

# Turniermanagement

Ein Webinterface zur einfachen Verwaltung von Turnieren.

Funktionsumfang: Abspeichern der Turniere in einer Datenbank, Mailversand an zutreffende Sportler und das Abspeichern von Turnieren in "Google Calendar".

<b>Turnier hinzufügen</b>	<b>Fechter hinzufügen</b>	<b>Turnierlisten exportieren</b>
---------------------------	---------------------------	----------------------------------

**Name des Turniers:**

**Link zur Ausschreibung:**

**Veranstaltungsort:**

**Datum:**

**Altersklasse(n):**

Schüler: <input type="checkbox"/>	Junioren: <input type="checkbox"/>
B-Jugend: <input type="checkbox"/>	Aktive: <input type="checkbox"/>
A-Jugend: <input type="checkbox"/>	Senioren: <input type="checkbox"/>

**Waffe(n):**

Säbel: <input type="checkbox"/>	Florett: <input type="checkbox"/>
------------------------------------	--------------------------------------

**Pflichtturnier:**

Anklicken für "Ja"  
☐

**Abschicken**

## Inhaltsverzeichnis

Turniermanagment.....	1
Nutzungshinweise.....	2
Anlegen von Turnieren:.....	2
Einen neuen Fechter anlegen:.....	2
Export der angelegten Turniere.....	2
Python.....	2
Emails versenden:.....	2
Kalender ausfüllen:.....	3
Installation und Updates.....	4
Voraussetzungen:.....	4
Installation.....	4
Vorbereitung:.....	4
Datenbank Grundlagen installieren:.....	4
Webinterface installieren:.....	5
Pythonscripte:.....	5
Datenbank importieren:.....	6
Zeitsteuerung aktivieren.....	6
Aktualisieren.....	6
Datenbankaufbau.....	7
Beziehungsdiagramm .....	8

## Nutzungshinweise

### **Anlegen von Turnieren:**

Neue Turniere können im Webinterface im Reiter "Turnier hinzufügen" angelegt werden. Die Felder mit dem Namen, der Ausschreibung und dem Veranstaltungsort sind Pflichtfelder. Sind diese nicht ausgefüllt schlägt das Hinzufügen fehl. Das Datum lässt sich einfach mithilfe des Kalenderformulars eingeben, sobald man auf die Eingabezeile hierfür klickt. Bei den Altersklassen und den Waffen muss mindestens ein Kästchen angekreuzt werden, ansonsten schlägt auch hier der Prozess fehl. Anders sieht es beim Pflichtturnier aus. Dieses kann ankreuzt werden, es ist jedoch nicht zwingend erforderlich. Sofern es ausgewählt ist wird sich im generierten Mailtext ein zusätzlicher Hinweis auf die Relevanz dieses Turniers befinden.

### **Einen neuen Fechter anlegen:**

Das Anlegen eines neuen Fechter ist unter "Fechter hinzufügen" möglich. Hier muss in jedem Feld eine Angabe gemacht werden und auch hier wieder mindestens ein Kästchen angekreuzt werden. Auch dieses Formular gibt Rückmeldung ob das hinzufügen erfolgreich gewesen ist oder nicht. Sofern Eltern informiert werden sollen sind diese mit dem Jahrgang des Kindes in der Datenbank zu erfassen, um eine Benachrichtigung nach Altersklassen zu ermöglichen.

### **Export der angelegten Turniere**

Es ist möglich die erfassten Turniere nach Waffe, Altersklasse und Datum sortiert in ein Tabellendokument zu exportieren. Hierfür ist ein Klick auf den Button im Reiter "Turnierlisten exportieren" nötig. Nach kurzer Zeit sollte sich ein Download öffnen, welche eine CSV Tabelle herunterlädt. Diese lässt sich mit herkömmlichen Tabellenkalkulationsprogrammen öffnen und dort grafisch aufarbeiten und ausdrucken.

# Python

## **Emails versenden:**

Der in Python verfasste "Turnierkurier" ist ein Script, welches die kommenden Turniere der nächsten 6 Wochen aus der Datenbank abfragt, die Altersklassen und Waffen für dieses Turnier ermittelt und anschließend mit den gespeicherten Daten der Fechter abgleicht um diese bei Übereinstimmung per Mail zu informieren.

Sollte dieses Script zu Demonstrationszwecken gestartet werden empfiehlt es sich den Testmodus durch setzen der Variable "test = 1" am Anfang des Scriptes zu setzen. Die Folge ist, dass es die Mail nicht zu versenden versucht und stattdessen in einer Konsole den Mailtext ausgibt.

