

Installationsanleitung Synology - DSM 7.0

Kimai2 - Zeiterfassung

Anleitung zur Installation des Zeiterfassungsprogramm Kimai2 (Open-Source) unter einer Synology NAS mit DSM 7.0

Was ist Kimai? Quelle <https://www.kimai.org/de/ueber-kimai.html>

Basis

Auf folgender Basis wurde das Programm installiert und eingerichtet:

- Synology DS720+
- Virtual DSM 7.0.1 (Virtual Machine)
- Stable Release: Kimai2-1.17.1.zip (<https://www.kimai.org/de/download.html>)
- Composer v2.2.7 (<https://getcomposer.org/download/>)
- Anleitung vom <https://www.kimai.org/documentation/synology.html>
- Grundkenntnisse Linux und dem vi oder vim Editor
- Putty SSH Client (<https://www.putty.org/>)

Server Anforderungen

- PHP 7.3 oder höher (über das Synology Paketzentrum installiert)
- Datenbank: MariaDB 10 über das Synology Paketzentrum installiert
- Webserver: nginx (Web Station über das Synology Paketzentrum installiert)
- phpMyAdmin über das Synology Paketzentrum installiert
- SSH Aktivieren über die Systemsteuerung „Terminal & SNMP“

Vorbereitungen zur Installation

1. Zuerst sind die unter Pkt. „Server Anforderungen“ erforderliche Pakete über das Synology Paketzentrum zu installieren und der SSH zu aktivieren.
2. Als erstes legen wir unter der Web Station ein neues PHP-Profil an. Ich benenne es **Kimai2** und nutze die **PHP 7.4 Version**

The screenshot displays the Synology Web Station interface. On the left, a sidebar contains navigation options: Status, Webdienstportal, Skript-Spracheinstellungen (highlighted), and Fehlerseite-Einstellungen. The main content area is titled 'PHP' and includes buttons for 'Erstellen', 'Bearbeiten', and 'Löschen'. Below these is a table listing PHP profiles:

