
Labfolder at FHI Documentation

Release 1.0

Stefan Weiher

Jun 04, 2018

CONTENTS:

1	Installation of labfolder	1
2	Groups, ownership and sharing in labfolder	5
2.1	Private projects	5
2.2	Group projects	5
2.3	Group settings	6
3	Import and export of projects and entries	7
3.1	Import	7
3.2	Export	7
4	Importing projects from e.g. the ELOGbook to Labfolder at the FHI	9
4.1	Creating a group project	9
4.2	Entries from ELOG	9
4.3	Using python to import from ELOG to labfolder	10
4.4	Alternatives to manipulating the labfolder DB on MySQL level	11
5	Printing via CUPS and attaching files to labfolder logbooks	13
5.1	Set up CUPS as labfolder admin	13
5.2	Printing to labfolder projects	15

INSTALLATION OF LABFOLDER

Labfolder is easy to install. Follow the instructions given in the *labfolder Installation Manual*. The broad steps are: **install MySQL, docker and labfolder; configure labfolder via the server.cnf file**. This labfolder config file could look like this:

```
# Network settings
DEFAULT_HTTP_PROTOCOL=https://
DEFAULT_DOMAIN=labfolder.rz-berlin.mpg.de

# JDBC Properties
JDBC_SERVER_TIMEZONE=Europe/Berlin
JDBC_DATABASE_URL=jdbc:mysql://localhost:3306/labfolder?useUnicode=true&
↪characterEncoding=UTF-8
JDBC_USERNAME=labfolder
JDBC_PASSWORD=fhilab
JDBC_SCHEMA_NAME=labfolder
#REMOVE_ABANDONED_TIMEOUT=60

# Mail client properties
MAIL_STARTTLS=false
MAIL_AUTHENTICATION_ENABLE=false
MAIL_HOST=mail.fhi-berlin.mpg.de
MAIL_PORT=25
#MAIL_USERNAME=
#MAIL_PASSWORD=
MAIL_EMAIL=ppb@fhi-berlin.mpg.de

# Server Event logging
LOG_TO_FILE=true

#Maximum upload file size
FILEUPLOAD_MAXUPLOADSIZE=25000000

# User and group control
DEFAULT_GROUP_SIZE=50
DEFAULT_GROUP_TYPE_MAXI=true
DEFAULT_USER_STORAGE=3221225472
FEATURE_GLOBAL_PREVENT_DELETE_CONTENT=false

# Terms and Privacy links
TERMS_OF_USE_LINK=https://www.labfolder.com/terms-external-servers/
PRIVACY_LINK=https://www.labfolder.com/privacy-external-servers/

# LDAP Authentication
FEATURE_LDAP_AUTHENTICATION=true
```

```
LDAP_URL=ldap://ldap.rz-berlin.mpg.de:389
LDAP_BASE=ou=people,dc=ppb,dc=rz-berlin,dc=mpg,dc=de
#LDAP_SERVER_TYPE=
LDAP_USER_DN=cn=pwCheck,dc=rz-berlin,dc=mpg,dc=de
LDAP_PASSWORD=ProstetnikVogonJeltz
#LDAP_ANONYMOUS_READ_ONLY=
LDAP_USER_DN_PATTERNS=uid={0}
#LDAP_IS_TLS_ENABLED=
#LDAP_IS_ATTRIBUTE_SEARCH_ENABLED=
#LDAP_ATTRIBUTE_SEARCH_NAME=

# Usage monitoring
ACTIVE_USER_REPORT_USE_DEFAULT_MAIL_CLIENT=true
CUSTOMER_IDENTIFIER='Fritz-Haber-Institut'

# Mendeley
FEATURE_MENDELEY=false
#MENDELEY_CLIENT_ID=
#MENDELEY_CLIENT_SECRET=
#MENDELEY_CLIENT_CALLBACKURL=http://localhost:9091/el/mendeley/oauthCallback

# Dropbox
FEATURE_DROPBOX=false
#DROPBOX_CONSUMER_KEY=
#DROPBOX_CONSUMER_SECRET=

# Figshare
FEATURE_FIGSHARE=false

# XHTML Export
FEATURE_XHTML_EXPORT=true
EXPORT_DOCUMENT_REPOSITORY_TYPE=fileSystem
```

The configuration is explained in detail in the installation manual. Now, **create the labfolder database and run labfolder** (see the manual for details). Additionally, you could install Apache and set it up as a reverse proxy in order to make labfolder only available through https. The exact steps are:

```
labfolder@lf:~$ sudo apt-get install build-essential
labfolder@lf:~$ sudo apt install apache2
labfolder@lf:~$ sudo a2enmod proxy proxy_ajp proxy_http rewrite deflate headers proxy_
↪balancer proxy_connect proxy_html xml2enc
labfolder@lf:~$ sudo a2enmod ssl

### Use certificates from e.g. let's encrypt. Please see below for more information.

### create /etc/apache2/sites-available/labfolder.conf with virtual hosts for http_
↪and https:
labfolder@lf:~$ sudo a2ensite labfolder.conf
labfolder@lf:~$ sudo a2dissite 000-default.conf
labfolder@lf:~$ sudo service apache2 restart
```