## **Kalender ausfüllen:**

Das Script "calendar-push" schreibt die Turniere nach Waffen sortiert in eigene Google Kalender, welche auf nahezu allen Plattformen eingebunden werden können. (Outlook, Thunderbird, iOS, Android). Dies ist auch der Grund, weshalb die Entscheidung auf ein proprietäres Kalenderformat gefallen ist. Es ist unter Android nicht ohne erweiterten Aufwand einen anderen Kalender importieren. Dies ist mit vorinstallierten Apps nur mit Google Calendar und den integrierten Freigaben möglich.

# Installation und Updates

## **Voraussetzungen:**

- Ein Webserver wie Apache oder Nginx (getestet)
- PHP Unterstützung (Version 5.5 oder höher)
- MySQL und die notwendigen PHP-MySQL Erweiterungen
- Python Version 2.7
- Python MySQL-Connector (→ Installation)
- Python Google Oauth Pakete (→ Installation)
- Ein lokaler Mailserver

## **Installation**

### **Vorbereitung:**

1. Git installieren (Nicht notwendig, vereinfacht aber Installation und Updates)
2. Den Webserver mit den zugehörigen (PHP-)Paketen nach der offiziellen Dokumentation installieren.
3. MySQL Server installieren und einen „root“-Benutzer inklusive Passwort anlegen.
4. Python 2.7 inklusive „pip“ installieren.
5. Die für den Kalender notwendigen Python Bibliotheken installieren:  
`pip install --upgrade google-api-python-client python-gflags`
6. Den Python MySQL Connector von <https://dev.mysql.com/downloads/connector/python/> für das entsprechende Betriebssystem herunterladen und installieren

### **Datenbank Grundlagen installieren:**

1. Öffnen einer Kommandozeile und MySQL starten (`mysql -u root -p` und bei der Aufforderung das Passwort eingeben)
2. Einen neuen Datenbankbenutzer anlegen und ein Passwort zuweisen:  
`CREATE USER 'turniere'@'localhost' IDENTIFIED BY '$password';`
3. Erstelle die Datenbank:

```
CREATE DATABASE IF NOT EXISTS `turniermanagment` DEFAULT CHARACTER SET `utf8`  
COLLATE `utf8_unicode_ci`;
```

4. Weise dem neu erstellten Nutzer die Berechtigungen für die Datenbank zu:  
`GRANT ALL PRIVILEGES ON `turniermanagment`. * TO 'turniere'@'localhost';`

5. Aus MySQL ausloggen:

```
\q
```

### **Webinterface installieren:**

1. Das öffentliche Verzeichnis des Webserver aufrufen und eine Kommandozeile öffnen.

2. "git clone https://git.scimeda.de/thore/turniermanagment.git" ausführen und die aktuelle Version herunterladen.

3. Die Datenbankeinstellungen im Ordner subpages von Tournament.php und AddFencer.php sowie in libs/ExportCSV.php Nutzernamen und Passwort durch die zuvor angelegten Nutzerdaten ersetzen.

4. Die Änderungen in git sichern für spätere Updates:  
`git commit -am „Datenbankdaten der Website angepasst“`

### **Pythonscripte:**

Die Scripte befinden sich im Unterordner Python

1. Die Datenbankparameter in calendar-push.py und turnierkurier.py ebenfalls anpassen.

2. In calendar-push.py müssen zudem die Kalendereinstellungen angepasst werden.

Zunächst werden Google API Zugangsdaten aus der Developers Console benötigt:

<https://developers.google.com/google-apps/calendar/firstapp#register>

Die dort erstellten Zugangsdaten für die Calendar API müssen bei

```
client_id = "your client ID here"
```

```
client_secret = "your client secret here"
```

eingefügt werden.

3. Außerdem sind zwei öffentliche Google Kalender notwendig. Deren IDs sind müssen in jeweils eines der:

```
Calendar*ID = "your calendar ID here"
```

Felder eingefügt werden. Die KalenderID sit in der jeweiligen Kalendereinstellung unter "Kalenderadresse" aufzufinden.

4. Auch diese Änderungen wieder in git sichern für spätere Updates:  
git commit -am "Variablen der Pythonscripte gesetzt"

### **Datenbank importieren:**

1. Im Ordner "Datenbanken" befinden sich zwei Dateien:

- db-sample.sql
- db.sql

Erstgenannte enthält Beispieldatensätze.

