Transactions

Overview

Fedora 4 supports the ability to wrap multiple REST API calls into a single transaction that can be committed or rolled back as an atomic operation.

After retrieving a transaction resource using the REST API, clients can send additional REST API calls within the scope of that transaction. After committing the transaction, the pending changes will be persisted in a single atomic operation. Validation, event-driven workflows, and other asynchronous operations are not triggered until the transaction is committed, which may provide a performance increase for some types of iterative tasks. Transactions can also be rolled back, as if they never occurred.

Example

Creating a new transaction resource

```
curl -v -X POST "http://localhost:8080/rest/fcr:tx"
```

Create a transaction. In the response, the Location header will contain a path to the created transaction

Create transaction response

```
Status: 201 Created

Headers:
Location: http://localhost:8080/rest/tx:83e34464-144e-43d9-af13-b3464a1fb9b5
Expires: Sat, 16 Nov 2013 00:32:57 GMT
```

The expires header indicates when the transaction will be automatically rolled back due to inactivity.

Transactions are automatically closed and rolled back after 3 minutes of inactivity. Transactions can be refreshed by POSTing to /rest/{tx:id}/fcr:tx

API requests can be prefixed with the transaction identifier:

Modifying resources in the transaction

```
curl -X POST "http://localhost:8080/rest/tx:83e34464-144e-43d9-af13-b3464a1fb9b5/path/to/object/to/create"
curl -X DELETE "http://localhost:8080/rest/tx:83e34464-144e-43d9-af13-b3464a1fb9b5/path/to/resource/to/delete"
```

You can send most API requests scoped by the transaction. API requests that are not prefixed by a path to the node do not occur within the scope of a transaction and cannot be used (e.g. fcr:nodetypes, fcr:namespaces, etc)

Committing a transaction

```
curl -X POST "http://localhost:8080/rest/tx:83e34464-144e-43d9-af13-b3464a1fb9b5/fcr:tx/fcr:commit"
```

Until the transaction is committed, no other client will see the changes made within the scope of the transaction. After committing, the changes will be applied to the workspace and all clients will be able to see the change.

Fedora 4's transactions do NOT support checkpointing. After committing a transaction, it is closed, and any further operations on that transaction will be rejected.

Rolling back a transaction

```
curl -X POST "http://localhost:8080/rest/tx:83e34464-144e-43d9-af13-b3464a1fb9b5/fcr:tx/fcr:rollback"
```

The changes made within the scope of the transaction will be erased, and the transaction closed and removed.