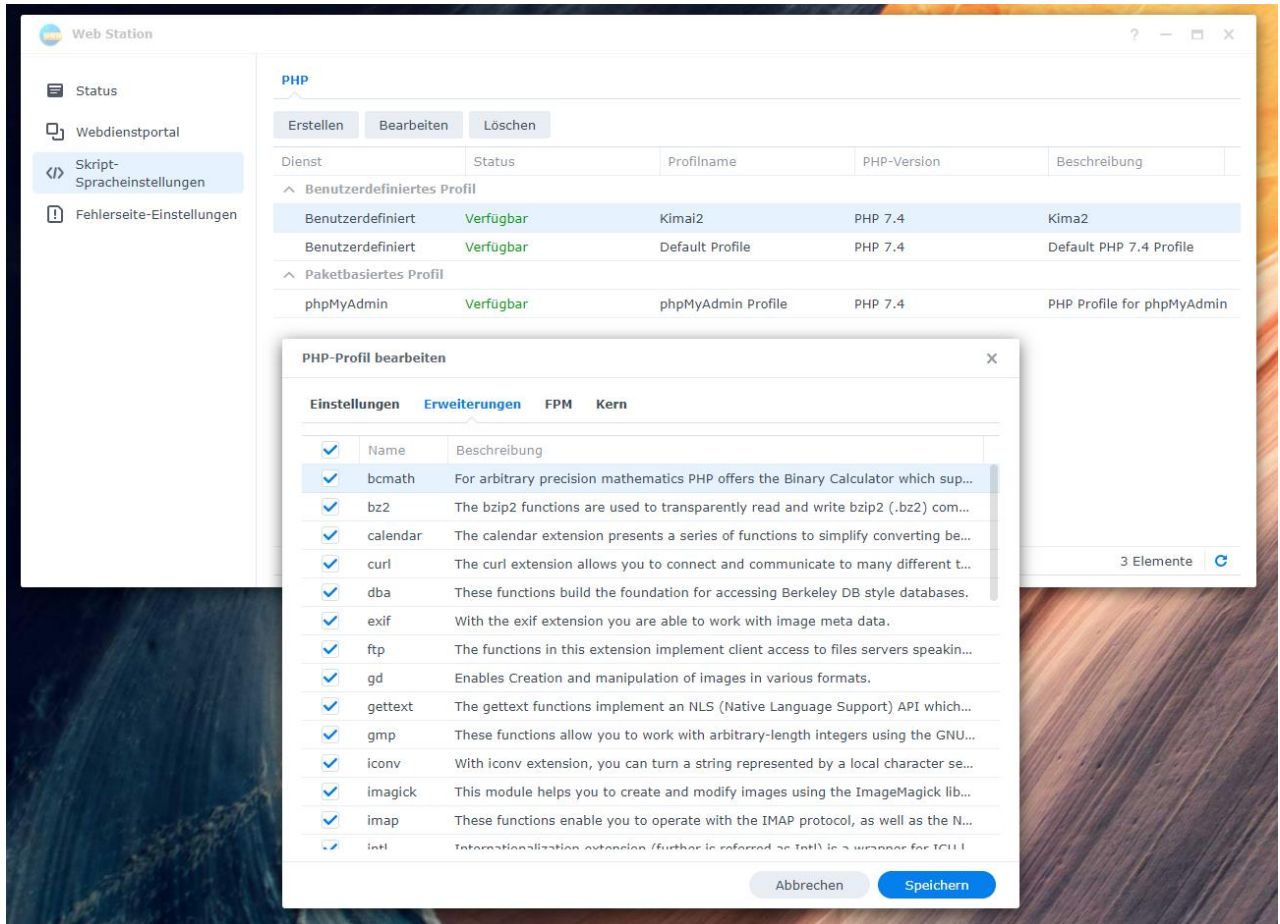
Dienst	Status	Profilname	PHP-Version	Beschreibung
Benutzerdefiniertes Profil				
Benutzerdefiniert	Verfügbar	Kimai2	PHP 7.4	Kimai2
Benutzerdefiniert	Verfügbar	Default Profile	PHP 7.4	Default
Paketbasiertes Profil				
phpMyAdmin	Verfügbar	phpMyAdmin Profile	PHP 7.4	PHP Pr

The 'PHP-Profil bearbeiten' modal window is open, showing the following settings:

- Profilname:** Kimai2
- Beschreibung:** Kimai2
- PHP-Version:** PHP 7.4
- PHP-Cache aktivieren
- Xdebug aktivieren
- display_errors aktivieren, um PHP-Fehlermeldungen anzuzeigen
- open_basedir:**
 - Benutzerdefiniert
 - Standard (None)

Buttons for 'Abbrechen' and 'Speichern' are visible at the bottom of the modal.

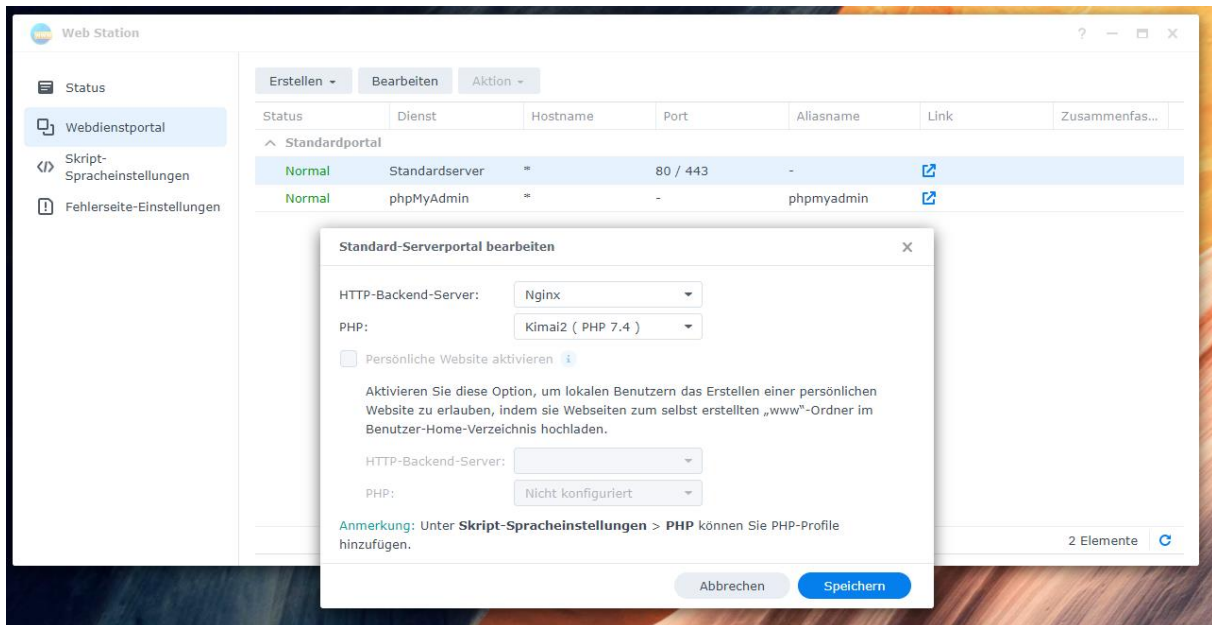
Unter „Erweiterung“ habe ich vorerst **alle** verfügbaren **Erweiterungen** aktiviert:



In den Einstellungen „Kern“ habe ich folgende Einstellungen vorgenommen und den Speicher für PHP erhöht:

Memory_limit **512M**

Danach wird unter Webdienstportal der Web Station der http-Backend-Server: **nginx** und PHP: **Kimai2 (PHP 7.4)** eingestellt.



3. Die Software Kimai2 auf den PC herunterladen (<https://www.kimai.org/de/download.html>)
Die Zip auf die NAS kopieren nach **/volume1/web** und extrahieren. Den extrahierten kimai Ordner benenne ich auf **kimai2** um.
4. So jetzt laden wir uns den composer.phar unter <https://getcomposer.org/download/> herunter und kopieren diesen nach **/volume1/web/kimai2**
5. Über Putty SSH loggen wir uns auf die Konsole der NAS ein und gehen ins Verzeichnis **/volume1/web/kimai2** und installieren erstmals den Composer mit folgendem Befehl:

```
php74 composer.phar install --no-dev --optimize-autoloader
```

Nach der Installation solltet ihr hoffentlich den Status erhalten das alles OK ist:

```
- Installing symfony/yaml (v4.4.36): Extracting archive
- Installing tijsverkoyen/css-to-inline-styles (2.2.4): Extracting archive
- Installing twig/cssinliner-extra (v3.3.5): Extracting archive
- Installing symfony/twig-bundle (v4.4.36): Extracting archive
- Installing twig/extra-bundle (v3.3.7): Extracting archive
- Installing lorenzo/pinky (1.0.5): Extracting archive
- Installing twig/inky-extra (v3.3.5): Extracting archive
- Installing twig/intl-extra (v3.3.5): Extracting archive
- Installing symfony/string (v5.4.2): Extracting archive
- Installing twig/string-extra (v3.3.5): Extracting archive
Package symfony/inflector is abandoned, you should avoid using it. Use EnglishInflector
Package phpunit/php-token-stream is abandoned, you should avoid using it. No replacemen
Generating optimized autoload files
composer/package-versions-deprecated: Generating version class...
composer/package-versions-deprecated: ...done generating version class
102 packages you are using are looking for funding.
Use the `composer fund` command to find out more!

Run composer recipes at any time to see the status of your Symfony recipes.

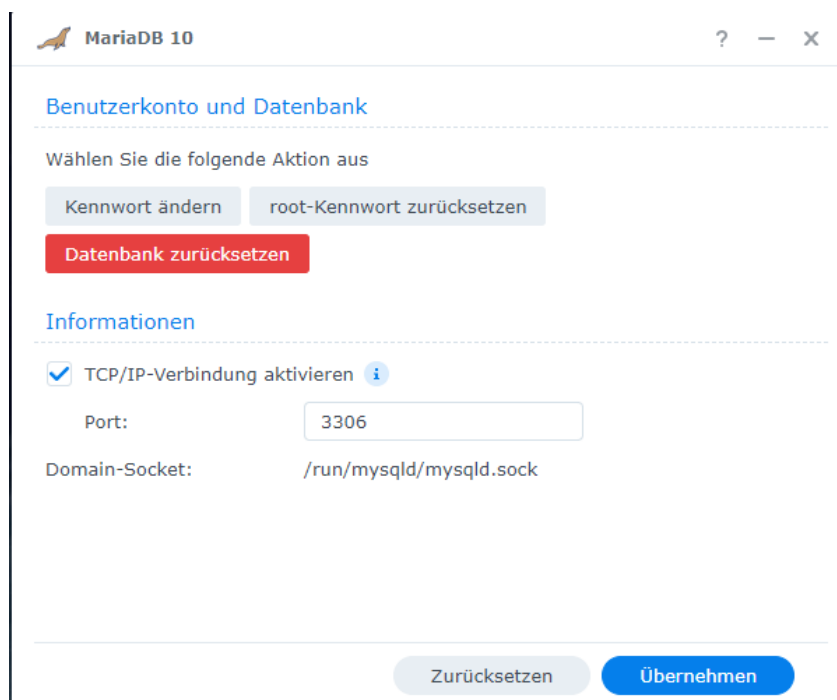
Executing script cache:clear [OK]
Executing script assets:install [OK]
```

