Core Knowledge Platform 2.0 Documentation
Release 0

Florian Bergmann

July 20, 2011

CONTENTS

Contents:

OVERVIEW

Every interaction with the web service should be conducted through HTTP. To allow the web service to respond with the appropriate data format HTTP-headers are used: all queries to the web service *must* set the following headers:

ACCEPT: tells the web service which format to return.

CONTENT-TYPE: tells the web service which format was sent.

Currently *most* resources of the web service allow only one format: `application/xml`.

1.1 Data retrieval

To obtain data from the service issue a GET request to one of the following addresses:

publication/ Retrieve a list of all publications.

publication/{id} Retrieve all information for a specific publication.

authors/ Retrieve a list of all authors.

authors/{id} Retrieve all information for a specific authors.

comment/{id} Retrieve a specific comment.

peerreviewtemplate/ Retrieve a list of available peer review templates.

peerreviewtemplate/{id} Retrieve a concrete peer review template.

tag/ Retrieve a list of all tags.

tag/{id} Retrieve a specific tag.

These resources can be accessed without being logged in. The following resources require a user to be logged in:

peerreview/{id} Retrieve a concrete peer review.

1.1.1 Login & Logout

If a resource requires a login the client can login via a `POST` request to the following address: `user/login/`

To logout from the system a simple `GET` request to `/user/logout/` is enough.

1.1.2 Searching

Searching authors and publications is possible:

Publications

Just append key-value-pairs to the url according to the following schema:

`/publication/{name=value&name2=value2}/author/{name=value&...}/keyword/{..}`

Sub-searches will be executed via OR: all publications with name = value OR name2=value2

The results of the sub-searches will be concatenated using the AND operator: all publications where AND where the authors is AND where the keyword is

Authors

Do not forget that in this case the search uses a query string.

`/authors/?{name=value}`

1.2 Data Insertion

When inserting data it is important to distinguish between an initial insertion of a modification of data:

1.2.1 Initial insert

The initial insertion is performed via `POST` requests to the root urls of a resource.

To insert a publication `POST` the appropriate data to:

`/publication/`

To insert an author `POST` the appropriate data to:

`/author/`

This will insert a new object into the database and return the location the object can be retrieved from.

1.2.2 Updating data

To update data the `PUT` request is used - it will be send to the specific resource url.

To update an existing publication the `PUT` request needs to be sent to:

`/publication/{id}`

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*