Zotero Data Server Installation on Debian Linux

Altynbek Isabekov, Eric Fernandez

June 20, 2013

Abstract

This tutorial explains how to install Zotero data server on a host running Debian Wheezy (7.0RC1).

1 Introduction

2 Installing Zotero Data Server

2.1 Retrieving Sources

Change directory to `/srv/http`:

```
$ mkdir /srv/http
$ cd /srv/http/
```

Dowload Zotero data server source code from github repository:

```
$ sudo git clone --recursive https://github.com/Panzerkampfwagen/
    dataserver.git
```

Rename the directory (important! the directory named is used in the source code):

```
$ sudo mv dataserver ZoteroDataServer
```

Download Zend framework:

```
$ sudo wget -vc "http://framework.zend.com/releases/ZendFramework
    -1.12.3/ZendFramework-1.12.3.tar.gz"
```

Decompress part of the archive (ZendFramework-1.12.2/library/Zend) to the "include" directory (`/srv/http/ZoteroDataServer/include/Zend`):

```
$ sudo tar -xvf ZendFramework-1.12.3.tar.gz --strip=3  -C "/srv/http/
    ZoteroDataServer/include/Zend" "ZendFramework-1.12.3/library/Zend"
```

2.2 Configuring Apache HTTP Server and PHP Engine

Install Apache server:

```
$ sudo aptitude install apache2 php5
```

Create a virtual host for Zoter data server. To do that, append the following configuration into `/etc/apache2/sites-available/default`. The directory for data server is `/srv/http/ZoteroDataServer/`.

```
NameVirtualHost *:85
<VirtualHost *:85>
ServerName *:85
        DocumentRoot "/srv/http/ZoteroDataServer/htdocs"
        ErrorLog "${APACHE_LOG_DIR}/Zotero-Data-Server-Error.log"
        CustomLog "${APACHE_LOG_DIR}/Zotero-Data-Server-Access.log"
            common
```

```
        <Directory "/srv/http/ZoteroDataServer/htdocs">
                Options FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                Allow from all
        </Directory>
        AllowEncodedSlashes On
</VirtualHost>
```

Enable the following modules:

```
$ sudo a2enmod rewrite
$ sudo a2enmod vhost_alias
```

Make Apache server listen to port number 85 which points to Zotero virtual host by adding the following line to `/etc/apache2/ports.conf`:

```
NameVirtualHost *:85
Listen 85
```

Start Apache server and check it runs OK:

```
$ sudo service apache2 start
```

2.3 Configuring MySQL

Install MySQL:

```
$ sudo aptitude install php5-mysql mysql-server
```

Zotero data server is configured to use 'SecurePassword' as a root password. You have to reset root password to "SecurePassword" or modify source codes of the data server. To reset the root password, stop the MySQL daemon if it is running:

```
$ sudo service mysql stop
```

Restart MySQL daemon and bypass authentication:

```
$ sudo mysqld_safe --skip-grant-tables &
```

Connect to the mysql server

```
$ sudo mysql -u root mysql
```

Change root password:

```
mysql> UPDATE mysql.user SET Password=PASSWORD('SecurePassword') WHERE
    User='root';
mysql> FLUSH PRIVILEGES;
mysql> EXIT
```

Scripts that create Zotero database need authentication information. When script logs into mysql shell, the credentials will be taken from `/etc/mysql/my.cnf` file. To add password information, modify `/etc/mysql/my.cnf`:

```
[client]
user=root
password=SecurePassword
```

Change MySQL time zone to UTC by modifying the following line in `/etc/mysql/my.cnf`:

```
[mysqld]
...
default-time-zone = '+0:00'
```

Install phpMyAdmin, a convenient tool for MySQL administration using web interface:

```
$ sudo aptitude install phpmyadmin
```

Restart Apache server:

```
$ sudo service apache2 restart
```

2.4 Configuring Zotero Data Server

Main settings are stored in `ZoteroDataServer/include/config/config.inc.php` file:

```
...
public static $SYNC_DOMAIN = '127.0.0.1:85';
...
public static $CLI_PHP_PATH = '/usr/bin/php';
public static $CLI_DOCUMENT_ROOT = "/srv/http/ZoteroDataServer/";
...
```

Sync domain is configured to localhost so it won't accept connections from other IPs. But the goal is to make data server work at least on local host. Later, when the server is working properly, you will need to change the IP address to your machine IP on the network to make the server available to other client machines on the same network.

If you want to change the data server root directory by editting `$CLI_DOCUMENT_ROOT` do not forget the trailing slash "/" at the end of the path.