Datenbank erstellen

Um die Datenbank und Datenbank-User für kimai anzulegen benutze ich am liebsten phpMyAdmin.

Als erstes wird unter MariaDB die TCP/IP Verbindung aktivieren und den Port belasse ich auf 3306.

Danach starten wir phpMyAdmin und loggen uns als root mit dem Passwort von der MariaDB Installation ein.



Unter Benutzerkonto legen wir einen neuen Benutzer mit folgenden Optionen an:

(Achtung: nutze für das Sonderzeichen im Passwort eher ein - oder _)

The screenshot shows the 'Benutzerkonto hinzufügen' (Add User) interface in the Synology MySQL tool. It is divided into three main sections:

- Anmeldeinformation (Login Information):**
 - Benutzername: Textfeld verwenden:
 - Hostname: Lokal
 - Passwort: Textfeld verwenden: Strength: Gut
 - Wiederholen:
 - Authentifizierungs Plugin: Native MySQL-Authentifizierung
 - Passwort generieren:
- Datenbank für Benutzerkonto (Database for User Account):**
 - Erstelle eine Datenbank mit gleichem Namen und gewähre alle Rechte.
 - Gewähre alle Rechte auf Datenbanken die mit dem Benutzernamen beginnen (username_%).
- Globale Rechte (Global Privileges):** Alle auswählen
 - Hinweis: MySQL-Rechte werden auf Englisch angegeben.*
 - Daten:** SELECT, INSERT, UPDATE, DELETE, FILE
 - Struktur:** CREATE, ALTER, INDEX, DROP, CREATE TEMPORARY TABLES, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE, CREATE VIEW, EVENT, TRIGGER
 - Administration:** GRANT, SUPER, PROCESS, RELOAD, SHUTDOWN, SHOW DATABASES, LOCK TABLES, REFERENCES, REPLICATION CLIENT, REPLICATION SLAVE, CREATE USER
 - Ressourcenbeschränkungen (Resource Limits):**
 - MAX QUERIES PER HOUR:
 - MAX UPDATES PER HOUR:
 - MAX CONNECTIONS PER HOUR:
 - MAX USER_CONNECTIONS:
 - SSL:**
 - REQUIRE NONE
 - REQUIRE SSL
 - REQUIRE X509
 - SPECIFIED
 - REQUIRE CIPHER:
 - REQUIRE ISSUER:
 - REQUIRE SUBJECT:

Wenn die Datenbank und der User erfolgreich angelegt sind müssen wir die „.env“ Datei im **/volume1/web/kimai2** noch anpassen. Hierzu loggen wir uns per SSH mit Putty auf die Konsole der NAS ein und bearbeiten die Datei mit dem „vi“ Editor:

```
vi /volume1/web/kimai2/.env
```



Die Zeile „DATABASE_URL“ muss wie folgt geändert werden:

```
DATABASE_URL="mysql://kimai2:password@localhost:3306/kimai2?unix_socket=/run/mysqld/mysqld.sock"
```

unter „Password“ tragen sie in Klartext das zuvor von dir festgelegte Passwort der kimai2 Datenbank ein.

Installation des Programms Kimai

Über die Konsole gehen wir in das Verzeichnis **/volume1/web/kimai2** und starten die Kimai-Installation mit folgendem Befehl:

```
php74 bin/console kimai:install -n
```

Die Installation dauert und es wird hierzu kein Status angezeigt, daher um ein wenig Geduld. Bei mir hat die Installation gute 15-20min benötigt unter der Virtuellen Maschine (2 CPU und 2 GB Ram).

```
Kimai installation running ...
=====
Created database `unix_socket=/run/mysqld/mysqld.sock` for connection named default
[notice] Migrating up to DoctrineMigrations\Version20220101204501
█
```

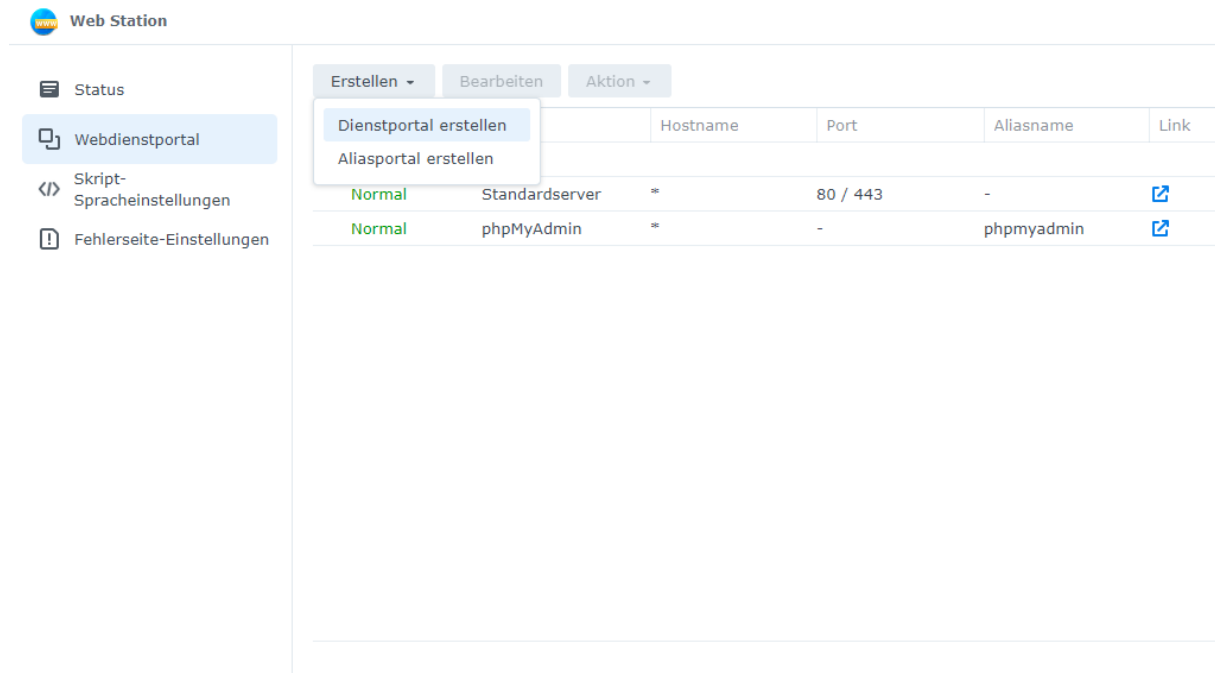
Sollte alle gut gegangen sein, sollte folgende Meldungen erscheinen 😊:

```
Rebuilding your cache, please be patient ...
// Clearing the cache for the prod environment with debug false
[OK] Cache for the "prod" environment (debug=false) was successfully cleared.
// Warming up the cache for the prod environment with debug false
[OK] Cache for the "prod" environment (debug=false) was successfully warmed.
[OK] Congratulations! Successfully installed Kimai version 1.17.1
```

Virtual Host

Damit wir das Programm starten bzw. öffnen können, legen wir in der **Web Station** einen Virtual Host für kimai an.

