

# Backup and Restore

- [Overview](#)
- [Usage](#)
  - [Backup](#)
  - [Restore](#)
- [Configurations](#)
- [Backup Format](#)

## Overview

The Fedora 4 Backup capability allows a user, such as the repository manager, make a REST call to have the repository binaries and metadata exported to the local file system. Inversely, the Restore capability allows a user to make a REST call to have the repository restored from the contents of a previous Backup operation.

## Usage

### Backup

If a POST body specifying a writeable directory (local to Fedora 4 server) is not included in the request, the backup will be written to the system temp directory.

Perform a backup of a running Fedora 4 repository

#### Request

```
POST /rest/fcr:backup  
> optional POST body
```

#### Response

On success

- HTTP/1.1 200 OK
- Path where the backup was written

Response body

- Absolute path of local backup directory

### Restore

Note: Restoring a backup replaces the repository content with the contents of the backup, so any data in the repository will be lost.

Perform a restore of a running Fedora 4 repository

#### Request

```
POST /rest/fcr:restore  
> with POST body
```

A POST body containing the full path to a previous backup.

#### Response

On success

- HTTP/1.1 204 No Content

## Configurations

The following configurations have been successfully tested with the Backup and Restore functionality

- Non-clustered Fedora, using Infinispan cache backed by LevelDB ([config](#))

# Backup Format

Regardless of the repository configuration, the output of the backup process creates resources of the same format. Further details on backup contents and the underlying implementation can be found in ModeShape's [documentation](#).

The backup directory will contain

- 'binaries' directory that contains the repository "content" binaries stored in a pair-tree like structure. The filename of the binary is the SHA-1 of the content with the extension '.bin'. The directory structure in which each binary is found is three levels deep based on the SHA-1.
  - For example, binary content in the repository with a SHA-1 of "5613537644c4d081c1dc3530fdb1a2fe843e570170d3d054", would look like

```
binaries
  44
    c4
      d0
        44c4d081c1dc3530fdb1a2fe843e570170d3d054.bin
```

- One or more "documents\_00000n.bin.gz" files which contains a concatenated listing of the metadata for each of the repository objects in JSON format
  - For example

```
{ "metadata" :
  { "id" : "87a0a8c317f1e7/jcr:system/jcr:nodeTypes/nt:unstructured//undefined/1" ,
    "contentType" : "application/json" } ,
  "content" :
  { "key" : "87a0a8c317f1e7/jcr:system/jcr:nodeTypes/nt:unstructured//undefined/1" ,
    "parent" : "87a0a8c317f1e7/jcr:system/jcr:nodeTypes/nt:unstructured" ,
    "properties" :
    { "http://www.jcp.org/jcr/1.0" :
      { "primaryType" :
        { "$name" : "nt:propertyDefinition" } ,
        "onParentVersion" : "COPY" ,
        "multiple" : false ,
        "protected" : false ,
        "availableQueryOperators" :
        [ "jcr.operator.equal.to" ,
          "jcr.operator.greater.than" ,
          "jcr.operator.greater.than.or.equal.to" ,
          "jcr.operator.less.than" ,
          "jcr.operator.less.than.or.equal.to" ,
          "jcr.operator.like" ,
          "jcr.operator.not.equal.to" ] ,
        "requiredType" : "UNDEFINED" ,
        "mandatory" : false ,
        "autoCreated" : false }
      }
    }
  }
}
```