

---

## NAME

**ticket-check.pl** - Ein treE Ticketsystem auf neue Tickets überprüfen.

## BESCHREIBUNG

Dieses Script überprüft ein oder mehrere treE Ticketsystem(e) auf das Vorhandensein neuer Tickets. Dies erfolgt durch das anmelden beim Ticketsystem als User und Überprüfung der Seite für neue Tickets.

## KONFIGURATION

Die Konfiguration erfolgt über die Konfigurationsdatei 'ticket-check.yml', welche in dem selben Ordner wie das Script selbst erwartet wird. Die Konfigurationsdatei ist im YAML Format.

Bei dem YAML Format können mehrere Werte "zusammengefasst" werden. Hierbei ist die Anzahl der Leerzeichen oder Tabulatoren für einen Abschnitt immer gleich zu halten.

Kommentare beginnen in YAML mit einer Raute (#).

## Beispiel-Konfiguration

```
config:      # Die Basis-Konfiguration ist unter 'config:'
  interval: 3 # Wie lange nach dem Check aller Ticketsysteme gewartet
              # werden soll, bis diese
              # erneut gecheckt werden. (Sekunden)
  interval_betw_inst: 0 # Wie viel Zeit zwischen einzelnen Ticketsystemen
                       # vergehen soll. (Sekunden)
  maxretrys: 2 # Maximale Anzahl der erneuten versuche, bis ein
               # 'ticketdown' gesendet werden soll.
  verbose: 0   # Debug Informationen ausgeben. (1)

alarmd:      # Konfiguration für den AlarmClient genaueres siehe
              # Dokumentation zu AlarmClient.pm.
  host: '127.0.0.1'
  port: 5061
  client: ticket
  pass: test
  maxretrys: 5
  after: 5
  separator: '::'

instances:   # Unter diesem Punkt werden die einzelnen Ticketsysteme
              # definiert.
  example:   # Name der Instanz
    url: 'https://ticket.example.com/index/status/1/' # URL zur 'Neue
    Tickets' Seite.
    user: 'checkuser' # Benutzername
    pass: 'Pa$$w0rD.' # und Passwort für das Ticketsystem.

  ticket2:   # Beispiel für ein zweites Ticketsystem
    url: 'https://ticket2.example.net/index/status/1/'
    user: 'checkuser'
    pass: 'pwTest'
```

## GESENDETE ALARME

**Folgende Alarme werden an den alarmd.pl Server übertragen:**

\* neu

Falls eine Instanz einen neues Ticket enthält.

Erst wenn keine neuen Tickets mehr da sind, also kein Ticket mehr in keinem der Ticketsysteme unter der Kategorie 'neu' ist, wird der Alarm aufgehoben.

\* `ticketdown`

Falls eine Instanz nach vordefinierter Anzahl erneuter versuche nicht erreichbar ist.

## FUNKTIONSWEISE

In dem Script wird eine Endlosschleife gestartet, in welcher für jedes Ticketsystem die jeweilige Seite für neue Tickets aufgerufen wird. In dieser Seite wird nach Ticket-ID's gesucht. Ticket-ID's fangen immer mit einer Raute (#) an und haben dahinter eine fortlaufende Zahl. (Beispiel: #47615). Alle gefundenen Tickets werden in einem Hash zu dem jeweiligen Ticketsystem gespeichert. Ist ein Ticket dabei, welches bei der letzten Überprüfung nicht dabei war (neues Ticket) so wird ein 'neu' Alarm an den **alarmd.pl** Server gesendet. Hierfür wird das Perl-Modul **AlarmClient.pm** benutzt.

Kann ein Ticketsystem nicht erreicht werden, so wird dessen Retry-Counter (`$retrys`) für um eins erhöht. Überschreitet ein Retry-Counter das vordefinierte Limit ('`maxretrys`'), so wird ein 'ticketdown' Alarm an den Server übermittelt.

## AUTOR

**ticket-check.pl** - Geschrieben von Florian Schießl (florian@trans.net) im Rahmen des Projektes zur IHK Abschlussprüfung zum Fachinformatiker für Systemintegration Sommer 2013. Entwicklungsbeginn ist der 12.3.2013.

## SIEHE AUCH

**AlarmClient.pm** - Das verwendete Perl Modul zur Kommunikation mit dem **alarmd.pl** Server.

**alarmd.pl** - Der Alarm-Server.

<http://www.deam.org/tree/> - Das treE Ticketsystem.