Hierzu öffnen wir die Web Station und legen einen neuen vHost unter „Dienstportal erstellen“ an:



The screenshot shows the Synology Web Station interface. On the left, there is a sidebar with navigation options: Status, Webdienstportal (selected), Skript-Spracheinstellungen, and Fehlerseite-Einstellungen. The main area displays a table of virtual hosts. At the top, there are buttons for 'Erstellen', 'Bearbeiten', and 'Aktion'. A dropdown menu is open under 'Erstellen', showing 'Dienstportal erstellen' and 'Aliasportal erstellen'. The table below has columns for 'Normal', 'Standardserver', 'Hostname', 'Port', 'Aliasname', and 'Link'. Two entries are visible: 'Standardserver' with port '80 / 443' and 'phpMyAdmin' with port '-'. Both have an external link icon.

	Hostname	Port	Aliasname	Link
Normal	Standardserver	* 80 / 443	-	🔗
Normal	phpMyAdmin	* -	phpmyadmin	🔗

Die vHost für kimai nutzen wir Portbasiert und mit folgenden Einstellungen:

- Port: 7777 mit HTTPS
- Document root: /volume1/web/kimai2/public
- HTTP backend server: nginx
- PHP backend: Kimai2 (PHP 7.4)



Portal-Erstellungsassistent - Einen virtuellen Host einrichten ✕

Namensbasiert

Hostname

Port 80 / 443

HTTP

HTTPS

Portbasiert

Port HTTP

HTTPS

Dokument-Root

HTTPS-Einstellungen HSTS

HTTP-Backend-Server

PHP i

Zugangskontrollprofil i

Fehlerseitenprofil i

Im vHost müssen noch Anpassungen für kimai durchgeführt werden. Hierzu loggen wir uns als Root-Admin, mit dem Befehl:

```
sudo -i
```

ein und navigieren ins Verzeichnis:

```
cd /etc/nginx/sites-enabled
```

öffnen mit „vi“ folgende Datei:

```
vi /etc/nginx/sites-enabled/server.webstation-vhost.conf
```

suchen sie die Zeile, am Ende, die mit „**include**“ beginnt und kopieren sie folgenden Abschnitt und beenden sie „vi“:

```
/usr/local/etc/nginx/conf.d/1d16b269-904e-41c9-bd23-cbdf761e305e
```

Die Zahlen- und Buchstabenkombination ist bei jedem unterschiedlich!



Danach erstellen wir eine neue Datei mit folgendem Befehl:

```
vi /usr/local/etc/nginx/conf.d/1d16b269-904e-41c9-bd23-  
cbdf761e305e/user.conf-kimai2
```

und fügen folgende Zeilen ein:

```
index index.php;  
  
access_log off;  
log_not_found off;  
  
location ~ /\.ht {  
    deny all;  
}  
  
location / {  
    try_files $uri /index.php$is_args$args;  
}  
  
location ~ ^/index\.php(/|$) {  
    fastcgi_pass unix:/run/php/php7.2-fpm.sock;  
    fastcgi_split_path_info ^(.+\.(php|php5|php7|php8|php9|html|htm|jsp|jsp2|jsp3|perl|asp|aspx|ascx|aspx?|asp3|aspx3|ascx3|aspx3?|aspx3?)$);  
    include fastcgi.conf;  
    fastcgi_param PHP_ADMIN_VALUE "open_basedir=$document_root/../../tmp/";  
    internal;  
}  
  
location ~ \.php$ {  
    return 404;  
}
```