Interaction with MySQL database through PHP is done using credentials in `ZoteroDataServer/include/config/config.inc.php` file. You can change default root password ("SecurePassword") only by editting this file.

2.5 Setting Up Zotero Data Server

Start the MySQL daemon: Start Apache server and check it runs OK:

```
$ sudo service mysql start
```

Install memcached:

```
$ sudo aptitude install memcached
```

Change directory to `/srv/http/ZoteroDataServer/misc`

```
$ cd /srv/http/ZoteroDataServer/misc
```

Run `test_reset` script which deletes all existing Zotero data server databases (if they exist) and creates new databases from scratch.

```
$ sudo ./test_reset
Deleting databases
Creating databases
Deleting users
Creating users
Updating user privileges
Filling databases with default fields
Reset is successfull. Now run ./test_setup
```

You should see "Reset is successfull. Now run ./test_setup" message. Now you can run `test_setup` script which adds some items to the `zoterotest1` database:

```
$ sudo ./test_setup
```

If you see "Test setup is successfull." this means items were successfully added to the database.

To check this, log into MySQL:

```
$ mysql -u root -pSecurePassword
```

Now check for the created databases:

```
$ mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| mysql              |
| zotero_www         |
| zotero_www_test    |
| zoterotest1        |
| zoterotest2        |
```

```
| zoterotest_ids     |
| zoterotest_master  |
+--------------------+
8 rows in set (0.01 sec)
```

```
$ mysql> USE zoterotest1;
Database changed
$ mysql> SELECT * FROM items;
+--------+-----------+------------++----------+---------+
| itemID | libraryID | itemTypeID || key      | version |
+--------+-----------+------------++----------+---------+
|      1 |         1 |          2 || AAAA2222 |       0 |
|      2 |         2 |          2 || BBBB2222 |       0 |
|      3 |         2 |         14 || CCCC4444 |       0 |
|      4 |         3 |          2 || CCCC2222 |       0 |
|      5 |         3 |         14 || CCCC3333 |       0 |
+--------+-----------+------------++----------+---------+
5 rows in set (0.00 sec)
```

Change permissions for Zotero dataserver tmp directory:

```
$ sudo chmod 777 /srv/http/ZoteroDataServer/tmp
```

Now it is possible to open three terminals and start upload, download and error daemons separately in foreground: In the 1st terminal:

```
$ cd /srv/http/ZoteroDataServer/processor/upload/
$ php daemon.php
```

In the 2nd terminal

```
$ cd /srv/http/ZoteroDataServer/processor/download/
$ php daemon.php
```

In the 3rd terminal

```
$ cd /srv/http/ZoteroDataServer/processor/error/
$ php daemon.php
```

Verify that each of the processor daemon has started (in this case, the download daemon):

```
2013-06-20 11:16:48.5637 Starting download processor daemon
2013-06-20 11:16:48.5754 0 processors, 0 queued processes
```

Note that in case of an exception, error or crash, these daemons will stop and Zotero data server will be unavailable for sync until they are manually restarted. In particular, the download and upload daemons crash if MySQL database access has timed out, which occurs after 8 hours of inactivity by default. To prevent this, it is possible to increase the `wait_timeout` setting in `/etc/mysql/my.cnf` to a maximum of 1 year (31536000 seconds):

```
[mysqld]
...
wait_timeout = 31536000
```

Alternatively to changing wait_timeout, install daemontools:

```
$ sudo aptitude install daemontools
```

Then within each download, upload and error directory, type: In the 1st terminal:

```
$ cd /srv/http/ZoteroDataServer/processor/upload/
$ sudo supervise .
```

In the 2nd terminal

```
$ cd /srv/http/ZoteroDataServer/processor/download/
$ sudo supervise .
```

In the 3rd terminal

```
$ cd /srv/http/ZoteroDataServer/processor/error/
$ sudo supervise .
```

The program `supervise` will automatically run the ./run file in each directory and restart the daemon in case of crash. This has also the advantage to restart the daemons for other crash reasons.

