

Kalenderdateien aus Webkalendern erstellen

Ein interessanter Kalender im Web, aber keine iCal-Datei. Schade, da muss man wohl alles abtippen, um es in den eigenen Kalender zu übernehmen. Oder auch nicht! Für den Deutschen Perl-Workshop im letzten Jahr, wollten die Orgas eine Kalenderdatei mit den Vorträgen zur Verfügung stellen. Damit ich das nicht abtippen muss, habe ich ein kleines Skript geschrieben, das diese Dateien erzeugt.

Als erstes wird der Zeitplan des Workshops geholt. Dafür benutze ich einfach `LWP::Simple` mit der `get` Methode

```
my $url = 'http://www.perl-workshop.de/
de/2009/zeitplan.html';
my $content = LWP::Simple::get( $url );
```

Danach müssen erstmal die Informationen aus der Webseite geholt werden. Ich persönlich bevorzuge für solche Aufgaben das Modul `Web::Scraper`, da es einfach zu benutzen ist und der Perl-Code schlank und lesbar bleibt. In der Ausgabe 7 von `$foo` habe ich das Modul schonmal genauer vorgestellt.

```
my $scraper = scraper {
  process 'div[class="zeitplan"]',
  'tage' => scraper {
    process 'table',
    'vortraege[]' => scraper {
      process 'td',
      'eintraege[]' => 'TEXT';
    };
  };
  result 'tage';
};

my $all_days = $scraper->scrape( $content )
->{vortraege};
```

Aber das Modul soll hier ja nicht Hauptthema sein, deshalb nur eine kurze Erläuterung: Der Zeitplan an sich ist innerhalb eines `div`s mit der CSS-Klasse "zeitplan". Die Vorträge wiederum sind in einer Tabelle (eine Tabelle pro Tag) und alle benötigten Informationen in den Zellen der Tabelle.

Dazu muss man noch wissen, dass immer vier Tabellenzellen zusammen die Informationen zu einem Vortrag liefern: Beginn, Dauer, Vortragender und Titel.

Danach beginnt die eigentliche Arbeit für die Kalenderdatei. Damit ich mich nicht näher mit dem iCal-Format beschäftigen muss, benutze ich das Modul `Data::ICal`. Für die Zeit-Felder empfiehlt es sich, noch das Modul `Date::ICal` zu verwenden, da man sonst genau wissen muss, wie Zeitangaben im iCal-Format aussehen. Ich bin da anfangs drübergestolpert, und habe mich dann gewundert, warum mein Kalender bei den Zeiten nur Murks angezeigt hat.

Als erstes muss ein neues iCal-Objekt erzeugt werden, das dann die komplette Kalenderdatei repräsentiert.

```
my $ical_all = Data::ICal->new;
```

Danach werden die einzelnen Tage respektive Tabellen durchgegangen und dort wiederum alle Einträge (Listing 1). Wie schon weiter oben geschrieben, gehören immer vier Zeilen zu einander, die dann die Informationen über den Vortrag ergeben.

Die Zeilen, die "--" enthalten werden rausgenommen, da die Pausen nicht mit in der Kalenderdatei auftauchen sollen. Danach werden die Zeitangaben berechnet. Die Funktion `_get_epoche` berechnet mit Hilfe der Funktion `timelocal` aus `Time::Local` die Epochensekunden für einen Vortrag.

Für die einzelnen Einträge innerhalb der Kalenderdatei wird ein Event-Objekt erzeugt, das als Attribut eine Zusammenfassung, Startzeitpunkt und Endzeitpunkt hat. Für dieses Event-Objekt benötigen wir auch das `Date::ICal`-Modul und die Epochensekunden. Danach muss das Event noch mittels `add_entry` zu der Kalenderdatei hinzugefügt werden.



Zum Schluss werden die Kalenderinformationen in eine .ics-Datei geschrieben und schon ist die Kalenderdatei fertig.

```
open my $fh, '>', 'gpw2009.ics' or die $!;
print {$fh} $ical_all->as_string;
close $fh;
```

Natürlich muss das Skript für einen anderen Webkalender entsprechend angepasst werden. Aber das Beispiel zeigt, dass mit `Data::ICal` leicht Kalenderdateien erstellt werden können.

Renée Bäcker

```
for my $i ( 0 .. $#all_days ) {
    my $date = $days[$i];
    my $day = $all_days->[$i];

    my (undef, $entries) = %$day;
    while( @$entries ) {
        my $time = shift @$entries;
        my $duration = shift @$entries;
        my $presenter = shift @$entries;
        my $title = shift @$entries;

        next if $presenter =~ /--/;

        my $dtstart = _get_epoche( $date, $time );
        $duration =~ s/\s*min\s*(?:\(Tutorial\))?//;
        $duration = 30 if $duration =~ /BOF/;

        my $event = Data::ICal::Entry::Event->new;
        $event->add_properties(
            summary => sprintf( "%s: %s", $presenter, $title ),
            dtstart => Date::ICal->new( epoch => $dtstart )->ical,
            dtend => Date::ICal->new( epoch => $dtstart + ( $duration * 60 ) )->ical,
        );

        $ical_all->add_entry( $event );
    }
}
```

Listing 1



Perlfragen.de

Das Team von "\$foo - Perl Magazin" stellt auf <http://perlfragen.de> eine Übersetzung der aktuellen Perl FAQ zur Verfügung. Damit soll Perl-Einsteigern ein Anlaufpunkt gegeben werden. Da noch nicht viel übersetzt ist, wird es vorübergehend einen Deutsch-Englisch-Mischmasch geben, der aber nach und nach abgebaut wird. Mithilfe ist natürlich gerne gesehen.

Wir versuchen die möglichst aktuelle Perl FAQ zu verwenden und schauen uns einmal im Monat die Änderungen am Original an. Wer selbst einen Blick auf das Original werfen will,

kann sich die Quellen unter <http://github.com/briandfoy/perlfaq> anschauen. Die Links in den perlfaq-Dokumenten wurden für die Übersetzung nicht angepasst. So finden sich im Abschnitt "Wie kann ich etwas zu den perlfaq beitragen" die Links auf das Repository der englischen perlfaq.

Die Übersetzungen stellen wir auch im Wiki von Perl-Community.de und auf Github unter http://github.com/reneeb/perlfaq_de zur Verfügung. So kann jeder die Quellen holen und an einer eigenen Version arbeiten und/oder Fehler beheben.