2. Aufruf des Ordners Datenbanken in der Konsole

3. Import der Datenbank mit

mysql -u turniere -p < db.sql

oder für die Beispieldatenbank

mysql -u turniere -p < db-sample.sql

Nach einer Eingabe des Passworts importiert MySQL die Datenbank inklusive Tabellen und gegebenenfalls mit Daten

### **Zeitsteuerung aktivieren**

1. Für das jeweilige Betriebssystem unterschiedliche zeitgesteuerte Aufrufe der Pythonscripte aktivieren (Empfohlen: wöchentlich)

## **Aktualisieren**

Der bevorzugte Weg ein Update einzuspielen ist ebenfalls über git. Die alternative Möglichkeit ist ein erneutes Herunterladen der Scripte und die erneute Eingabe der Parameter. Eine Datenbankeinrichtung entfällt jedoch.

1. Neue Version herunterladen:

git fetch origin

2. Updates in die aktuelle Konfiguration eingliedern (\$Version durch die zu installierende Version ersetzen)

git merge update-v.\$Version

## Datenbankaufbau

Die Datenbank besteht aus vier Tabellen, welche Daten beinhalten und drei weiteren Zwischentabellen, welche Verknüpfungen abspeichern.

Kerntabellen:

- turnier
  - Allgemeine Daten des Turniers (Ort, Name, Datum, Ausschreibung) und ob es bereits per Mail bearbeitet wurde.
- fechter
  - Daten die über den Fechter erforderlich sind (Name, Email) sowie das Geburtsjahr. Die Tabelle ist nicht mit "jahrgaenge" verknüpft
- waffe
  - Name der Waffe und eine zugehörige ID
- jahrgaenge
  - Die einzelnen Jahrgänge mit Beginn und Ende definiert nach Jahren. Die Daten verschieben sich jedes Jahr.

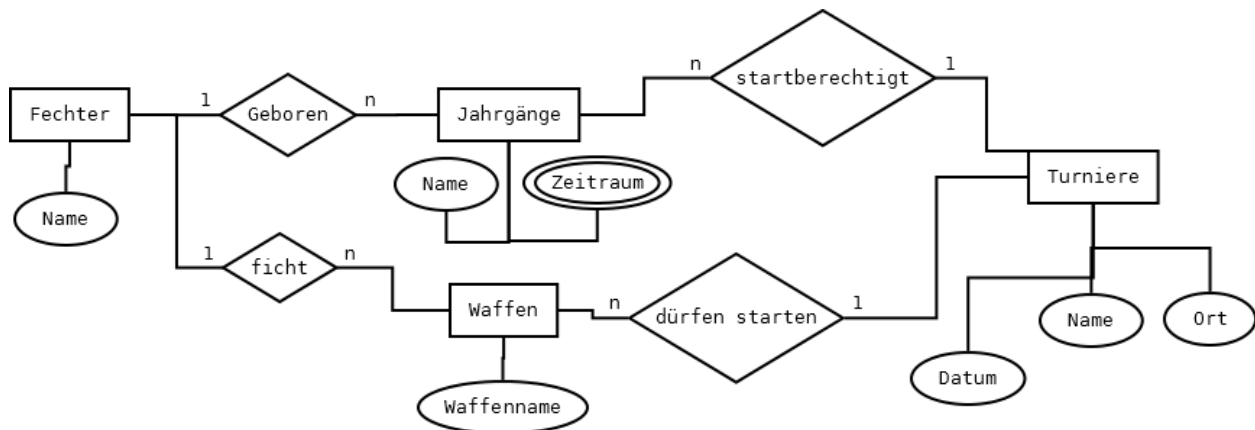
Zwischentabellen:

- fechterwaffe
  - Die FechterID und die zugehörige(n) WaffeID(s)
- waffetur
  - Die TurnierID und die IDs der zugewiesenen Waffen. Darüber hinaus wird in dieser Tabelle abgespeichert, ob es bereits einen Schreibvorgang in den Kalender gegeben hat.
- altersklassen
  - Die Verknüpfung zwischen den Turnieren und den möglichen Jahrgängen.

Zwischen der Tabelle Fechter und Jahrgänge gibt es keine Verknüpfung, da die Jahrgänge sich jährlich verschieben. Deshalb wird bei den Fechtern ein Geburtsjahr gespeichert und anschließend ein Abgleich ausgeführt, ob dieses innerhalb des definierten Bereiches des Jahrganges liegt.



## Entity-Relationship-Diagramm:



## Beziehungsdiagramm:

