

Wettkampf-Manager

Dokumentation und Anleitung

Dieses Dokument erklärt den Benutzung des Programms „Wettkampf-Manager“ für Feuerwehrsport-Wettkämpfe.

Inhaltsverzeichnis

1 Installation Windows.....	2
1.1 Download.....	2
1.2 Installation.....	2
1.3 Programm starten.....	3
1.4 Weitere Skripte.....	4
2 Installation Linux.....	5
3 Externes Gerät auslesen.....	7
3.1 Timy.....	8
3.2 Messcomputer vom Team-MV und Landesanlage MV.....	8
4 Lizenz der Programmdateien.....	9
5 Über die Entwicklung.....	9

Abbildungsverzeichnis

Abbildung 1: Entpacktes Verzeichnis unter Windows.....	2
Abbildung 2: Vollständig gestarteter Server.....	3
Abbildung 3: Firewallanfrage unter Windows.....	3
Abbildung 4: Passwort vergeben.....	4
Abbildung 5: Seriell-zu-USB-Adapter.....	7
Abbildung 6: Kabel zwischen Timy und seriellen Anschluss.....	8
Abbildung 7: Nullmodem-Schaltung.....	8

1 Installation Windows

1.1 Download

Über die Webseite Feuerwehreport-Statistik.de → Feuerwehreport → Wettkampf-Manager kann man sich die aktuelle Version herunterladen. Diese enthält eine Datenbank mit Personen, Mannschaften und Serienergebnissen, die täglich aktualisiert wird.

1.2 Installation

Nach dem Herunterladen wird die ZIP-Datei entpackt („Alle extrahieren“).

Der Pfad darf keine Leerzeichen enthalten!

Der Wettkampf-Manager ist nun einsatzbereit.

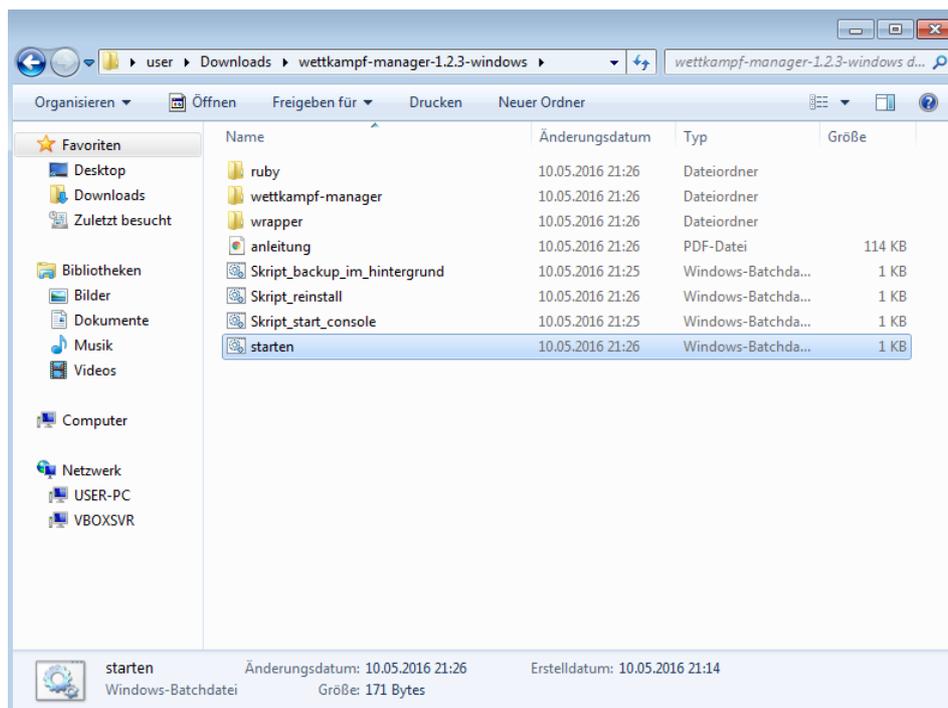
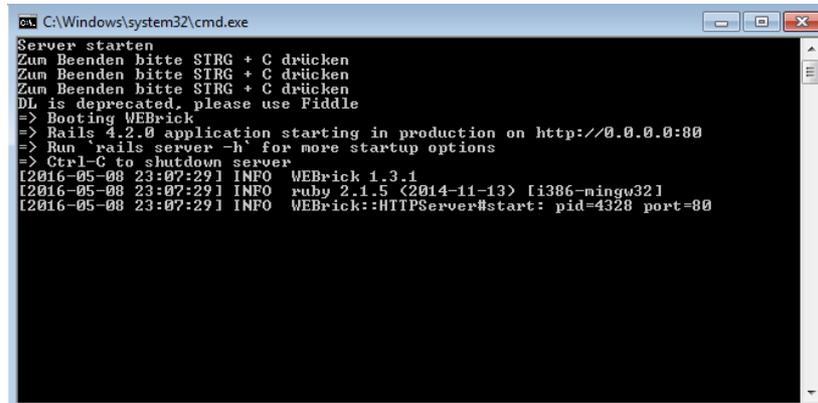


Abbildung 1: Entpacktes Verzeichnis unter Windows

Will man einen weiteren Wettkampf auswerten, reicht ein weiteres Entpacken in einen separates Verzeichnis.

