

Services

All the services present on the [server](#) or outside.

- [Website](#)
- [Nextcloud](#)
- [SSO - KeyCLIC](#)
- [Clicketing](#)
- [Utility](#)
- [Roboclic](#)
- [IC Plays Pokemon](#)
- [Overclicked](#)
- [Clicketing V1 \(outdated\)](#)
- [Save the Date](#)

Website

The site is developed by hand, using [Directus](#) as CMS (content management system).

More information on the [repository](#).

Nextcloud

Manually adding files

It is possible to add files directly in the server, by transferring them in the Nextcloud's app volume. The volumes are located under `/var/lib/docker/volumes`, and the files are in the `data/___groupfolders/1/` subfolder of the volume.

To correctly add the files, you may need to convert the filename encoding, the easiest being using `rsync`. For example: `rsync -ahr --info=progress2 --iconv=ISO-8859-1,utf8 /var/gdrive/03_É've'nementiel/ /var/lib/docker/volumes/nextcloud_nextcloud-app/_data/data/___groupfolders/1/GDrive/03_É've'nementiel/`.

Once added, you first need to update the files ownership to `www-data` with `chmod www-data:www-data <FOLDER> -R`. Then index them in Nextcloud for them to be available in the web application. To do so, run the command `php ./occ files:scan --all` from the `/var/www/html` directory as the user `www-data` within the Nextcloud's container. The easy way is to get the container's name using `docker ps`, then run `docker exec -it -u www-data <CONTAINER> php ./occ files:scan --all`. This may take a while, so I recommend to use `screen` to start it in background, such that you can close the connexion without stopping the indexing.

Fixing index.php

After restarting nextcloud, it may happen that the base path of the installation become `/nextcloud/index.php`. In that case, you should run the command `php occ maintenance:update:htaccess` in the nextcloud container when logged in as `www-data`, thus usually something like `docker exec -u www-data nextcloud_nextcloud.<smth> php occ maintenance:update:htaccess` should be sufficient.

OnlyOffice file size issues

OnlyOffice restricts the maximum size of files that can be edited live by the server. It can be increased in the server's configuration (`default.json` in the `onlyoffice_config` docker volume, full path is `/var/lib/docker/volumes/onlyoffice_config/_data/default.json`), through the `FileConverter.converter.maxDownloadBytes` and `FileConverter.inputLimits.[type=your_file_type].uncompressed` fields.

We have increased the maximum size for PPTX and related files to 512MB, as well as XSLX and related to 300MB.

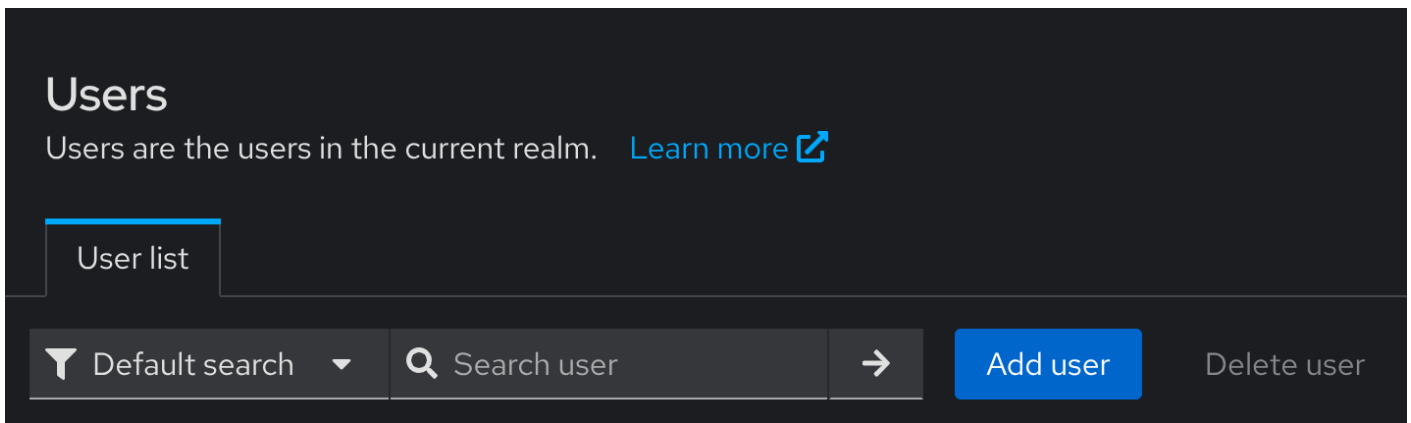
SSO - KeyCLIC

The [SSO](#) (or Single-Sign On) is handled by a [Keycloak](#) instance hosted on the server. The admin panel is available at <https://cllic.epfl.ch/keycllic> and users can manage their profile at <https://cllic.epfl.ch/me>.

