</td>
<td>Destination path already exists</td>
</tr>
<tr>
<td>502</td>
<td>Destination URI isn't a valid resource path</td>
</tr>
</tbody>
</table>