
ILMO Documentation

Release 1.3.0

Julian-Samuel Gebühr

Sep 16, 2021

Contents:

1	Users guide	1
1.1	Registration	1
1.2	Lending	1
1.3	Login	1
1.4	E-Mail	2
2	Administration	3
2.1	Create user	3
2.2	Lending	3
2.3	Returning	4
2.4	Opening hours	4
2.5	Add Items to lend	4
2.6	Monitoring	6
3	Installation, customization and contributing	9
3.1	Installation	9
3.2	Customization	10
3.3	Release	11
3.4	Contributing	12
4	API Documentation	13
4.1	Make Data Publicly Available	13
4.2	Track Working Hours and Opening Status	13
4.3	Access Control	14

1.1 Registration

To register you have to visit the library. An administrator will then set up an account for you. You will need to provide an valid E-Mail Address and a password.

1.2 Lending

You can check the available books or material online with your account.

To lend any books or material you visit the library and search for the item you want to lend. The location can be found in ILMO.

An administrator will then check the code of the item. You will provide them with the user ID you got during the registration or your E-Mail Address. The item will then be marked as lent.

You have to return the item according to the rules of the library.

1.3 Login

After your [Registration](#) you are able to log into your account with your E-Mail Address or user-ID and your password.

You can then access:

- your lendings
- your profile data
- the available books and items
- the opening hours

If you want t make changes to your profile ask an Administrator.

1.4 E-Mail

With your *Registration* you provide a valid E-Mail address. For every lending you will be reminded via E-Mail. The interval of the reminder is defined by the library.

2.1 Create user

Users are created by the administrators manually. Select *Create user* from the navigation.

You will need to provide the following information:

- E-Mail Address
- Forename
- Surname
- Password
- Admin [default = No]
- Language [default = library default]

Optional fields:

- UID (only needed for admins that want to track working hours)

After clicking *Send* you will be redirected to a list of all user, where you can check the user ID.

2.2 Lending

As a user wants to lend an item you need to know the following:

- Item category [book or material]
- Item ID [e.g. CH42 d]
- User ID or e-mail address

As many users do not remember their user ID or which e-mail address they used there is the possibility to search for a user. Select *Search user* from the navigation and search e.g. for their forename and then use the found information to continue the lending.

2.3 Returning

To return an item either visit the page *All loans* and search for the loan there or you search for the user, click on *Show loans* a search the loan there.

If you found the loan, you can simply click on the button *Return* and you are finished.

2.4 Opening hours

The opening hours can be changed by selecting the page *Change opening hours* in the navigation menu.

Just type in your new opening hours and click *send*.

Note: It is advised to fill empty time cells with a “-”.

If you want to change the opening day you can do so by changing the `config/settings.ini`.

2.5 Add Items to lend

In the current state ILMO only supports one indexing system for books and one for material. If you have an existing numbering system it is advised to reach out to a developer to see if the system can be added or to adapt the supported indexing system.

2.5.1 Books

ID System

As most libraries do often have multiple copies of the same book an identification system was found that accounts for that.

For books the ID format therefore consists of a `stem` of category and number and a `numbering` with characters of copies.

```
CC[number] [ii]

"CH42 c"      # Titel 42 in category Chemistry, copy number 3

"XY132 af"    # Titel 132 in category XY, copy number 33

"a16792 c"    # bad example as categories should be two capital letters and the
               # titel number should be ascending
```

- **CC:** Category abbreviation e. CH for chemistry. It is advised to use a two letter abbreviation, but there is no technical limit.
- **Number:** Indicates which title the book has. The number is ascending incremented for every title added to the category.
- **ii:** index of the copy. For the first copy the index is a for the second it is b and for the 27th copy it is aa.

Adding books

When adding books to the library you have to provide the following:

- **Book-ID:**
 - **You add only one book to the library:** Give a full ID e.g. CH4 a and no number or the number one.
 - **You add multiple books to the library:** Give only an ID stem. The index of the copy will be added automatically.
- Number of copies
- Title
- Author [optional]
- Location [optional]

Give all of that information and click send.

