

RESTful HTTP API - Export and Import

- [GET fcr:export](#) Retrieve serialized form of a resource. By default, no subtree and binary content will be exported
- [POST fcr:import](#) Import a serialized resource

Request URI: `/path/to/some/resource/fcr:export`

Methods: GET, POST

 Exporting large trees of resources may fail because of memory limitations. Exporting 250,000 simple objects (only the system-defined properties) was successful and resulted in a 400MB export file, but exporting larger sets of resources failed. To work around this limitation, you can segment your repository (for example, into several top-level directories each containing 250,000 or fewer resources) and export each top-level directory as a separate export file.

Deprecated

These HTTP endpoints have been deprecated and should not be depended upon as they may get removed in future (in favor of Apache Camel powered tools).

 Objects are exported and imported relative to the request URI path given. By default, the exported jcr/xml won't contain any subtrees and binary contents.

E.g., if you run:

```
curl "http://localhost:8080/rest/path/to/resource/fcr:export" > export.xml
curl -X POST --data-binary "@export.xml" "http://localhost:8080/rest/somewhere/else"
```

The resource with no subtree and binary content will be imported to `http://localhost:8080/rest/somewhere/else/resource`.

GET

fcr:export Retrieve serialized form of a resource. By default, no subtree and binary content will be exported

Request Headers:

FORMAT

Export formats; the exact formats available depend on the repository configuration.

Parameters:

`recurse`: To export subtree with its value set to true. For example, `recurse=true`. Default is false.

`skipBinary`: To export binary content with its value set to false. For example, `skipBinary=false`. Default is true.

 The export formats that are available are shown as properties on resources, e.g. this resource exports jcr/xml:

```
<> <http://fedora.info/definitions/v4/rest-api#exportsAs> <./fcr:export?format=jcr/xml> ;
```

Example:

To export a resource with no subtree and no binary content:

```
curl "http://localhost:8080/rest/path/to/some/resource/fcr:export"
```

To export a resource with subtree and binary content:

```
curl "http://localhost:8080/rest/path/to/some/resource/fcr:export?recurse=true&skipBinary=false"
```

Response:

Status: 200

Headers:

Content-Type: application/xml

Body:

```
<sv:node xmlns:dc="http://purl.org/dc/terms/" xmlns:nt="http://www.jcp.org/jcr/nt/1.0" xmlns:ns001="http://purl.org/dc/elements/1.1/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:test="info:fedora/test/" xmlns:image="http://www.modeshape.org/images/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:fedorarelsext="http://fedora.info/definitions/v4/rels-ext#" xmlns:mix="http://www...
```

Status:

200

OK: if the resource is exported successfully

404

Not Found: if the resource does not exist

POST

fcr:import Import a serialized resource

Request Headers:

FORMAT

Import formats. The exact formats available depend on the repository configuration.

CONTENT-LOCATION

(Optional) A URI to a resource to use instead of the request body

Example:

```
curl -X POST --data-binary "@export.xml" "http://localhost:8080/rest/path/to/some/fcr:import?format=jcr/xml"
```

Body:

```
<sv:node xmlns:dc="http://purl.org/dc/terms/" xmlns:nt="http://www.jcp.org/jcr/nt/1.0" xmlns:ns001="http://purl.org/dc/elements/1.1/" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:test="info:fedora/test/" xmlns:image="http://www.modeshape.org/images/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:fedorarelsext="http://fedora.info/definitions/v4/rels-ext#" xmlns:mix="http://www...
```

Response:

Status: 201 Created

Headers:

Location: http://localhost:8080/rest/path/to/some/resource/

Status:

201

No Content: if the resource is imported successfully

404

Not Found: if the path does not exist

409

Conflict: if a node with the same UUID exists

Response:

Item already exists