2.6 Testing Zotero Data Server

Install curl:

```
$ sudo aptitude install curl
```

To test authentication on the server, post the following request:

```
$ curl -X POST -d "version=9&username=testuser&password=testuser"
     http://127.0.0.1:85/login
```

It should return something like:

```
<?xml version="1.0"?>
<response version="9" timestamp="1341134959">
  <sessionID>
     da802280ce0bfc2e90cb1ad0747ff642
  </sessionID>
</response>
```

Now using provided "sessionID" you can post request to "updated" action:

```
$ curl -X POST -d "version=9&sessionid=da802280ce0bfc2e90cb1ad0747ff642&
    lastsync=1" http://127.0.0.1:85/updated
```

the answer will be similar to:

```
<?xml version="1.0"?>
<response version="9" timestamp="1341135305" userID="1"
      defaultLibraryID="1" updateKey="43d4eaa497ab8cbfc8f4d201d955fd70"
         earliest="1341131740">
<updated/>
</response>
```

```
$ cadaver http://127.0.0.1/zotero
Authentication required for WebDAV on server `127.0.0.1':
Username: testuser
Password:
dav:/zotero/> ls
Listing collection `/zotero/': succeeded.
        3QBQSD38.prop                       117  Aug 12 13:21
        3QBQSD38.zip                     465149  Aug 12 13:21
        5BA5I3IP.prop                       117  Aug 12 13:21
        5BA5I3IP.zip                        809  Aug 12 13:21
        8IF963XB.prop                       117  Aug 12 13:21
        8IF963XB.zip                     860702  Aug 12 13:21
        lastsync                              1  Aug 12 13:21
dav:/zotero/>
```

3 Installing Zotero Attachment Server

In order to be able to upload attachment files you need to provide your Zotero client with a WebDAV server. Distributed authoring and versioning (WebDAV). For example, you can use Yandex.Disk with WebDAV protcocol. It provides 5 GB disc space for free! Files are also accessible through its e-mail web-client. If you are paranoid enough not to trust anyone, you can set up a WebDAV server by yourself.

Enable the following modules:

```
$ sudo a2enmod auth_digest
$ sudo a2enmod dav
$ sudo a2enmod dav_fs
$ sudo a2enmod dav_lock
```

Now create WebDAV lock directory and file:

```
$ sudo mkdir /srv/http/DAVLock
$ sudo chmod -R 777 /srv/http/DAVLock
$ sudo chown -R www-data:www-data /srv/http/DAVLock
```

Create directory to store uploaded Zotero attachments:

```
$ sudo mkdir /srv/http/zotero
$ sudo chown -R www-data:www-data /srv/http/zotero
$ sudo chmod -R 777 /srv/http/zotero
```

Add the following configuration into `/etc/apache2/sites-available/default`:

```
DavLockDB /srv/http/DAVLock/DAVLockDB
```

Then put the following configuration into `/etc/apache2/sites-available/default` under the `<VirtualHost *:80>` section:

```
DocumentRoot /srv/http
<Directory "/srv/http/zotero">
    Dav On
    Order Allow,Deny
    Allow from all
    AllowOverride None
    AuthType Digest
    AuthName "WebDAV"
    AuthUserFile "/var/www/.DAVlogin"
    AuthDigestProvider file
    Require user "testuser"
</Directory>
```

Now create an MD5 hash for user "testuser" in realm "WebDAV" authorized by some password and store it in `/etc/httpd/conf/extra/AuthWebDAV.passwd` :

```
$ sudo htdigest -c /etc/httpd/conf/extra/AuthWebDAV.passwd WebDAV
   testuser
Adding password for testuser in realm WebDAV.
New password:
Re-type new password:
```

Restart Apache server:

```
$ sudo service apache2 restart
```

In order to test your WebDAV server you have to install "cadaver" package:

```
$ sudo aptitude install cadaver
```

Now connect to your WebDAV server using "cadaver" program:

```
$ cadaver http://127.0.0.1/zotero
Authentication required for WebDAV on server `127.0.0.1':
Username: testuser
Password:
dav:/zotero/> mkcol SomeCollection
Creating `SomeCollection': succeeded.
dav:/zotero/>
```

4 Installing Zotero Client

4.1 Zotero Firefox Extension

Install Firefox extension from `ZoteroDataServer/misc/zotero-3.0.14-patched.xpi`. It is patched such that you can add custom data servers as shown in Figure 1.

5 Troubleshooting

5.1 Lost connection to MySQL server during query

In case you need to restore a very large library to your server, e.g. from a backup, upload can fail after sync attempt with this error: “Lost connection to MySQL server during query" (visible in the Zotero client plugin log). To prevent this, modify the following line in `/etc/mysql/my.cnf`:

```
[mysqld]
...
max_allowed_packet = 64M # default is 16M
```

5.2 MySQL server has gone away

This error, which appears in the download or upload daemon output, happens if MySQL database access has timed out, which occurs after 8 hours of inactivity by default. To prevent this, it is possible to increase the `wait_timeout` setting in `/etc/mysql/my.cnf` to a maximum of 1 year (31536000 seconds):

```
[mysqld]
...
wait_timeout = 31536000
```

Alternatively, use supervise from daemontools to restart daemons in case of a crash, as described above. This will also solve issues where the daemons crash for alternate reasons.