For now, each user has a `nextcloud_id` attribute, to remain compatible with accounts that were created directly on Nextcloud. New users should have their username as `nextcloud_id`, with the format `{name}.{surname}`. When there will be no remaining users with the `nextcloud_id` different than their `username`, you can update Nextcloud to use the username (`preferred_username`) and get rid of the `nextcloud_id` field.

Adding new users

Go to the KeyCLIC admin console here: [KeyCLIC Users](#)



Click add user and fill in the following information:

- **Required user actions:** Update Password, Verify Email
- **Email verified:** leave this OFF
- **Username:** `name.surname`
- **Email:** EPFL email address if available (may be non EPFL user address for a user who is not at EPFL)
- **First Name & Last Name:** you can fill in the user's name, or leave it for them to fill in later
- **Nextcloud ID:** `name.surname`, same as username
- **Groups:** select only the relevant groups for the user, for example members of a commission should only be in their commission group.

Optionally, once the user is created, you can set their password to a temporary value (such as their email address). Leave the "Temporary" option on when creating the password, and ensure the options requiring them to verify their email and update their password are active. Otherwise, they can set their password by selecting the "Forgot password" option on the KeyCLIC login page.

Clicketing

Clicketing registration management system, created by CLIC, for CLIC.

It is available on <https://clic.epfl.ch/clicketing>, and its codebase on the [GitHub repository](#).

TODO

Possible improvements:

- [Camipro](#)

Utility

Various small services and projects for CLIC that are not crucial, but come in handy.

QR Bill Generator

A handy QR Bill Generator that can be used to generate personalized bills for event participants. For example, this is used for the [Faculty Diner](#) and [IC Boost Day](#).

It is hosted at <http://clic.epfl.ch/qrbill-generator> and the source code can be found on the [repository](#).

Mail Sender

A simple mail sender that supports Liquid templates and ICS attachments. It was mainly created to use from Directus flows (that do not support ICS attachments easily). The source code can be found on the [repository](#).

Crab CLIC

A self hosted and themed instance of crab.fit. The source code can be found on the [repository](#).

KeyCLIC Theme

A simple theming for our self hosted instance of Keycloak. The source code can be found on the [repository](#).

Roboclic

Roboclic is CLIC's Telegram bot. It is written in Rust and hosted on server.

More information on the [repository](#).

IC Plays Pokemon

A collaborative game derived from the concept of [Twitch Plays Pokemon](#) (but without the Twitch part). It comes with an interface for players to select a command, then the backend plays the majority command each cycle in the actual Pokemon game. The source code can be found here [IC Plays Pokemon](#).

Overclicked

Overclicked is a web application created by CLIC, used for tracking and managing orders at student-organized events. It provides simple interfaces for event staff in various roles (at the register, in the preparation area, etc) to monitor incoming orders and update their status in real time.

It was mainly used for Subsonic, to manage Croque-Monsieur orders in real time.

The source code can be found here: [Overclicked](#)

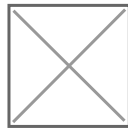
Clicketing V1 (outdated)

Clicketing registration management system, created by CLIC, for CLIC.

“ Important Note

This page is about the first version of Clicketing, which was last used in Spring 2024 and has now been replaced by an entirely new version. You can find the codebase for this older version on the [GitHub repository](#).

See more about the newest version of Clicketing here : [Clicketing](#).



How it works

Before an event, Clicketing sends an e-mail to each participant containing a QR Code to facilitate entry. At the start of the event, everyone presents themselves at reception with their QR Code, which staff can scan to confirm registration and attendance.

Login

The admin password is stored on the [Vaultwarden](#).

Utilisation

The service allows you to manage registrations by event. On the [main page](#), you can create and edit events. Each event is assigned a name, a date and an e-mail template.

The mail can be sent to all participants exactly once, using the `Send mail` button. To check the email, you can send it to a specific person, by entering their address in the `Preview email recipient` field and pressing `Send preview email`. This email will contain a random QR Code, and is only to be used to view the visual rendering of the template.

To add participants, you need to use the HTTP API. The ideal way is to create a script on the registration sheet, in order to generate a JSON. For more information, see dev's documentation on the [repository](#).

Possible improvements

- Use camipro readers instead of QR Code (see [Camipro API](#)).
- Security: Sessions with expiry date.
- Additional data to be inserted in e-mail (name, date, menu, etc.).
- Automatic insertion of participants when registering on form.
- Insert/edit/archive participants (indicate payment, etc.).
- Automatic sending of email upon registration/payment.

Save the Date

todo update ça