To get SSL certificates from letsencrypt is quite straight forward. For example this guide shows how to set up SSL certificates:

<https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-16-04>

The Apache labfolder config file could look like this:

```
<VirtualHost *:80>
    ErrorLog /var/log/apache2/error.log
    LogLevel warn

    ServerName labfolder.rz-berlin.mpg.de
    Redirect / https://labfolder.rz-berlin.mpg.de/
</VirtualHost>

<VirtualHost *:443>
    ErrorLog /var/log/apache2/error.log
    LogLevel warn

    ServerName labfolder.rz-berlin.mpg.de

    SSLEngine On
    SSLCertificateFile /etc/letsencrypt/live/labfolder.rz-berlin.mpg.de/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/labfolder.rz-berlin.mpg.de/privkey.pem

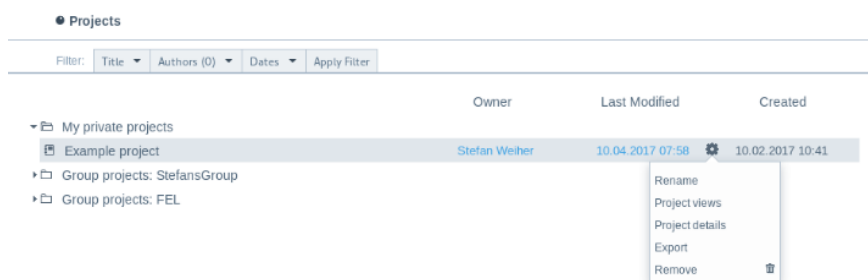
    ProxyPass / http://141.14.138.230:9091/
    ProxyPassReverse / http://141.14.138.230:9091/
</VirtualHost>
```


GROUPS, OWNERSHIP AND SHARING IN LABFOLDER

As labfolder user you can create groups (ultimately: group projects) and private projects. In *Manage > Projects* you see all projects that you own and that you are a member of.

2.1 Private projects

You alone are the owner of these projects in *My Private Projects*. Their ownership cannot be transferred from you to anyone else via the webinterface. Only via accepting the invitation to a group with the policy that private projects are not allowed the content of the former private project would be visible (and administrable) also to the group's admin(s). By directly manipulating the database the ownership could be changed to an active user as well.



Private projects are possible through a back door if a user creates their own group which doesn't necessarily mean that they invite other users to this group. In their own group they could create as many (private) projects as they like.

2.2 Group projects

Any labfolder user can create groups and subgroups in *Manage > Groups > Button "+Add"*. But within an already existing group a member can only create subgroups if they are set as an administrator of this group. In the role of the admin you can do the following:

- Create (sub)groups
- Invite users to your (sub)group via e-mail
- Move users between subgroups
- Set and remove (sub)group members as admins
- Remove users from a (sub)group e.g. when they are leaving the institute (**In this case the ownership of this user's projects within the (sub)group needs to be transferred first!**)
- Delete (sub)groups (before deleting it you need to remove all group members)

2.3 Group settings

Administrators of groups (but not the ones of the subgroups) can edit the general settings of groups.

labfolder • Groups > StefansGroup

NOTEBOOK MANAGE

General settings

Name StefansGroup [edit](#)

Description

Members rights [edit](#)

☐ Prevent group members from deleting content

☒ Prevent group members from having private projects

☐ Users can be members of several subgroups

- Through the option *Prevent group members from deleting content* all users (including the admin) are prevented from deleting projects or any other data from the labfolder database. Projects will merely be hidden if a user decides to *Remove* it from their list of projects.
- By checking the option *Prevent group members from having private projects*, group members will not be allowed to have private projects. All private projects of group members will be moved into the group projects, but will not be shared with any other group members.
- By switching on *Users can be members of several subgroups*, users which are already members of a group can be invited again to another subgroup within the group. If switched to off members cannot be invited to subgroups of their current group.