2.5.2 Material

ID System

Material has a different ID system. It consists of an abbreviation and an index.

```
AA [index]

"LC 42"           # Labcoat number 42
"SG 132"          # Safety glasses number 132

"Labcoat 16792"   # bad example as abbreviation should be two capital letters and ↵
↵index           # number should be ascending
```

- **AA:** Abbreviation of the items name
- **Index:** Ascending number. The highest index of an item is the number of items of this name.

Adding Material

To add material add provide the following:

- Material abbreviation
- Number of items to add
- Name of the item type [e.g. Labcoat]
- Location [optional]

Give all of that information and click send.

2.6 Monitoring

ILMO should, like every other software, be easy to monitor. Therefore a basic metrics are exposed to https://example.com/metrics_json/. The data is encoded in JSON format and is therefore suitable to be read by humans and it is easy to use it as data source for further processing.

2.6.1 Exposed Metrics

```
users: number of users (all roles combined)
loan: number of loans
loan_length_mean: Mean loan length in days
admin_num: Number of admins
lent_books_num: number of books that are lent
avail_books_num: number of books that are available
lent_material_num: number of materials are lent
avail_material_num: number of materials are available
```

2.6.2 Example workflow

To use the exposed metrics you will usually need a time series database and a visualization tool.

As time series database we will utilize InfluxDB, the visualization tool will be Grafana.

InfluxDB and Telegraf

First we install InfluxDB (e.g. with docker, be aware of the security risks!).

```
# Pull the image
$ sudo docker pull influxdb

# Start influxdb
$ sudo docker run -d -p 8086:8086 -v influxdb:/var/lib/influxdb --name influxdb_
↪influxdb

# Start influxdb console
$ docker exec -it influxdb influx
Connected to http://localhost:8086 version 1.8.3
InfluxDB shell version: 1.8.3
> create database monitoring
> create user "telegraf" with password 'mypassword'
> grant all on monitoring to telegraf
```

Note: When creating the user telegraf check the double and single quotes for username and password.

Now install telegraf and configure *etc/telegraf/telegraf.conf*. Modify the domain and your password for the InfluxDB database.

```
1 # Global tags can be specified here in key="value" format.
2 [global_tags]
3
4
```

(continues on next page)

(continued from previous page)

