

## NAME

**AlarmClient.pm** - Perl Modul, um mit einem **alarmd.pl** über Netzwerk zu kommunizieren.

## GEBRAUCH

```
use AlarmClient;

my $ac = AlarmClient->new(
    'alerthost.example.com',
    5061,
    'Testclient',
    'Serverpasswort',
    5,
    15,
    '::',
);

$ac->alert('EinAlarm', 1); # Alarm tritt auf

$ac->alert('EinAlarm', 0); # Alarm vorbei
```

## OPTIONEN

Alle Optionen werden im Konstruktor in nachfolgender Reihenfolge übergeben:

Host

Der Hostname oder die IP-Adresse des **alarmd.pl** Servers.

Port

Der Port, auf welchem der Server lauscht.

Clientname

Der Name, mit dem sich die Instanz des **AlarmClient** gegenüber dem Server meldet.

Passwort

Das Passwort des **alarmd.pl** Servers.

Erneute Versuche

Die maximale Anzahl der erneuten Versuche, bis aufgegeben wird, den Server zu kontaktieren.

Warten

Wie viele Sekunden nach einem erfolglosem Verbindungsversuch gewartet werden soll.

Trenner

Die Werte, welche an den Server geschickt werden, werden durch diese Zeichenkette getrennt.

## FUNKTIONEN

**alert(\$name, \$neu)**

**Übergabeparameter:**

name

Der Name des Alarms.

neu

Ob der Alarm neu ist (1) oder vorbei ist (0).

**Rückgabewerte:**

0 = Erfolg; 1 = Fehler

**FUNKTIONSWEISE**

Tritt in dem verwendenden Script ein Alarm auf, so wird von diesem die Funktion `alert()` aufgerufen und der Name des Alarms, sowie eine Eins (1) übergeben. Die Eins markiert, dass ein Alarm auftritt. Eine Null (0) hingegen markiert, dass ein Alarm vorbei ist.

Die Funktion `alert()` baut zunächst aus den übergebenen Werten die Nachricht für den Server auf. Genaueres Siehe Abschnitt PROTOKOLL. Daraufhin wird eine Verbindung zum Server aufgebaut und die Nachricht übermittelt. Sollte ein Verbindungsaufbau nicht möglich sein, oder die Antwort vom Server nicht "ok", "err" oder "unknown" lauten, so wird es nach der vordefinierten Wartezeit erneut versucht. Sind alle erneuten Versuche auch fehlgeschlagen, so wird eine Eins an den Aufrufer zurückgegeben. Wurde die Nachricht erfolgreich übermittelt, so wird eine Null zurückgegeben.

**PROTOKOLL****Client zu Server**

Folgende Nachricht wird zum auslösen eines Alarms an den Server gesendet:

```
Passwort::Clientname::Alarmname::1
```

Bei dem Passwort handelt es sich um das Serverpasswort. Der Clientname ist der Name, mit welchem sich die Instanz dieses Modules gegenüber dem Server meldet. Alarmname ist Name des Alarms, welcher ausgelöst oder aufgehoben werden soll. Die Eins markiert, dass der Alarm ausgelöst wird. Bei den doppelten Doppelpunkten handelt es sich um den vordefinierten Trenner. Dieser kann je nach Konfiguration abweichen, muss aber immer beim Server und Client der gleiche sein.

Folgende Nachricht hebt den ausgelösten Alarm wieder auf:

```
Passwort::Clientname::Alarmname::0
```

Die Null markiert das Aufheben des Alarms.

**Server zu Client**

Die Antwort des Servers beschränkt sich auf ein einfaches

```
'ok'
```

falls der Alarm erfolgreich aufgenommen wurde,

```
'err'
```

falls die Anfrage ungültig war (sollte mit diesem Modul nicht auftreten) oder

```
'unknown'
```

falls der Alarm auf dem Server nicht definiert ist. Ein 'err' und ein 'unknown' lösen ein akustisches Signal auf dem Server aus, um über die Fehlkonfiguration zu benachrichtigen.

Ist das Passwort falsch, wird der Server sofort und ohne Quittierung die Verbindung beenden.

**AUTOR**

**AlarmClient.pm** - 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

[alarmd.pl](#) - Der Alarm-Server.