

Setting Up Your Own Mozilla Sync Server

Mozilla Sync is a feature which is supported in Firefox since version 4. It allows you to store your bookmarks, browsing history, browser preferences, and saved passwords on a remote server. It even allows you to work on multiple computers and have the same Firefox configuration on all these computers. That's a pretty cool feature. You can for instance setup sync on your Android phone's Firefox browser and quickly configure it by entering your account-specific sync key. Moments later you will have your synced desktop browser configuration on your phone.

Mozilla offers public sync servers. If you don't trust your passwords and browsing history in the hands of Mozilla (though your data is supposedly being encrypted with your unique sync key before it gets uploaded to Mozilla), then you can use this HOWTO for setting up your own Sync Server at home (or at work).

Luckily, Mozilla is open as always and makes the Open Source Sync Server code available to everyone who wants to play with it. And setting it up is not difficult at all.

Note: Mozilla have re-designed and re-written the original Sync Server (it was called "*Mozilla Weave Server*" at the time, and was almost impossible to setup). It is now written in Python, and the steps to create your own Sync Server are [described on this page](#). That page is what you should follow. The rest of this HOWTO tries to bring more detail to the process of configuring your Slackware box to become a real Sync Server.

Of course you want to install this on a Slackware server!

Requirements

There are some explicit dependencies as mentioned on the [Mozilla page](#):

- python 2.6 (part of Slackware 13.37. I have not tested with Slackware 14 which upgraded to python 2.7)
- sqlite3 (already supplied by Slackware as well)
- [virtualenv](#). Slackware packages can be downloaded here: <http://slackware.com/~alien/slackbuilds/virtualenv/> or built using the [SlackBuilds.org script](#) .

The Sync Server code is hosted in a Mercurial repository and since Slackware ships a Mercurial client, that code is easy to check out to a local directory where you continue to build the server binary:

```
$ mkdir -p /usr/local/server-full
$ cd /usr/local/server-full
$ hg clone https://hg.mozilla.org/services/server-full/ .
$ make build
```

Now, this server is ready to run, but unless you make some configuration changes it is not going to do much good:

- By default, the server stores your sync data in a sqlite database (you can use a real MySQL server database if you want), but it will create the sqlite database in /tmp which is entirely

inappropriate. Further down you will find what you have to modify so that your clients' sync data go somewhere less "volatile".

- The python-based Sync Server can run all by itself on a configurable port number (5000 by default) which is just fine if you are its only user in your little LAN. But even Mozilla advises to put a real webserver between you and the Sync Server. In Slackware you can use the apache webserver for this. You must install an additional apache module for this to work: [mod_wsgi](#). Slackware packages plus SlackBuild script can be downloaded here: http://slackware.com/~alien/slackbuilds/mod_wsgi/ or get a alternative SlackBuild script from slackbuilds.org.

Configuration

- Create a separate system user and group (weave:sync) which is going to run the Sync Server so that it can not interfere with Apache if anything goes crazy:

```
# groupadd sync
# useradd -g sync weave
```

If you omitted the creation of the home directory for this "weave" user account, then create it like this:

```
# mkdir /home/weave
# chown weave:sync /home/weave
# chmod 711 /home/weave
```

- Create a new file "/etc/httpd/extra/httpd-wsgi.conf" with the following content (adapt to your actual sync server hostname):

```
<Directory /usr/local/server-full>
    Order deny,allow
    Allow from all
</Directory>

<VirtualHost *:80>
    ServerName sync.myserver.net
    DocumentRoot /usr/local/server-full
    WSGIProcessGroup sync
    WSGIDaemonProcess sync user=weave group=sync processes=2
threads=25
    WSGIPassAuthorization On
    WSGIScriptAlias / /usr/local/server-full/sync.wsgi
    CustomLog /var/log/httpd/sync.myserver.net-access.log combined
    ErrorLog /var/log/httpd/sync.myserver.net-error.log
</VirtualHost>
```

Apache 2.4 uses a different syntax for access control:

```
<Directory /usr/local/server-full>
    Require all granted
```

```

</Directory>
<VirtualHost *:80>
    ServerName sync.myserver.net
    DocumentRoot /usr/local/server-full
    WSGIProcessGroup sync
    WSGIDaemonProcess sync user=weave group=sync processes=2
threads=25
    WSGIPassAuthorization On
    WSGIScriptAlias / /usr/local/server-full/sync.wsgi
    CustomLog /var/log/httpd/sync.myserver.net-access.log combined
    ErrorLog /var/log/httpd/sync.myserver.net-error.log
</VirtualHost>

```

This tells apache that a script called “sync.wsgi” is going to be spawned as user “weave:sync” whenever a client connects to `http://sync.myserver.net/`.