6 Debugging

6.1 Capturing Traffic

You can capture the traffic using Wireshark. Select loopback interface (lo) to filter all other IP addresses except localhost (127.0.0.1). You should change HTTP port preferences in Wireshark to be able to analyze traffic. In Wireshark -> Preferences -> Protocols -> HTTP -> TCP Ports add port 85. In the capture filter field, enter “http”. Turn on capture.

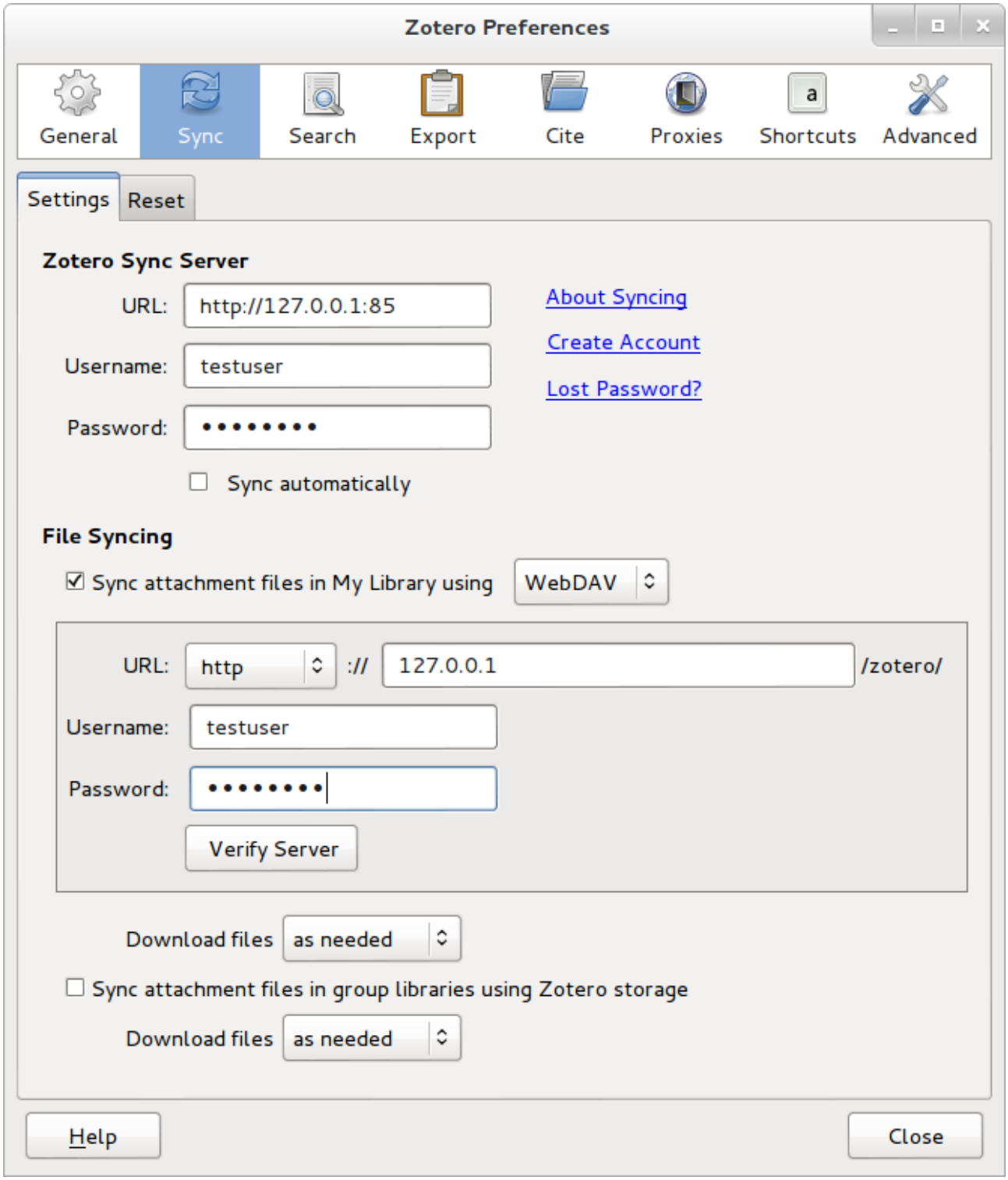

Figure 1: Zotero preferences for data server.