RESTful HTTP API - Transactions

- GET Get the current status of the repository in a transaction
- POST Create a new transaction
- POST Keep an existing transaction alive
- POST Save and commit an open transaction
- POST Rollback and close an open transaction

Request URI: /fcr:tx; /fcr:tx/{transaction_id}
Methods: GET, POST, PUT, DELETE

**GET** Get the current status of the repository in a transaction

Example:

```bash
curl -i "http://localhost:8080/rest/fcr:tx/86dd0891-d975-42d8-8837-a24ad6041b59"
```

Response:

- Status: 204 NO CONTENT
- Atomic-Expires: Sat, 16 Nov 2013 00:32:57 GMT

**POST** Create a new transaction

Example:

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

Response:

- Status: 201 Created
- Location: http://localhost:8080/rest/fcr:tx/83e34464-144e-43d9-af13-b3464a1fb9b5
- Expires: Sat, 16 Nov 2013 00:32:57 GMT

Usage:

After retrieving a transaction resource, the client can execute any REST API method with the transaction resource sent in the Atomic-ID header.

Transactions are automatically closed and rolled back after 3 minutes of inactivity. Transactions can be refreshed by POSTing to /rest/fcr:tx/{tx: id}
When a transaction has been created, it will return a Location header. Use this location as Atomic-ID header when performing REST API operations within the transaction. When you are done with the transaction, either PUT to the transaction location to commit the transaction to the repository or DELETE the transaction location to discard the changes.

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

**Status:**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>Created: if the transaction is created successfully</td>
</tr>
</tbody>
</table>

**POST**  
Keep an existing transaction alive

**Example:**

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

**Response:**

Status: 204 No Content

Headers:
- Location: http://localhost:8080/rest/tx:83e34464-144e-43d9-af13-b3464a1fb9b5
- Atomic-Expires: Sat, 16 Nov 2013 03:35:43 GMT

**Status:**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>No Content: if the transaction is renewed successfully</td>
</tr>
<tr>
<td>404</td>
<td>Not Found: Transaction doesn't exist</td>
</tr>
<tr>
<td>410</td>
<td>Gone: Transaction expired</td>
</tr>
</tbody>
</table>

**POST**  
Save and commit an open transaction

Any operations you made within the scope of the transaction will be applied together, meaning if any of them fail, the whole transaction will fail.

**Example:**

```bash
curl -i -X PUT "http://localhost:8080/rest/fcr:tx/123456789"
```

**Response:**

Status: 204 No Content

**Status:**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>No Content: if the transaction is committed successfully</td>
</tr>
<tr>
<td>Status Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>404</td>
<td>Not Found: Transaction doesn't exist</td>
</tr>
<tr>
<td>409</td>
<td>Conflict: Transaction did not commit successfully</td>
</tr>
<tr>
<td>410</td>
<td>Gone: Transaction expired</td>
</tr>
</tbody>
</table>

**POST** Rollback and close an open transaction

**Example:**

```
curl -i -X DELETE "http://localhost:8080/rest/fcr:tx/123456789"
```

**Response:**

```
Status: 204 No Content
```

**Status:**

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>204</td>
<td>No Content: if the transaction is discarded successfully</td>
</tr>
<tr>
<td>404</td>
<td>Not Found: Transaction doesn't exist</td>
</tr>
<tr>
<td>409</td>
<td>Conflict: Transaction did not commit successfully</td>
</tr>
<tr>
<td>410</td>
<td>Gone: Transaction expired</td>
</tr>
</tbody>
</table>