IMPORT AND EXPORT OF PROJECTS AND ENTRIES

3.1 Import

The import of projects (from another labfolder instance or from another logbook software) is only possible via the labfolder API. An import of labfolder-own projects via webinterface is not possible to date. You can find the API documentation here:

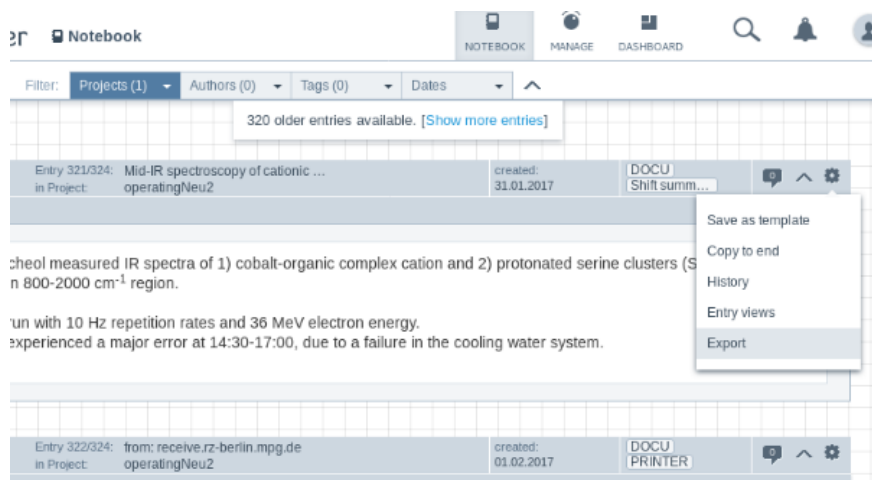
Version 1: <http://labfolder.rz-berlin.mpg.de:9091/api/v1>

Version 2: <https://labfolder.rz-berlin.mpg.de/api/v2/docs/development.html>

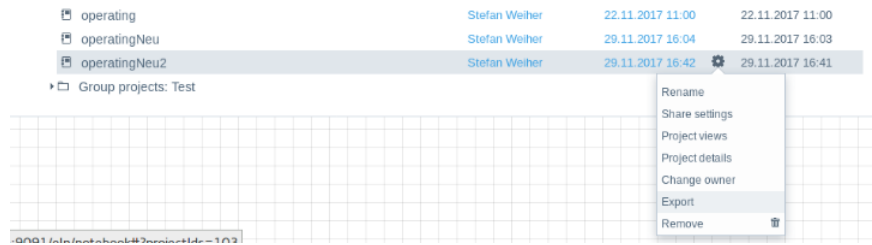
Version 2 is still in the development phase.

3.2 Export

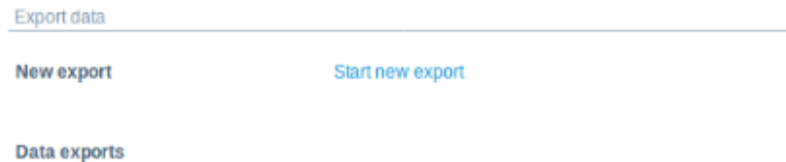
Exporting projects (or just single entries) is possible in three different formats: PDF, XHTML and JSON. The **export as PDF** can be done from within a project ...



... or from the project overview (*Manage > Projects*):



The **XHTML export** can be done via the settings:



However, to use this type of export it has to be installed first in the *Manage > Apps* section. By means of the XHTML export all projects that one owns will be exported. A detailed selection of projects or entries to be exported is not possible. As soon as the export process is finished a download link appears in the settings in *Data exports*. The downloadable ZIP file contains all images and files along with an `index.html` file that contains information about all entries. You could open and view the `index.html` file in a browser similar to the view in labfolder.

The **JSON export** is only useable via the API endpoint `GET /entries`. However, at the moment the response only contains useful information such as dates and tags. The author name is hidden behind the author ID and the entry text content is completely missing.

IMPORTING PROJECTS FROM E.G. THE ELOGBOOK TO LABFOLDER AT THE FHI

This is how you generally do it:

1. In labfolder, create a new group project.
2. Get the ELOG logbook "operating" and export it as RAW. Remember where you've put
→the text file with the raw output and the directory with attachments files.
3. Only once: Get the elog2labfolder repository from github.
4. Adapt elog2labfolder.py to your needs. Specify the name of the project that you
→created previously.
4. Run elog2labfolder.py

These steps are explained in more detail in the following.

4.1 Creating a group project

As the group admin (or one of the admins) in labfolder create a group project, for example, "operating" in a group named "FEL". You are now the owner of this project. Make sure you don't have several projects of the same name in one group.

4.2 Entries from ELOG

The ELOG entries of a certain project need to be exported as *Raw* which looks like this:

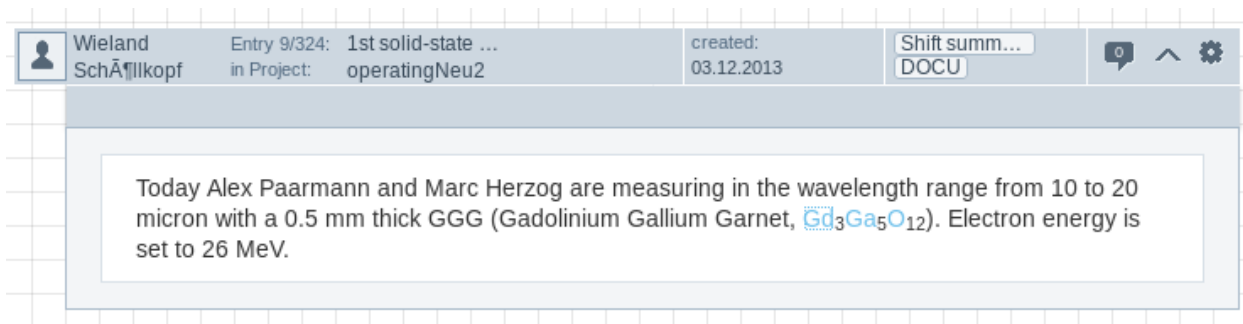
```
$@MID@$: 12
Date: Tue, 03 Dec 2013 12:15:14 +0100
Author: Wieland SchÃ¶llkopf
Author Email: mailto:wschoell@fhi-berlin.mpg.de
Category: Shift summary
Subject: 1st solid-state experiment with FHI FEL
Severity: DOCU
Keywords:
Record date: 1386069171
Attachment:
Encoding: HTML
=====
<p>Today Alex Paarmann and Marc Herzog are measuring in the wavelength range from 10_
→to 20 micron with a 0.5 mm thick GGG (Gadolinium Gallium Garnet, <a title=
→"Gadolinium" href="http://en.wikipedia.org/wiki/Gadolinium">Gd</a><sub>3</sub><a_
→title="Gallium" href="http://en.wikipedia.org/wiki/Gallium">Ga</a><sub>5</sub><a_
→title="Oxygen" href="http://en.wikipedia.org/wiki/Oxygen">O</a><sub>12</sub></sub>)._
→Electron energy is set to 26 MeV.</p>
```

The entry shown above looks like this in the ELOG webinterface:

Message ID: 12	Entry time: Tue Dec 3 12:15:14 2013
Author:	Wieland SchÄ¶llkopf
Author Email:	wschoell@fhi-berlin.mpg.de
Category:	Shift summary
Subject:	1st solid-state experiment with FHI FEL
Severity:	DOCU
Keywords:	
Record date:	Tue Dec 3 12:12:51 2013
Today Alex Paarmann and Marc Herzog are measuring in the wavelength range from 10 to 20 micron with a 0.5 mm thick GGG (Gadolinium Gallium Garnet, Gd3Ga5O12). Electron energy is set to 26 MeV.	
ELOG V2.9.2-2455	

What is noticeable is that the ELOG export comes with an “Ä¶” instead of an “ö”. Interestingly, if only entries done by Wieland are exported the export file has UTF-8 encoding which can actually print German “Umlaute” (ö, ä, ü). If, however, all entries are exported the encoding changes to an encoding that cannot handle e.g. “ö”.

After importing the above ELOG entry into labfolder this entry appears like that:



4.3 Using python to import from ELOG to labfolder

There is a GitHub project containing a python program for importing a raw (see the export example above) ELOG project. Get the project files by typing:

```
$ git clone https://github.molgen.mpg.de/weiher/elog2labfolder.git
```

The script *elog2labfolder.py* is meant to be executed from remote, that is, not on the machine on which labfolder is running. A prerequisite is the python MySQLDB package because for a few methods the labfolder API cannot be used, instead, direct manipulation of the labfolder MySQL database is needed. The actual labfolder API is applied in the methodes collected in *myPythonAPIv1.py* and *myPythonAPIv2.py*. Mostly, API version 2 is used but adding text at the end of an entry can only be accomplished by using the version 1 API call. *getListOfAuthors.py* gives you a list of all authors that are mentioned in a raw ELOG export file.

To finally import an ELOG logbook to a labfolder project you need to run `$ python elog2labfolder.py`. Prior to this you need to make some changes to the header of the file, that is, these lines:

```
### Set the hostname
#hostname = 'lftest'
hostname = 'labfolder'

if hostname == 'lftest':
    server = 'http://' + hostname + '.rz-berlin.mpg.de:9091'
elif hostname == 'labfolder':
    server = 'https://' + hostname + '.rz-berlin.mpg.de:9091'
```

```

### The entries will be created using the following labfolder account which specifies
↳ the author of the entry. During the creation process, however, the entry's author
↳ ID will be changed to the real author's ID if the real author is already
↳ contained in the user table of the labfolder DB.
email = 'weiher@fhi-berlin.mpg.de'
password = getpass.getpass('Enter the labfolder password associated with ' +
↳ emailAddress+': ')

### Specify the absolute path to the raw ELOG export file.
elogExportFile = '/home/stefan/Dokumente/labfolder/OperatingLogbook_exportRaw_
↳ ORIGINAL.txt'

### Full path to where the attachments of the logbook to be imported are stored:
attachmentsDirectory = '/home/stefan/Dokumente/labfolder/Operating/'

### Project name in labfolder
projectName = 'operating'

