IslandoraFedoraAPI

1.1

Generated by Doxygen 1.8.2

Tue Mar 12 2013 14:50:48

Contents

1 Todo List 1

2 Hierarchical Index 3
2.1 Class Hierarchy 3

3 Class Index 5
3.1 Class List 5

4 File Index 7
4.1 File List 7

5 Class Documentation 9
5.1 AbstractCache Class Reference 9
5.1.1 Detailed Description 9
5.1.2 Member Function Documentation 9
5.1.2.1 add 9
5.1.2.2 delete 10
5.1.2.3 get 10
5.1.2.4 set 10
5.2 AbstractDatastream Class Reference 10
5.2.1 Detailed Description 11
5.2.2 Constructor & Destructor Documentation 12
5.2.2.1 __construct 12
5.2.3 Member Function Documentation 12
5.2.3.1 getContent 12
5.2.3.2 setContentFromFile 12
5.2.3.3 setContentFromString 12
5.2.3.4 setContentFromUrl 12
5.3 AbstractExistingFedoraDatastream Class Reference 13
5.3.1 Detailed Description 13
5.3.2 Constructor & Destructor Documentation 13
5.3.2.1 __construct 13
5.3.3 Member Function Documentation 14

5.3.3.1 getDatastreamContent 14
5.3.3.2 getDatastreamHistory 14
5.3.3.3 modifyDatastream 14
5.3.3.4 purgeDatastream 14
5.4 AbstractFedoraDatastream Class Reference 15
5.4.1 Detailed Description 16
5.4.2 Constructor & Destructor Documentation 16
5.4.2.1 __construct 16
5.4.3 Member Function Documentation 16
5.4.3.1 controlGroupMagicProperty 16
5.4.3.2 delete 16
5.4.3.3 idMagicProperty 16
5.4.3.4 isDatastreamPropertySet 16
5.4.3.5 validateChecksumType 17
5.4.3.6 validateMimetype 17
5.4.3.7 validateState 17
5.4.3.8 validateVersionable 17
5.5 AbstractFedoraObject Class Reference 18
5.5.1 Detailed Description 19
5.5.2 Member Function Documentation 19
5.5.2.1 __toString 19
5.5.2.2 constructDatastream 19
5.5.2.3 delete 19
5.5.2.4 idMagicProperty 19
5.5.2.5 labelMagicProperty 19
5.5.2.6 logMessageMagicProperty 19
5.5.2.7 modelsMagicProperty 19
5.5.2.8 ownerMagicProperty 20
5.5.2.9 stateMagicProperty 20
5.6 AbstractObject Class Reference 20
5.6.1 Detailed Description 21
5.6.2 Constructor & Destructor Documentation 21
5.6.2.1 __construct 21
5.6.3 Member Function Documentation 21
5.6.3.1 constructDatastream 21
5.6.3.2 getDatastream 22
5.6.3.3 purgeDatastream 22
5.7 AbstractRepository Class Reference 22
5.7.1 Detailed Description 23
5.7.2 Member Function Documentation 23

5.7.2.1 constructObject 23

5.7.2.2 findObjects 23

5.7.2.3 getObject 23

5.7.2.4 ingestObject 24

5.7.2.5 purgeObject 24

5.8 CurlConnection Class Reference 24

5.8.1 Detailed Description 25

5.8.2 Constructor & Destructor Documentation 26

5.8.2.1 __construct 26

5.8.2.2 __destruct 26

5.8.3 Member Function Documentation 26

5.8.3.1 deleteRequest 26

5.8.3.2 doCurlRequest 26

5.8.3.3 getRequest 26

5.8.3.4 patchRequest 26

5.8.3.5 postRequest 26

5.8.3.6 putRequest 27

5.9 FedoraApi Class Reference 27

5.9.1 Detailed Description 27

5.9.2 Constructor & Destructor Documentation 27

5.9.2.1 __construct 27

5.10 FedoraApiA Class Reference 28

5.10.1 Detailed Description 28

5.10.2 Constructor & Destructor Documentation 28

5.10.2.1 __construct 28

5.10.3 Member Function Documentation 29

5.10.3.1 describeRepository 29

5.10.3.2 findObjects 29

5.10.3.3 getDatastreamDissemination 31

5.10.3.4 getDissemination 31

5.10.3.5 getObjectHistory 31

5.10.3.6 getObjectProfile 32

5.10.3.7 listDatastreams 32

5.10.3.8 listMethods 33

5.10.3.9 resumeFindObjects 34

5.11 FedoraApiM Class Reference 34

5.11.1 Detailed Description 35

5.11.2 Constructor & Destructor Documentation 35

5.11.2.1 __construct 35

5.11.3 Member Function Documentation 35

5.11.3.1 addDatastream 35
5.11.3.2 addRelationship 37
5.11.3.3 export 37
5.11.3.4 getDatastream 38
5.11.3.5 getDatastreamHistory 38
5.11.3.6 getNextPid 39
5.11.3.7 getObjectXml 40
5.11.3.8 getRelationships 40
5.11.3.9 ingest 41
5.11.3.10 modifyDatastream 42
5.11.3.11 modifyObject 43
5.11.3.12 purgeDatastream 43
5.11.3.13 purgeObject 44
5.11.3.14 validate 44
5.12 FedoraApiSerializer Class Reference 45
5.12.1 Detailed Description 46
5.12.2 Member Function Documentation 46
5.12.2.1 domDocumentExceptionHandler 46
5.12.2.2 flattenDocument 47
5.12.2.3 loadDomDocument 47
5.12.2.4 loadSimpleXml 47
5.13 FedoraDatastream Class Reference 48
5.13.1 Detailed Description 49
5.13.2 Constructor & Destructor Documentation 49
5.13.2.1 __construct 49
5.13.3 Member Function Documentation 50
5.13.3.1 checksumMagicProperty 50
5.13.3.2 checksumTypeMagicProperty 50
5.13.3.3 contentMagicProperty 50
5.13.3.4 controlGroupMagicProperty 50
5.13.3.5 count 50
5.13.3.6 createdDateMagicProperty 50
5.13.3.7 formatMagicProperty 50
5.13.3.8 getContent 50
5.13.3.9 labelMagicProperty 51
5.13.3.10 locationMagicProperty 51
5.13.3.11 logMessageMagicProperty 51
5.13.3.12 mimetypeMagicProperty 51
5.13.3.13 offsetExists 51
5.13.3.14 offsetGet 51

5.13.3.15 offsetSet 51
5.13.3.16 offsetUnset 51
5.13.3.17 setContentFromFile 51
5.13.3.18 setContentFromString 52
5.13.3.19 setContentFromUrl 52
5.13.3.20 sizeMagicProperty 52
5.13.3.21 stateMagicProperty 52
5.13.3.22 urlMagicProperty 52
5.13.3.23 versionableMagicProperty 52
5.14 FedoraDatastreamVersion Class Reference 52
5.14.1 Detailed Description 53
5.14.2 Constructor & Destructor Documentation 54
5.14.2.1 __construct 54
5.14.3 Member Function Documentation 54
5.14.3.1 checksumMagicProperty 54
5.14.3.2 contentMagicProperty 54
5.14.3.3 createdDateMagicProperty 54
5.14.3.4 formatMagicProperty 54
5.14.3.5 getContent 54
5.14.3.6 labelMagicProperty 54
5.14.3.7 logMessageMagicProperty 54
5.14.3.8 mimetypeMagicProperty 55
5.14.3.9 setContentFromFile 55
5.14.3.10 setContentFromString 55
5.14.3.11 setContentFromUrl 55
5.14.3.12 sizeMagicProperty 55
5.14.3.13 stateMagicProperty 55
5.14.3.14 versionableMagicProperty 55
5.15 FedoraDate Class Reference 55
5.15.1 Detailed Description 56
5.15.2 Constructor & Destructor Documentation 56
5.15.2.1 __construct 56
5.15.3 Member Function Documentation 56
5.15.3.1 __sleep 56
5.15.3.2 __toString 56
5.16 FedoraObject Class Reference 57
5.16.1 Detailed Description 58
5.16.2 Constructor & Destructor Documentation 58
5.16.2.1 __construct 58
5.16.3 Member Function Documentation 58

5.16.3.1 constructDatastream 58
5.16.3.2 count 58
5.16.3.3 createdDateMagicProperty 58
5.16.3.4 getDatastream 58
5.16.3.5 getParents 59
5.16.3.6 ingestDatastream 59
5.16.3.7 labelMagicPropertySet 59
5.16.3.8 lastModifiedDateMagicProperty 59
5.16.3.9 logMessageMagicPropertySet 59
5.16.3.10 offsetExists 59
5.16.3.11 offsetGet 59
5.16.3.12 offsetSet 59
5.16.3.13 offsetUnset 60
5.16.3.14 ownerMagicPropertySet 60
5.16.3.15 purgeDatastream 60
5.16.3.16 stateMagicPropertySet 60
5.17 FedoraRelationships Class Reference 60
5.17.1 Detailed Description 61
5.17.2 Constructor & Destructor Documentation 61
5.17.2.1 __construct 61
5.17.3 Member Function Documentation 61
5.17.3.1 changeObjectID 61
5.17.3.2 getDom 62
5.17.3.3 getXpath 62
5.17.3.4 getXpathResults 62
5.17.3.5 internalAdd 62
5.17.3.6 internalGet 62
5.17.3.7 internalRemove 63
5.17.3.8 registerNamespace 63
5.17.4 Member Data Documentation 64
5.17.4.1 $namespaces 64
5.18 FedoraRelsExt Class Reference 64
5.18.1 Constructor & Destructor Documentation 65
5.18.1.1 __construct 65
5.18.2 Member Function Documentation 65
5.18.2.1 add 65
5.18.2.2 get 65
5.18.2.3 initializeDatastream 66
5.18.2.4 remove 66
5.19 FedoraRelsInt Class Reference 66

5.19.1 Constructor & Destructor Documentation 67
5.19.1.1 __construct 67
5.19.2 Member Function Documentation 67
5.19.2.1 add 67
5.19.2.2 get 67
5.19.2.3 remove 68
5.20 FedoraRepository Class Reference 68
5.20.1 Detailed Description 69
5.20.2 Constructor & Destructor Documentation 69
5.20.2.1 __construct 69
5.20.3 Member Function Documentation 69
5.20.3.1 constructObject 69
5.20.3.2 findObjects 70
5.20.3.3 getObject 70
5.20.3.4 ingestObject 70
5.20.3.5 purgeObject 70
5.21 FoxmlDocument Class Reference 70
5.22 HttpConnection Class Reference 71
5.22.1 Detailed Description 72
5.22.2 Member Function Documentation 72
5.22.2.1 deleteRequest 72
5.22.2.2 getRequest 72
5.22.2.3 patchRequest 73
5.22.2.4 postRequest 73
5.22.2.5 putRequest 74
5.23 HttpConnectionException Class Reference 74
5.23.1 Detailed Description 75
5.23.2 Constructor & Destructor Documentation 75
5.23.2.1 __construct 75
5.23.3 Member Function Documentation 75
5.23.3.1 getResponse 75
5.24 MagicProperty Class Reference 75
5.24.1 Detailed Description 76
5.24.2 Member Function Documentation 77
5.24.2.1 __get 77
5.24.2.2 __isset 77
5.24.2.3 __set 77
5.24.2.4 __unset 77
5.24.2.5 getGeneralMagicPropertyMethodName 78
5.25 NewFedoraDatastream Class Reference 78

5.25.1 Detailed Description 79
5.25.2 Constructor & Destructor Documentation 79
5.25.2.1 __construct 79
5.25.3 Member Function Documentation 79
5.25.3.1 checksumMagicProperty 79
5.25.3.2 checksumTypeMagicProperty 79
5.25.3.3 contentMagicProperty 79
5.25.3.4 controlGroupMagicProperty 80
5.25.3.5 formatMagicProperty 80
5.25.3.6 getContent 80
5.25.3.7 labelMagicProperty 80
5.25.3.8 logMessageMagicProperty 80
5.25.3.9 mimetypeMagicProperty 80
5.25.3.10 setContentFromFile 80
5.25.3.11 setContentFromString 80
5.25.3.12 setContentFromUrl 81
5.25.3.13 stateMagicProperty 81
5.25.3.14 urlMagicProperty 81
5.25.3.15 validateControlGroup 81
5.25.3.16 validateType 81
5.25.3.17 versionableMagicProperty 81
5.26 NewFedoraObject Class Reference 82
5.26.1 Detailed Description 82
5.26.2 Member Function Documentation 83
5.26.2.1 constructDatastream 83
5.26.2.2 count 83
5.26.2.3 getDatastream 83
5.26.2.4 getIterator 83
5.26.2.5 idMagicProperty 83
5.26.2.6 ingestDatastream 83
5.26.2.7 offsetExists 84
5.26.2.8 offsetGet 84
5.26.2.9 offsetSet 84
5.26.2.10 offsetUnset 84
5.26.2.11 purgeDatastream 84
5.27 RepositoryBadArguementException Class Reference 84
5.27.1 Detailed Description 84
5.28 RepositoryConfigInterface Interface Reference 85
5.28.1 Detailed Description 85
5.29 RepositoryConnection Class Reference 85

5.29.1 Detailed Description 86
5.29.2 Constructor & Destructor Documentation 86
5.29.2.1 __construct 86
5.29.3 Member Function Documentation 86
5.29.3.1 addParam 86
5.29.3.2 addParamArray 87
5.29.3.3 deleteRequest 87
5.29.3.4 getRequest 87
5.29.3.5 parseFedoraExceptions 87
5.29.3.6 patchRequest 87
5.29.3.7 postRequest 88
5.29.3.8 putRequest 88
5.30 RepositoryException Class Reference 88
5.30.1 Detailed Description 88
5.31 RepositoryQuery Class Reference 88
5.31.1 Constructor & Destructor Documentation 89
5.31.1.1 __construct 89
5.31.2 Member Function Documentation 89
5.31.2.1 itqlQuery 89
5.31.2.2 parseSparqlResults 89
5.31.2.3 pidUriToBarePid 89
5.31.2.4 query 90
5.31.2.5 sparqlQuery 90
5.32 RepositoryRelationshipsException Class Reference 90
5.32.1 Detailed Description 91
5.33 RepositoryXmlError Class Reference 91
5.33.1 Detailed Description 91
5.34 SimpleCache Class Reference 91
5.34.1 Detailed Description 92
5.34.2 Member Function Documentation 92
5.34.2.1 add 92
5.34.2.2 delete 92
5.34.2.3 get 92
5.34.2.4 set 93
5.34.2.5 setCacheSize 93

6 File Documentation 95

6.1 Cache.php File Reference 95
6.1.1 Detailed Description 95
6.2 Datastream.php File Reference 95

6.2.1 Detailed Description 96
6.3 FedoraApi.php File Reference 96
6.3.1 Detailed Description 96
6.4 FedoraApiSerializer.php File Reference 96
6.4.1 Detailed Description 96
6.5 FedoraRelationships.php File Reference 96
6.5.1 Detailed Description 97
6.6 HttpConnection.php File Reference 97
6.6.1 Detailed Description 97
6.7 MagicProperty.php File Reference 97
6.7.1 Detailed Description 98
6.8 Object.php File Reference 98
6.8.1 Detailed Description 98
6.9 Repository.php File Reference 98
6.9.1 Detailed Description 98
6.10 RepositoryConnection.php File Reference 98
6.10.1 Detailed Description 99
6.11 RepositoryException.php File Reference 99
6.11.1 Detailed Description 99
6.12 RepositoryQuery.php File Reference 99
6.12.1 Detailed Description 99

Index 99

Chapter 1

Todo List

Member AbstractFedoraDatastream::validateMimetype ($mime)

test if this covers all cases.

Member AbstractRepository::findObjects (array $search)

Flesh out the function definition for this.

Member FedoraApiM::ingest ($params=array())

This function is a problem in Fedora < 3.5 where ownerId does not properly get set. `https://jira.duraspace.org/browse/FCREPO-963`. We should deal with this.

Member FedoraDatastream::checksumMagicProperty ($function, $value)

maybe add functionality to set it to auto

Member FedoraDatastream::contentMagicProperty ($function, $value)

We should perhaps cache this? depending on size?

Class FedoraRelationships

potentially we should validate the predicate URI

Member FedoraRepository::constructObject ($id=NULL)

validate the ID

catch the getNextPid errors

Member FedoraRepository::findObjects (array $search)

this needs to be implemented!

Member FedoraRepository::getObject ($id)

perhaps we should check if an object exists instead of catching the exception

Member FedoraRepository::ingestObject (NewFedoraObject &$object)

error handling

Member HttpConnection::postRequest ($url, $type= 'none', $data=NULL, $content_type=NULL)

Test this for posting general form data. (Other then files.)

Member NewFedoraDatastream::__construct ($id, $control_group, AbstractFedoraObject $object, FedoraRepository $repository)

test for valid identifiers. it can't start with a number etc.

Member NewFedoraDatastream::checksumMagicProperty ($function, $value)

this should be refined a bit

Class SimpleCache

Replace this with something more interesting like memcached

Try some other intersting caching strategies like LRU.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- AbstractCache 9
  - SimpleCache 91
- ArrayAccess
  - AbstractObject 20
    - AbstractFedoraObject 18
      - FedoraObject 57
      - NewFedoraObject 82
  - FedoraDatastream 48
- Countable
  - AbstractObject 20
  - FedoraDatastream 48
- DateTime
  - FedoraDate 55
- DOMDocument
  - FoxmlDocument 70
- Exception
  - HttpConnectionException 74
  - RepositoryException 88
    - RepositoryBadArguementException 84
    - RepositoryRelationshipsException 90
    - RepositoryXmlError 91
- FedoraApi 27
- FedoraApiA 28
- FedoraApiM 34
- FedoraApiSerializer 45
- FedoraRelationships 60
  - FedoraRelsExt 64
  - FedoraRelsInt 66
- HttpConnection 71
  - CurlConnection 24
    - RepositoryConnection 85
- IteratorAggregate
  - AbstractObject 20
  - FedoraDatastream 48
- MagicProperty 75
  - AbstractDatastream 10
    - AbstractFedoraDatastream 15

AbstractExistingFedoraDatastream 13

FedoraDatastream 48

FedoraDatastreamVersion 52

NewFedoraDatastream 78

AbstractObject 20

AbstractRepository 22

FedoraRepository 68

RepositoryConfigInterface 85

RepositoryConnection 85

RepositoryQuery 88

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AbstractCache
Simple abstract Cache defintion providing basic key value caching functionality 9
AbstractDatastream
This abstract class can be overriden by anything implementing a datastream 10
AbstractExistingFedoraDatastream
This abstract class defines some shared functionality between all classes that implement exising fedora datastreams 13
AbstractFedoraDatastream
Abstract base class implementing a datastream in Fedora 15
AbstractFedoraObject
This is the base class for a Fedora Object 18
AbstractObject
An abstract class defining a Object in the repository 20
AbstractRepository
An abstract repository interface 22
CurlConnection
This class defines a abstract HttpConnection using the PHP cURL library 24
FedoraApi
This is a simple class that brings FedoraApiM and FedoraApiA together 27
FedoraApiA
This class implements the Fedora API-A interface 28
FedoraApiM
This class implements the Fedora API-M interface 34
FedoraApiSerializer
A class to Serialize the XML responses from Fedora into PHP arrays 45
FedoraDatastream
This class implements a fedora datastream 48
FedoraDatastreamVersion
This class implements an old version of a fedora datastream 52
FedoraDate
This file represents a Fedora Date 55
FedoraObject
Represents a Fedora Object 57
FedoraRelationships
This is the base class for Fedora Relationships 60
FedoraRelsExt 64
FedoraRelsInt 66

FedoraRepository
Concrete implementation of the AbstractRepository for Fedora 68
FoxmlDocument 70
HttpConnection
Abstract class defining functions for HTTP connections 71
HttpConnectionException
HTTP Exception 74
MagicProperty
This abstract class allows us to implement PHP magic properties by defining a private method in the class that entends it 75
NewFedoraDatastream
This defines a new fedora datastream 78
NewFedoraObject
This represents a new fedora object that hasn't been ingested yet 82
RepositoryBadArguementException
This is thrown when a bad arguement is passed 84
RepositoryConfigInterface
The general interface for a RepositoryConfig object 85
RepositoryConnection
Specific RepositoryConfig implementation that extends the CurlConnection class so that we can do specific processing on Curl requests for Fedora 85
RepositoryException
The top level exception for the Islandora Fedora API 88
RepositoryQuery 88
RepositoryRelationshipsException
This is thrown from the relationships class when something goes wrong 90
RepositoryXmlError
This is thrown when there is an error parsing XML 91
SimpleCache
This is a simple cache that uses a static array to hold the cached values 91

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Cache.php
Simple set of classes defining caches 95
Datastream.php
This file defines all the classes used to manipulate datastreams in the repository 95
FedoraApi.php
The RAW API wrappers for the Fedora interface 96
FedoraApiSerializer.php
This file contains functions to serialize the XML responses from Fedora into something more easily dealt with in PHP 96
FedoraRelationships.php
This file defines the classes that are used for manipulaing the fedora relationships datastreams 96
HttpConnection.php
This file defines the classes used to make HTTP requests 97
MagicProperty.php
This file contains the class MagicProperty 97
Object.php
This file contains all of the functionality for objects in the repository 98
Repository.php
This file defines an abstract repository that can be overridden and also defines a concrete implementation for Fedora 98
RepositoryConnection.php
This file contains the implementation of a connection to Fedora 98
RepositoryException.php
Contains all of the exceptions thrown by Tuque 99
RepositoryQuery.php
This file provides some methods for doing RDF queries 99

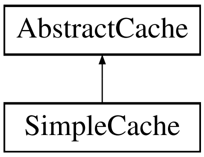
Chapter 5

Class Documentation

5.1 AbstractCache Class Reference

Simple abstract Cache defintion providing basic key value caching functionality.

Inheritance diagram for AbstractCache:



Public Member Functions

- add ($key, $data)

  Add data to the cache.
- get ($key)

  Retrieve data from the cache.
- set ($key, $data)

  Set data in the cache.
- delete ($key)

  Delete key from the cache.

5.1.1 Detailed Description

Simple abstract Cache defintion providing basic key value caching functionality.

5.1.2 Member Function Documentation

5.1.2.1 AbstractCache::add (*$key*, *$data*) `[abstract]`

Add data to the cache.

Parameters

string	*$key*	The key to add to the cache.
mixed	*$data*	The data to store with the key.

Returns

boolean TRUE if the data was added to the cache. FALSE if $key already exists in the cache or if there was an error.

5.1.2.2 AbstractCache::delete (*$key*) `[abstract]`

Delete key from the cache.

Parameters

string	*$key*	The key to delete.

Returns

boolean TRUE if they key existed and was deleted. FALSE otherwise.

5.1.2.3 AbstractCache::get (*$key*) `[abstract]`

Retrieve data from the cache.

Parameters

string	*$key*	They key to retrieve from the cache.

Returns

mixed FALSE if the data wasn't found in the cache. Otherwise it returns the data assoctiated with the key.

5.1.2.4 AbstractCache::set (*$key*, *$data*) `[abstract]`

Set data in the cache.

This will create new keys if they don't already exist, or update existing keys.

Parameters

string	*$key*	The key to add/update.
mixed	*$data*	The data to store with the key.

Returns

boolean TRUE on success. FALSE on failure.

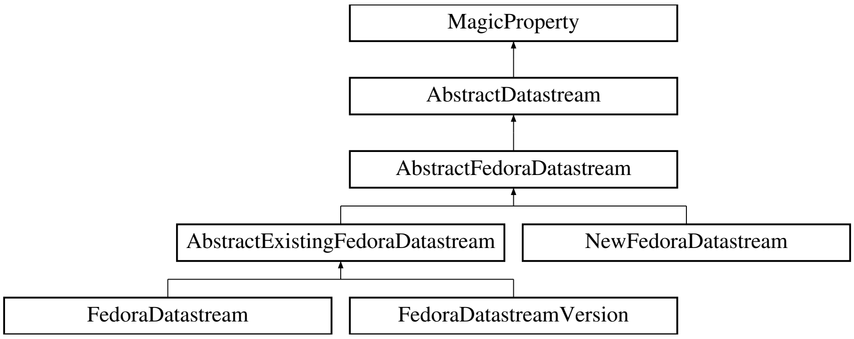
The documentation for this class was generated from the following file:

- Cache.php

5.2 AbstractDatastream Class Reference

This abstract class can be overriden by anything implementing a datastream.

Inheritance diagram for AbstractDatastream:



Public Member Functions

- delete ()

  This will set the state of the datastream to deleted.
- setContentFromFile ($file)

  Set the contents of the datastream from a file.
- setContentFromUrl ($url)

  Set the contents of the datastream from a URL.
- setContentFromString ($string)

  Set the contents of the datastream from a string.
- getContent ($file)

  Get the contents of a datastream and output it to the file provided.
- __construct ()

  Unsets public members.
- __wakeup ()

  Upon unserialization unset any public members.

Public Attributes

- **$id**
- **$label**
- **$location**
- **$controlGroup**
- **$versionable**
- **$state**
- **$mimetype**
- **$format**
- **$size**
- **$checksum**
- **$checksumType**
- **$createdDate**
- **$content**
- **$url**
- **$logMessage**

Additional Inherited Members

5.2.1 Detailed Description

This abstract class can be overriden by anything implementing a datastream.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 AbstractDatastream::__construct ( )

Unsets public members.

We only define the public members of the object for Doxygen, they aren't actually accessed or used, and if they are not unset, they can cause problems after unserialization.

5.2.3 Member Function Documentation

5.2.3.1 AbstractDatastream::getContent (*$file*) `[abstract]`

Get the contents of a datastream and output it to the file provided.

Parameters

string	*$file*	The path of the file to output the contents of the datastream to.

Returns

TRUE on success or FALSE on failure.

5.2.3.2 AbstractDatastream::setContentFromFile (*$file*) `[abstract]`

Set the contents of the datastream from a file.

Parameters

string	*$file*	The full path of the file to set to the contents of the datastream.

5.2.3.3 AbstractDatastream::setContentFromString (*$string*) `[abstract]`

Set the contents of the datastream from a string.

Parameters

string	*$string*	The string whose contents will become the contents of the datastream.

5.2.3.4 AbstractDatastream::setContentFromUrl (*$url*) `[abstract]`

Set the contents of the datastream from a URL.

The contents of this URL will be fetched, and the datastream will be updated to contain the contents of the URL.

Parameters

string	*$url*	The full URL to fetch.

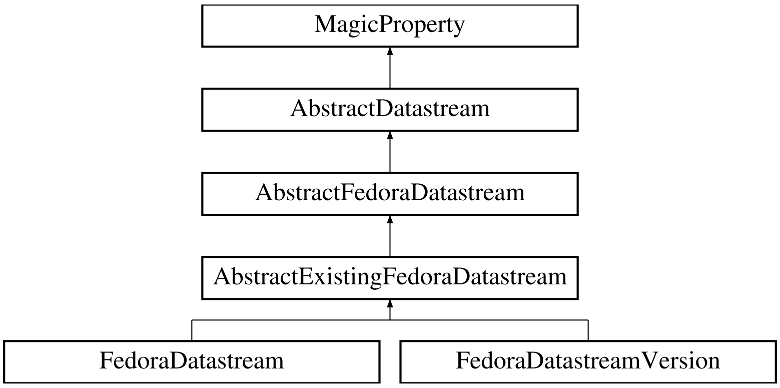
The documentation for this class was generated from the following file:

- Datastream.php

5.3 AbstractExistingFedoraDatastream Class Reference

This abstract class defines some shared functionality between all classes that implement exising fedora datastreams.

Inheritance diagram for AbstractExistingFedoraDatastream:



Public Member Functions

- __construct ($id, FedoraObject $object, FedoraRepository $repository)

  Class constructor.

Protected Member Functions

- getDatastreamContent ($version=NULL, $file=NULL)

  Wrapper for the APIA getDatastreamDissemination function.
- getDatastreamHistory ()

  Wrapper around the APIM getDatastreamHistory function.
- modifyDatastream (array $args)

  Wrapper around the APIM modifyDatastream function.
- purgeDatastream ($version)

  Wrapper around the APIM Purge function.

Additional Inherited Members

5.3.1 Detailed Description

This abstract class defines some shared functionality between all classes that implement exising fedora datastreams.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 AbstractExistingFedoraDatastream::__construct (*$id,* FedoraObject *$object,* FedoraRepository *$repository*)

Class constructor.

Parameters

string	*$id*	Unique identifier for the DS.
FedoraObject	*$object*	The FedoraObject that this DS belongs to.
FedoraRepository	*$repository*	The FedoraRepository that this DS belongs to.

5.3.3 Member Function Documentation

5.3.3.1 AbstractExistingFedoraDatastream::getDatastreamContent (*$version* = NULL, *$file* = NULL) [protected]

Wrapper for the APIA getDatastreamDissemination function.

Parameters

string	*$version*	The version of the content to retreve.
string	*$file*	The file to put the content into.

Returns

string String containing the content.

5.3.3.2 AbstractExistingFedoraDatastream::getDatastreamHistory ( ) [protected]

Wrapper around the APIM getDatastreamHistory function.

Returns

array Array containing datastream history.

5.3.3.3 AbstractExistingFedoraDatastream::modifyDatastream (array *$args*) [protected]

Wrapper around the APIM modifyDatastream function.

Parameters

array	*$args*	Args to pass to the function.

Returns

array Datastream history array.

5.3.3.4 AbstractExistingFedoraDatastream::purgeDatastream (*$version*) [protected]

Wrapper around the APIM Purge function.

Parameters

string	*$version*	The version to purge.

Returns

array The versions purged.

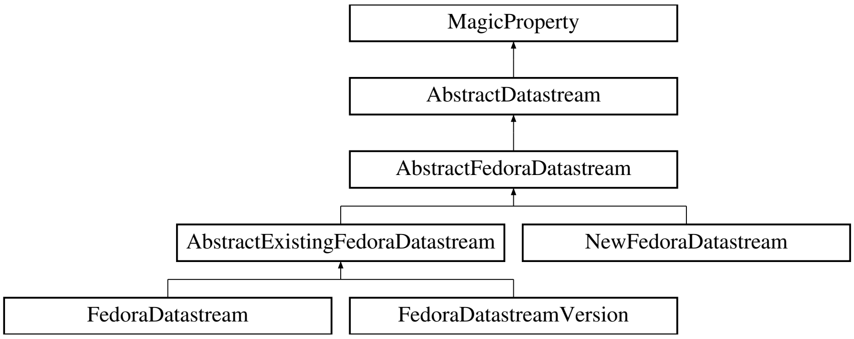
The documentation for this class was generated from the following file:

- Datastream.php

5.4 AbstractFedoraDatastream Class Reference

Abstract base class implementing a datastream in Fedora.

Inheritance diagram for AbstractFedoraDatastream:



Public Member Functions

- __construct ($id, AbstractFedoraObject $object, FedoraRepository $repository)

  The constructor for the datastream.
- delete ()

Public Attributes

- **$repository**
- **$parent**
- **$relationships**

Protected Member Functions

- idMagicProperty ($function, $value)
- isDatastreamProperySet ($actual, $unsetval)

  This is a replacement for isset when things can't be unset.
- validateMimetype ($mime)

  Validates a mimetype using a regular expression.
- validateState ($state)

  Validates and normalizes the datastream state.
- validateVersionable ($versionable)

  Validates the versionable setting of a datastream.
- validateChecksumType ($type)

  Validates and normalizes the checksumType arguement.
- controlGroupMagicProperty ($function, $value)

Protected Attributes

- **$datastreamId** = NULL
- **$datastreamInfo** = NULL
- **$fedoraRelsIntClass** = 'FedoraRelsInt'
- **$fedoraDatastreamVersionClass** = 'FedoraDatastreamVersion'

5.4.1 Detailed Description

Abstract base class implementing a datastream in Fedora.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 AbstractFedoraDatastream::__construct (*$id*, **AbstractFedoraObject** *$object*, **FedoraRepository** *$repository*)

The constructor for the datastream.

Parameters

string	*$id*	The identifier of the datastream.

5.4.3 Member Function Documentation

5.4.3.1 AbstractFedoraDatastream::controlGroupMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::controlGroup

5.4.3.2 AbstractFedoraDatastream::delete ( )

See Also

AbstractDatastream::delete()

5.4.3.3 AbstractFedoraDatastream::idMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::id

5.4.3.4 AbstractFedoraDatastream::isDatastreamProperySet (*$actual*, *$unsetval*) [protected]

This is a replacement for isset when things can't be unset.

So we define a default value, then return TRUE or FALSE based on if it is set.

Parameters

anything	*$actual*	The value we are testing.
anything	*$unsetval*	The value it would be if it was unset.

Returns

boolean TRUE or FALSE

5.4.3.5 AbstractFedoraDatastream::validateChecksumType (*$type*) [protected]

Validates and normalizes the checksumType arguement.

Parameters

string	*$type*	The input string

Returns

mixed FALSE if validation fails. The checksumType string otherwise.

5.4.3.6 AbstractFedoraDatastream::validateMimetype (*$mime*) [protected]

Validates a mimetype using a regular expression.

Parameters

string	*$mime*	A string representing a mimetype

Returns

boolean TRUE if the string looks like a mimetype.

Todo test if this covers all cases.

5.4.3.7 AbstractFedoraDatastream::validateState (*$state*) [protected]

Validates and normalizes the datastream state.

Parameters

string	*$state*	The input state

Returns

string Returns FALSE if validation fails, otherwise it returns the normalized datastream state.

5.4.3.8 AbstractFedoraDatastream::validateVersionable (*$versionable*) [protected]

Validates the versionable setting of a datastream.

Parameters

mixed	*$versionable*	The input versionable arguement.

Returns

boolean Returns TRUE if the arguement is a boolean, FALSE otherwise.

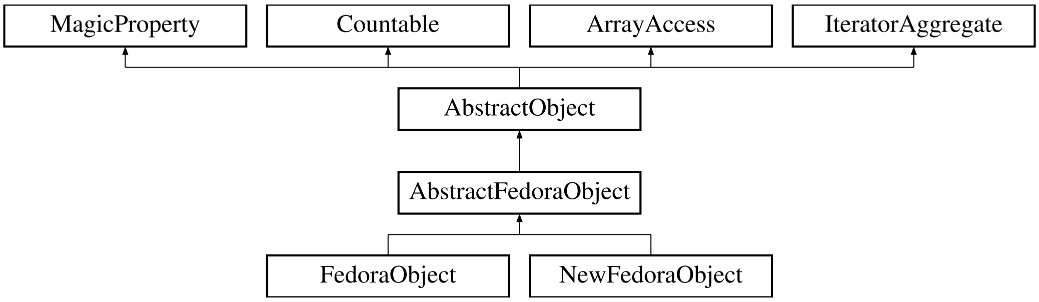
The documentation for this class was generated from the following file:

- Datastream.php

5.5 AbstractFedoraObject Class Reference

This is the base class for a Fedora Object.

Inheritance diagram for AbstractFedoraObject:



Public Member Functions

- __construct ($id, FedoraRepository $repository)

  Constructosaurus.
- __toString ()

  Implements Magic Method.
- delete ()
- constructDatastream ($id, $control_group= 'M')

Public Attributes

- **$relationships**
- **$repository**

Protected Member Functions

- idMagicProperty ($function, $value)
- stateMagicProperty ($function, $value)
- labelMagicProperty ($function, $value)
- ownerMagicProperty ($function, $value)
- logMessageMagicProperty ($function, $value)
- modelsMagicProperty ($function, $value)

Protected Attributes

- **$objectId**
- **$objectProfile**
- **$fedoraDatastreamClass** = 'FedoraDatastream'
- **$newFedoraDatastreamClass** = 'NewFedoraDatastream'
- **$fedoraRelsExtClass** = 'FedoraRelsExt'

5.5.1 Detailed Description

This is the base class for a Fedora Object.

5.5.2 Member Function Documentation

5.5.2.1 AbstractFedoraObject::__toString ( )

Implements Magic Method.

Returns PID of object when object is printed

Returns

string

5.5.2.2 AbstractFedoraObject::constructDatastream (*$id*, *$control_group* = 'M')

See Also

AbstractObject::constructDatastream()

5.5.2.3 AbstractFedoraObject::delete ( )

See Also

AbstractObject::delete()

5.5.2.4 AbstractFedoraObject::idMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractObject::id

5.5.2.5 AbstractFedoraObject::labelMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractObject::label

5.5.2.6 AbstractFedoraObject::logMessageMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractObject::logMessage

5.5.2.7 AbstractFedoraObject::modelsMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractObject::models

5.5.2.8 AbstractFedoraObject::ownerMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractObject::owner

5.5.2.9 AbstractFedoraObject::stateMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractObject::state

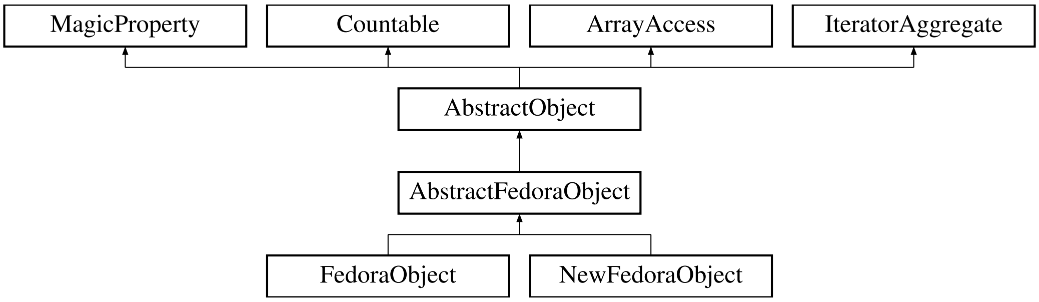
The documentation for this class was generated from the following file:

- Object.php

5.6 AbstractObject Class Reference

An abstract class defining a Object in the repository.

Inheritance diagram for AbstractObject:



Public Member Functions

- delete ()

  Set the state of the object to deleted.
- getDatastream ($id)

  Get a datastream from the object.
- purgeDatastream ($id)

  Purges a datastream.
- constructDatastream ($id, $control_group= 'M')

  Factory to create new datastream objects.
- ingestDatastream (&$ds)

  Ingests a datastream object into the repository.
- __construct ()

  Unsets public members.
- __wakeup ()

  Upon unserialization unset any public members.

Public Attributes

- **$label**
- **$owner**
- **$state**
- **$id**
- **$createdDate**
- **$lastModifiedDate**
- **$logMessage**
- **$models**

Additional Inherited Members

5.6.1 Detailed Description

An abstract class defining a Object in the repository.

This is the class that needs to be implemented in order to create new repository backends that can be accessed using Tuque.

These classes implement the php object array interfaces so that the object can be accessed as an array. This provides access to datastreams. The object is also traversable with foreach, so that each datastream can be accessed.

```
$object = new AbstractObject()

// access every object
foreach ($object as $dsid => $dsObject) {
  // print dsid and set contents to "foo"
  print($dsid);
  $dsObject->content = 'foo';
}

// test if there is a datastream called 'DC'
if (isset($object['DC'])) {
  // if there is print its contents
  print($object['DC']->content);
}
```

5.6.2 Constructor & Destructor Documentation

5.6.2.1 AbstractObject::__construct ( )

Unsets public members.

We only define the public members of the object for Doxygen, they aren't actually accessed or used, and if they are not unset, they can cause problems after unserialization.

5.6.3 Member Function Documentation

5.6.3.1 AbstractObject::constructDatastream (*$id*, *$control_group* = 'M') [abstract]

Factory to create new datastream objects.

Creates a new datastream object, this object is not ingested into the repository until you call ingestDatastream.

Parameters

string	*$id*	The identifier of the new datastream.
string	*$control_group*	The control group the new datastream will be created in.

Returns

AbstractDatastream Returns an instantiated Datastream object.

5.6.3.2 AbstractObject::getDatastream (*$id*) [abstract]

Get a datastream from the object.

Parameters

string	*$id*	The id of the datastream to retreve.

Returns

AbstractDatastream Returns FALSE if the datastream could not be found. Otherwise it return an instantiated Datastream object.

5.6.3.3 AbstractObject::purgeDatastream (*$id*) [abstract]

Purges a datastream.

Parameters

string	*$id*	The id of the datastream to purge.

Returns

boolean TRUE on success. FALSE on failure.

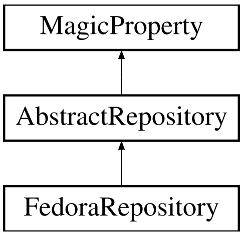
The documentation for this class was generated from the following file:

- Object.php

5.7 AbstractRepository Class Reference

An abstract repository interface.

Inheritance diagram for AbstractRepository:



Public Member Functions

- constructObject ($id=NULL)

  This method is a factory that will return a new repositoryobject object that can be manipulated and then ingested into the repository.
- ingestObject (NewFedoraObject &$object)

This ingests a new object into the repository.

- getObject ($id)

  Gets a object from the repository.
- purgeObject ($id)

  Removes an object from the repository.
- findObjects (array $search)

  Search the repository for objects.

Additional Inherited Members

5.7.1 Detailed Description

An abstract repository interface.

This can be used to override the implementation of the Repository.

5.7.2 Member Function Documentation

5.7.2.1 AbstractRepository::constructObject (*$id* = NULL) [abstract]

This method is a factory that will return a new repositoryobject object that can be manipulated and then ingested into the repository.

Parameters

string	*$id*	The ID to assign to this object. There are three options: • NULL: An ID will be assigned. • A namespace: An ID will be assigned in this namespace. • A whole ID: The whole ID must contains a namespace and a identifier in the form NAMESPACE:IDENTIFIER

Returns

AbstractObject Returns an instantiated AbstractObject object that can be manipulated. This object will not actually be created in the repository until the ingest method is called.

5.7.2.2 AbstractRepository::findObjects (array *$search*) [abstract]

Search the repository for objects.

This function isn't implemented yet.

Todo Flesh out the function definition for this.

5.7.2.3 AbstractRepository::getObject (*$id*) [abstract]

Gets a object from the repository.

Parameters

string	*$id*	The identifier of the object.

Returns

AbstractObject The requested object.

5.7.2.4 AbstractRepository::ingestObject (NewFedoraObject & *$object*) [abstract]

This ingests a new object into the repository.

Parameters

Abstract-Object	*&$object*	The instantiated AbstractObject to ingest into the repository. This object is passed by reference, and the reference will be replaced by an object representing the ingested AbstractObject.

Returns

AbstractObject The ingested abstract object.

5.7.2.5 AbstractRepository::purgeObject (*$id*) [abstract]

Removes an object from the repository.

This function removes an object from the repository premenenty. It is a dangerous function since it remvoes an object and all of its history from the repository permenently.

Parameters

string	*$id*	The identifier of the object.

Returns

boolean TRUE if object was purged.

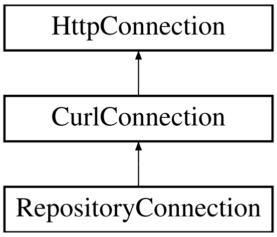
The documentation for this class was generated from the following file:

- Repository.php

5.8 CurlConnection Class Reference

This class defines a abstract HttpConnection using the PHP cURL library.

Inheritance diagram for CurlConnection:



Public Member Functions

- __construct ()

Constructor for the connection.

- __sleep ()

  Save the cookies to the sessions and remember all of the parents members.

- __wakeup ()

  Restore the cookies file and initialize curl.

- __destruct ()

  Destructor for the connection.

- patchRequest ($url, $type= 'none', $data=NULL, $content_type=NULL)
- postRequest ($url, $type= 'none', $data=NULL, $content_type=NULL)
- putRequest ($url, $type= 'none', $file=NULL)
- getRequest ($url, $headers_only=FALSE, $file=NULL)
- deleteRequest ($url)

Public Attributes

- const **COOKIE_LOCATION** = 'curl_cookie'

Protected Member Functions

- createCookieFile ()

  Create a file to store cookies.

- saveCookiesToSession ()

  Save the contents of the cookie file to the session.

- getCurlContext ()

  This function sets up the context for curl.

- unallocateCurlContext ()

  Unallocate curl context.

- setupCurlContext ($url)

  This sets the curl options.

- doCurlRequest ($file=NULL)

  This function actually does the cURL request.

Protected Attributes

- **$cookieFile** = NULL

Static Protected Attributes

- static **$curlContext** = NULL

5.8.1 Detailed Description

This class defines a abstract HttpConnection using the PHP cURL library.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 CurlConnection::__construct ( )

Constructor for the connection.

Exceptions

HttpConnectionException	

5.8.2.2 CurlConnection::__destruct ( )

Destructor for the connection.

Save the cookies to the session unallocate curl, and free the cookies file.

5.8.3 Member Function Documentation

5.8.3.1 CurlConnection::deleteRequest (*$url*)

See Also

HttpConnection::deleteRequest

5.8.3.2 CurlConnection::doCurlRequest (*$file* = NULL) [protected]

This function actually does the cURL request.

It is a private function meant to be called by the public get, post and put methods.

Exceptions

HttpConnectionException	

Returns

array Array has keys: (status, headers, content).

5.8.3.3 CurlConnection::getRequest (*$url*, *$headers_only* = FALSE, *$file* = NULL)

See Also

HttpConnection::getRequest

5.8.3.4 CurlConnection::patchRequest (*$url*, *$type* = 'none', *$data* = NULL, *$content_type* = NULL)

See Also

HttpConnection::patchRequest

5.8.3.5 CurlConnection::postRequest (*$url*, *$type* = 'none', *$data* = NULL, *$content_type* = NULL)

See Also

HttpConnection::postRequest

5.8.3.6 CurlConnection::putRequest (*$url*, *$type* = 'none', *$file* = NULL)

See Also

HttpConnection::putRequest

The documentation for this class was generated from the following file:

- HttpConnection.php

5.9 FedoraApi Class Reference

This is a simple class that brings FedoraApiM and FedoraApiA together.

Public Member Functions

- __construct (RepositoryConnection $connection=NULL, FedoraApiSerializer $serializer=NULL)

  Constructor for the FedoraApi object.

Public Attributes

- **$a**
- **$m**
- **$connection**

5.9.1 Detailed Description

This is a simple class that brings FedoraApiM and FedoraApiA together.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 FedoraApi::__construct (RepositoryConnection *$connection* = NULL, FedoraApiSerializer *$serializer* = NULL)

Constructor for the FedoraApi object.

Parameters

RepositoryConnection	*$connection*	(Optional) If one isn't provided a default one will be used.
FedoraApiSerializer	*$serializer*	(Optional) If one isn't provided a default will be used.

The documentation for this class was generated from the following file:

- FedoraApi.php

5.10 FedoraApiA Class Reference

This class implements the Fedora API-A interface.

Public Member Functions

- __construct (RepositoryConnection $connection, FedoraApiSerializer $serializer)

  Constructor for the new FedoraApiA object.
- describeRepository ()

  Returns basic information about the Repository.
- findObjects ($type, $query, $max_results=NULL, $display_fields=array('pid', 'title'))

  Query fedora to return a list of objects.
- resumeFindObjects ($session_token)

  Given a session key returned from findObject, this will return the next set of objects.
- getDatastreamDissemination ($pid, $dsid, $as_of_date_time=NULL, $file=NULL)

  Get the default dissemination of a datastream.
- getDissemination ($pid, $sdef_pid, $method, $method_parameters=NULL)

  Get a datastream dissemination from Fedora.
- getObjectHistory ($pid)

  Get the change history for the object.
- getObjectProfile ($pid, $as_of_date_time=NULL)

  Implements the getObjectProfile Fedora API-A method.
- listDatastreams ($pid, $as_of_date_time=NULL)

  List all the datastreams that are associated with this PID.
- listMethods ($pid, $sdef_pid= '', $as_of_date_time=NULL)

  Implements the listMethods Fedora API-A method.

Protected Attributes

- **$connection**
- **$serializer**

5.10.1 Detailed Description

This class implements the Fedora API-A interface.

This is a light wrapper around the Fedora interface. Very little attempt is put into putting things into native PHP datastructures.

See this page for more information: `https://wiki.duraspace.org/display/FEDORA35/REST+API`

5.10.2 Constructor & Destructor Documentation

5.10.2.1 FedoraApiA::__construct (RepositoryConnection *$connection*, FedoraApiSerializer *$serializer*)

Constructor for the new FedoraApiA object.

Parameters

RepositoryConnection	*$connection*	Takes the Respository Connection object for the Respository this API should connect to.
FedoraApiSerializer	*$serializer*	Takes the serializer object to that will be used to serialze the XML Fedora returns.

5.10.3 Member Function Documentation

5.10.3.1 FedoraApiA::describeRepository ( )

Returns basic information about the Repository.

This is listed as an unimplemented function in the official API for Fedora. However other libraries connecting to the Fedora REST interaface use this so we are including it here. It may change in the future.

Exceptions

RepositoryException	

Returns

array() An array describing the repository.

```
Array
(
    [repositoryName] => Fedora Repository
    [repositoryBaseURL] => http://localhost:8080/fedora
    [repositoryVersion] => 3.4.1
    [repositoryPID] => Array
        (
            [PID-namespaceIdentifier] => changeme
            [PID-delimiter] => :
            [PID-sample] => changeme:100
            [retainPID] => *
        )

    [repositoryOAI-identifier] => Array
        (
            [OAI-namespaceIdentifier] => example.org
            [OAI-delimiter] => :
            [OAI-sample] => oai:example.org:changeme:100
        )

    [sampleSearch-URL] => http://localhost:8080/fedora/objects
    [sampleAccess-URL] => http://localhost:8080/fedora/objects/demo:5
    [sampleOAI-URL] => http://localhost:8080/fedora/oai?verb=Identify
    [adminEmail] => Array
        (
            [0] => bob@example.org
            [1] => sally@example.org
        )

)
```

5.10.3.2 FedoraApiA::findObjects (*$type*, *$query*, *$max_results* = NULL, *$display_fields* = array('pid', 'title'))

Query fedora to return a list of objects.

Parameters

string	*$type*	The type of query. Decides the format of the next parameter. Valid options are: • query: specific query on certain fields • terms: search in any field
string	*$query*	The format of this parameter depends on what was passed to type. The formats are: • query: A sequence of space-separated conditions. A condition consists of a metadata element name followed directly by an operator, followed directly by a value. Valid element names are (pid, label, state, ownerId, cDate, mDate, dcmDate, title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage, rights). Valid operators are: contains (∼), equals (=), greater than (>), less than (<), greater than or equals (>=), less than or equals (<=). The contains (∼) operator may be used in combination with the ? and ∗ wildcards to query for simple string patterns. Values may be any string. If the string contains a space, the value should begin and end with a single quote character ('). If all conditions are met for an object, the object is considered a match. • terms: A phrase represented as a sequence of characters (including the ? and ∗ wildcards) for the search. If this sequence is found in any of the fields for an object, the object is considered a match.
int	*$max_results*	(optional) Default: 25. The maximum number of results that the server should provide at once.
array	*$display_fields*	(optional) Default: array('pid', 'title'). The fields to be returned as an indexed array. Valid element names are the same as the ones given for the query parameter.

Exceptions

RepositoryException	

Returns

array() The results are returned in an array key called 'results'. If there are more results that aren't returned then the search session information is contained in a key called 'session'. Note that it is possible for some display fields to be multivalued, such as identifier (DC allows multiple DC identifier results) in the case there are multiple results an array is returned instread of a string, this indexed array contains all of the values.

```
Array
(
   [session] => Array
       (
           [token] => 96b2604f040067645f45daf029062d6e
           [cursor] => 0
           [expirationDate] => 2012-03-07T14:28:24.886Z
       )

   [results] => Array
       (
           [0] => Array
               (
                   [pid] => islandora:collectionCModel
                   [title] => Islandora Collection Content Model
                   [identifier] => Contents of DC:Identifier
               )

           [1] => Array
               (
                   [pid] => islandora:testCModel
                   [title] => Test content model for Ari
                   [identifier] => Array
                       (
                           [0] => Contents of first DC:Identifier
```

```
                              [1] => Contents of seconds DC:Identifier
                          )

                  )

          )

    )
```

5.10.3.3 FedoraApiA::getDatastreamDissemination (*$pid*, *$dsid*, *$as_of_date_time* = NULL, *$file* = NULL)

Get the default dissemination of a datastream.

(Get the contents.)

Parameters

String	*$pid*	Persistent identifier of the digital object.
String	*$dsid*	Datastream identifier.
array	*$as_of_date_-time*	(optional) Indicates that the result should be relative to the digital object as it existed at the given date and time. Defaults to the most recent version.
array	*$file*	(optional) A file to retrieve the dissemination into.

Exceptions

RespositoryException	

Returns

string The response from Fedora with the contents of the datastream if file isn't set. Returns TRUE if the file parameter is passed.

5.10.3.4 FedoraApiA::getDissemination (*$pid*, *$sdef_pid*, *$method*, *$method_parameters* = NULL)

Get a datastream dissemination from Fedora.

Parameters

String	*$pid*	Persistent identifier of the digital object.
String	*$sdef_pid*	Persistent identifier of the sDef defining the methods.
String	*$method*	Method to invoke.
String	*$method_-parameters*	A key-value paired array of parameters required by the method.

Exceptions

RespositoryException	

Returns

string The response from Fedora.

5.10.3.5 FedoraApiA::getObjectHistory (*$pid*)

Get the change history for the object.

Parameters

String	*$pid*	Persistent identifier of the digital object.

Exceptions

RepositoryException	

Returns

array() An array containing the different revisions of the object.

```
Array
(
    [0] => 2011-07-08T18:01:40.384Z
    [1] => 2011-07-08T18:01:40.464Z
    [2] => 2011-07-08T18:01:40.552Z
    [3] => 2011-07-08T18:01:40.694Z
    [4] => 2012-02-22T15:07:15.305Z
    [5] => 2012-02-29T14:20:28.857Z
    [6] => 2012-02-29T14:22:18.239Z
    [7] => 2012-02-29T14:22:46.545Z
    [8] => 2012-02-29T20:52:33.069Z
)
```

5.10.3.6 FedoraApiA::getObjectProfile (*$pid*, *$as_of_date_time* = NULL)

Implements the getObjectProfile Fedora API-A method.

Parameters

String	*$pid*	Persistent identifier of the digital object.
String	*$as_of_date_time*	(Optional) Indicates that the result should be relative to the digital object as it existed on the given date. Date Format: yyyy-MM-dd or yyyy-MM-ddTHH:mm-:ssZ

Exceptions

RepositoryException	

Returns

array() Returns information about the digital object.

```
Array
(
    [objLabel] => Islandora strict PDF content model
    [objOwnerId] => fedoraAdminnnn
    [objModels] => Array
        (
            [0] => info:fedora/fedora-system:ContentModel-3.0
            [1] => info:fedora/fedora-system:FedoraObject-3.0
        )

    [objCreateDate] => 2011-07-08T18:01:40.384Z
    [objLastModDate] => 2012-03-02T20:50:13.534Z
    [objDissIndexViewURL] => http://localhost:8080/fedora/objects/
      islandora%3Astrict_pdf/methods/fedora-system%3A3/viewMethodIndex
    [objItemIndexViewURL] => http://localhost:8080/fedora/objects/
      islandora%3Astrict_pdf/methods/fedora-system%3A3/viewItemIndex
    [objState] => A
)
```

5.10.3.7 FedoraApiA::listDatastreams (*$pid*, *$as_of_date_time* = NULL)

List all the datastreams that are associated with this PID.

Parameters

string	*$pid*	Persistent identifier of the digital object.
string	*$as_of_date_-time*	(optional) Indicates that the result should be relative to the digital object as it existed on the given date. Date Format: yyyy-MM-dd or yyyy-MM-ddTHH:mm-:ssZ.

Exceptions

RepositoryException	

Returns

array() An associative array with the dsid of the datastreams as the key and the mimetype and label as the value.

```
Array
(
    [DC] => Array
        (
            [label] => Dublin Core Record for this object
            [mimetype] => text/xml
        )

    [RELS-EXT] => Array
        (
            [label] => Fedora Object-to-Object Relationship Metadata
            [mimetype] => text/xml
        )

    [ISLANDORACM] => Array
        (
            [label] => ISLANDORACM
            [mimetype] => text/xml
        )

)
```

5.10.3.8 FedoraApiA::listMethods (*$pid*, *$sdef_pid* = ' ', *$as_of_date_time* = NULL)

Implements the listMethods Fedora API-A method.

Parameters

String	*$pid*	Persistent identifier of the digital object.
String	*$sdef_pid*	(Optional) Persistent identifier of the SDef defining the methods.
String	*$as_of_date_-time*	(Optional) Indicates that the result should be relative to the digital object as it existed on the given date. Date Format: yyyy-MM-dd or yyyy-MM-ddTHH:mm-:ssZ.

Exceptions

RepositoryException	

Returns

array() An array containing data about the methods that can be called. The result array is an associative array where the sdef pid is the key and the value is a indexed array of methods.

```
Array
(
    [ilives:viewerSdef] => Array
        (
            [0] => getViewer
        )

    [ilives:jp2Sdef] => Array
```

```
        (
            [0] => getMetadata
            [1] => getRegion
        )

    [fedora-system:3] => Array
       (
            [0] => viewObjectProfile
            [1] => viewMethodIndex
            [2] => viewItemIndex
            [3] => viewDublinCore
        )

)
```

5.10.3.9 FedoraApiA::resumeFindObjects (*$session_token*)

Given a session key returned from findObject, this will return the next set of objects.

Parameters

string	*$session_token*	Session token returned from previous search call.

Exceptions

RespositoryException	

Returns

array() The result format is the same as findObjects.

See Also

FedoraApiA::findObjects

The documentation for this class was generated from the following file:

- FedoraApi.php

5.11 FedoraApiM Class Reference

This class implements the Fedora API-M interface.

Public Member Functions

- __construct (RepositoryConnection $connection, FedoraApiSerializer $serializer)

  Constructor for the new FedoraApiM object.
- addDatastream ($pid, $dsid, $type, $file, $params)

  Add a new datastream to a fedora object.
- addRelationship ($pid, $relationship, $is_literal, $datatype=NULL)

  Add a RDF relationship to a Fedora object.
- export ($pid, $params=array())

  Export a Fedora object with the given PID.
- getDatastream ($pid, $dsid, $params=array())

  Returns information about the datastream.
- getDatastreamHistory ($pid, $dsid)

Get information on the different versions of a datastream that are avilable in Fedora.

- getNextPid ($namespace=NULL, $numpids=NULL)

  Get a new unused PID.
- getObjectXml ($pid)

  Get the Fedora Objects XML (Foxml)
- getRelationships ($pid, $relationship=array())

  Query relationships for a particular fedora object.
- ingest ($params=array())

  Create a new object in Fedora.
- modifyDatastream ($pid, $dsid, $params=array())

  Update a datastream.
- modifyObject ($pid, $params=NULL)

  Update Fedora Object parameters.
- purgeDatastream ($pid, $dsid, $params=array())

  Purge a datastream from from Fedora.
- purgeObject ($pid, $log_message=NULL)

  Purge an object.
- validate ($pid, $as_of_date_time=NULL)

  Validate an object.
- **upload** ($file)

5.11.1 Detailed Description

This class implements the Fedora API-M interface.

This is a light wrapper around the Fedora interface. Very little attempt is put into putting things into native PHP datastructures.

See this page for more information: `https://wiki.duraspace.org/display/FEDORA35/REST+API`

5.11.2 Constructor & Destructor Documentation

5.11.2.1 FedoraApiM::__construct (RepositoryConnection *$connection*, FedoraApiSerializer *$serializer*)

Constructor for the new FedoraApiM object.

Parameters

RepositoryConnection	*$connection*	Takes the Respository Connection object for the Respository this API should connect to.
FedoraApiSerializer	*$serializer*	Takes the serializer object to that will be used to serialze the XML Fedora returns.

5.11.3 Member Function Documentation

5.11.3.1 FedoraApiM::addDatastream (*$pid*, *$dsid*, *$type*, *$file*, *$params*)

Add a new datastream to a fedora object.

The datastreams are sent to Fedora using a multipart post if a string or file is provided otherwise Fedora will go out and fetch the URL

Parameters

string	*$pid*	Persistent identifier of the digital object.
string	*$dsid*	Datastream identifier.
string	*$type*	This parameter tells the function what type of arguement is given for file. It must be one of: • string: The datastream is passed as a string. • file: The datastream is contained in a file. • url: The datastream is located at a URL, which is passed as a string. this is the only option that can be used for R and E type datastreams.
string	*$file*	This parameter depends on what is selected for $type. • string: A string containing the datastream. • file: A string containing the file name that contains the datastream. The file name must be a full path. • url: A string containing the publically accessable URL that the datastream is located at.
	array()	$params (optional) An array that can have one or more of the following elements: • controlGroup: one of "X", "M", "R", or "E" (Inline ∗X∗ML, ∗M∗anaged Content, ∗R∗edirect, or ∗E∗xternal Referenced). Default: X. • altIDs: alternate identifiers for the datastream. A space seperated list of alternate identifiers for the datastream. • dsLabel: the label for the datastream. • versionable: enable versioning of the datastream (boolean). • dsState: one of "A", "I", "D" (∗A∗ctive, ∗I∗nactive, ∗D∗eleted). • formatURI: the format URI of the datastream. • checksumType: the algorithm used to compute the checksum. One of DEFAULT, DISABLED, MD5, SHA-1, SHA-256, SHA-385, SHA-512. • checksum: the value of the checksum represented as a hexadecimal string. • mimeType: the MIME type of the content being added, this overrides the Content-Type request header. • logMessage: a message describing the activity being performed.