Checke dann die nginx config mit dem Befehl:

```
sudo nginx -t
```

wenn alles OK gegangen ist sollte folgendes Angezeigt werden

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

So jetzt starten wir nginx mit der Config neu:

```
sudo nginx -s reload
```

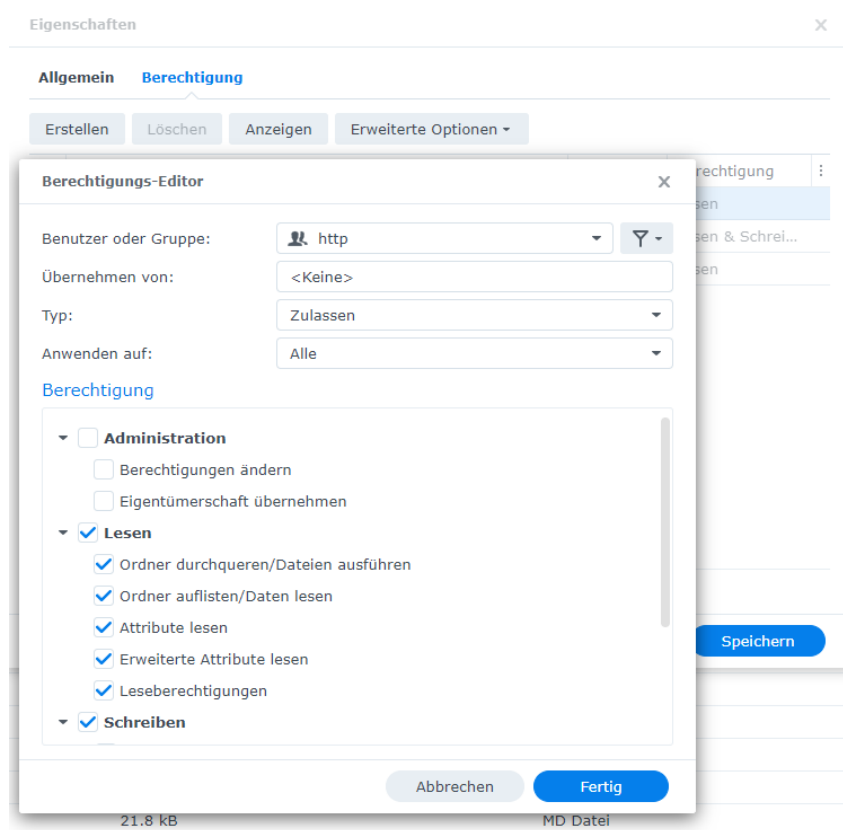


Berechtigungen

Jetzt müssen die Berechtigungen noch angepasst werden. Das machst du am Besten über die Filestation.

Gehe dazu ins Verzeichnis `Volume1/web/kimai2` und klicke mit der rechten Maustaste auf `/var`

Erstellen hierzu eine neue Berechtigung mit dem Benutzer/Gruppe **http** und vergeben Lese und Schreibrechte.



Dasselbe machen wir für das Verzeichnis `/public`

Hier nehmen wir die bestehenden **http** Gruppe (sollte nicht ausgegraut sein) und vergeben zusätzlich die Schreibrechte.

Erstelle den ersten User

Um überhaupt in das Programm einzuloggen muss zuerst ein User über das Terminal/Konsole angelegt werden:

```
cd /volume1/web/kimai2
```

```
php74 bin/console kimai:user:create username admin@example.com  
ROLE_SUPER_ADMIN
```

- als „username“ trage euch den gewünschte Usernamen in der Befehlszeile ein
- als „admin@example.com“ kannst du eine E-Mail-Adresse angeben – diese kann im Programm auch nachher geändert werden
- „ROLE_SUPER_ADMIN“ erhältst du die Rolle des Super Admins (nicht Ändern)

Danach wirst du nach einem Passwort gefragt für den Super Admin.

Programmaufruf

So endlich geschafft! Viel SPASS..... 😊

Das Programm rufst du im Browser mit deiner IP und den im vHost eingetragenen Port auf (<https://192.xxx.x.xx0:7777/>). Login mit dem zuvor angelegten User und Passwort.