```

4.4 Alternatives to manipulating the labfolder DB on MySQL level

As mentioned previously: a few methods in the python API use a direct manipulation of the labfolder DB via MySQL queries. These methods are:

- *createUserAccount(email, author, entryContent, verboseOutput, cursor, db)*: If an author of an ELOG logbook entry does not yet have a labfolder account this account is created on the fly by setting a default (encrypted) password. This password is irrelevant because labfolder checks the entered password with the one from the LDAP server. Note: when setting a password in the DB labfolder's LDAP authentication needs to be disabled (FEATURE_LDAP_AUTHENTICATION=false) beforehand and enabled again after the setting process is finished.
- *updateAuthor(userID, blockID, cursor, db)*: Update the author of an entry. With the script *elog2labfolder.py* an entry is created with the author that is the person whose labfolder credentials are used to get the login token. Via this method the true author can be set to the passed in *userID*.
- *updateDateAndTime(date, blockID, cursor, db)*: Similar to *updateAuthor()* this method sets the creation and modification date to the creation time and date of the entry to be imported. This is necessary because otherwise creation and modification date in labfolder are the time and date when the import took place.

The labfolder support suggested the following alternative involving using the labfolder API:

- **User account**: A user account can be created via the API 2 endpoint `/auth/signup`. The password will be an encrypted dummy password. In order for this to work the LDAP authentication must be turned off when this endpoint is called. After the account creation LDAP authentication should be turned on again. Once the true user of this account logs in labfolder will ignore the dummy password and check the entered password against the password stored in LDAP.
- **Author**: To create a labfolder entry by using the original entry's author needs a special work-around. **First, make sure to prevent labfolder users from accessing labfolder for the time of the import.** Then, manually change all user passwords in the DB to the same (encrypted) password. Switch off the LDAP authentication and restart labfolder. Now, import entry by entry and get a new login token for each new entry. It is only possible to get login tokens for all users because they all have the same known password. If the author of the entry to be imported does not exist yet in the DB you need to create an account via the API 2 endpoint `/auth/signup`. Now you reopen the access to the labfolder instance and switch on LDAP again restarting labfolder afterwards. The users can login using their FHI credentials no matter the "universal" password in the DB. If LDAP authentication is not turned on the users could do a password recovery on the login screen and re-set their password to their liking.

- Date and time: Since you cannot change creation and modification time stamp of an entry via the API you could instead create custom dates during the import of an entry. The original entry creation timestamp could be “Original entry creation” followed by the creation time and date of the import. All entries can be filtered and searched by the custom dates.

PRINTING VIA CUPS AND ATTACHING FILES TO LABFOLDER LOGBOOKS

Follow this guide to set up CUPS virtual printers to be able to print webpages and pictures/files (PDF, PNG, JPG, PS) to one of labfolder's logbooks, thus, creating a new entry with an attachment. If you are an admin of the labfolder VM read the section *Set up CUPS as labfolder admin*. Users of labfolder read the section *Printing to labfolder projects*.

5.1 Set up CUPS as labfolder admin

First, you need to install CUPS on the virtual machine that runs labfolder:

```
sudo apt-get install cups cups-client cups-bsd
```

Next, configure `/etc/cups/cupsd.conf`:

```
# Sample configuration file for the CUPS scheduler.  See "man cupsd.conf" for a
# complete description of this file.

# Log general information in error_log - change "warn" to "debug"
# for troubleshooting...
LogLevel debug

# Deactivate CUPS' internal logrotating, as we provide a better one, especially
# LogLevel debug2 gets usable now
MaxLogSize 0

# Administrator user group...
SystemGroup lpadmin
ServerAdmin <name>@fhi-berlin.mpg.de

HostNameLookups On
ServerAlias *

# Only listen for connections from the local machine.
Listen lftest.rz-berlin.mpg.de:631
Listen /var/run/cups/cups.sock

# Show shared printers on the local network.
Browsing Off
BrowseOrder allow,deny
BrowseAllow all
BrowseLocalProtocols CUPS dnssd
BrowseAddress @LOCAL
```

```
# Default authentication type, when authentication is required...
DefaultAuthType Basic

# Web interface setting...
WebInterface Yes

# Restrict access to the server...
<Location />
    Satisfy All
    Allow localhost
    Allow 141.14.*
    Allow 172.16.*
    Allow 192.*
    Allow all
    # Allow shared printing and remote administration...
    Order allow,deny
    Allow all
</Location>

# Restrict access to the admin pages...
<Location /admin>
    Order deny,allow
    Satisfy All
    AuthType Basic
    Require user @SYSTEM
    Deny All
    Allow localhost
    Allow 141.14.*
    Allow 172.16.*
</Location>

# Restrict access to configuration files...
<Location /admin/conf>
    Order deny,allow
    AuthType Default
    Require user @SYSTEM
    Satisfy All
    Require user root
    Deny All
    Allow 127.0.0.1
    Allow 141.14.*
    Allow 172.16.*
</Location>

# Set the default printer/job policies...
<Policy default>
    ... # probably as in the original
</Policy>
```

Change the *Server Admin*'s e-mail address and the server that shall be listened to (see the `Listen` statement). Set `FileDevice Yes` in `/etc/cups/cups-files.conf`. Now, create a directory where all printers will go, e.g. `/usr/local/labfolder/printers`.