- Modify your apache server's main configuration file “/etc/httpd/httpd.conf”, in order to make it load the mod_wsgi module and use its configuration file. Add the following lines to the `httpd.conf`:

```

# Activate the WSGI module (serving python binaries to the web):
Include /etc/httpd/extra/mod_wsgi.conf
# Act as a reverse proxy to the Mozilla Sync server:
Include /etc/httpd/extra/httpd-wsgi.conf

```

- Check your Apache configuration for syntax errors which you may have introduced:

```
# apachectl configtest
```

- Restart Apache:

```

# /etc/rc.d/rc.httpd stop
# pause 3
# /etc/rc.d/rc.httpd stop
# /etc/rc.d/rc.httpd start

```

- Configure the Sync Server configuration files.
 - Inside the directory “/usr/local/server-full” (where the Sync Server code has been downloaded and built if you followed this HOWTO to the letter) you will find the aforementioned “sync.wsgi” script. Open the file in an editor. Read through the file and change it to something which you find better suited. For instance, change “/tmp” to “/var/tmp” in the line “`os.environ['PYTHON_EGG_CACHE'] = '/tmp/python-eggs'`”:

```
os.environ['PYTHON_EGG_CACHE'] = '/tmp/python-eggs'
```

changes to

```
os.environ['PYTHON_EGG_CACHE'] = '/var/tmp/python-eggs'
```

- Next, edit the file “development.ini” (that filename is configured in “sync.wsgi” in case you want to use another name). In that file change the location of the error logfile so that it is created in “/var/tmp”. The line

```
args = ('/tmp/sync-error.log',)
```

changes to:

```
args = ('/var/tmp/sync-error.log',)
```

- The “development.ini” file also mentions the name of the actual sync configuration file “etc/sync.conf”. Note that this path is relative to the Weave server’s root directory. It requires some editing! For instance, create the SQL database file in the “weave” user’s own home directory. The line

```
sqluri = sqlite:///tmp/test.db
```

becomes:

```
sqluri = sqlite:///home/weave/my_weave_server.db
```

The default value for the quota assigned to each sync account is too low (it’s set to 5 MB). Mozilla states that 99.9% of all clients which connect to its public server use less than 25 MB of quota, so use that value as a default. The line

```
quota_size = 5120
```

changes to:

```
quota_size = 25600
```

Then, look for the following line and make absolutely certain that it is changed to the Apache http:// URL of your Sync Server, i.e. the URL that the sync clients are going to use:

```
fallback_node = http://localhost:5000/
```

You can play with other options in this file to see what they can offer. For instance, you can disable “captcha” and not enable SSL, but you may decide that this is important to your setup.

Don’t forget to restart your apache server if you have not yet done so.