1.3 Programm starten

Die Datei „starten.sh“ bzw. „starten.bat“ durch Doppelklick ausführen. Es öffnet sich ein schwarzes Fenster, welches ein paar englische Ausgaben anzeigt. Dies ist ein Webserver, der die ganze Zeit im Hintergrund laufen muss. Man muss nun warten, bis die Ausgabe „Booting WEBrick“ erscheint.



```
C:\Windows\system32\cmd.exe
Server starten
Zum Beenden bitte STRG + C drücken
Zum Beenden bitte STRG + C drücken
Zum Beenden bitte STRG + C drücken
DL is deprecated, please use Fiddle
=> Booting WEBrick
=> Rails 4.2.0 application starting in production on http://0.0.0.0:80
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
[2016-05-08 23:07:29] INFO WEBrick 1.3.1
[2016-05-08 23:07:29] INFO ruby 2.1.5 (2014-11-13) [i386-mingw32]
[2016-05-08 23:07:29] INFO WEBrick::HTTPServer#start: pid=4328 port=80
```

Abbildung 2: Vollständig gestarteter Server

An diese Stelle muss die Firewall bestätigt werden.

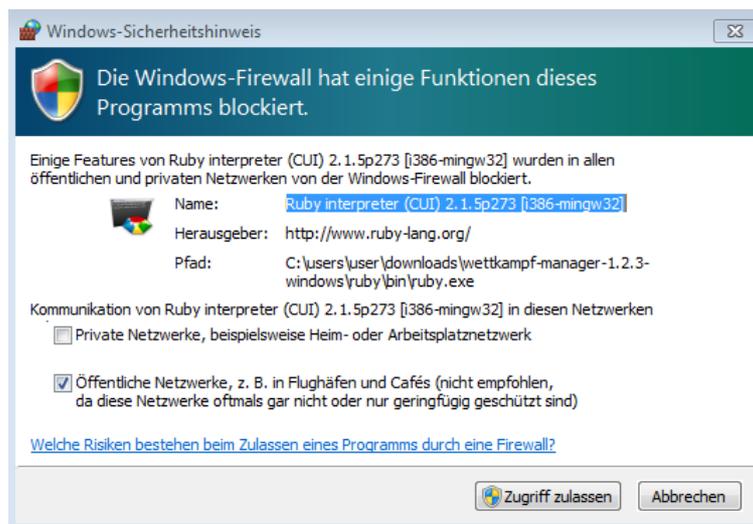


Abbildung 3: Firewallanfrage unter Windows

Nun öffnet man ein Browser seiner Wahl, vorzugsweise Firefox oder Chrome und gibt <http://localhost/> in die Adresszeile ein.

Es erscheint nach kurzer Zeit die Anmeldemaske des Wettkampf-Managers.