Get the package for creating new printers from git:

```
labfolder@lftest:~$ git clone https://github.molgen.mpg.de/weiher/cupsPrinters
labfolder@lftest:~$ ls
```

```
cupsPrinters labfolder
```

The only file you need to adapt is `newPrinter.sh`. It is not a must but it is suggested to name the printers like the projects. For example, the FEL projects *operating* and *installation* could be placed at `.../printers/FEL/operating` and `.../printers/FEL/installation`. Execute the script with `sudo` rights:

```
labfolder@lftest:~/cupsPrinters$ sudo ./newPrinter.sh
```

To prevent DENIED errors from `apparmor` install and set the following and restart the CUPS server:

```
$ sudo apt-get install apparmor-utils apparmor-profiles
$ sudo aa-complain cupsd
$ sudo service cups restart
```

In case you need to debug some logfiles can be found here: `/var/log/syslog`, `/var/log/cups/error_log` and `/var/log/cups/acces_log`

One last change is that you need to once add a user named *Printer Print* (firstname lastname) to the labfolder database. This can be done manually inside the DB by this command:

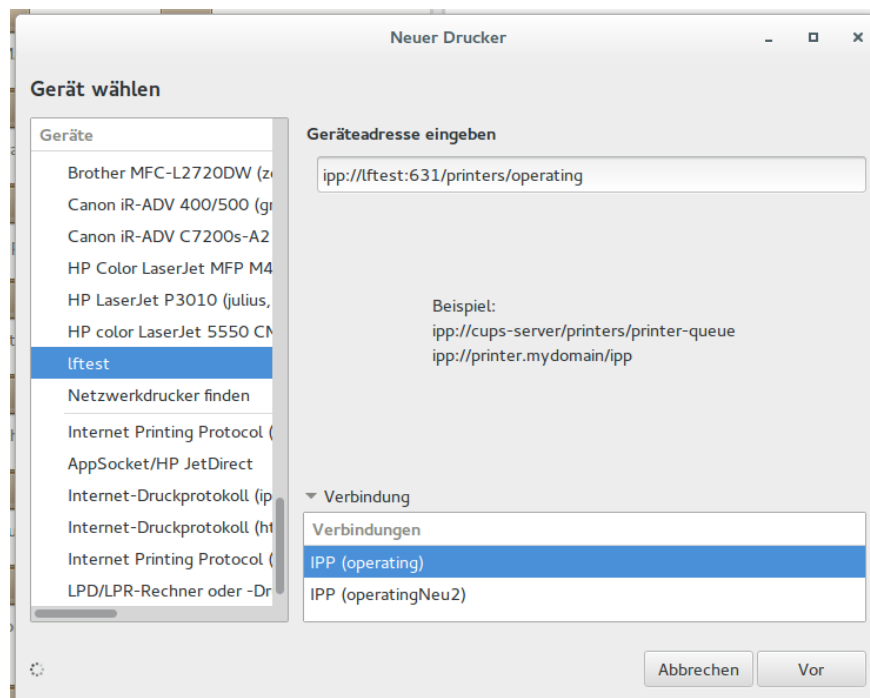
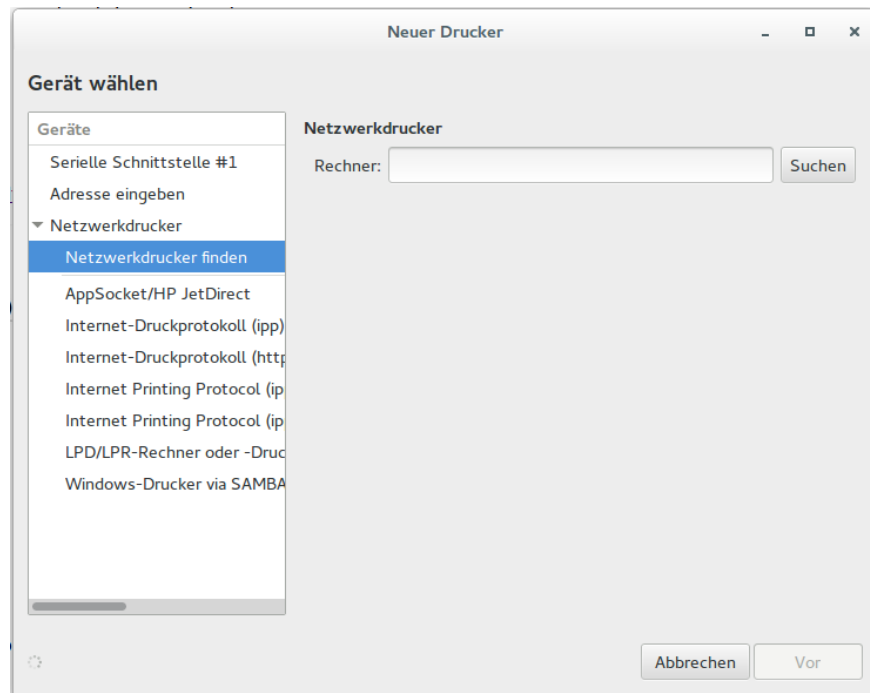
```
$ mysql -u root -p
Enter password:
...
mysql> use labfolder;
mysql> INSERT INTO user (email, password, firstname, lastname, signupTimestamp, ↵
↵maxStorage) VALUES ('printer@fhi-berlin.mpg.de',
↵'91a13f2aba4a35b00c35dfacadd1d75f4a5171dfa7a465fca9c40e6b007ce982', 'Printer',
↵'Print', '2018-03-21 10:54:00', '10000000');
```

