Earlier Implementation Description - GSoC09

This is the list of methods/endpoints implemented in earlier development stage.

Name and description	Value and notes
Base URI:	`/communities?topLevelOnly=true&idOnly=false`
Description:	Returns a list of all communities on the system or return just top level communities.
HTTP method:	`GET`
Optional parameters:	`topLevelOnly=true`: returns only top level communities `idOnly=false`: if true return only the identifiers for the record, by default return the full record
Sorting fields supported:	`id, name, countitems`: sorting by id, community name and item count
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	By default contains item count, identifier, handle and name and other referencing data, or 204 if none. Additionaly, based on parameter idOnly=true the method returns only identifiers.

Name and description	Value and notes
Base URI:	/communities/{id}/parents?idOnly=false&immediateOnly=true
Description:	Returns a list of all parent communities of the `id` community.
HTTP method:	`GET`
Optional parameters:	idOnly=false: if true return only the identifiers for the record, by default return the full record immediateOnly=true: if false then return all parents for this community, by default return only the immediate parents
Sorting fields supported:	`id, name, countitems`: sorting by id, community name and item count
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains item count, identifiers, handle and name or 204 if id is already top-level. Additionaly, based on parameter idOnly=true the method returns only identifiers.

Name and description	Value and notes
Base URI:	/communities/`id`/children?idOnly=false
Description:	Returns a list of immediate sub-communities (children) of the `id` community.
HTTP method:	`GET`
Optional parameters:	idOnly=false: if true return only the identifiers for the record, by default return the full record
Sorting fields supported:	`id, name, countitems`: sorting by id, community name and item count
Response formats:	`json`, `xml`

Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains item count, identifiers, handle and name or 204 if none. Additionaly, based on parameter idOnly=true the method returns only identifiers.

Name and description	Value and notes
Base URI:	/communities/`id`/collections?idOnly=false
Description:	Returns a list of collections in the `id` community
HTTP method:	`GET`
Optional parameters:	idOnly=false: if true return only the identifiers for the record, by default return the full record
Sorting fields supported:	`id, name, countitems`: sorting by id, collection name and item count
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains item count, identifiers, name, archival status, last modification and handle of collections, or 204 if there are none of them. Additionaly with idOnly=true only identifiers are returned.

Name and description	Value and notes
Base URI:	/communities/`id`/recent?idOnly=false
Description:	Returns a list of recent submissions to a community
HTTP method:	`GET`
Optional parameters:	idOnly=false: if true return only the identifiers for the record, by default return the full record
Sorting fields supported:	`id, name, lastmodified, submitter`: sorting by id, name(title), last modified date and submitter(name) of item
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains complete items from recent submissions in community. Additionaly, contains only identifiers when idOnly=true is used.

Name and description	Value and notes
Base URI:	/collections?idOnly=false
Description:	Returns a list of all collections in the system
HTTP method:	`GET`
Optional parameters:	idOnly=false: if true return only the identifiers for the record, by default return the full record
Sorting fields supported:	`id, name, countitems`: sorting by id, collection name and item count
Response formats:	`json`, `xml`

Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains item count, identifiers, name and handle of collections, or 204 if there are none of them. In the case idOnly=true is used, returns only identifiers of collections.

Name and description	Value and notes
Base URI:	/collections/`id`/communities?idOnly=false
Description:	Returns a list of all communities a collection with `id` belongs to
HTTP method:	`GET`
Optional parameters:	idOnly=false: if true return only the identifiers for the record, by default return the full record
Sorting fields supported:	`id, name, countitems`: sorting by id, community name and item count
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains item count, identifier, name and handle of collections, or 204 if there are none of them. Additionaly, when idOnly=true is used, return only identifiers.

Name and description	Value and notes
Base URI:	/collections/`id`/items?idOnly=false&in_archive=false
Description:	Returns a list of all items from the collection `id`
HTTP method:	`GET`
Optional parameters:	idOnly=false: if true, return only identifiers of result records in_archive=false: defines should it return in_archive items only or all items if true
Sorting fields supported:	`id, name, lastmodified, submitter`: sorting by id, name, lastmodified date and submitter of item
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains full information info including name, submitter, collections related or 204 if there are none of them In the case idOnly=true is used, return only identifiers of results.

Name and description	Value and notes
Base URI:	/search?query=`query`&(community=`id`!collection=`id`)&idOnly=false
Description:	Returns a list of all objects found by searching criteria
HTTP method:	`GET`
Optional parameters:	`query`: string query for searching, without query method behaves like browsing with additional sorting/ordering capabilities community=`id`: the id of community limiting search to collection=`id`: the id of collection limiting search to Notice: community and collection are mutually exclusive options idOnly=false: using true returns only identifiers of the results

Sorting fields supported:	`id, name, lastmodified, submitter`: sorting by id, name, last modifed date or submitter of item
Sorting/ordering modifiers:	`title, issueDate, author, subject, submitter`
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Item info with basic metadata for the search results. Additionally return only identifiers when idOnly=true is used.

