

# Alternative REST API

## Alternative REST API

⚠️ Aaron Birkland pointed out a severe, and unrecoverably problem with this proposal. For a versionable datastream, a new version is created for each property or content change. As such, to not make 10 versions with a datastream update, a proper transaction system, or methods that allow you to change all the properties in one invocation is nessessary. For that reason, a modifyDatastream method like in the current API is still required.

In this I will propose an alternative REST api to the one in the Fedora Documentation. It is made to support CRUD (Create, Retrieve, Update, Delete) workflows more natively than the current Fedora APIs. It is based around the idea that every conceivable url should be gettable, and probably puttable. There should not be big methods with lots of parameters, but rather a bunch of small resources with very clear purpose.

For CRUD purposes, foxml is not the format of Fedora. Foxml is a very useful and welldesigned storage format, but it is intricate. There will be no ingest or exportToXML methods in this api, as the user should not have to understand foxml. Rather, the Fedora objects is decomposed into resources, available through through URLs.

- Alternative REST API
  - General principles
  - Create methods
    - Object resources
      - Create new object with given pid
      - Create new object with autogenerated pid
    - Datastream resources
      - Create a new datastream
  - Retrieve methods
    - Object resources
      - Present Object
      - Get the object properties
      - Get a specific object property
      - Get content models for a object
      - Get all the relations of an object
      - Get a specific relation from an object
      - List the methods on an object
      - Invoke a method on an object
    - Datastream resources
      - Get Contents of a datastream
      - Get the properties of a datastream
      - Get a specific property of a datastream
      - Get the relations of the datastream
      - Get version timestamps of a datastream
      - Get the contents of a specific version of a datastream
      - Get the versioned properties of a specific version
      - Get a specific versioned property of a specific version
      - Get the relations of a specific version
  - Update methods
    - Object resources
      - Write a specific property
      - Add a relation to an object
      - Add a content model to this object
      - Add a literal relation
    - Datastream resources
      - Update the contents of an Inline Datastream
      - Update a datastream property
      - Update an External, managed or Redirect Datastream
      - Add a relation to the datastream
      - Add a literal relation to the datastream
  - Delete methods
    - Object resources
      - Purge object
      - Remove a content model from this object
      - Purge a relation to an object
      - Delete a literal relation
    - Datastream resources
      - Delete a datastream
      - Purge a relation from the datastream
      - Purge a literal relation from a datastream

## General principles

| Content             | Location       |
|---------------------|----------------|
| Objects             | /objects/      |
| Object presentation | /objects/{pid} |

|                       |                                                   |
|-----------------------|---------------------------------------------------|
| Object properties     | /objects/{pid}/properties                         |
| Object relations      | /objects/{pid}/relations                          |
| Object content models | /objects/{pid}/contentmodels                      |
| Datastreams           | /objects/{pid}/datastreams                        |
| Datastream            | /objects/{pid}/datastreams/{dsID}                 |
| Datastream content    | /objects/{pid}/datastreams/{dsID}/content         |
| Datastream properties | /objects/{pid}/datastreams/{dsID}/properties      |
| Datastream relations  | /objects/{pid}/datastreams/{dsID}/relations       |
| Datastream versions   | /objects/{pid}/datastreams/{dsID}/versions        |
| Object Methods        | /objects/{pid}/methods                            |
| Method invocation     | /objects/{pid}/methods/{sdef}/{methodName}?params |

## Create methods

### Object resources

#### Create new object with given pid

PUT /objects/{pid}

Return: Nothing or pid of the new object

Creates a new, empty object

#### Create new object with autogenerated pid

POST /objects/

Return: pid of the new object (/objects/demo:newObject2)

Creates a new, empty object

### Datastream resources

#### Create a new datastream

PUT /objects/{pid}/datastreams/{dsID}/{withControlGroup/{cg}}

Creates a new datastream in the object with controlgroup X or {cg} if specified. Datastream is empty.

## Retrieve methods

### Object resources

#### Present Object

GET /objects/{pid}

Return: XML that presents the object....