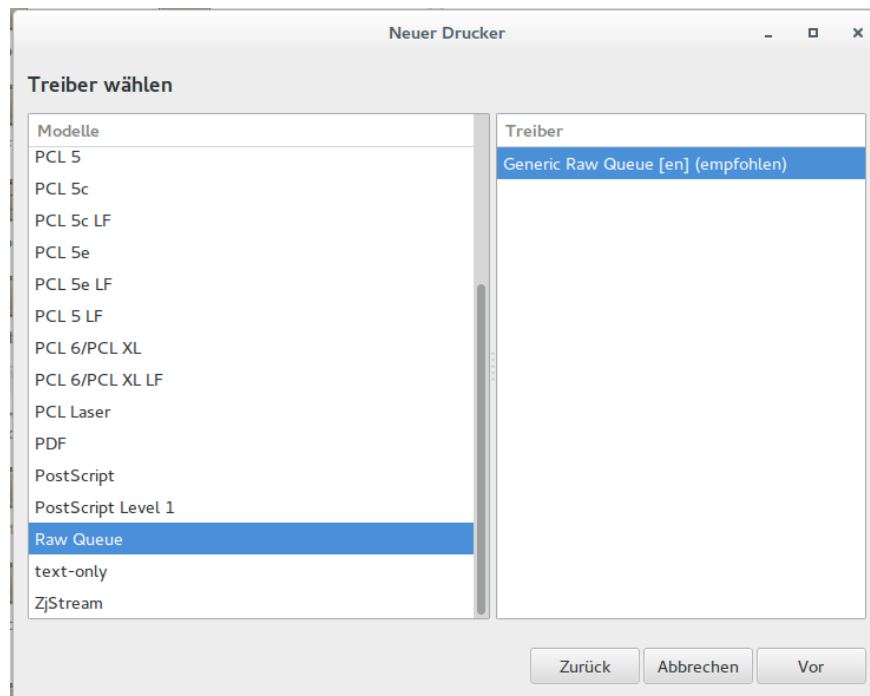
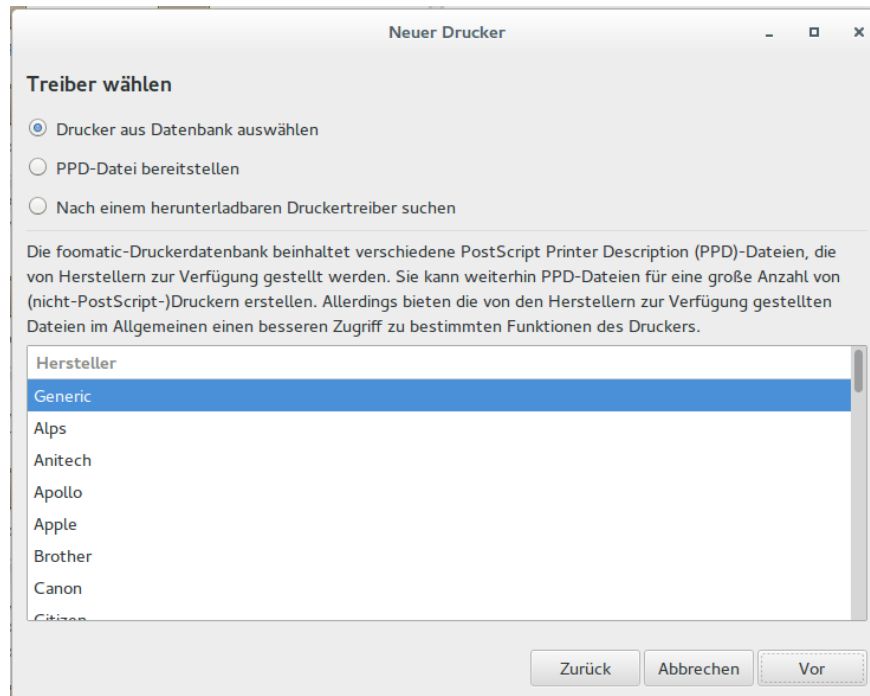
Das Passwort steht für “test1234”. Der entsprechende Hash-Wert im MySQL-Kommando wurde von Florian Hauer (labfolder) erstellt.