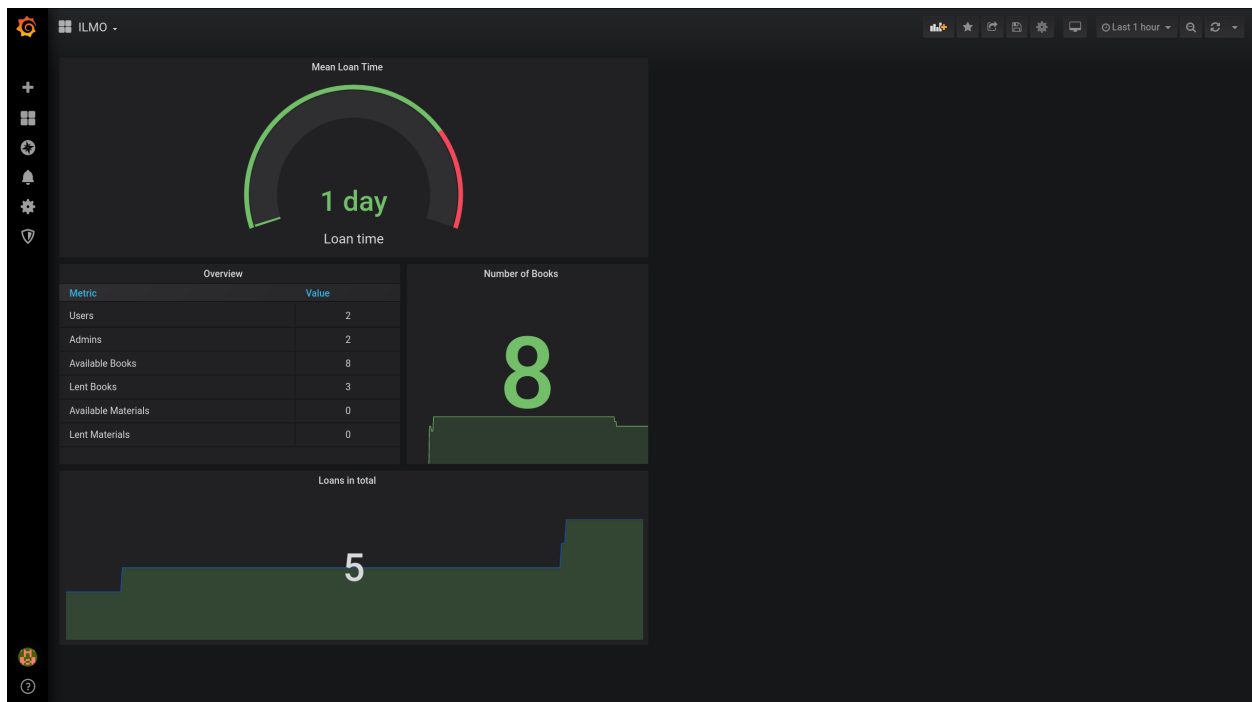
```

5 # Configuration for telegraf agent
6 [agent]
7   interval = "10s"
8   round_interval = true
9   metric_batch_size = 1000
10  metric_buffer_limit = 10000
11  collection_jitter = "0s"
12  flush_interval = "10s"
13  flush_jitter = "0s"
14
15 # Configuration for sending metrics to InfluxDB
16 [[outputs.influxdb]]
17   urls = ["http://:::8086"]
18   database = "telegraf"
19   skip_database_creation = true
20   username = 'telegraf'
21   password = 'yourpassword'
22
23 [[inputs.http]]
24   urls = ["https://example.com/metrics_json/"]
25   name_override = "ilmo"
26   #Data from HTTP in JSON format
27   data_format = "json"
28

```

Graphana

Now we can simply use the InfluxDB as data source in Grafana and configure until you have beautiful plots!



Installation, customization and contributing

3.1 Installation

3.1.1 Requirements

- A webserver (tested on Nginx)
- PHP version > 5.6.38
- **SQL database (tested with MariaDB)** As many hosting providers restrict the number of databases you can have, it can be necessary to share a database with another project. To use this you have to set a table prefix during installation.

Warning: This is not recommended as other projects could influence your library!

3.1.2 Download the project

You can download the newest release at *Github* <<https://github.com/GerJuli/ILMO/releases>>.

As an alternative you can clone the complete project from GitHub using:

```
$ git clone https://github.com/GerJuli/ILMO/
```

3.1.3 Build the documentation

If you want you can build the documentation. In order to do this navigate in the folder docs/ and execute

```
$ make
```

or on windows the make.bat file.

3.1.4 Move it to your server

Now move the project to the servers document root (or any other directory that can be accessed by your webserver).

Warning: Exclude the docs/ folder from the server copy. They are not needed for the project to work.

3.1.5 Basic configuration

The basic configuration can be done in the browser. Open <https://yourdomain.com> and follow the given instructions:

1. Configure database
2. Create first user
3. Delete the install/ directory

3.2 Customization

3.2.1 Settings

When opening the file `config/settings.ini` with a text editor you can adjust the following:

- **timezone** = **'Europe/Berlin'** Timezone as supported by php see the [PHP documentation](#) e.g. *'Europe/Berlin'*
- **enable_status** = **1** enable/disable status bar that indicates if the library is open
- **enable_presence_API** = **1** enable/disable presence API (used for tracking working hours)
- **max_loan_time** = **0** Maximum time of loan: Set to 0 for infinite
- **mail_reminder_interval** = **90** Interval of the e-mail reminder: Days after the user is reminded of a loan which is not returned
- **default_language** = **german** Default language. Pick between `german` and `english`
- **log_mail** = **0** If the e-mails that are sent should be logged
- **path_to_mail_log** = **"log/mail.log"** The path to the logfile for sent mails
- **opening_days[]** = **monday** For every day of the week where the library is open, add a line

3.2.2 Images

To customize the images just replace the images in `images/`.