Undetailed

#### Get the object properties

GET /objects/{pid}/properties

Return: list of property names (/object/demo:myPid2/properties/label)

Provides a list of property names that can be queried

#### Get a specific object property

GET /objects/{pid}/properties/{property}  
Return: The value

Give the value of the specific property, or 404 if property not found

Example: /objects/demo:myPid2/properties/label

### Get content models for a object

GET /objects/{pid}/contentmodels  
Return: Get the content models of the object (/objects/demo:contentmodel)

Get a list of the content models of this object

### Get all the relations of an object

GET /objects/{pid}/relations  
Return: Return a list of relation names

### Get a specific relation from an object

GET /objects/{pid}/relations/{rel-name}  
Return: List of pids of objects referred by this relation

### List the methods on an object

GET /objects/{pid}/methods  
Return: a list of objects

### Invoke a method on an object

GET/POST /objects/{pid}/methods/{sdef}/{methodName}?params

## Datastream resources

### Get Contents of a datastream

GET /objects/{pid}/datastreams/{dsID}

### Get the properties of a datastream

GET /objects/{pid}/datastreams/{dsID}/properties

### Get a specific property of a datastream

GET /objects/{pid}/datastreams/{dsID}/properties/{prop}  
Return: The value

### Get the relations of the datastream

GET /objects/{pid}/datastreams/{dsID}/relations

The RELS-INT block "about" this datastream

### Get version timestamps of a datastream

The idea is that the old version of the datastream should support the same getter methods as the current version

GET /objects/{pid}/datastreams/{dsID}/versions

### Get the contents of a specific version of a datastream

GET /objects/{pid}/datastreams/{dsID}/versions/{timestamp}/contents

### Get the versioned properties of a specific version

GET /objects/{pid}/datastreams/{dsID}/versions/{timestamp}/properties

### Get a specific versioned property of a specific version

GET /objects/{pid}/datastreams/{dsID}/versions/{timestamp}/properties/{propertyname}

## Get the relations of a specific version

GET /objects/{pid}/datastreams/{dsID}/versions/{timestamp}/relations/

## Update methods

### Object resources

#### Write a specific property

PUT /objects/{pid}/properties/{property}

Contents: The new value

Return nothing, or 404 if property not found. New properties cannot be created.

#### Add a relation to an object

PUT /objects/{pid}/relations/{rel-name}/to/{objpid}

#### Add a content model to this object

PUT /objects/{pid}/contentmodels/{cmpid}

Contents: Nothing

Add cmpid to the list of content model for this object

#### Add a literal relation

PUT /objects/{pid}/relations/{rel-name}/toLiteral/{value}/{withType/{type}}

### Datastream resources

#### Update the contents of an Inline Datastream

POST /objects/{pid}/datastreams/{dsID}/contents

Contents: The new content of the datastream

#### Update a datastream property

PUT /objects/{pid}/datastreams/{dsID}/properties/{prop}

Contents: The new value

examples: versionable, state and so on.

#### Update an External, managed or Redirect Datastream

ContentLocation is just a property of these datastreams

PUT /objects/{pid}/datastreams/{dsID}/properties/contentLocation

Contents: The URL to the contents of the datastream

#### Add a relation to the datastream

PUT /objects/{pid}/datastreams/{dsID}/relations/{rel-name}/to/{objPid}

#### Add a literal relation to the datastream

PUT /objects/{pid}/datastreams/{dsID}/relations/{rel-name}/toLiteral/{value}/{withType/{type}}

## Delete methods

### Object resources

#### Purge object

DELETE /objects/{pid}

#### Remove a content model from this object

DELETE /objects/{pid}/contentmodels/{cmpid}

Contents: Nothing

Remove cmpid from the list of content models for this object

### **Purge a relation to an object**

DELETE /objects/{pid}/relations/{rel-name}/to/{objpid}

### **Delete a literal relation**

DELETE /objects/{pid}/relations/{rel-name}/toLiteral/{value}

## **Datastream resources**

### **Delete a datastream**

DELETE /objects/{pid}/datastreams/{dsID}

Purge the datastream from the object from the repository

### **Purge a relation from the datastream**

DELETE /objects/{pid}/datastreams/{dsID}/relations/{rel-name}/to/{objPid}

### **Purge a literal relation from a datastream**

DELETE /objects/{pid}/datastreams/{dsID}/relations/{rel-name}/toLiteral/{value}