Exceptions

RepositoryException	

Returns

array() Returns an array describing the new datastream. This is the same array returned by getDatastream. This may also contain an dsAltID key, that contains any alternate ids if any are specified.

```
Array
(
    [dsLabel] =>
    [dsVersionID] => test.3
    [dsCreateDate] => 2012-03-07T18:03:38.679Z
    [dsState] => A
    [dsMIME] => text/xml
    [dsFormatURI] =>
    [dsControlGroup] => M
    [dsSize] => 22
    [dsVersionable] => true
```

```
        [dsInfoType] =>
        [dsLocation] => islandora:strict_pdf+test+test.3
        [dsLocationType] => INTERNAL_ID
        [dsChecksumType] => DISABLED
        [dsChecksum] => none
        [dsLogMessage] =>
    )
```

See Also

FedoraApiM::getDatastream

5.11.3.2 FedoraApiM::addRelationship (*$pid*, *$relationship*, *$is_literal*, *$datatype* = `NULL`)

Add a RDF relationship to a Fedora object.

Parameters

string	*$pid*	Persistent identifier of the digital object.
array	*$relationship*	An array containing the subject, predicate and object for the relationship. • subject: (optional) Subject of the relationship. Either a URI for the object or one of its datastreams. If none is given then the URI for the current object is used. • predicate: Predicate of the relationship. • object: Object of the relationship.
boolean	*$is_literal*	true if the object of the relationship is a literal, false if it is a URI
string	*$datatype*	(optional) if the object is a literal, the datatype of the literal.

Exceptions

RepositoryException	

See Also

FedoraApiM::getRelationships
FedoraApiM::purgeRelationships

5.11.3.3 FedoraApiM::export (*$pid*, *$params* = `array()`)

Export a Fedora object with the given PID.

Parameters

string	*$pid*	Persistent identifier of the digital object.
array	*$params*	(optional) An array that can have one or more of the following elements: • format: The XML format to export. One of info:fedora/fedora-system:FOXML-1.1 (default), info:fedora/fedora-system:FOXML-1.0, info:fedora/fedora-system:METSFedoraExt-1.1, info:fedora/fedora-system:METSFedoraExt-1.0, info:fedora/fedora-system:ATOM-1.1, info:fedora/fedora-system:ATOMZip-1.1 • context: The export context, which determines how datastream URLs and content are represented. Options: public (default), migrate, archive. • encoding: The preferred encoding of the exported XML.

Exceptions

RepositoryException	

Returns

string A string containing the requested XML.

5.11.3.4 FedoraApiM::getDatastream (*$pid, $dsid, $params* = `array()`)

Returns information about the datastream.

Parameters

string	*$pid*	Persistent identifier of the digital object.
string	*$dsid*	Datastream identifier.
array	*$params*	(optional) An array that can have one or more of the following elements: • asOfDateTime: Indicates that the result should be relative to the digital object as it existed on the given date. • validateChecksum: verifies that the Datastream content has not changed since the checksum was initially computed.

Exceptions

RepositoryException	

Returns

array() An array containing information about the datastream. This may also contains a key dsAltID which contains alternate ids if any are specified.

```
Array
(
    [dsLabel] =>
    [dsVersionID] => test.3
    [dsCreateDate] => 2012-03-07T18:03:38.679Z
    [dsState] => A
    [dsMIME] => text/xml
    [dsFormatURI] =>
    [dsControlGroup] => M
    [dsSize] => 22
    [dsVersionable] => true
    [dsInfoType] =>
    [dsLocation] => islandora:strict_pdf+test+test.3
    [dsLocationType] => INTERNAL_ID
    [dsChecksumType] => DISABLED
    [dsChecksum] => none
    [dsChecksumValid] => true
)
```

5.11.3.5 FedoraApiM::getDatastreamHistory (*$pid, $dsid*)

Get information on the different versions of a datastream that are avilable in Fedora.

Parameters

string	*$pid*	Persistent identifier of the digital object.
string	*$dsid*	Datastream identifier.

Exceptions

RepositoryException	

Returns

array Returns a indexed array with the same keys as getDatastream.

```
Array
(
    [0] => Array
        (
            [dsLabel] =>
            [dsVersionID] => test.3
            [dsCreateDate] => 2012-03-07T18:03:38.679Z
            [dsState] => A
            [dsMIME] => text/xml
            [dsFormatURI] =>
            [dsControlGroup] => M
            [dsSize] => 22
            [dsVersionable] => true
            [dsInfoType] =>
            [dsLocation] => islandora:strict_pdf+test+test.3
            [dsLocationType] => INTERNAL_ID
            [dsChecksumType] => DISABLED
            [dsChecksum] => none
        )

    [1] => Array
        (
            [dsLabel] =>
            [dsVersionID] => test.2
            [dsCreateDate] => 2012-03-07T18:03:13.722Z
            [dsState] => A
            [dsMIME] => text/xml
            [dsFormatURI] =>
            [dsControlGroup] => M
            [dsSize] => 22
            [dsVersionable] => true
            [dsInfoType] =>
            [dsLocation] => islandora:strict_pdf+test+test.2
            [dsLocationType] => INTERNAL_ID
            [dsChecksumType] => DISABLED
            [dsChecksum] => none
        )

)
```

See Also

FedoraApiM::getDatastream

5.11.3.6 FedoraApiM::getNextPid (*$namespace* = NULL, *$numpids* = NULL)

Get a new unused PID.

Parameters

string	*$namespace*	The namespace to get the PID in. This defaults to default namespace of the repository. This should not contain the PID seperator, for example it should be islandora not islandora:.
int	*$numpids*	The number of pids being requested.

Exceptions

RepositoryException	

Returns

array/string If one pid is requested it is returned as a string. If multiple pids are requested they they are returned in an array containg strings.

```
Array
(
    [0] => test:7
    [1] => test:8
)
```

5.11.3.7 FedoraApiM::getObjectXml (*$pid*)

Get the Fedora Objects XML (Foxml)

Parameters

string	*$pid*	Persistent identifier of the digital object.

Exceptions

RepositoryException	

Returns

string A string containing the objects foxml

See Also

FedoraApiM::export

5.11.3.8 FedoraApiM::getRelationships (*$pid*, *$relationship* = array())

Query relationships for a particular fedora object.

Parameters

string	*$pid*	Persistent identifier of the digital object.
array	*$relationship*	(Optional) An array defining the relationship: • subject: subject of the relationship(s). Either a URI for the object or one of its datastreams. defaults to the URI of the object. • predicate: predicate of the relationship(s), if missing returns all predicates.

Exceptions

RepositoryException	

Returns

array An indexed array with all the relationships.

```
Array
(
    [0] => Array
        (
            [subject] => islandora:strict_pdf
```

```
                [predicate] => Array
                    (
                        [predicate] => hasModel
                        [uri] => info:fedora/fedora-system:def/model#
                        [alias] =>
                    )

                [object] => Array
                    (
                        [literal] => FALSE
                        [value] => fedora-system:FedoraObject-3.0
                    )

            )

        [1] => Array
            (
                [subject] => islandora:strict_pdf
                [predicate] => Array
                    (
                        [predicate] => bar
                        [uri] => http://woot/foo#
                        [alias] =>
                    )

                [object] => Array
                    (
                        [literal] => TRUE
                        [value] => thedude
                    )

            )

    )
```

5.11.3.9 FedoraApiM::ingest (*$params* = `array()`)

Create a new object in Fedora.

This could be ingesting a XML file as a string or a file. Executing this request with no XML file content will result in the creation of a new, empty object (with either the specified PID or a system-assigned PID). The new object will contain only a minimal DC datastream specifying the dc:identifier of the object.

Parameters

array	*$params*	(optional) An array that can have one or more of the following elements: • pid: persistent identifier of the object to be created. If this is not supplied then either a new PID will be created for this object or the PID to be used is encoded in the XML included as the body of the request • string: The XML file defining the new object as a string • file: The XML file defining the new object as a string containing the full path to the XML file. This must not be used with the string parameter • label: the label of the new object • format: the XML format of the object to be ingested. One of info:fedora/fedora-system:FOXML-1.1, info:fedora/fedora-system:FOXML-1.0, info:fedora/fedora-system:METSFedoraExt-1.1, info:fedora/fedora-system:METSFedoraExt-1.0, info:fedora/fedora-system:ATOM-1.1, info::fedora/fedora-system:ATOMZip-1.1 • encoding: the encoding of the XML to be ingested. If this is specified, and given as anything other than UTF-8, you must ensure that the same encoding is declared in the XML. • namespace: The namespace to be used to create a PID for a new empty object: if a 'string' parameter is included with the request, the namespace parameter is ignored. • ownerId: the id of the user to be listed at the object owner. • logMessage: a message describing the activity being performed.

Exceptions

RepositoryException	

Returns

string The PID of the newly created object.

Todo This function is a problem in Fedora < 3.5 where ownerId does not properly get set. `https://jira.duraspace.org/browse/FCREPO-963`. We should deal with this.

5.11.3.10 FedoraApiM::modifyDatastream (*$pid*, *$dsid*, *$params* = `array()`)

Update a datastream.

Either changing its metadata, updaing the datastream contents or both.

Parameters

string	*$pid*	Persistent identifier of the digital object.
string	*$dsid*	Datastream identifier.
array	*$params*	(optional) An array that can have one or more of the following elements: • dsFile: String containing the full path to a file that will be used as the new contents of the datastream. • dsString: String containing the new contents of the datastream. • dsLocation: String containing a URL to fetch the new datastream from. Only ONE of dsFile, dsString or dsLocation should be used. • altIDs: alternate identifiers for the datastream. This is a space seperated string of alternate identifiers for the datastream. • dsLabel: the label for the datastream. • versionable: enable versioning of the datastream. • dsState: one of "A", "I", "D" (∗A∗ctive, ∗I∗nactive, ∗D∗eleted) • formatURI: the format URI of the datastream • checksumType: the algorithm used to compute the checksum. This has to be one of: DEFAULT, DISABLED, MD5, SHA-1, SHA-256, SHA-384, SHA-512. If this parameter is given and no checksum is given the checksum will be computed. • checksum: the value of the checksum represented as a hexadecimal string. This checksum must be computed by the algorithm defined above. • mimeType: the MIME type of the content being added, this overrides the Content-Type request header. • logMessage: a message describing the activity being performed • lastModifiedDate: date/time of the last (known) modification to the datastream, if the actual last modified date is later, a 409 response is returned. This can be used for opportunistic object locking.

Exceptions

RepositoryException	

Returns

array An array contianing information about the updated datastream. This array is the same as the array returned by getDatastream.

See Also

FedoraApiM::getDatastream

5.11.3.11 FedoraApiM::modifyObject (*$pid*, *$params* = `NULL`)

Update Fedora Object parameters.

Parameters

string	*$pid*	Persistent identifier of the digital object.
array	*$params*	(optional) An array that can have one or more of the following elements: • label: object label. • ownerId: the id of the user to be listed at the object owner. • state: the new object state - ∗A∗ctive, ∗I∗nactive, or ∗D∗eleted. • logMessage: a message describing the activity being performed. • lastModifiedDate: date/time of the last (known) modification to the datastream, if the actual last modified date is later, a 409 response is returned. This can be used for opportunistic object locking.

Exceptions

RepositoryException	

Returns

string A string containg the timestamp of the object modification.

5.11.3.12 FedoraApiM::purgeDatastream (*$pid*, *$dsid*, *$params* = `array()`)

Purge a datastream from from Fedora.

This perminatly removes the datastream and all its associated data.

Parameters

string	*$pid*	Persistent identifier of the digital object.
string	*$dsid*	Datastream identifier
array	*$params*	(optional) An array that can have one or more of the following elements: • startDT: the (inclusive) start date-time stamp of the range. If not specified, this is taken to be the lowest possible value, and thus, the entire version history up to the endDT will be purged. • endDT: the (inclusive) ending date-time stamp of the range. If not specified, this is taken to be the greatest possible value, and thus, the entire version history back to the startDT will be purged. • logMessage: a message describing the activity being performed.

Exceptions

RepositoryException	

Returns

array An array containing the timestamps of the datastreams that were removed.

```
Array
(
    [0] => 2012-03-08T18:44:15.214Z
    [1] => 2012-03-08T18:44:15.336Z
)
```

5.11.3.13 FedoraApiM::purgeObject (*$pid*, *$log_message* = NULL)

Purge an object.

Parameters

string	*$pid*	Persistent identifier of the digital object.
string	*$log_message*	(optional) A message describing the activity being performed.

Exceptions

RepositoryException	

Returns

string Timestamp when object was deleted.

5.11.3.14 FedoraApiM::validate (*$pid*, *$as_of_date_time* = NULL)

Validate an object.

Parameters

string	*$pid*	Persistent identifier of the digital object.
array	*$as_of_date_time*	(optional) Indicates that the result should be relative to the digital object as it existed at the given date and time. Defaults to the most recent version.

Exceptions

RepositoryException	

Returns

array An array containing the validation results.

```
Array
(
    [valid] => false
    [contentModels] => Array
        (
            [0] => "info:fedora/fedora-system:FedoraObject-3.0"
        )
    [problems] => Array
        (
            [0] => "Problem description"
        )
    [datastreamProblems] => Array
```

```
            (
                [dsid] => Array
                (
                    [0] => "Problem description"
                )
            )
    )
```

The documentation for this class was generated from the following file:

- FedoraApi.php

5.12 FedoraApiSerializer Class Reference

A class to Serialize the XML responses from Fedora into PHP arrays.

Public Member Functions

- domDocumentExceptionHandler ($errno, $errstr, $errfile= ”, $errline=NULL, $errcontext=NULL)

  This is a simple exception handler, that will throw an exception if there is a problem parsing XML from Fedora.
- describeRepository ($request)

  Serializes the data returned in FedoraApiA::describeRepository()
- findObjects ($request)

  Serializes the data returned in FedoraApiA::findObjects()
- resumeFindObjects ($request)

  Serializes the data returned in FedoraApiA::resumeFindObjects()
- getDatastreamDissemination ($request, $file)

  Serializes the data returned in FedoraApiA::getDatastreamDissemination()
- getDissemination ($request)

  Serializes the data returned in FedoraApiA::getDissemination()
- getObjectHistory ($request)

  Serializes the data returned in FedoraApiA::getObjectHistory()
- getObjectProfile ($request)

  Serializes the data returned in FedoraApiA::getObjectProfile()
- listDatastreams ($request)

  Serializes the data returned in FedoraApiA::listDatastreams()
- listMethods ($request)

  Serializes the data returned in FedoraApiA::listMethods()
- addDatastream ($request)

  Serializes the data returned in FedoraApiM::addDatastream()
- addRelationship ($request)

  Serializes the data returned in FedoraApiM::addRelationship()
- export ($request)

  Serializes the data returned in FedoraApiM::export()
- getDatastream ($request)

  Serializes the data returned in FedoraApiM::getDatastream()
- getDatastreamHistory ($request)

  Serializes the data returned in FedoraApiM::getDatastreamHistory()
- getNextPid ($request)

  Serializes the data returned in FedoraApiM::getNextPid()
- getObjectXml ($request)

  Serializes the data returned in FedoraApiM::getObjectXml()

- getRelationships ($request)

    Serializes the data returned in FedoraApiM::getRelationships()
- ingest ($request)

    Serializes the data returned in FedoraApiM::ingest()
- modifyDatastream ($request)

    Serializes the data returned in FedoraApiM::modifyDatastream()
- modifyObject ($request)

    Serializes the data returned in FedoraApiM::modifyObject()
- purgeDatastream ($request)

    Serializes the data returned in FedoraApiM::purgeDatastream()
- purgeObject ($request)

    Serializes the data returned in FedoraApiM::purgeObject()
- validate ($request)

    Serializes the data returned in FedoraApiM::validate()
- **upload** ($request)

Protected Member Functions

- loadSimpleXml ($xml)

    Simple function that takes an XML string and returns a SimpleXml object.
- loadDomDocument ($xml)

    Simple function that takes an XML string and returns a DomDocument object.
- flattenDocument ($xml, $make_array=array())

    Flatten a simplexml object returning an array containing the text from the XML.

5.12.1 Detailed Description

A class to Serialize the XML responses from Fedora into PHP arrays.

5.12.2 Member Function Documentation

5.12.2.1 FedoraApiSerializer::domDocumentExceptionHandler (*$errno*, *$errstr*, *$errfile* = ' ', *$errline* = NULL, *$errcontext* = NULL)

This is a simple exception handler, that will throw an exception if there is a problem parsing XML from Fedora.

This is nice so we can catch it.

Parameters

int	*$errno*	The error number
string	*$errstr*	String describing an error.
string	*$errfile*	(optional) The third parameter is optional, errfile, which contains the filename that the error was raised in, as a string.
int	*$errline*	(optional) The fourth parameter is optional, errline, which contains the line number the error was raised at, as an integer.
array	*$errcontext*	(optional) The fifth parameter is optional, errcontext, which is an array that points to the active symbol table at the point the error occurred. In other words, errcontext will contain an array of every variable that existed in the scope the error was triggered in. User error handler must not modify error context.

Returns

boolean TRUE if we through an exception FALSE otherwise

See Also

php.net/manual/en/function.set-error-handler.php

5.12.2.2 FedoraApiSerializer::flattenDocument (*$xml*, *$make_array* = `array()`) `[protected]`

Flatten a simplexml object returning an array containing the text from the XML.

This is often used to get data back from fedora. It also makes sure to cast everything to string.

Parameters

SimpleXmlElement	*$xml*	The simplexml elemnt to be processed.
	array()	$make_array (optional) This parameter specifies tags that should become an array instead of an element in an array. This is used to get consistant values for things that are multivalued when there is only one value returned.

Returns

array An array representation of the XML.

5.12.2.3 FedoraApiSerializer::loadDomDocument (*$xml*) `[protected]`

Simple function that takes an XML string and returns a DomDocument object.

It makes sure no PHP errors or warnings are issued and instead throws an exception if the XML parse failed.

Parameters

string	*$xml*	The XML as a string

Exceptions

RepositoryXmlError	

Returns

DomDocument Return an istantiated DomDocument

5.12.2.4 FedoraApiSerializer::loadSimpleXml (*$xml*) `[protected]`

Simple function that takes an XML string and returns a SimpleXml object.

It makes sure no PHP errors or warnings are issued and instead throws an exception if the XML parse failed.

Parameters

string	*$xml*	The XML as a string

Exceptions

RepositoryXmlError	

Returns

SimpleXmlElement Return an istantiated simplexml

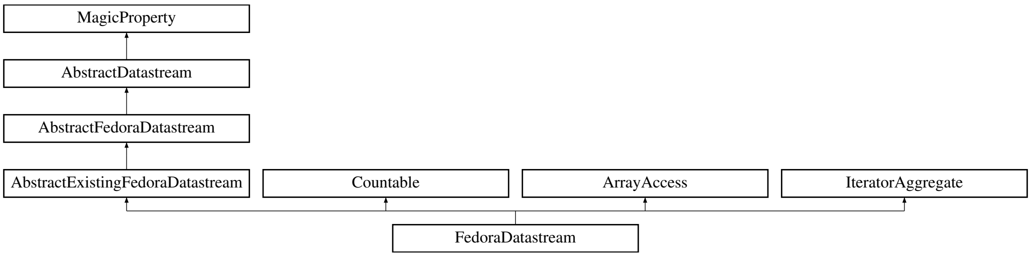
The documentation for this class was generated from the following file:

- FedoraApiSerializer.php

5.13 FedoraDatastream Class Reference

This class implements a fedora datastream.

Inheritance diagram for FedoraDatastream:



Public Member Functions

- __construct ($id, FedoraObject $object, FedoraRepository $repository, array $datastream_info=NULL)

    Domo arigato, Mr.
- refresh ()

    This function clears the datastreams caches, so everything will be requested directly from fedora again.
- setContentFromFile ($file)
- setContentFromUrl ($url)
- setContentFromString ($string)
- count ()
- offsetExists ($offset)
- offsetGet ($offset)
- offsetSet ($offset, $value)
- offsetUnset ($offset)
- getIterator ()

    IteratorAggregate::getIterator()
- getContent ($file)

Public Attributes

- **$forceUpdate** = FALSE

Protected Member Functions

- populateDatastreamHistory ()

  This populates datastream history if it needs to be populated.
- populateDatastreamInfo ()

  This function uses datastream history to populate datastream info.
- modifyDatastream (array $args)

  This function modifies the datastream in fedora while adding the parameters needed to respect datastram locking and making sure that we keep the internal class cache of the datastream up to date.
- controlGroupMagicProperty ($function, $value)
- locationMagicProperty ($function, $value)
- stateMagicProperty ($function, $value)
- labelMagicProperty ($function, $value)
- versionableMagicProperty ($function, $value)
- mimetypeMagicProperty ($function, $value)
- formatMagicProperty ($function, $value)
- sizeMagicProperty ($function, $value)
- checksumMagicProperty ($function, $value)
- checksumTypeMagicProperty ($function, $value)
- createdDateMagicProperty ($function, $value)
- contentMagicProperty ($function, $value)
- urlMagicProperty ($function, $value)
- logMessageMagicProperty ($function, $value)

Protected Attributes

- **$datastreamHistory** = NULL

5.13.1 Detailed Description

This class implements a fedora datastream.