E.g. if you want to replace the header image replace the file `images/logo_library.png`.

3.3 Release

3.3.1 What qualifies as release?

A new release should be announced when a significant number functions, bugfixes or other improvements to the software is made. Usually this indicates a minor release. Major releases are yet to be determined.

3.3.2 What should be done before a release?

Tested basic functions

- Installation
- User, book and material management (create, view, change, delete)
- Logging (e.g. mail log)
- E-Mails
- Monitoring

Test upgrade on a copy of a production database

Warning: You have to prevent e-mails from beeing sent, otherwise users could recive duplicate e-mails!

- Ensure correct conversion if neccessary
- Views correct?

3.3.3 Release

After testing everything you are good to go. Open the file `config/settings.ini` with a text editor you can adjust the version number:

```
...
"path_mail_log": "log/mail.log",
"timezone": "Europe/Berlin",
"version": "1.3"
...
```

Do a final commit on this change, and tag the commit as release with appropriate version number.

```
git tag -a v1.0.0 -m "Releasing version v1.0.0"
git push origin v1.0.0
```

Make sure the tag is visible on GitHub and celebrate

3.4 Contributing

3.4.1 Report a bug

To report a bug, file an issue on [Github](#)

Try to include the following information:

- The information needed to reproduce the problem
- What you would expect to happen
- What did actually happen
- Error messages

You are also invited to include:

- Screenshots
- Which browser you are using
- The URL of the site
- How urgent it is
- Any additional information you consider useful

3.4.2 Get involved!

To contribute simply clone the directory, make your changes and file a pull request.

If you want to know what can be done, have a look at the current [Issues](#).

3.4.3 Get in touch!

If you have questions, want to contribute or want to message me regarding something else you can find contact information at <https://hyteck.de/about/> or directly write an [E-Mail](#)

ILMOs API serves two different purposes. The first is to make data available to the public. The seconde is to track working hours and opening status.

4.1 Make Data Publicly Available

4.1.1 Example Use-Case:

You have a department/club/etc.. homepage that is run by a Content Management System like Wordpress and want to show which books are in the library available. ILMO is hosted on a different Website. One line of plain html code is enough to do what you want!

Simply include

```
<iframe src="https://example.com/index.php?ac=open_show_small_plain" width="100%"  
↪height="270"></iframe>
```

4.1.2 Show all books

4.1.3 Show all material

4.2 Track Working Hours and Opening Status

This part of the API offers the possibility to track working hours via a suitable IoT Device (e.g. a Raspberry Pi with NFC Reader).

Hint: This part of the API has to be activated in the settings!

4.2.1 Example Use Case

You have a library in your that is run by two students that need to track the hours they worked. Writing this down is way too easy and you want a live update when they are present so ILMO shows that someone is there. ILMO offers the possibility to checkin someone via an API. You therefore adapt the provided example script, so the students can use their student ID and a NFC reader that is connected to a Raspberry Pi to identify and checkin themselves.

4.3 Access Control

This part of the API offers the possibility to control the access for an optional key locker via a suitable device. This device sends a message to the server and get back whether the person is allowed to lend a key or not. All access tries are logged and can be shown by an administrator. The permission of each user can be set during the user registration and in the user settings. It is also necessary to set the UID of each user as it is necessary to identify the user.

Hint: This part of the API has to be activated in the settings!

4.3.1 Settings

You can send the UID of the RFID card coded in the URL. For safety reasons it is also necessary to send an access key in the request. The access key can be set in the settings. Optional you can send the status of the key (key available or not). The default value is 0.

Arguments: `ac=check_access` `UID="Number of the UID"` `access_key="Access key as set in settings` `key_available="0` or `1`, if not set `0`" The URL could be: http://localhost/ilmo/index.php?ac=check_aces&UID=1234&access_key=1234568&key_available=1



Read the manual, Image by [Mike Powell](#) (CC-BY).