Name and description	Value and notes
Base URI:	/harvest?startdate=`startdate`&enddate=`enddate`?(community=`id`!collection=`id`) &idOnly=false&withdrawn=false
Description:	Returns a list of all objects that have been created, modified or withdrawn within specified time range
HTTP method:	`GET`
Optional parameters:	`startdate`: start date `enddate`: end date date is defined with ISO 8601 format, e.g. YYYY-MM-DD or 2009-08-20T21:15Z; if the date parameters are ommited, then from_beginning/to_current_time are assumed community=`id`: the id of community limiting search to collection=`id`: the id of collection limiting search to Notice: community and collection are mutually exclusive options idOnly=false: using true returns only identifiers of the results withdrawn=false: with true includes withdrawn items too
Sorting/ordering modifiers:	`id, name, lastmodified, submitter`: information on item returned
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains item info including id, name, handle, metadata, bitstreams according to the defined requirements. Additionally when idOnly=true only identifiers of results are returned. If the date is in incompatible format, error 400 is returned.

Item status and retrieval

Name and description	Value and notes
Base URI:	`/items/{id}`
Description:	Returns detailed information about an item
HTTP method:	`GET`
Required parameters:	{`id`}: item id
Sorting fields supported:	`id, name, lastmodified, submitter`: sorting by id, name, last modifed date or submitter of item
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	Contains an information about an item including resource name, metadata, owning collection, collections stored in, communities stored in, bundle ids, last modified date, archival/withdrawn status and submitter of an item

Name and description	Value and notes
Base URI:	`/items/{id}/permissions`
Description:	Returns status of user permissions on this item
HTTP method:	`GET`
Required parameters:	`{id}`: item id
Response formats:	`json`, `xml`
Status codes	200: OK 400: bad request 500: internal server error
Response details	Boolean variable, stating can user edit the listed item

Name and description	Value and notes
Base URI:	/items/`id`/communities?idOnly=false&immediateOnly=true
Description:	Returns communities this item is part of
HTTP method:	`GET`
Required parameters:	
Sorting fields supported:	`id, name, countitems`: community properties used for sorting
Response formats:	`json`, `xml`
Status codes	200: OK 400: bad request 500: internal server error
Response details	Communities listed

Name and description	Value and notes
Base URI:	/items/`id`/collections?idOnly=false
Description:	Returns collections this item is part of
HTTP method:	`GET`
Required parameters:	
Sorting fields supported:	`id, name, countitems`: collection parameters
Response formats:	`json`, `xml`
Status codes	200: OK 400: bad request 500: internal server error
Response details	Collections listed

Name and description	Value and notes
Base URI:	`/bitstream/{id}`
Description:	Returns bitstream object - usually the library item file
HTTP method:	`GET`
Required parameters:	`{id}`: bitstream item id
Response formats:	`json`, `xml` (not yet complete)

Status codes	200: OK 404: Not found 400: bad request 401: Unauthorized 403: Forbidden 500: internal server error
Response details	Includes all information about referenced bitstream, including file name, licence, corresponding ittem etc. It is possible only to get information for particular bitstreams. When the request is made without parameters/references, the blank list is presented (there is no list of all bitstreams in the system available).

Name and description	Value and notes
Base URI:	`/bitstream/{id}/receive`
Description:	Returns checksum of bitstream
HTTP method:	`GET`
Required parameters:	`{id}`: bitstreamitem id
Response formats:	`binary`
Status codes	200: OK 400: bad request 401: Unauthorized 403: Forbidden 500: internal server error
Response details	Receive full bitstream

Name and description	Value and notes
Base URI:	/users?query=`query`&idOnly=false
Description:	Returns list containing id, name and email of persons (optionally matching a query)
HTTP method:	`GET`
Optional parameters:	
Sorting fields supported:	`id, name, lastname, fullname, language`: sorting properties of user(submitter) supported
Response formats:	`json`, `xml`
Status codes	200: OK 204: no content 400: bad request 500: internal server error
Response details	List with information on particular user. Additionaly only identifiers are sent if idOnly is true.

Name and description	Value and notes
Base URI:	`/stats`
Description:	Returns general statistics
HTTP method:	`GET`
Response formats:	`json`, `xml`
Status codes	200: OK 400: bad request (if there is no stats package available) 500: internal server error
Response details	Returns cummulative list of statistics data for the system currently available