It also lets old versions of datastreams be accessed using array notation. For example to see how many versions of a datastream there are:

```
count($datastream)
```

Old datastreams are indexed newest to oldest. The current version is always index 0, and older versions are indexed from that. Old versions can be discarded using the unset command.

These functions respect datastream locking. If a datastream changes under your feet then an exception will be raised.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 FedoraDatastream::__construct (*$id,* FedoraObject *$object,* FedoraRepository *$repository,* array *$datastream_info* = NULL)

Domo arigato, Mr.

Roboto. Constructor.

5.13.3 Member Function Documentation

5.13.3.1 FedoraDatastream::checksumMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::checksum

Todo maybe add functionality to set it to auto

5.13.3.2 FedoraDatastream::checksumTypeMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::checksumType

5.13.3.3 FedoraDatastream::contentMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::content

Todo We should perhaps cache this? depending on size?

5.13.3.4 FedoraDatastream::controlGroupMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::controlGroup

5.13.3.5 FedoraDatastream::count ( )

See Also

Countable::count

5.13.3.6 FedoraDatastream::createdDateMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::createdDate

5.13.3.7 FedoraDatastream::formatMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::format

5.13.3.8 FedoraDatastream::getContent (*$file*)

See Also

AbstractDatastream::getContent()

5.13.3.9 FedoraDatastream::labelMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::label

5.13.3.10 FedoraDatastream::locationMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::location

5.13.3.11 FedoraDatastream::logMessageMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::logMessage

5.13.3.12 FedoraDatastream::mimetypeMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::mimetype

5.13.3.13 FedoraDatastream::offsetExists (*$offset*)

See Also

ArrayAccess::offsetExists

5.13.3.14 FedoraDatastream::offsetGet (*$offset*)

See Also

ArrayAccess::offsetGet

5.13.3.15 FedoraDatastream::offsetSet (*$offset*, *$value*)

See Also

ArrayAccess::offsetSet

5.13.3.16 FedoraDatastream::offsetUnset (*$offset*)

See Also

ArrayAccess::offsetUnset

5.13.3.17 FedoraDatastream::setContentFromFile (*$file*)

See Also

AbstractDatastream::setContentFromFile

5.13.3.18 FedoraDatastream::setContentFromString (*$string*)

See Also

AbstractDatastream::setContentFromString

5.13.3.19 FedoraDatastream::setContentFromUrl (*$url*)

See Also

AbstractDatastream::setContentFromUrl

5.13.3.20 FedoraDatastream::sizeMagicProperty (*$function, $value*) [protected]

See Also

AbstractDatastream::size

5.13.3.21 FedoraDatastream::stateMagicProperty (*$function, $value*) [protected]

See Also

AbstractDatastream::state

5.13.3.22 FedoraDatastream::urlMagicProperty (*$function, $value*) [protected]

See Also

AbstractDatastream::url

5.13.3.23 FedoraDatastream::versionableMagicProperty (*$function, $value*) [protected]

See Also

AbstractDatastream::versionable

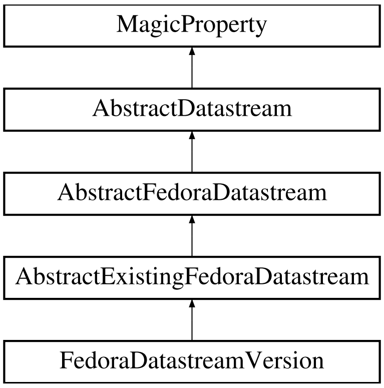
The documentation for this class was generated from the following file:

- Datastream.php

5.14 FedoraDatastreamVersion Class Reference

This class implements an old version of a fedora datastream.

Inheritance diagram for FedoraDatastreamVersion:



Public Member Functions

- __construct ($id, array $datastream_info, FedoraDatastream $datastream, FedoraObject $object, FedoraRepository $repository)

    The Constructor! Sounds like a superhero doesn't it.
- setContentFromFile ($file)
- setContentFromString ($string)
- setContentFromUrl ($url)
- getContent ($file)

Public Attributes

- **$parent**

Protected Member Functions

- error ()

    This function gives us a consistant error across this whole clas.
- generalReadOnly ($offset, $unset_val, $function, $value)

    Since this whole class is read only, this is a general implementation of the MagicPropery function that is ready only.
- stateMagicProperty ($function, $value)
- labelMagicProperty ($function, $value)
- versionableMagicProperty ($function, $value)
- mimetypeMagicProperty ($function, $value)
- formatMagicProperty ($function, $value)
- sizeMagicProperty ($function, $value)
- checksumMagicProperty ()
- createdDateMagicProperty ($function, $value)
- contentMagicProperty ($function, $value)
- logMessageMagicProperty ($function, $value)

Additional Inherited Members

5.14.1 Detailed Description

This class implements an old version of a fedora datastream.

Its properties are the same of a normal fedora datastream, except since its an older verion everything is read only.

5.14.2 Constructor & Destructor Documentation

5.14.2.1 FedoraDatastreamVersion::__construct (*$id*, array *$datastream_info*, **FedoraDatastream** *$datastream*, **FedoraObject** *$object*, **FedoraRepository** *$repository*)

The Constructor! Sounds like a superhero doesn't it.

Constructor away!

5.14.3 Member Function Documentation

5.14.3.1 FedoraDatastreamVersion::checksumMagicProperty ( ) [protected]

See Also

AbstractDatastream::checksum

5.14.3.2 FedoraDatastreamVersion::contentMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::content

5.14.3.3 FedoraDatastreamVersion::createdDateMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::createDate

5.14.3.4 FedoraDatastreamVersion::formatMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::format

5.14.3.5 FedoraDatastreamVersion::getContent (*$file*)

See Also

AbstractDatastream::getContent()

5.14.3.6 FedoraDatastreamVersion::labelMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::label

5.14.3.7 FedoraDatastreamVersion::logMessageMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::logMessage

5.14.3.8 FedoraDatastreamVersion::mimetypeMagicProperty (*$function,* *$value*) [protected]

See Also

AbstractDatastream::mimetype

5.14.3.9 FedoraDatastreamVersion::setContentFromFile (*$file*)

See Also

AbstractDatastream::setContentFromFile()

5.14.3.10 FedoraDatastreamVersion::setContentFromString (*$string*)

See Also

AbstractDatastream::setContentFromString()

5.14.3.11 FedoraDatastreamVersion::setContentFromUrl (*$url*)

See Also

AbstractDatastream::setContentFromUrl()

5.14.3.12 FedoraDatastreamVersion::sizeMagicProperty (*$function,* *$value*) [protected]

See Also

AbstractDatastream::size

5.14.3.13 FedoraDatastreamVersion::stateMagicProperty (*$function,* *$value*) [protected]

See Also

AbstractDatastream::state

5.14.3.14 FedoraDatastreamVersion::versionableMagicProperty (*$function,* *$value*) [protected]

See Also

AbstractDatastream::versionable

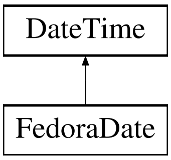
The documentation for this class was generated from the following file:

- Datastream.php

5.15 FedoraDate Class Reference

This file represents a Fedora Date.

Inheritance diagram for FedoraDate:



Public Member Functions

- __toString ()

  Get the date in a format that Fedora can use.
- __construct ($time)

  Instantiate FedoraDate.
- __sleep ()

  Serialize this object.
- __wakeup ()

  Unserialize this object.

5.15.1 Detailed Description

This file represents a Fedora Date.

It wraps the PHP datetime functions specifically for the date format used in Fedora. This allows the easy comparison of dates for example.

5.15.2 Constructor & Destructor Documentation

5.15.2.1 FedoraDate::__construct (*$time*)

Instantiate FedoraDate.

Parameters

string	*$time*	The date as you would pass to create a DateTime object.

5.15.3 Member Function Documentation

5.15.3.1 FedoraDate::__sleep ( )

Serialize this object.

Returns

array The class members to be serialized.

5.15.3.2 FedoraDate::__toString ( )

Get the date in a format that Fedora can use.

Returns

string A fedora formated iso8601 date.

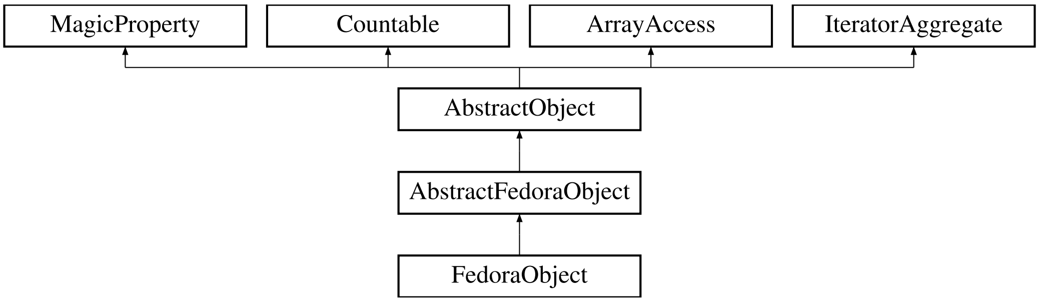
The documentation for this class was generated from the following file:

- FedoraDate.php

5.16 FedoraObject Class Reference

Represents a Fedora Object.

Inheritance diagram for FedoraObject:



Public Member Functions

- __construct ($id, FedoraRepository $repository)

  The class constructor.
- refresh ()

  This function clears the object cache, so everything will be requested directly from fedora again.
- purgeDatastream ($id)

  Purge a datastream.
- getDatastream ($id)
- constructDatastream ($id, $control_group= 'M')
- ingestDatastream (&$ds)
- count ()
- offsetExists ($offset)
- offsetGet ($offset)
- offsetSet ($offset, $value)
- offsetUnset ($offset)
- getIterator ()

  IteratorAggregate::getIterator()
- getParents ()

  Returns IDs of collections of which object is a member.

Public Attributes

- **$forceUpdate** = FALSE

Protected Member Functions

- populateDatastreams ()

  this will populate the datastream list the first time it is needed.
- modifyObject ($params)

  This is a wrapper on APIM modify object to make sure we respect locking.
- stateMagicPropertySet ($value)
- labelMagicPropertySet ($value)

- ownerMagicPropertySet ($value)
- createdDateMagicProperty ($function, $value)
- lastModifiedDateMagicProperty ($function, $value)
- logMessageMagicPropertySet ($value)

Protected Attributes

- **$datastreams** = NULL

5.16.1 Detailed Description

Represents a Fedora Object.

Should be created by the factory method in the repository. Respects object locking. Will throw an exception if an object is modified under us.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 FedoraObject::__construct (*$id*, **FedoraRepository** *$repository*)

The class constructor.

Should be instantiated by the repository.

5.16.3 Member Function Documentation

5.16.3.1 FedoraObject::constructDatastream (*$id*, *$control_group* = 'M')

See Also

AbstractObject::constructDatastream()

5.16.3.2 FedoraObject::count ( )

See Also

Countable::count

5.16.3.3 FedoraObject::createdDateMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractObject::createdDate

5.16.3.4 FedoraObject::getDatastream (*$id*)

See Also

AbstractObject::getDatastream()

5.16.3.5 FedoraObject::getParents ( )

Returns IDs of collections of which object is a member.

Returns

array

5.16.3.6 FedoraObject::ingestDatastream (& *$ds*)

See Also

AbstractObject::ingestDatastream()

5.16.3.7 FedoraObject::labelMagicPropertySet (*$value*) [protected]

See Also

AbstractObject::label

5.16.3.8 FedoraObject::lastModifiedDateMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractObject::lastModifiedDate

5.16.3.9 FedoraObject::logMessageMagicPropertySet (*$value*) [protected]

See Also

AbstractObject::logMessage

5.16.3.10 FedoraObject::offsetExists (*$offset*)

See Also

ArrayAccess::offsetExists

5.16.3.11 FedoraObject::offsetGet (*$offset*)

See Also

ArrayAccess::offsetGet

5.16.3.12 FedoraObject::offsetSet (*$offset*, *$value*)

See Also

ArrayAccess::offsetSet

5.16.3.13 FedoraObject::offsetUnset (*$offset*)

See Also

ArrayAccess::offsetUnset

5.16.3.14 FedoraObject::ownerMagicPropertySet (*$value*) `[protected]`

See Also

AbstractObject::owner

5.16.3.15 FedoraObject::purgeDatastream (*$id*)

Purge a datastream.

Parameters

string	*$id*	The id of hte datastream to purge.

Returns

boolean Returns TRUE on success and FALSE on failure.

5.16.3.16 FedoraObject::stateMagicPropertySet (*$value*) `[protected]`

See Also

AbstractObject::state

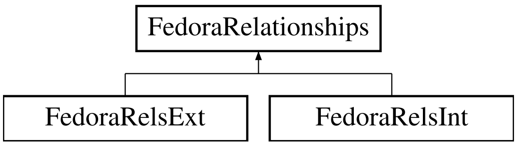
The documentation for this class was generated from the following file:

- Object.php

5.17 FedoraRelationships Class Reference

This is the base class for Fedora Relationships.

Inheritance diagram for FedoraRelationships:



Public Member Functions

- __construct (array $namespaces=NULL)

  The constructor.
- registerNamespace ($alias, $uri)

  Add a new namespace to the relationship xml.
- changeObjectID ($id)

  This function allows you to change the ID referenced in the rdf:about attribute.

Public Attributes

- **$datastream** = NULL

Protected Member Functions

- getXpath ($document)

  This function returns a domXPath object with all the current namespaces already registered.
- getDom ()

  Sets up a domdocument for the functions.
- updateDatastream ($document)

  This updates the associated datastreams content.
- internalAdd ($subject, $predicate_uri, $predicate, $object, $type=RELS_TYPE_URI)

  Add a new relationship.
- getXpathResults ($xpath_object, $subject, $predicate_uri, $predicate, $object, $literal)

  This function is used to create an xpath expression based on the input.
- internalGet ($subject, $predicate_uri=NULL, $predicate=NULL, $object=NULL, $literal=FALSE)

  This function queries the relationships in the assocaited datastream.
- internalRemove ($subject, $predicate_uri, $predicate, $object, $literal=FALSE)

  This function removes relationships that match the pattern from the datastream.

Protected Attributes

- **$namespaces**

5.17.1 Detailed Description

This is the base class for Fedora Relationships.

Todo potentially we should validate the predicate URI

5.17.2 Constructor & Destructor Documentation

5.17.2.1 FedoraRelationships::__construct (array *$namespaces* = `NULL`)

The constructor.

This will usually be called by one of its subclasses.

Parameters

array	*$namespaces*	An array of default namespaces.

5.17.3 Member Function Documentation

5.17.3.1 FedoraRelationships::changeObjectID (*$id*)

This function allows you to change the ID referenced in the rdf:about attribute.

This allows the updating of all the about attribures if the datastream is being attached to another object.

Parameters

string	*$id*	The new ID

5.17.3.2 FedoraRelationships::getDom ( ) `[protected]`

Sets up a domdocument for the functions.

Returns

DomDocument The domdocument to modify

5.17.3.3 FedoraRelationships::getXpath (*$document*) `[protected]`

This function returns a domXPath object with all the current namespaces already registered.

Returns

DomXPath The object

5.17.3.4 FedoraRelationships::getXpathResults (*$xpath_object, $subject, $predicate_uri, $predicate, $object, $literal*) `[protected]`

This function is used to create an xpath expression based on the input.

Returns

DOMNodeList The node list

5.17.3.5 FedoraRelationships::internalAdd (*$subject, $predicate_uri, $predicate, $object, $type* = `RELS_TYPE_URI`) `[protected]`

Add a new relationship.

Parameters

string	*$subject*	The subject. This can be a PID, or a PID/DSID combo. This string does not contain the info:fedora/ part of the URI this is added automatically.
string	*$predicate_uri*	The URI to use as the namespace of the predicate. If you would like the XML to use a prefix instead of the full predicate call the FedoraRelationships::registerNamespace() function first.
string	*$predicate*	The predicate tag to add.
string	*$object*	The object for the relationship that is being created.
boolean	*$literal*	Specifies if the object is a literal or not.

5.17.3.6 FedoraRelationships::internalGet (*$subject, $predicate_uri* = `NULL`, *$predicate* = `NULL`, *$object* = `NULL`, *$literal* = `FALSE`) `[protected]`

This function queries the relationships in the assocaited datastream.

Any parameter except for $subject can be set to NULL to act as a wildcard. Calling with just $subject will return all relationships.

Parameters

string	*$subject*	The subject. This can be a PID, or a PID/DSID combo. This string does not contain the info:fedora/ part of the URI this is added automatically.
string	*$predicate_uri*	The URI to use as the namespace of the predicate. This is ignored if predicate is NULL.
string	*$predicate*	The predicate tag to filter by.
string	*$object*	The object for the relationship to filter by.
boolean	*$literal*	Defines if the $object is a literal or not.

Returns

array This returns an indexed array with all the matching relationships. The array is of the form:

```
Array
(
    [0] => Array
        (
            [predicate] => Array
                (
                    [value] => thepredicate
                    [alias] => thexmlprefix
                    [namespace] => http://crazycool.com#
                )

            [object] => Array
                (
                    [literal] => TRUE
                    [value] => test
                )

        )

)
```

5.17.3.7 FedoraRelationships::internalRemove (*$subject*, *$predicate_uri*, *$predicate*, *$object*, *$literal* = FALSE) [protected]

This function removes relationships that match the pattern from the datastream.

Any parameter can be given as NULL which will make it a wildcard.

Parameters

string	*$subject*	The subject. This can be a PID, or a PID/DSID combo. This string does not contain the info:fedora/ part of the URI this is added automatically.
string	*$predicate_uri*	The URI to use as the namespace of the predicate. This is ignored if predicate is NULL.
string	*$predicate*	The predicate tag to filter removed results by.
string	*$object*	The object for the relationship to filter by.
boolean	*$literal*	Defines if the $object is a literal or not.

Returns

boolean TRUE if relationships were removed, FALSE otherwise.

5.17.3.8 FedoraRelationships::registerNamespace (*$alias*, *$uri*)

Add a new namespace to the relationship xml.

Doing this before adding new predicates with differnt URIs makes the XML look a little prettier.

Parameters

string	*$alias*	The alias to add.
string	*$uri*	The URI to associate with the alias.

5.17.4 Member Data Documentation

5.17.4.1 FedoraRelationships::$namespaces [protected]

Initial value:

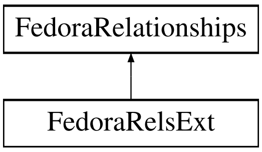
```
= array(
    'rdf' => RDF_URI,
  )
```

The documentation for this class was generated from the following file:

- FedoraRelationships.php

5.18 FedoraRelsExt Class Reference

Inheritance diagram for FedoraRelsExt:



Public Member Functions

- __construct (AbstractFedoraObject $object)

    Objects Construct!
- add ($predicate_uri, $predicate, $object, $type=RELS_TYPE_URI)

    Add a new relationship.
- remove ($predicate_uri=NULL, $predicate=NULL, $object=NULL, $literal=FALSE)

    This function removes relationships that match the pattern from the datastream.
- get ($predicate_uri=NULL, $predicate=NULL, $object=NULL, $literal=FALSE)

    This function queries the relationships in the assocaited datastream.
- **changeObjectID** ($id)

Protected Member Functions

- initializeDatastream ()

    Initialize the datastrem that we are using.

Protected Attributes

- **$new** = FALSE
- **$initialized** = FALSE

Additional Inherited Members

5.18.1 Constructor & Destructor Documentation

5.18.1.1 FedoraRelsExt::__construct (AbstractFedoraObject *$object*)

Objects Construct!

Parameters

Abstract-Fedora-Object	*$object*	The object whose relationships we are manipulating

5.18.2 Member Function Documentation

5.18.2.1 FedoraRelsExt::add (*$predicate_uri*, *$predicate*, *$object*, *$type* = RELS_TYPE_URI)

Add a new relationship.

Parameters

string	*$predicate_uri*	The URI to use as the namespace of the predicate. If you would like the XML to use a prefix instead of the full predicate call the FedoraRelationships::registerNamespace() function first.
string	*$predicate*	The predicate tag to add.
string	*$object*	The object for the relationship that is being created.
boolean	*$literal*	Specifies if the object is a literal or not.

5.18.2.2 FedoraRelsExt::get (*$predicate_uri* = NULL, *$predicate* = NULL, *$object* = NULL, *$literal* = FALSE)

This function queries the relationships in the assocaited datastream.

Any parameter except for $subject can be set to NULL to act as a wildcard. Calling with just $subject will return all relationships.

Parameters

string	*$predicate_uri*	The URI to use as the namespace of the predicate. This is ignored if predicate is NULL.
string	*$predicate*	The predicate tag to filter by.
string	*$object*	The object for the relationship to filter by.
boolean	*$literal*	Defines if the $object is a literal or not.

Returns

array This returns an indexed array with all the matching relationships. The array is of the form:

```
Array
(
    [0] => Array
        (
            [predicate] => Array
                (
                    [value] => thepredicate
                    [alias] => thexmlprefix
                    [namespace] => http://crazycool.com#
                )

            [object] => Array
                (
                    [literal] => TRUE
```

```
                        [value] => test
                    )

            )

    )
```

5.18.2.3 FedoraRelsExt::initializeDatastream ( ) [protected]

Initialize the datastrem that we are using.

We use this function to delay this as long as possible, in case it never has be be called.

5.18.2.4 FedoraRelsExt::remove (*$predicate_uri* = NULL, *$predicate* = NULL, *$object* = NULL, *$literal* = FALSE)

This function removes relationships that match the pattern from the datastream.

Any parameter can be given as NULL which will make it a wildcard.

Parameters

string	*$predicate_uri*	The URI to use as the namespace of the predicate. This is ignored if predicate is NULL.
string	*$predicate*	The predicate tag to filter removed results by.
string	*$object*	The object for the relationship to filter by.
boolean	*$literal*	Defines if the $object is a literal or not.

Returns

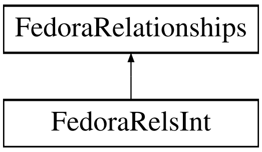
boolean TRUE if relationships were removed, FALSE otherwise.

The documentation for this class was generated from the following file:

- FedoraRelationships.php

5.19 FedoraRelsInt Class Reference

Inheritance diagram for FedoraRelsInt:



Public Member Functions

- __construct (AbstractFedoraDatastream $datastream)

  Objects Construct!
- add ($predicate_uri, $predicate, $object, $type=RELS_TYPE_URI)

  Add a new relationship.
- remove ($predicate_uri=NULL, $predicate=NULL, $object=NULL, $type=RELS_TYPE_URI)

  This function removes relationships that match the pattern from the datastream.
- get ($predicate_uri=NULL, $predicate=NULL, $object=NULL, $type=RELS_TYPE_URI)

  This function queries the relationships in the assocaited datastream.
- **changeObjectID** ($id)

Protected Member Functions

- initializeDatastream ()

  Delay initialization by waiting to set datastream with this function.

Protected Attributes

- **$new** = FALSE
- **$initialized** = FALSE
- **$aboutDs**

Additional Inherited Members

5.19.1 Constructor & Destructor Documentation

5.19.1.1 FedoraRelsInt::__construct (**AbstractFedoraDatastream** *$datastream*)

Objects Construct!

Parameters

AbstractFedoraObject	*$datastream*	The datastream whose relationships we are manipulating

5.19.2 Member Function Documentation

5.19.2.1 FedoraRelsInt::add (*$predicate_uri*, *$predicate*, *$object*, *$type* = RELS_TYPE_URI)

Add a new relationship.

Parameters

string	*$predicate_uri*	The URI to use as the namespace of the predicate. If you would like the XML to use a prefix instead of the full predicate call the FedoraRelationships::registerNamespace() function first.
string	*$predicate*	The predicate tag to add.
string	*$object*	The object for the relationship that is being created.
string	*$type*	Specifies if the object is a literal or if not, what the atttribute type should be.

5.19.2.2 FedoraRelsInt::get (*$predicate_uri* = NULL, *$predicate* = NULL, *$object* = NULL, *$type* = RELS_TYPE_URI)

This function queries the relationships in the assocaited datastream.

Any parameter except for $subject can be set to NULL to act as a wildcard. Calling with just $subject will return all relationships.

Parameters

string	*$predicate_uri*	The URI to use as the namespace of the predicate. This is ignored if predicate is NULL.
string	*$predicate*	The predicate tag to filter by.
string	*$object*	The object for the relationship to filter by.
string	*$type*	Specifies if the object is a literal or if not, what the atttribute type should be..

Returns

array This returns an indexed array with all the matching relationships. The array is of the form:

```
Array
(
    [0] => Array
        (
            [predicate] => Array
                (
                    [value] => thepredicate
                    [alias] => thexmlprefix
                    [namespace] => http://crazycool.com#
                )

            [object] => Array
                (
                    [literal] => TRUE
                    [value] => test
                )

        )

)
```

5.19.2.3 FedoraRelsInt::remove (*$predicate_uri* = NULL, *$predicate* = NULL, *$object* = NULL, *$type* = RELS_TYPE_URI)

This function removes relationships that match the pattern from the datastream.

Any parameter can be given as NULL which will make it a wildcard.

Parameters

string	*$predicate_uri*	The URI to use as the namespace of the predicate. This is ignored if predicate is NULL.
string	*$predicate*	The predicate tag to filter removed results by.
string	*$object*	The object for the relationship to filter by.
string	*$type*	Specifies if the object is a literal or if not, what the atttribute type should be.

Returns

boolean TRUE if relationships were removed, FALSE otherwise.

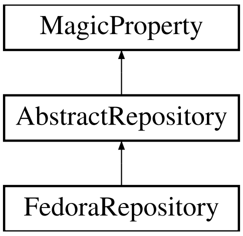
The documentation for this class was generated from the following file:

- FedoraRelationships.php

5.20 FedoraRepository Class Reference

Concrete implementation of the AbstractRepository for Fedora.

Inheritance diagram for FedoraRepository:



Public Member Functions

- __construct (FedoraApi $api, AbstractCache $cache)

  Constructor for the FedoraRepository Object.
- findObjects (array $search)
- constructObject ($id=NULL)
- ingestObject (NewFedoraObject &$object)
- getObject ($id)
- purgeObject ($id)

Public Attributes

- **$ri**
- **$api**

Protected Attributes

- **$cache**
- **$queryClass** = 'RepositoryQuery'
- **$newObjectClass** = 'NewFedoraObject'
- **$objectClass** = 'FedoraObject'

Additional Inherited Members

5.20.1 Detailed Description

Concrete implementation of the AbstractRepository for Fedora.

The parent class has more detailed documentation about how this class can be called as an Array.

See Also

AbstractRepository

5.20.2 Constructor & Destructor Documentation

5.20.2.1 FedoraRepository::__construct (FedoraApi *$api*, AbstractCache *$cache*)

Constructor for the FedoraRepository Object.

Parameters

FedoraApi	*$api*	An instantiated FedoraAPI which will be used to connect to the repository.
Abstract-Cache	*$cache*	An instantiated AbstractCache which will be used to cache fedora objects.

5.20.3 Member Function Documentation

5.20.3.1 FedoraRepository::constructObject (*$id* = NULL)

Todo validate the ID

catch the getNextPid errors

See Also

AbstractRepository::constructObject

5.20.3.2 FedoraRepository::findObjects (array *$search*)

See Also

AbstractRepository::findObjects

Todo this needs to be implemented!

5.20.3.3 FedoraRepository::getObject (*$id*)

See Also

AbstractRepository::getObject()

Todo perhaps we should check if an object exists instead of catching the exception

5.20.3.4 FedoraRepository::ingestObject (NewFedoraObject & *$object*)

See Also

AbstractRepository::ingestObject()

Todo error handling

5.20.3.5 FedoraRepository::purgeObject (*$id*)

See Also

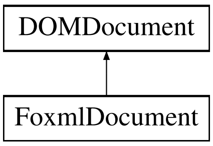
AbstractRepository::purgeObject()

The documentation for this class was generated from the following file:

- Repository.php

5.21 FoxmlDocument Class Reference

Inheritance diagram for FoxmlDocument:



Public Member Functions

- **__construct** (NewFedoraObject $object)
- createDocumentDatastreams ()

  Passes each datastream to the appropriate ds create function.

Public Attributes

- const **FOXML** = 'info:fedora/fedora-system:def/foxml#'
- const **xlink** = 'http://www.w3.org/1999/xlink'
- const **xsi** = 'http://www.w3.org/2001/XMLSchema-instance'
- const **xmlns** = 'http://www.w3.org/2000/xmlns/'
- const **rdf** = 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'
- const **rdfs** = 'http://www.w3.org/2000/01/rdf-schema#'
- const **fedora** = 'info:fedora/fedora-system:def/relations-external#'
- const **dc** = 'http://purl.org/dc/elements/1.1/'
- const **oai_dc** = 'http://www.openarchives.org/OAI/2.0/oai_dc/'
- const **fedora_model** = 'info:fedora/fedora-system:def/model#'

Protected Attributes

- **$root**
- **$object**

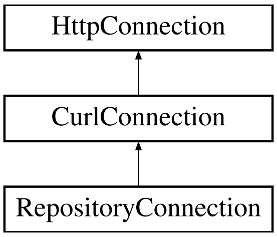
The documentation for this class was generated from the following file:

- FoxmlDocument.php

5.22 HttpConnection Class Reference

Abstract class defining functions for HTTP connections.

Inheritance diagram for HttpConnection:



Public Member Functions

- **__sleep** ()
- postRequest ($url, $type= 'none', $data=NULL, $content_type=NULL)

    Post a request to the server.
- patchRequest ($url, $type= 'none', $data=NULL, $content_type=NULL)

    Do a patch request, used for partial updates of a resource.
- getRequest ($url, $headers_only=FALSE, $file=FALSE)

    Send a HTTP GET request to URL.
- putRequest ($url, $type= 'none', $file=NULL)

    Send a HTTP PUT request to URL.
- deleteRequest ($url)

    Send a HTTP DELETE request to URL.

Public Attributes

- **$cookies** = TRUE
- **$username** = NULL
- **$password** = NULL
- **$verifyHost** = TRUE
- **$verifyPeer** = TRUE
- **$timeout** = NULL
- **$connectTimeout** = 5
- **$userAgent** = NULL
- **$reuseConnection** = TRUE
- **$sslVersion** = NULL
- **$debug** = FALSE

5.22.1 Detailed Description

Abstract class defining functions for HTTP connections.

5.22.2 Member Function Documentation

5.22.2.1 HttpConnection::deleteRequest (*$url*) [abstract]

Send a HTTP DELETE request to URL.

Parameters

string	*$url*	The URL to post the request to. Should start with the protocol. For example: `http://`.

Exceptions

HttpConnectionException	

Returns

array Associative array containing:

- $return['status'] = The HTTP status code
- $return['headers'] = The HTTP headers of the reply
- $return['content'] = The body of the HTTP reply

5.22.2.2 HttpConnection::getRequest (*$url*, *$headers_only* = FALSE, *$file* = FALSE) [abstract]

Send a HTTP GET request to URL.

Parameters

string	*$url*	The URL to post the request to. Should start with the protocol. For example: `http://`.
boolean	*$headers_only*	This will cause curl to only return the HTTP headers.
string	*$file*	A file to output the content of request to. If this is set then headers are not returned and the 'content' and 'headers' keys of the return isn't set.

Exceptions

HttpConnectionException	

Returns

array Associative array containing:

- $return['status'] = The HTTP status code
- $return['headers'] = The HTTP headers of the reply
- $return['content'] = The body of the HTTP reply

5.22.2.3 HttpConnection::patchRequest (*$url*, *$type* = `'none'`, *$data* = `NULL`, *$content_type* = `NULL`) `[abstract]`

Do a patch request, used for partial updates of a resource.

Parameters

string	*$url*	The URL to post the request to. Should start with the protocol. For example: `http://`.
string	*$type*	This paramerter must be one of: string, file.
string	*$data*	What this parameter contains is decided by the $type parameter.

Exceptions

HttpConnectionException	

Returns

array Associative array containing:

- $return['status'] = The HTTP status code
- $return['headers'] = The HTTP headers of the reply
- $return['content'] = The body of the HTTP reply

5.22.2.4 HttpConnection::postRequest (*$url*, *$type* = `'none'`, *$data* = `NULL`, *$content_type* = `NULL`) `[abstract]`

Post a request to the server.

This is primarily used for sending files.

Todo Test this for posting general form data. (Other then files.)

Parameters

string	*$url*	The URL to post the request to. Should start with the protocol. For example: `http://`.
string	*$type*	This paramerter must be one of: string, file.
string	*$data*	What this parameter contains is decided by the $type parameter.
string	*$content_type*	The content type header to set for the post request.

Exceptions

HttpConnectionException	

Returns

array Associative array containing:

- $return['status'] = The HTTP status code
- $return['headers'] = The HTTP headers of the reply
- $return['content'] = The body of the HTTP reply

5.22.2.5 HttpConnection::putRequest (*$url, $type* = 'none', *$file* = NULL) [abstract]

Send a HTTP PUT request to URL.

Parameters

string	*$url*	The URL to post the request to. Should start with the protocol. For example: `http://`.
string	*$type*	This paramerter must be one of: string, file, none.
string	*$file*	What this parameter contains is decided by the $type parameter.

Exceptions

HttpConnectionException	

Returns

array Associative array containing:

- $return['status'] = The HTTP status code
- $return['headers'] = The HTTP headers of the reply
- $return['content'] = The body of the HTTP reply

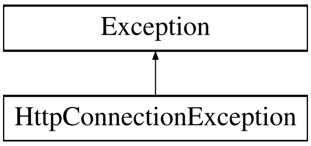
The documentation for this class was generated from the following file:

- HttpConnection.php

5.23 HttpConnectionException Class Reference

HTTP Exception.

Inheritance diagram for HttpConnectionException:



Public Member Functions

- __construct ($message, $code, $response=NULL, $previous=NULL)

The constructor for the exception.

- getResponse ()

  Get the HTTP response that caused the exception.

Protected Attributes

- **$response**

5.23.1 Detailed Description

HTTP Exception.

This is thrown when a status code other then 2XX is returned.

Parameters

string	*$message*	A message describing the exception.
int	*$code*	The error code. These are often the HTTP status codes, however less then 100 is defined by the class extending HttpConnection, for eample cURL.
array	*$response*	The array containing: status, headers, and content of the HTTP request causing the error. This is only set if there was a HTTP response sent.

5.23.2 Constructor & Destructor Documentation

5.23.2.1 HttpConnectionException::__construct (*$message*, *$code*, *$response* = `NULL`, *$previous* = `NULL`)

The constructor for the exception.

Adds a response field.

Parameters

string	*$message*	The error message
int	*$code*	The error code
array	*$response*	The HTTP response
Exception	*$previous*	The previous exception in the chain

5.23.3 Member Function Documentation

5.23.3.1 HttpConnectionException::getResponse ( )

Get the HTTP response that caused the exception.

Returns

array Array containing the HTTP response. It has three keys: status, headers and content.

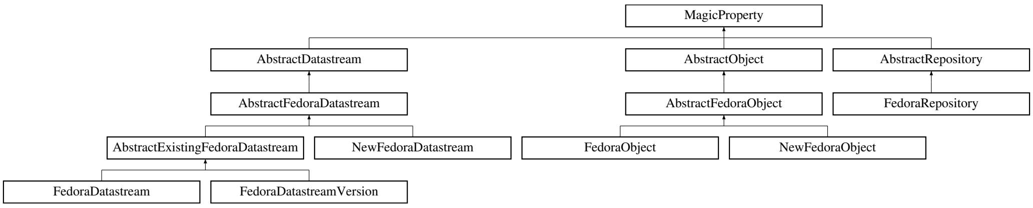
The documentation for this class was generated from the following file:

- HttpConnection.php

5.24 MagicProperty Class Reference

This abstract class allows us to implement PHP magic properties by defining a private method in the class that entends it.

Inheritance diagram for MagicProperty:



Public Member Functions

- __get ($name)

    This implements the PHP __get function which is utilized for reading data from inaccessible properties.
- __isset ($name)

    This implements the PHP __isset function which is utilized for testing if data in inaccessable properties is set.
- __set ($name, $value)

    This implements the PHP __set function which is utilized for setting inaccessable properties.
- __unset ($name)

    This implements the PHP __unset function which is utilized for unsetting inaccessable properties.

Protected Member Functions

- getGeneralMagicPropertyMethodName ($name)

    Returns the name of the magic property.

5.24.1 Detailed Description

This abstract class allows us to implement PHP magic properties by defining a private method in the class that entends it.

It attemtps to make the magic properties behave as much like normal PHP properties as possible.

This code lets the user define a new method that will be called when a property is accessed. Any method that ends in MagicProperty is code that implements a magic property.

Usage Example

```
class MyClass extends MagicProperty {
  private $secret;

  protected function myExampleMagicProperty($function, $value) {
    switch($function) {
      case 'set':
        $secret = $value;
        return;
      case 'get':
        return $secret;
      case 'isset':
        return isset($secret);
      case 'unset':
        return unset($secret);
    }
  }
}

$test = new MyClass();
$test->myExample = 'woot';
print($test->myExample);
```

5.24.2 Member Function Documentation

5.24.2.1 MagicProperty::__get (*$name*)

This implements the PHP __get function which is utilized for reading data from inaccessible properties.

It wraps it by calling the appropriatly named method in the inherteting class. http://php.net/manual/en/language.oop5.overloading.php

Parameters

string	*$name*	The name of the function being called.

Returns

void The data returned from the property.

5.24.2.2 MagicProperty::__isset (*$name*)

This implements the PHP __isset function which is utilized for testing if data in inaccessable properties is set.

This function calls the approprietly named method in the inhereting class. http://php.net/manual/en/language.oop5.overloading.php

Parameters

string	*$name*	The name of the function being called.

Returns

boolean If the variable is set.

5.24.2.3 MagicProperty::__set (*$name*, *$value*)

This implements the PHP __set function which is utilized for setting inaccessable properties.

http://php.net/manual/en/language.oop5.overloading.php

Parameters

string	*$name*	The property to set.
void	*$value*	The value it should be set with.

5.24.2.4 MagicProperty::__unset (*$name*)

This implements the PHP __unset function which is utilized for unsetting inaccessable properties.

http://php.net/manual/en/language.oop5.overloading.php

Parameters

string	*$name*	The property to unset

5.24.2.5 MagicProperty::getGeneralMagicPropertyMethodName (*$name*) [protected]

Returns the name of the magic property.

Makes it easy to change what we use as the name.

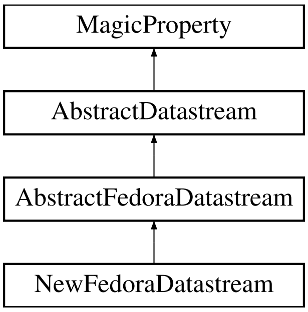
The documentation for this class was generated from the following file:

- MagicProperty.php

5.25 NewFedoraDatastream Class Reference

This defines a new fedora datastream.

Inheritance diagram for NewFedoraDatastream:



Public Member Functions

- __construct ($id, $control_group, AbstractFedoraObject $object, FedoraRepository $repository)

  The constructor for a new fedora datastream.
- setContentFromFile ($file, $copy=TRUE)
- setContentFromUrl ($url)
- setContentFromString ($string)
- getContent ($file)

Protected Member Functions

- validateControlGroup ($value)

  Validates and normalizes the control group.
- validateType ($type)

  Validates and normalizes the contentType.
- controlGroupMagicProperty ($function, $value)
- stateMagicProperty ($function, $value)
- labelMagicProperty ($function, $value)
- versionableMagicProperty ($function, $value)
- mimetypeMagicProperty ($function, $value)
- formatMagicProperty ($function, $value)
- checksumMagicProperty ($function, $value)
- checksumTypeMagicProperty ($function, $value)
- contentMagicProperty ($function, $value)
- urlMagicProperty ($function, $value)
- logMessageMagicProperty ($function, $value)

Protected Attributes

- **$copied** = FALSE

Additional Inherited Members

5.25.1 Detailed Description

This defines a new fedora datastream.

This is the class used to contain the inforamtion for a new fedora datastream before it is ingested.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 NewFedoraDatastream::__construct (*$id*, *$control_group*, AbstractFedoraObject *$object*, FedoraRepository *$repository*)

The constructor for a new fedora datastream.

Parameters

string	*$id*	The unique identifier of the DS.
FedoraObject	*$object*	The FedoraObject that this DS belongs to.
FedoraRepository	*$repository*	The FedoraRepository that this DS belongs to.
string	*$control_group*	The control group this DS will belong to.

Todo test for valid identifiers. it can't start with a number etc.

5.25.3 Member Function Documentation

5.25.3.1 NewFedoraDatastream::checksumMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::checksum

Todo this should be refined a bit

5.25.3.2 NewFedoraDatastream::checksumTypeMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::checksumType

5.25.3.3 NewFedoraDatastream::contentMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::content

5.25.3.4 NewFedoraDatastream::controlGroupMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::controlGroup

5.25.3.5 NewFedoraDatastream::formatMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::format

5.25.3.6 NewFedoraDatastream::getContent (*$file*)

See Also

AbstractDatastream::getContent

5.25.3.7 NewFedoraDatastream::labelMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::label

5.25.3.8 NewFedoraDatastream::logMessageMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::logMessage

5.25.3.9 NewFedoraDatastream::mimetypeMagicProperty (*$function*, *$value*) [protected]

See Also

AbstractDatastream::mimetype

5.25.3.10 NewFedoraDatastream::setContentFromFile (*$file*, *$copy* = TRUE)

See Also

AbstractDatastream::setContentFromFile

Parameters

boolean	*$copy*	If TRUE this object will copy and manage the given file, if FALSE the management of the files is up to the caller.

5.25.3.11 NewFedoraDatastream::setContentFromString (*$string*)

See Also

AbstractDatastream::setContentFromString

5.25.3.12 NewFedoraDatastream::setContentFromUrl (*$url*)

See Also

AbstractDatastream::setContentFromUrl

5.25.3.13 NewFedoraDatastream::stateMagicProperty (*$function, $value*) [protected]

See Also

AbstractDatastream::state

5.25.3.14 NewFedoraDatastream::urlMagicProperty (*$function, $value*) [protected]

See Also

AbstractDatastream::url

5.25.3.15 NewFedoraDatastream::validateControlGroup (*$value*) [protected]

Validates and normalizes the control group.

Parameters

string	*$value*	The passed in control group.

Returns

mixed The stirng for hte controlgroup or FALSE if validation fails.

5.25.3.16 NewFedoraDatastream::validateType (*$type*) [protected]

Validates and normalizes the contentType.

Parameters

string	*$type*	The passed in value for type.

Returns

mixed The stirng for the type or FALSE if validation fails.

5.25.3.17 NewFedoraDatastream::versionableMagicProperty (*$function, $value*) [protected]

See Also

AbstractDatastream::versionable

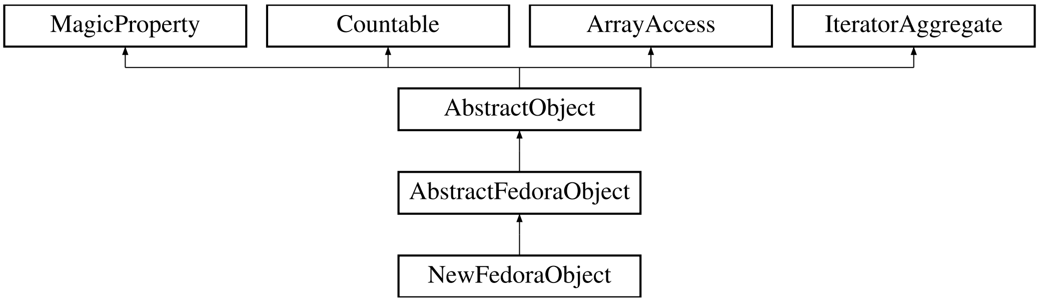
The documentation for this class was generated from the following file:

- Datastream.php

5.26 NewFedoraObject Class Reference

This represents a new fedora object that hasn't been ingested yet.

Inheritance diagram for NewFedoraObject:



Public Member Functions

- __construct ($id, FedoraRepository $repository)

  Constructoman!
- constructDatastream ($id, $control_group= 'M')
- ingestDatastream (&$ds)

  This function doesn't actually ingest the datastream, it just adds it to the queue to be ingested whenever this object is ingested.
- purgeDatastream ($id)
- getDatastream ($id)
- count ()
- offsetExists ($offset)
- offsetGet ($offset)
- offsetSet ($offset, $value)
- offsetUnset ($offset)
- getIterator ()

Protected Member Functions

- idMagicProperty ($function, $value)

  We override this as the object may need to manipulate its ID before ingestion.

Protected Attributes

- **$datastreams** = array()

Additional Inherited Members

5.26.1 Detailed Description

This represents a new fedora object that hasn't been ingested yet.

It lets us pass around the object and add datastreams before we ingest it. This shouldn't be constructed on its own, it should only be created from a factory class.

5.26.2 Member Function Documentation

5.26.2.1 NewFedoraObject::constructDatastream (*$id*, *$control_group* = 'M')

See Also

AbstractObject::constructDatastream()

5.26.2.2 NewFedoraObject::count ( )

See Also

Countable::count

5.26.2.3 NewFedoraObject::getDatastream (*$id*)

See Also

AbstractObject::getDatastream()

5.26.2.4 NewFedoraObject::getIterator ( )

See Also

IteratorAggregate::getIterator()

5.26.2.5 NewFedoraObject::idMagicProperty (*$function*, *$value*) [protected]

We override this as the object may need to manipulate its ID before ingestion.

See Also

AbstractObject::id

5.26.2.6 NewFedoraObject::ingestDatastream (& *$ds*)

This function doesn't actually ingest the datastream, it just adds it to the queue to be ingested whenever this object is ingested.

See Also

AbstractObject::ingestDatastream()

Parameters

NewFedoraDatastream	*$ds*	the datastream to be ingested

Returns

mixed FALSE on failure otherwise a instantiated FedoraDatastream.

5.26.2.7 NewFedoraObject::offsetExists (*$offset*)

See Also

ArrayAccess::offsetExists

5.26.2.8 NewFedoraObject::offsetGet (*$offset*)

See Also

ArrayAccess::offsetGet

5.26.2.9 NewFedoraObject::offsetSet (*$offset*, *$value*)

See Also

ArrayAccess::offsetSet

5.26.2.10 NewFedoraObject::offsetUnset (*$offset*)

See Also

ArrayAccess::offsetUnset

5.26.2.11 NewFedoraObject::purgeDatastream (*$id*)

See Also

AbstractObject::purgeDatastream()

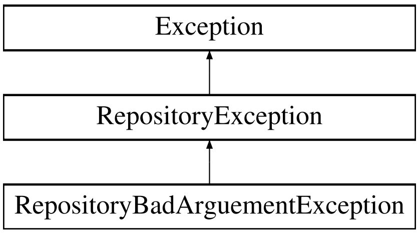
The documentation for this class was generated from the following file:

- Object.php

5.27 RepositoryBadArguementException Class Reference

This is thrown when a bad arguement is passed.

Inheritance diagram for RepositoryBadArguementException:



5.27.1 Detailed Description

This is thrown when a bad arguement is passed.

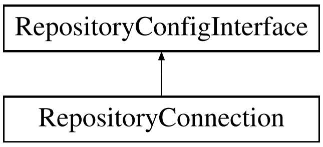
The documentation for this class was generated from the following file:

- RepositoryException.php

5.28 RepositoryConfigInterface Interface Reference

The general interface for a RepositoryConfig object.

Inheritance diagram for RepositoryConfigInterface:



Public Member Functions

- __construct ($url, $username, $password)

  Simple constructor defintion for the repository.

5.28.1 Detailed Description

The general interface for a RepositoryConfig object.

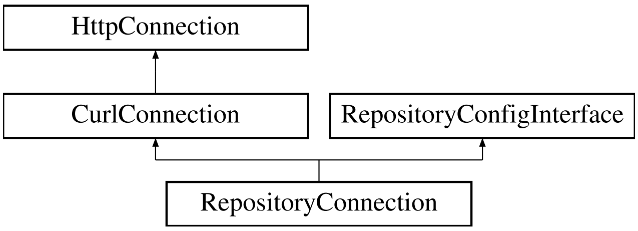
The documentation for this interface was generated from the following file:

- RepositoryConnection.php

5.29 RepositoryConnection Class Reference

Specific RepositoryConfig implementation that extends the CurlConnection class so that we can do specific processing on Curl requests for Fedora.

Inheritance diagram for RepositoryConnection:



Public Member Functions

- __construct ($url= 'http://localhost:8080/fedora', $username=NULL, $password=NULL)

  This constructor for the RepositoryConnection.
- addParamArray (&$request, &$seperator, $params, $name)

  These functions are used a lot when connecting to Fedora to create the correct arguements for REST calls.
- addParam (&$request, &$seperator, $name, $value)

  This function adds a specific parameter to a RESTful request.
- getRequest ($url, $headers_only=false, $file=NULL)

  Do a get request.
- postRequest ($url, $type= 'none', $data=NULL, $content_type=NULL)
- patchRequest ($url, $type= 'none', $data=NULL, $content_type=NULL)

- putRequest ($url, $type= 'none', $file=NULL)
- deleteRequest ($url)

Public Attributes

- **$url**
- **$username**
- **$password**

Protected Member Functions

- buildUrl ($url)

  This private function makes a fedora URL from a noraml URL.
- parseFedoraExceptions ($e)

  This function attempts to parse the exceptions that are recieved from Fedora into something more reasonable.

Additional Inherited Members

5.29.1 Detailed Description

Specific RepositoryConfig implementation that extends the CurlConnection class so that we can do specific processing on Curl requests for Fedora.

This also wraps the exceptions thrown by Curl, so that we keep our exception encapsulation.

It also doesn't take urls but instead uses relative paths that it adds the fedora URL to. This makes is a bit easier to use. It also makes sure that we always send usernames and passwords.

5.29.2 Constructor & Destructor Documentation

5.29.2.1 RepositoryConnection::__construct (*$url* = 'http://localhost:8080/fedora', *$username* = NULL, *$password* = NULL)

This constructor for the RepositoryConnection.

Parameters

string	*$url*	The URL to Fedora.
string	*$username*	The username to connect with.
string	*$password*	The password to connect with.

5.29.3 Member Function Documentation

5.29.3.1 RepositoryConnection::addParam (& *$request*, & *$seperator*, *$name*, *$value*)

This function adds a specific parameter to a RESTful request.

It makes sure that PHP booleans are changes into true and false and that the parameters are properly URL encoded.

Parameters

string	*$request*	The request that is being built.
character	*$seperator*	This is a helper to make sure that the first arguement gets a ? and the rest of them get a &.
string	*$name*	The name of the parameter that is being added
string	*$value*	the value of hte parameter.

5.29.3.2 RepositoryConnection::addParamArray (& *$request*, & *$seperator*, *$params*, *$name*)

These functions are used a lot when connecting to Fedora to create the correct arguements for REST calls.

This will encode and add an array of arguements to a request URL.

Parameters

string	*$request*	The request that is being built.
character	*$seperator*	This is a helper to make sure that the first arguement gets a ? and the rest of them get a &.
array	*$params*	An array of parameters.
string	*$name*	The name of the parameter that we are adding.

5.29.3.3 RepositoryConnection::deleteRequest (*$url*)

See Also

CurlConnection::deleteRequest()

5.29.3.4 RepositoryConnection::getRequest (*$url*, *$headers_only* = `false`, *$file* = `NULL`)

Do a get request.

Parameters

string	*$url*	The URL relative to the fedora path to use.
string	*$headers_only*	Returns only the headers
string	*$file*	The filename to output the request to. If this is set then no headers will be returned.

Returns

array The contents of the get request

See Also

CurlConnection::getRequest()

5.29.3.5 RepositoryConnection::parseFedoraExceptions (*$e*) `[protected]`

This function attempts to parse the exceptions that are recieved from Fedora into something more reasonable.

This it very hard since really things like this are garbage in garbage out, but we do what we can.

Parameters

Exception	*$e*	The exception being parsed

5.29.3.6 RepositoryConnection::patchRequest (*$url*, *$type* = `'none'`, *$data* = `NULL`, *$content_type* = `NULL`)

See Also

CurlConnection::patchRequest()

5.29.3.7 RepositoryConnection::postRequest (*$url*, *$type* = 'none', *$data* = NULL, *$content_type* = NULL)

See Also

CurlConnection::postRequest()

5.29.3.8 RepositoryConnection::putRequest (*$url*, *$type* = 'none', *$file* = NULL)

See Also

CurlConnection::putRequest()

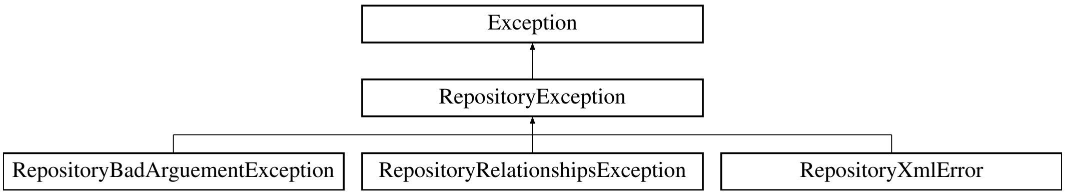
The documentation for this class was generated from the following file:

- RepositoryConnection.php

5.30 RepositoryException Class Reference

The top level exception for the Islandora Fedora API.

Inheritance diagram for RepositoryException:


Exception
RepositoryException
RepositoryBadArguementException
RepositoryRelationshipsException
RepositoryXmlError

5.30.1 Detailed Description

The top level exception for the Islandora Fedora API.

The documentation for this class was generated from the following file:

- RepositoryException.php

5.31 RepositoryQuery Class Reference

Public Member Functions

- __construct (RepositoryConnection $connection)

  Construct a new RI object.
- query ($query, $type= 'itql', $limit=-1)

  Performs the given Resource Index query and return the results.
- itqlQuery ($query, $limit=-1)

  Thin wrapper for self::_performRiQuery().
- sparqlQuery ($query, $limit=-1, $offset=0)

  Thin wrapper for self::performRiQuery().

Static Public Member Functions

- static parseSparqlResults ($sparql)

  Parse the passed in Sparql XML string into a more easily usable format.

Public Attributes

- **$connection**

Static Protected Member Functions

- static pidUriToBarePid ($uri)

  Utility function used in self::performRiQuery().

5.31.1 Constructor & Destructor Documentation

5.31.1.1 RepositoryQuery::__construct (RepositoryConnection *$connection*)

Construct a new RI object.

Parameters

Repository-Connection	*$connection*	The connection to connect to the RI with.

5.31.2 Member Function Documentation

5.31.2.1 RepositoryQuery::itqlQuery (*$query*, *$limit* = `-1`)

Thin wrapper for self::_performRiQuery().

See Also

self::performRiQuery()

5.31.2.2 static RepositoryQuery::parseSparqlResults (*$sparql*) `[static]`

Parse the passed in Sparql XML string into a more easily usable format.

Parameters

string	*$sparql*	A string containing Sparql result XML.

Returns

array Indexed (numerical) array, containing a number of associative arrays, with keys being the same as the variable names in the query. URIs beginning with 'info:fedora/' will have this beginning stripped off, to facilitate their use as PIDs.

5.31.2.3 static RepositoryQuery::pidUriToBarePid (*$uri*) `[static]`, `[protected]`

Utility function used in self::performRiQuery().

Strips off the 'info:fedora/' prefix from the passed in string.

Parameters

string	*$uri*	A string containing a URI.

Returns

string The input string less the 'info:fedora/' prefix (if it has it). The original string otherwise.

5.31.2.4 RepositoryQuery::query (*$query*, *$type* = 'itql', *$limit* = -1)

Performs the given Resource Index query and return the results.

Parameters

string	*$query*	A string containing the RI query to perform.
string	*$type*	The type of query to perform, as used by the risearch interface.
int	*$limit*	An integer, used to limit the number of results to return.

Returns

array Indexed (numerical) array, containing a number of associative arrays, with keys being the same as the variable names in the query. URIs beginning with 'info:fedora/' will have this beginning stripped off, to facilitate their use as PIDs.

5.31.2.5 RepositoryQuery::sparqlQuery (*$query*, *$limit* = -1, *$offset* = 0)

Thin wrapper for self::performRiQuery().

See Also

self::_performRiQuery()

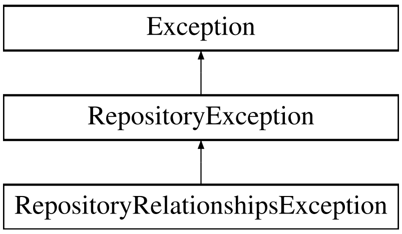
The documentation for this class was generated from the following file:

- RepositoryQuery.php

5.32 RepositoryRelationshipsException Class Reference

This is thrown from the relationships class when something goes wrong.

Inheritance diagram for RepositoryRelationshipsException:



5.32.1 Detailed Description

This is thrown from the relationships class when something goes wrong.

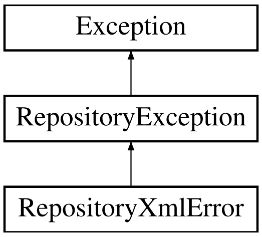
The documentation for this class was generated from the following file:

- RepositoryException.php

5.33 RepositoryXmlError Class Reference

This is thrown when there is an error parsing XML.

Inheritance diagram for RepositoryXmlError:



Public Member Functions

- __construct ($message, $code, $errors, $previous=NULL)

  Same as the default exception constructor except it takes another parameter errors, this is the error returned by the xml parser.

Public Attributes

- **$errors**

5.33.1 Detailed Description

This is thrown when there is an error parsing XML.

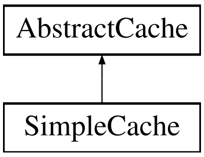
The documentation for this class was generated from the following file:

- RepositoryException.php

5.34 SimpleCache Class Reference

This is a simple cache that uses a static array to hold the cached values.

Inheritance diagram for SimpleCache:



Public Member Functions

- get ($key)
- add ($key, $data)
- set ($key, $data)
- delete ($key)

Static Public Member Functions

- static setCacheSize ($size)

  Set the cache size for the cache.
- static resetCache ()

  Reset the cache flushing it and returning it to its default size.

Public Attributes

- const **CACHESIZE** = 100

Static Protected Attributes

- static **$cache** = array()
- static **$entries** = array()
- static **$size** = SimpleCache::CACHESIZE

5.34.1 Detailed Description

This is a simple cache that uses a static array to hold the cached values.

This means that it will cache across instantiations in the same PHP runtime but not across runtimes. The cache has 100 slots and uses a simple LIFO caching strategy.

Todo Replace this with something more interesting like memcached

Try some other intersting caching strategies like LRU.

5.34.2 Member Function Documentation

5.34.2.1 SimpleCache::add (*$key*, *$data*)

See Also

AbstractCache::add

5.34.2.2 SimpleCache::delete (*$key*)

See Also

AbstractCache::delete

5.34.2.3 SimpleCache::get (*$key*)

See Also

AbstractCache::get

5.34.2.4 SimpleCache::set (*$key*, *$data*)

See Also

AbstractCache::set

5.34.2.5 static SimpleCache::setCacheSize (*$size*) [static]

Set the cache size for the cache.

If the size if bigger the cache size is just made bigger. If its smaller, the cache is flushed and the cache size is made smaller.

Parameters

int	*$size*	The new size of the cache.

The documentation for this class was generated from the following file:

- Cache.php

Chapter 6

File Documentation

6.1 Cache.php File Reference

Simple set of classes defining caches.

Classes

- class AbstractCache

  Simple abstract Cache defintion providing basic key value caching functionality.
- class SimpleCache

  This is a simple cache that uses a static array to hold the cached values.

6.1.1 Detailed Description

Simple set of classes defining caches. An abstract cache is defined so that we can use a more interesting caching setup in the future like memcached or the native drupal cache.

6.2 Datastream.php File Reference

This file defines all the classes used to manipulate datastreams in the repository.

Classes

- class AbstractDatastream

  This abstract class can be overriden by anything implementing a datastream.
- class AbstractFedoraDatastream

  Abstract base class implementing a datastream in Fedora.
- class NewFedoraDatastream

  This defines a new fedora datastream.
- class AbstractExistingFedoraDatastream

  This abstract class defines some shared functionality between all classes that implement exising fedora datastreams.
- class FedoraDatastreamVersion

  This class implements an old version of a fedora datastream.
- class FedoraDatastream

  This class implements a fedora datastream.

6.2.1 Detailed Description

This file defines all the classes used to manipulate datastreams in the repository.