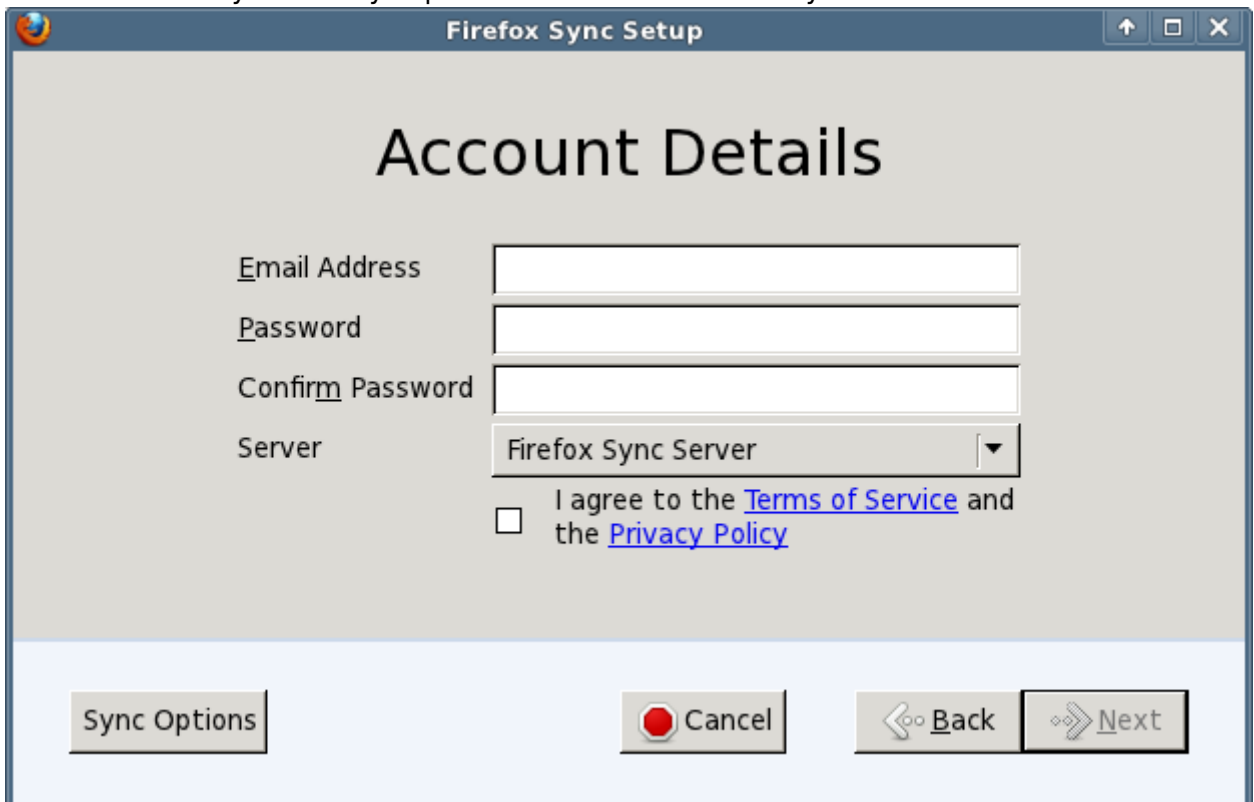
USing the Sync Server

And now it is time to tell your Firefox browser to use this new service!

- Select “Tools > Setup sync ...”:



- In case this is the first time you talk to your Sync Server, you need to setup an account, so click the button "Create a New Account". This opens a new dialog box where you can type your email address (this will become your account name), an associated password and then in the dropdown menu, do **not** select "Firefox Sync Server". Instead, you choose "Use a custom server". This allows you to enter the URL for your brand-new private Sync Server. Once the account has been created, Firefox will show you the sync key it will be using to encrypt your data. That key is important, you will need it if you want to add more computers to your sync account. And that's basically it - the sync process will start immediately:



- Else if you already have a Sync Account and want to add a second computer to it, you can click on "Connect" in the first dialog which will open a new dialog. There are two ways to add this computer to your existing sync account:
 - Either you enter the code which is being shown on another computer which is already syncing to the server;
 - Or else you click on "I don't have the device with me". This is what you would usually do. It is cumbersome to search for another computer or hand-held if all you need to know is your sync key:



- If you know your sync key you can enter that in the following dialog, along with your account details and the server URL:



The sync key which is associated with your sync account can be retrieved at any moment should you have forgotten to write it down. Just go to a computer which has been setup for Sync, and open your Firefox browser's Preferences at the "Sync" tab. Under "Manage your account" you will find a menu item to display your key (as well as a way to change your account's password). More help can be found at <http://support.mozilla.com/kb/what-firefox-sync> .

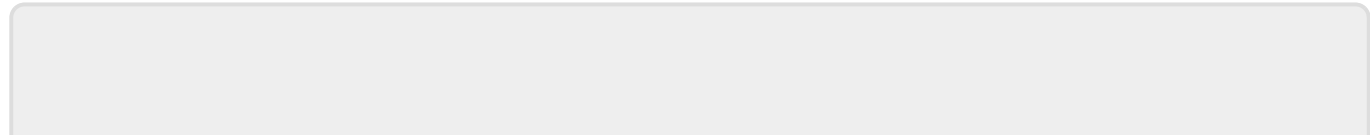
Securing your Sync Server with SSL

Fix Me! - This section is coming soon! You really want your Sync server to be accessible over https:// if you sync over the Internet.

Sources

- Originally written by [Eric Hameleers](#)
- Original source: [Alien Pastures Blog](#)

[howtos](#), [software](#), [sync](#), [weave](#), [author alienbob](#), [slackware 13.37](#)



From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

https://docs.slackware.com/howtos:network_services:weave

Last update: **2013/07/09 21:07 (UTC)**

