api

August 30, 2013

Contents

1 Overview

The Materials Commons Web Services (mcapi) allows users to interact with the Materials Commons service without using the Materials Commons Website. The website is composed using these services. This means that you can script all the functionality you see in the website.

The mcapi is accessed using RESTful urls. Many of the calls are privileged and require an apikey. You can specify an apikey even on services that don't require one.

Getting an apikey is simple. If you have an account at the website then you automatically have one. You can see your apikey by logging into the website and clicking on your account. Alternatively you can use mcapi itself to retrieve your key. This call is the only call you specify your username and password with.

The base URL for the mcapi services is `https://api.materialscommons.org`. All urls start with `/<version>` where version is currently `v1.0`. For example to get a list of all public data files you could use curl like this:

```
curl -XGET https://api.materialscommons.org/v1.0/datafiles
```

The apikey is specified as an argument. For example, if your *username* is `i@donotexist.com` and your apikey is `abc123`, to look at your list of data files loaded onto Materials Commons you would use:

```
curl -XGET https://api.materialscommons.org/v1.0/user/i@b.com/data?apikey=abc123
```

It might seem strange to have to specify your username when you are also using your apikey. However, your apikey is used to check your permissions. If another user has given you access to look at their private data files then you could put their username in place of your own.

1.1   API Calls

The api calls are broken into a number of different sections. Some calls relate to users, some to public data such as tags or publicly released data files, some are administrative in nature, such as changing your password. The set of services is constantly being updated so check back here often.

A url such as /usergroup/<usergroup>/data means that you specify the name of the usergroup in the <usergroup> parameter. For example:

```
curl -XGET https://api.materialscommons.org/v1.0/usergroup/mygroup/datafiles
```

The HTTP verbs are used in the following ways:

Verb	Meaning
GET(G)	Retrieve value
JSONP(J)	Retrieve value wrap in javascript call
PUT(P)	Update existing
POST(S)	Create new
DELETE(D)	Delete existing

In the tables below items between angled brackets, e.g. <u>, are parameters. If a call requires an APIKEY that means it is an authenticated call.

The calls below are subject to change. In particular calls may be changed to require an APIKEY, have their URL changed to better follow our naming convention, or changed to a different verb above to better reflect their. functionality.

This is a living document as we continue to expand our set of web services. The column specifies whether a user needs to specify an APIKEY for authorization/authentication purposes.

The current set of API calls and the verbs they support are listed below:

1.1.1 Tag Services

URL	VERBS	a?	Notes
/tag	S		
/tag/<tag>	D	x	
/tags	GJ		
/tags/count	GJ		

1.1.2 Misc calls

URL	VERBS	a?	Notes
/datafiles	GJ		Get public data files
/datadirs	GJ		Get public data dirs
/usergroups	GJ		Get public user groups
/usergroups/new	S		Create new public user group
/news	GJ		Get site news items
/news/new	S		Create new news item
/news/id	D		Delete news item
/users	GJ		Get list of users of site
/newuser	S		Create new user

1.1.3 Usergroups

NOTE: These calls are likely to change.

All URLs start with **/usergroup/<ug>**

URL	VERBS	a?	Notes
/datafiles	GJ		Get public dfiles for usergroup
/users	GJ		Get users in usergroup
/username/<u>	P		Add user to usergroup
/username/<u>/remove	P		Remove user from usergroup

1.1.4 User Account Services

All URLs start with **/user/<u>**

URL	VERBS	a?	Notes
/user/<u>	GJ	x	Retrieve user info
/<pw>/apikey	GJ		Retrieves the APIKEY
/password/<newpw>	P	x	Change password
/apikey/reset	P		Reset the API key
/usergroups	GJ	x	Get users usergroups
/all$_{usergroups}$	GJ	x	Retrieve all usergroups

1.1.5   User Datadir services

These calls do not download actual files or directories . They retrieve the data structures describing them.

All URLs start with **/user/<u>**

URL	VERBS	a?	Notes
/datadir/<ddid>	GJ	x	Get Datadir
/datadirs	GJ	x	Get all datadirs
/datadirs/datafiles	GJ	x	Get dirs and files
/datadirs/tree/groups	GJ	x	Get tree of groups datadirs
/datadirs/tree	GJ	x	Get tree of user's datadirs

1.1.6   User Datafile services

These calls do not download actual files or directories . They retrieve the data structures describing them.

All URLs start with **/user/<u>**

URL	GET	a?	Notes
/datafiles	GJ	x	Retrieves user's datafiles
/datafiles/tag/<tag>	GJ	x	Tags used on all users datafiles
/datafiles/update/<dfid>	P	x	Update data on datafile
/tags/count	GJ	x	Get tags/counts for datafiles
/datafile/<dfid>	GJ	x	Get datafile if user has access

1.1.7   User Review Services

All URLs start with **/user/<u>**

URL	VERBS	a?	Notes
/reviews	GJ	x	Reviews for user
/reviews/requested	GJ	x	Reviews requested by user
/review/<id>	D	x	Delete review
/review	S	x	Create a review
/datafile/reviews/<dfid>	GJ	x	Reviews for datafile
/review/<id>/mark/<as>	P		Mark review as done/not done

1.1.8 User Upload/Download Services

All URLs start with **/user/<u>**

URL	VERBS	a?	Notes
/udqueue	GJ	x	List user's upload/download queue
/upload/<ddid>	S	x	Upload a file to datadir
/download/file/<dfid>	G	x	Download a user's file