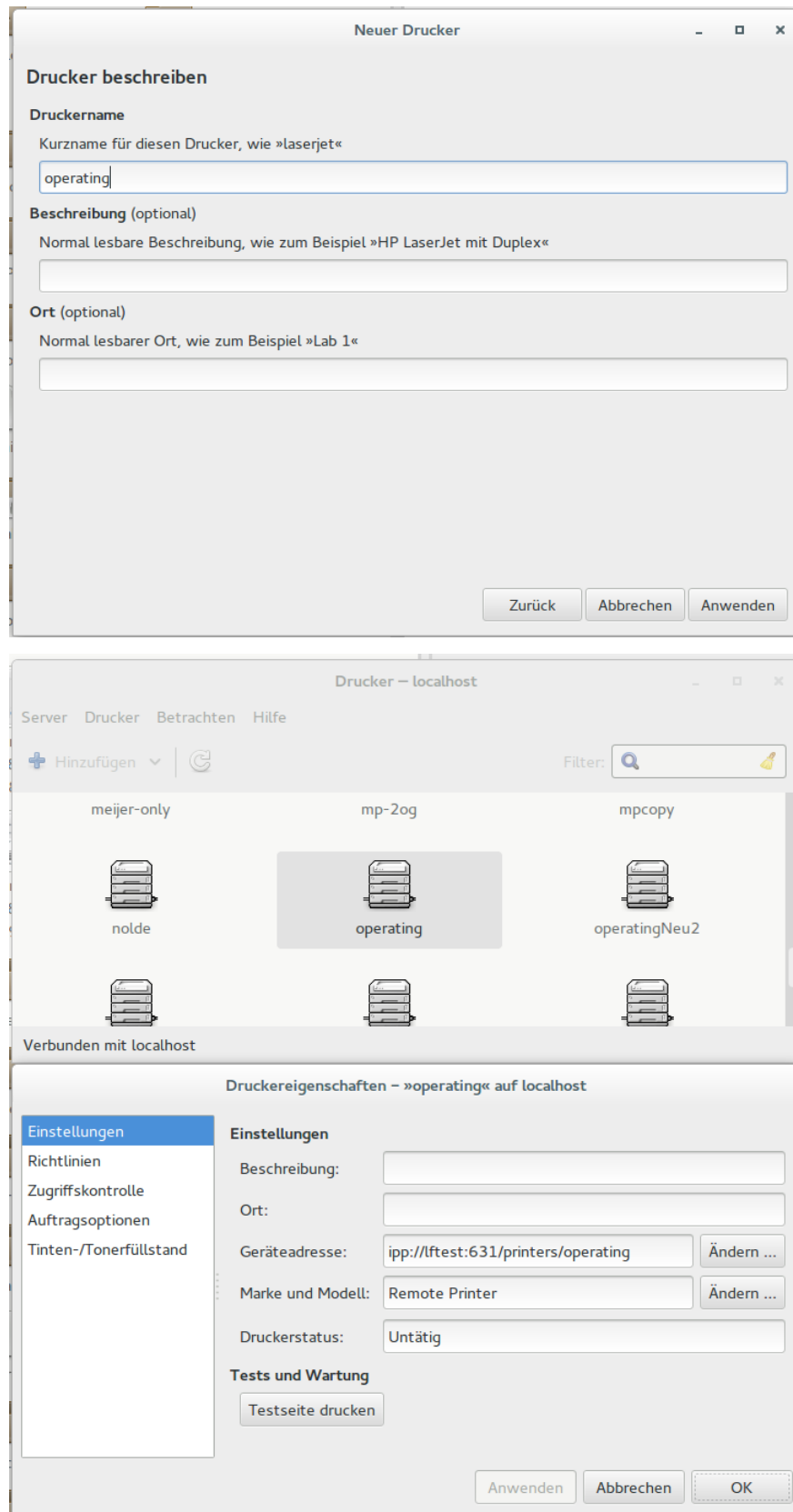
5.2 Printing to labfolder projects

The last thing you need to do is to set up a printer, for example, *operating* on your local machine. In Ubuntu Gnome hit `Alt+F2` keys and enter `system-config-printer`. In this printer config interface add a new printer *operating* as follows:









Now, just select this printer when printing a webpage or a picture to a labfolder project.