6.3 FedoraApi.php File Reference

The RAW API wrappers for the Fedora interface.

Classes

- class FedoraApi

  This is a simple class that brings FedoraApiM and FedoraApiA together.
- class FedoraApiA

  This class implements the Fedora API-A interface.
- class FedoraApiM

  This class implements the Fedora API-M interface.

6.3.1 Detailed Description

The RAW API wrappers for the Fedora interface. This file currently contains fairly raw wrappers around the Fedora REST interface. These could also be reinmplemented to use for example the Fedora SOAP interface. If there are version specific modifications to be made for Fedora, this is the place to make them.

6.4 FedoraApiSerializer.php File Reference

This file contains functions to serialize the XML responses from Fedora into something more easily dealt with in PHP.

Classes

- class FedoraApiSerializer

  A class to Serialize the XML responses from Fedora into PHP arrays.

6.4.1 Detailed Description

This file contains functions to serialize the XML responses from Fedora into something more easily dealt with in PHP.

6.5 FedoraRelationships.php File Reference

This file defines the classes that are used for manipulaing the fedora relationships datastreams.

Classes

- class FedoraRelationships

  This is the base class for Fedora Relationships.
- class FedoraRelsExt
- class FedoraRelsInt

Variables

- const **XMLNS** "http://www.w3.org/2000/xmlns/"
- const **FEDORA_RELS_EXT_URI** 'info:fedora/fedora-system:def/relations-external#'
- const **FEDORA_MODEL_URI** 'info:fedora/fedora-system:def/model#'
- const **ISLANDORA_RELS_EXT_URI** 'http://islandora.ca/ontology/relsext#'
- const **ISLANDORA_RELS_INT_URI** "http://islandora.ca/ontology/relsint#"
- const **RELS_TYPE_URI** 0
- const **RDF_URI** 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'
- const **RELS_TYPE_PLAIN_LITERAL** 1
- const **RELS_TYPE_STRING** 2
- const **RELS_STRING_NS** "http://www.w3.org/2001/XMLSchema#string"
- const **RELS_TYPE_INT** 3
- const **RELS_INT_NS** "http://www.w3.org/2001/XMLSchema#int"
- const **RELS_TYPE_DATETIME** 4
- const **RELS_DATETIME_NS** "http://www.w3.org/2001/XMLSchema#dateTime"

6.5.1 Detailed Description

This file defines the classes that are used for manipulaing the fedora relationships datastreams.

6.6 HttpConnection.php File Reference

This file defines the classes used to make HTTP requests.

Classes

- class HttpConnectionException

  HTTP Exception.
- class HttpConnection

  Abstract class defining functions for HTTP connections.
- class CurlConnection

  This class defines a abstract HttpConnection using the PHP cURL library.

6.6.1 Detailed Description

This file defines the classes used to make HTTP requests.

6.7 MagicProperty.php File Reference

This file contains the class MagicProperty.

Classes

- class MagicProperty

  This abstract class allows us to implement PHP magic properties by defining a private method in the class that entends it.

6.7.1 Detailed Description

This file contains the class MagicProperty.

6.8 Object.php File Reference

This file contains all of the functionality for objects in the repository.

Classes

- class AbstractObject

  An abstract class defining a Object in the repository.
- class AbstractFedoraObject

  This is the base class for a Fedora Object.
- class NewFedoraObject

  This represents a new fedora object that hasn't been ingested yet.
- class FedoraObject

  Represents a Fedora Object.

6.8.1 Detailed Description

This file contains all of the functionality for objects in the repository.

6.9 Repository.php File Reference

This file defines an abstract repository that can be overridden and also defines a concrete implementation for Fedora.

Classes

- class AbstractRepository

  An abstract repository interface.
- class FedoraRepository

  Concrete implementation of the AbstractRepository for Fedora.

6.9.1 Detailed Description

This file defines an abstract repository that can be overridden and also defines a concrete implementation for Fedora.

6.10 RepositoryConnection.php File Reference

This file contains the implementation of a connection to Fedora.

Classes

- interface RepositoryConfigInterface

  The general interface for a RepositoryConfig object.
- class RepositoryConnection

  Specific RepositoryConfig implementation that extends the CurlConnection class so that we can do specific processing on Curl requests for Fedora.

6.10.1 Detailed Description

This file contains the implementation of a connection to Fedora. And the interface for a repository configuration.

6.11 RepositoryException.php File Reference

Contains all of the exceptions thrown by Tuque.

Classes

- class RepositoryException

  The top level exception for the Islandora Fedora API.
- class RepositoryXmlError

  This is thrown when there is an error parsing XML.
- class RepositoryBadArguementException

  This is thrown when a bad arguement is passed.
- class RepositoryRelationshipsException

  This is thrown from the relationships class when something goes wrong.

6.11.1 Detailed Description

Contains all of the exceptions thrown by Tuque.

6.12 RepositoryQuery.php File Reference

This file provides some methods for doing RDF queries.

Classes

- class RepositoryQuery

6.12.1 Detailed Description

This file provides some methods for doing RDF queries. The essance of this file was taken from some commits that Adam Vessy made to Islandora 6.x, so I'd like to give him some credit here.

Index

$namespaces
 FedoraRelationships, 64
__construct
 AbstractDatastream, 12
 AbstractExistingFedoraDatastream, 13
 AbstractFedoraDatastream, 16
 AbstractObject, 21
 CurlConnection, 26
 FedoraApi, 27
 FedoraApiA, 28
 FedoraApiM, 35
 FedoraDatastream, 49
 FedoraDatastreamVersion, 54
 FedoraDate, 56
 FedoraObject, 58
 FedoraRelationships, 61
 FedoraRelsExt, 65
 FedoraRelsInt, 67
 FedoraRepository, 69
 HttpConnectionException, 75
 NewFedoraDatastream, 79
 RepositoryConnection, 86
 RepositoryQuery, 89
__destruct
 CurlConnection, 26
__get
 MagicProperty, 77
__isset
 MagicProperty, 77
__set
 MagicProperty, 77
__sleep
 FedoraDate, 56
__toString
 AbstractFedoraObject, 19
 FedoraDate, 56
__unset
 MagicProperty, 77

AbstractCache, 9
 add, 9
 delete, 10
 get, 10
 set, 10
AbstractDatastream, 10
 __construct, 12
 getContent, 12
 setContentFromFile, 12
 setContentFromString, 12
 setContentFromUrl, 12
AbstractExistingFedoraDatastream, 13
 __construct, 13
 getDatastreamContent, 14
 getDatastreamHistory, 14
 modifyDatastream, 14
 purgeDatastream, 14
AbstractFedoraDatastream, 15
 __construct, 16
 controlGroupMagicProperty, 16
 delete, 16
 idMagicProperty, 16
 isDatastreamProperySet, 16
 validateChecksumType, 17
 validateMimetype, 17
 validateState, 17
 validateVersionable, 17
AbstractFedoraObject, 18
 __toString, 19
 constructDatastream, 19
 delete, 19
 idMagicProperty, 19
 labelMagicProperty, 19
 logMessageMagicProperty, 19
 modelsMagicProperty, 19
 ownerMagicProperty, 19
 stateMagicProperty, 20
AbstractObject, 20
 __construct, 21
 constructDatastream, 21
 getDatastream, 22
 purgeDatastream, 22
AbstractRepository, 22
 constructObject, 23
 findObjects, 23
 getObject, 23
 ingestObject, 24
 purgeObject, 24
add
 AbstractCache, 9
 FedoraRelsExt, 65
 FedoraRelsInt, 67
 SimpleCache, 92
addDatastream
 FedoraApiM, 35
addParam
 RepositoryConnection, 86
addParamArray
 RepositoryConnection, 87
addRelationship

FedoraApiM, 37

Cache.php, 95
changeObjectID
FedoraRelationships, 61
checksumMagicProperty
FedoraDatastream, 50
FedoraDatastreamVersion, 54
NewFedoraDatastream, 79
checksumTypeMagicProperty
FedoraDatastream, 50
NewFedoraDatastream, 79
constructDatastream
AbstractFedoraObject, 19
AbstractObject, 21
FedoraObject, 58
NewFedoraObject, 83
constructObject
AbstractRepository, 23
FedoraRepository, 69
contentMagicProperty
FedoraDatastream, 50
FedoraDatastreamVersion, 54
NewFedoraDatastream, 79
controlGroupMagicProperty
AbstractFedoraDatastream, 16
FedoraDatastream, 50
NewFedoraDatastream, 79
count
FedoraDatastream, 50
FedoraObject, 58
NewFedoraObject, 83
createdDateMagicProperty
FedoraDatastream, 50
FedoraDatastreamVersion, 54
FedoraObject, 58
CurlConnection, 24
__construct, 26
__destruct, 26
deleteRequest, 26
doCurlRequest, 26
getRequest, 26
patchRequest, 26
postRequest, 26
putRequest, 27

Datastream.php, 95
delete
AbstractCache, 10
AbstractFedoraDatastream, 16
AbstractFedoraObject, 19
SimpleCache, 92
deleteRequest
CurlConnection, 26
HttpConnection, 72
RepositoryConnection, 87
describeRepository
FedoraApiA, 29
doCurlRequest
CurlConnection, 26
domDocumentExceptionHandler
FedoraApiSerializer, 46

export
FedoraApiM, 37

FedoraApi, 27
__construct, 27
FedoraApi.php, 96
FedoraApiA, 28
__construct, 28
describeRepository, 29
findObjects, 29
getDatastreamDissemination, 31
getDissemination, 31
getObjectHistory, 31
getObjectProfile, 32
listDatastreams, 32
listMethods, 33
resumeFindObjects, 34
FedoraApiM, 34
__construct, 35
addDatastream, 35
addRelationship, 37
export, 37
getDatastream, 38
getDatastreamHistory, 38
getNextPid, 39
getObjectXml, 40
getRelationships, 40
ingest, 41
modifyDatastream, 42
modifyObject, 43
purgeDatastream, 43
purgeObject, 44
validate, 44
FedoraApiSerializer, 45
domDocumentExceptionHandler, 46
flattenDocument, 47
loadDomDocument, 47
loadSimpleXml, 47
FedoraApiSerializer.php, 96
FedoraDatastream, 48
__construct, 49
checksumMagicProperty, 50
checksumTypeMagicProperty, 50
contentMagicProperty, 50
controlGroupMagicProperty, 50
count, 50
createdDateMagicProperty, 50
formatMagicProperty, 50
getContent, 50
labelMagicProperty, 50
locationMagicProperty, 51
logMessageMagicProperty, 51
mimetypeMagicProperty, 51
offsetExists, 51
offsetGet, 51

offsetSet, 51
offsetUnset, 51
setContentFromFile, 51
setContentFromString, 51
setContentFromUrl, 52
sizeMagicProperty, 52
stateMagicProperty, 52
urlMagicProperty, 52
versionableMagicProperty, 52
FedoraDatastreamVersion, 52
__construct, 54
checksumMagicProperty, 54
contentMagicProperty, 54
createdDateMagicProperty, 54
formatMagicProperty, 54
getContent, 54
labelMagicProperty, 54
logMessageMagicProperty, 54
mimetypeMagicProperty, 54
setContentFromFile, 55
setContentFromString, 55
setContentFromUrl, 55
sizeMagicProperty, 55
stateMagicProperty, 55
versionableMagicProperty, 55
FedoraDate, 55
__construct, 56
__sleep, 56
__toString, 56
FedoraObject, 57
__construct, 58
constructDatastream, 58
count, 58
createdDateMagicProperty, 58
getDatastream, 58
getParents, 58
ingestDatastream, 59
labelMagicPropertySet, 59
lastModifiedDateMagicProperty, 59
logMessageMagicPropertySet, 59
offsetExists, 59
offsetGet, 59
offsetSet, 59
offsetUnset, 59
ownerMagicPropertySet, 60
purgeDatastream, 60
stateMagicPropertySet, 60
FedoraRelationships, 60
$namespaces, 64
__construct, 61
changeObjectID, 61
getDom, 62
getXpath, 62
getXpathResults, 62
internalAdd, 62
internalGet, 62
internalRemove, 63
registerNamespace, 63
FedoraRelationships.php, 96
FedoraRelsExt, 64
__construct, 65
add, 65
get, 65
initializeDatastream, 66
remove, 66
FedoraRelsInt, 66
__construct, 67
add, 67
get, 67
remove, 68
FedoraRepository, 68
__construct, 69
constructObject, 69
findObjects, 70
getObject, 70
ingestObject, 70
purgeObject, 70
findObjects
AbstractRepository, 23
FedoraApiA, 29
FedoraRepository, 70
flattenDocument
FedoraApiSerializer, 47
formatMagicProperty
FedoraDatastream, 50
FedoraDatastreamVersion, 54
NewFedoraDatastream, 80
FoxmlDocument, 70

get
AbstractCache, 10
FedoraRelsExt, 65
FedoraRelsInt, 67
SimpleCache, 92
getContent
AbstractDatastream, 12
FedoraDatastream, 50
FedoraDatastreamVersion, 54
NewFedoraDatastream, 80
getDatastream
AbstractObject, 22
FedoraApiM, 38
FedoraObject, 58
NewFedoraObject, 83
getDatastreamContent
AbstractExistingFedoraDatastream, 14
getDatastreamDissemination
FedoraApiA, 31
getDatastreamHistory
AbstractExistingFedoraDatastream, 14
FedoraApiM, 38
getDissemination
FedoraApiA, 31
getDom
FedoraRelationships, 62
getGeneralMagicPropertyMethodName
MagicProperty, 77

getIterator
NewFedoraObject, 83
getNextPid
FedoraApiM, 39
getObject
AbstractRepository, 23
FedoraRepository, 70
getObjectHistory
FedoraApiA, 31
getObjectProfile
FedoraApiA, 32
getObjectXml
FedoraApiM, 40
getParents
FedoraObject, 58
getRelationships
FedoraApiM, 40
getRequest
CurlConnection, 26
HttpConnection, 72
RepositoryConnection, 87
getResponse
HttpConnectionException, 75
getXpath
FedoraRelationships, 62
getXpathResults
FedoraRelationships, 62

HttpConnection, 71
deleteRequest, 72
getRequest, 72
patchRequest, 73
postRequest, 73
putRequest, 74
HttpConnection.php, 97
HttpConnectionException, 74
__construct, 75
getResponse, 75

idMagicProperty
AbstractFedoraDatastream, 16
AbstractFedoraObject, 19
NewFedoraObject, 83
ingest
FedoraApiM, 41
ingestDatastream
FedoraObject, 59
NewFedoraObject, 83
ingestObject
AbstractRepository, 24
FedoraRepository, 70
initializeDatastream
FedoraRelsExt, 66
internalAdd
FedoraRelationships, 62
internalGet
FedoraRelationships, 62
internalRemove
FedoraRelationships, 63
isDatastreamPropertySet
AbstractFedoraDatastream, 16
itqlQuery
RepositoryQuery, 89

labelMagicProperty
AbstractFedoraObject, 19
FedoraDatastream, 50
FedoraDatastreamVersion, 54
NewFedoraDatastream, 80
labelMagicPropertySet
FedoraObject, 59
lastModifiedDateMagicProperty
FedoraObject, 59
listDatastreams
FedoraApiA, 32
listMethods
FedoraApiA, 33
loadDomDocument
FedoraApiSerializer, 47
loadSimpleXml
FedoraApiSerializer, 47
locationMagicProperty
FedoraDatastream, 51
logMessageMagicProperty
AbstractFedoraObject, 19
FedoraDatastream, 51
FedoraDatastreamVersion, 54
NewFedoraDatastream, 80
logMessageMagicPropertySet
FedoraObject, 59

MagicProperty, 75
__get, 77
__isset, 77
__set, 77
__unset, 77
getGeneralMagicPropertyMethodName, 77
MagicProperty.php, 97
mimetypeMagicProperty
FedoraDatastream, 51
FedoraDatastreamVersion, 54
NewFedoraDatastream, 80
modelsMagicProperty
AbstractFedoraObject, 19
modifyDatastream
AbstractExistingFedoraDatastream, 14
FedoraApiM, 42
modifyObject
FedoraApiM, 43

NewFedoraDatastream, 78
__construct, 79
checksumMagicProperty, 79
checksumTypeMagicProperty, 79
contentMagicProperty, 79
controlGroupMagicProperty, 79
formatMagicProperty, 80
getContent, 80

labelMagicProperty, 80
logMessageMagicProperty, 80
mimetypeMagicProperty, 80
setContentFromFile, 80
setContentFromString, 80
setContentFromUrl, 81
stateMagicProperty, 81
urlMagicProperty, 81
validateControlGroup, 81
validateType, 81
versionableMagicProperty, 81

NewFedoraObject, 82
constructDatastream, 83
count, 83
getDatastream, 83
getIterator, 83
idMagicProperty, 83
ingestDatastream, 83
offsetExists, 83
offsetGet, 84
offsetSet, 84
offsetUnset, 84
purgeDatastream, 84

Object.php, 98

offsetExists
FedoraDatastream, 51
FedoraObject, 59
NewFedoraObject, 83

offsetGet
FedoraDatastream, 51
FedoraObject, 59
NewFedoraObject, 84

offsetSet
FedoraDatastream, 51
FedoraObject, 59
NewFedoraObject, 84

offsetUnset
FedoraDatastream, 51
FedoraObject, 59
NewFedoraObject, 84

ownerMagicProperty
AbstractFedoraObject, 19

ownerMagicPropertySet
FedoraObject, 60

parseFedoraExceptions
RepositoryConnection, 87

parseSparqlResults
RepositoryQuery, 89

patchRequest
CurlConnection, 26
HttpConnection, 73
RepositoryConnection, 87

pidUriToBarePid
RepositoryQuery, 89

postRequest
CurlConnection, 26
HttpConnection, 73
RepositoryConnection, 88

purgeDatastream
AbstractExistingFedoraDatastream, 14
AbstractObject, 22
FedoraApiM, 43
FedoraObject, 60
NewFedoraObject, 84

purgeObject
AbstractRepository, 24
FedoraApiM, 44
FedoraRepository, 70

putRequest
CurlConnection, 27
HttpConnection, 74
RepositoryConnection, 88

query
RepositoryQuery, 90

registerNamespace
FedoraRelationships, 63

remove
FedoraRelsExt, 66
FedoraRelsInt, 68

Repository.php, 98
RepositoryBadArguementException, 84
RepositoryConfigInterface, 85
RepositoryConnection, 85
__construct, 86
addParam, 86
addParamArray, 87
deleteRequest, 87
getRequest, 87
parseFedoraExceptions, 87
patchRequest, 87
postRequest, 88
putRequest, 88

RepositoryConnection.php, 98
RepositoryException, 88
RepositoryException.php, 99
RepositoryQuery, 88
__construct, 89
itqlQuery, 89
parseSparqlResults, 89
pidUriToBarePid, 89
query, 90
sparqlQuery, 90

RepositoryQuery.php, 99
RepositoryRelationshipsException, 90
RepositoryXmlError, 91

resumeFindObjects
FedoraApiA, 34

set
AbstractCache, 10
SimpleCache, 92

setCacheSize
SimpleCache, 93

setContentFromFile

- AbstractDatastream, 12
- FedoraDatastream, 51
- FedoraDatastreamVersion, 55
- NewFedoraDatastream, 80

setContentFromString
- AbstractDatastream, 12
- FedoraDatastream, 51
- FedoraDatastreamVersion, 55
- NewFedoraDatastream, 80

setContentFromUrl
- AbstractDatastream, 12
- FedoraDatastream, 52
- FedoraDatastreamVersion, 55
- NewFedoraDatastream, 81

SimpleCache, 91
- add, 92
- delete, 92
- get, 92
- set, 92
- setCacheSize, 93

sizeMagicProperty
- FedoraDatastream, 52
- FedoraDatastreamVersion, 55

sparqlQuery
- RepositoryQuery, 90

stateMagicProperty
- AbstractFedoraObject, 20
- FedoraDatastream, 52
- FedoraDatastreamVersion, 55
- NewFedoraDatastream, 81

stateMagicPropertySet
- FedoraObject, 60

urlMagicProperty
- FedoraDatastream, 52
- NewFedoraDatastream, 81

validate
- FedoraApiM, 44

validateChecksumType
- AbstractFedoraDatastream, 17

validateControlGroup
- NewFedoraDatastream, 81

validateMimetype
- AbstractFedoraDatastream, 17

validateState
- AbstractFedoraDatastream, 17

validateType
- NewFedoraDatastream, 81

validateVersionable
- AbstractFedoraDatastream, 17

versionableMagicProperty
- FedoraDatastream, 52
- FedoraDatastreamVersion, 55
- NewFedoraDatastream, 81