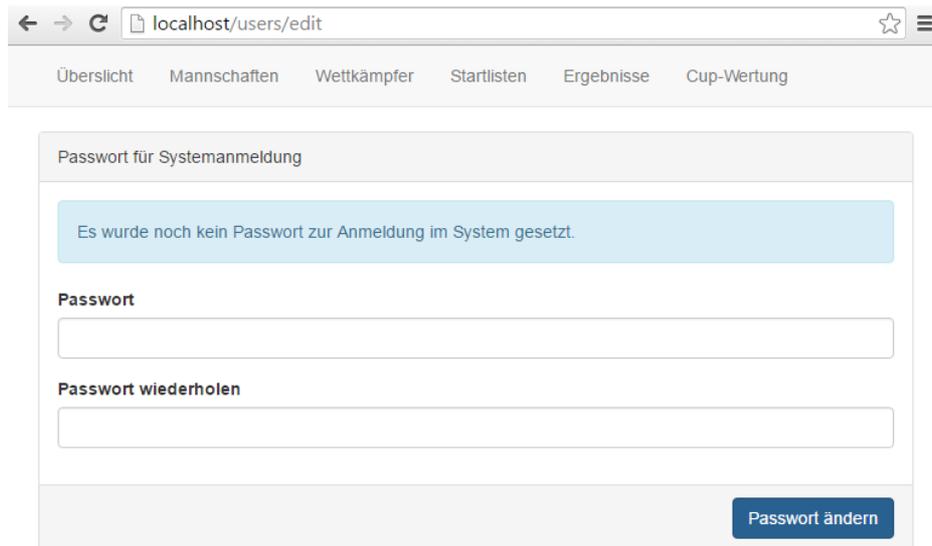


Abbildung 4: Passwort vergeben

Nach dem ersten Aufruf muss ein Administrator-Passwort vergeben werden. Dies ist nötig, da alle Rechner im Netzwerk auf den gestarteten Webserver zugreifen können. Nur Personen mit diesem Passwort können sich einloggen und Änderungen vornehmen. Alle anderen Personen haben nur lesenden Zugriff.

1.4 Weitere Skripte

Es gibt weitere nützliche Skripte, die hier kurz beschrieben werden:

<code>backup_im_hintergrund</code>	Führt alle 5 Minuten ein Backup aller Eingaben durch. Der Pfad kann im Wettkampf-Menü konfiguriert werden. Es ist sinnvoll diesen Pfad auf ein externes Medium, wie einen USB-Stick oder einer Speicherkarte zu legen, um einen Festplattenausfall zu kompensieren. Außerdem liefert diese Daten eine gewisse Änderungshistorie.
<code>start_console</code>	Führt eine Konsole zum direkten Ändern der Daten aus. Dies ist nur für Programmierer gedacht.
<code>start_api</code>	Startet ein Programm zum Auslesen der Daten aus einem externen Gerät. Siehe Externes Gerät auslesen.
<code>reinstall</code>	Führt eine erneute Installation mit Herunterladen der Datenbank durch.

2 Installation Linux

Es ist prinzipiell möglich, das Programm in jedem Betriebssystem mit Ruby-Unterstützung zum Laufen zu bekommen. Ich habe es mit Ubuntu und Debian in 32bit/64bit und arm-Variante getestet. Der folgende Ablauf ist ein Beispiel und sollte bei Bedarf angepasst werden:

1. System-Pakete installieren

```
$ sudo apt-get install nodejs curl git
```

2. RVM installieren

Siehe <https://rvm.io/>

1. PGP-Key hinzufügen:

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys
```

```
409B6B1796C275462A1703113804BB82D39DC0E3
```

```
7D2BAF1CF37B13E2069D6956105BD0E739499BDB
```

2. RVM installieren:

```
$ \curl -sSL https://get.rvm.io | bash -s stable -autolibs=read-fail
```

3. RVM laden und automatisch laden

```
$ echo 'source "$HOME/.rvm/scripts/rvm"' >> ~/.bashrc
```

```
$ source "$HOME/.rvm/scripts/rvm"
```

3. Ruby installieren

```
$ rvm install ruby-2.3.3
```

Hier wird RVM vermutlich darüber informieren, dass noch Pakete installiert werden müssen. (Zum Beispiel: Missing required packages: gawk libssl-dev make)

Dann müssen diese installiert werden:

```
$ sudo apt-get install gawk libssl-dev make
```

Danach führt man den Installationsprozess erneut durch.

Vermutlich müssen folgende Pakete installiert sein:

```
nodejs gawk libssl-dev make libc6-dev zlib1g-dev libyaml-dev libsqlite3-dev
```

```
sqlite3 autoconf libgmp-dev libgdbm-dev libncurses5-dev automake libtool
```

```
bison pkg-config libffi-dev libgmp-dev libreadline6-dev
```

4. Wettkampf-Manager herunterladen

```
$ git clone -b release --recursive
```

```
https://github.com/Feuerwehrsport/wettkampf-manager.git
```

```
$ cd wettkampf-manager/
```

5. Gems installieren und konfigurieren

```
$ doc/scripts/install.sh
```

6. Server starten

```
$ rails server -e production -p 3000 -b 0.0.0.0
```

Wie unter Windwos, gibt es weitere Aktionen, die ausführbar sind:

<pre>\$ RAILS_ENV=production rake backup_data_recurring</pre>	
	Führt alle 5 Minuten ein Backup aller Eingaben durch. Der Pfad kann im Wettkampf-Menü konfiguriert werden. Es ist sinnvoll diesen Pfad auf ein externes Medium, wie einen USB-Stick oder einer Speicherkarte zu legen, um einen Festplattenausfall zu kompensieren. Außerdem liefert diese Daten eine gewisse Änderungshistorie.
<pre>\$ rails console -e production</pre>	
	Führt eine Konsole zum direkten Ändern der Daten aus. Dies ist nur für Programmierer gedacht.
<pre>\$ rails runner -e production "API::Runner.new"</pre>	
	Startet ein Programm zum Auslesen der Daten aus einem externen Gerät. Siehe Externes Gerät auslesen.
<pre>\$ rails server -e production -p 3000 -b 0.0.0.0</pre>	
	Startet den Server auf Port 3000. Für privilegierte Ports (unter 1024) benötigt man Root-Rechte. Hierfür wäre eine Weiterleitung oder ein Proxy (Nginx) sinnvoll.

3 Externes Gerät auslesen

Zur Zeit kann man zwischen folgenden Geräten wählen:

- Timy von Alge-Timing (bis zu 8 Bahnen)
- Messcomputer vom Team-MV (2 Bahnen)
- Landesanlage Mecklenburg-Vorpommern (3 Bahnen)



Abbildung 5: Seriell-zu-USB-Adapter

Die Anschlüsse werden mittels seriellen Anschluss vorgenommen. Dabei kann man bei Laptops auf ein Seriell-zu-USB-Kabel zurückgreifen. Unter Windows muss man nach der Treiberinstallation die zugewiesene COM-Schnittstelle ermitteln.

Bei dem Start des Programms werden mehrere Abfragen getätigt. Hier eine Erläuterung:

Bitte das angeschlossene Gerät angeben:	Auswahl zwischen den bisher implementierten Geräten.
Angeschlossene Schnittstelle?	Windows z. B.: COM1 Linux z. B.: /dev/ttyUSB0
Zeiten der Disziplin:	Wird später als Hinweis auf der Webseite angezeigt.
URL zu Wettkampf-Manager?	Die vollständige URL zum Wettkampf-Manager ohne abschließenden Slash. Beispiel: http://192.168.100.30
Admin/API-Passwort für Wettkampf-Manager:	Passwort vom Admin oder vom API-Benutzer

3.1 Timy

Die Verbindung muss mit einem speziellen Kabel erfolgen. In der Abbildung 6: Kabel zwischen Timy und seriellen Anschluss kann man die Schaltung nachvollziehen.



Abbildung 6: Kabel zwischen Timy und seriellen Anschluss

3.2 Messcomputer vom Team-MV und Landesanlage MV

Die Verbindung muss mit einem Nullmodel-Kabel erfolgen. In der Abbildung 7: Nullmodem-Schaltung kann man die Schaltung nachvollziehen.

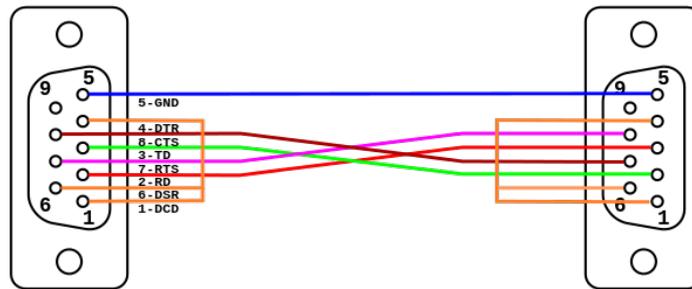


Abbildung 7: Nullmodem-Schaltung

4 Lizenz der Programmdateien

Der Quellcode für den Wettkampf-Manger steht unter der AGPLv3¹ und kann unter <https://github.com/Feuerwehrsport/wettkampf-manager> heruntergeladen werden.

Die Verfasser der Software übernehmen keine Garantie für Schäden jeglicher Art.

5 Über die Entwicklung

Der Wettkampf-Manager wird seit Frühjahr 2015 hauptsächlich von Georg Limbach als quelloffene Software entwickelt. Ziel ist es, die Ergebnisberechnung von Feuerwehrsportwettkämpfen so einfach wie möglich zu gestalten.

Als Basis wird das Framework Ruby on Rails 4.2 mit Ruby 2.1 benutzt. Weitere Informationen dazu finden sich auf Feuerwehrsport-Statistik.de² oder auf [Github](https://github.com/Feuerwehrsport/wettkampf-manager)³.

Die aktuelle Entwicklung wird durch die Lichtbit GmbH⁴ aus Rostock unterstützt.

Im Vorfeld entwickelte Daniel Grosche⁵ seit 2004 die Vorversion dieses Wettkampf-Managers. Es handelte sich um ein Datenbank-Management-System, entwickelt mit Microsoft Visual FoxPro 6.0.

1 GNU Affero General Public License - <https://www.gnu.org/licenses/agpl-3.0.de.html>

2 https://www.feuerwehrsport-statistik.de/wettkampf_manager

3 <https://github.com/Feuerwehrsport/wettkampf-manager>

4 <https://lichtbit.com>

5 <http://dgrosche.charlottenthal.de>