cal.sql

Schema for table to hold event data

```sql
CREATE TABLE events (
  id INT UNSIGNED NOT NULL,
  summary VARCHAR(255),
  location VARCHAR(255),
  start_dt DATETIME NOT NULL,
  end_dt DATETIME,
  duration VARCHAR(255),
  tz VARCHAR(255),
  rrule VARCHAR(255)
);
```

Event.php

Classes to generate iCalendar output

```php
<?php

require_once 'DB.php';

class Calendar {
    private $product = '-//Acme Software\, Inc./ExampleCalApp 1.2.3//EN';
    private $events;

    public function add(Event $e) {
        $this->events[] = $e;
    }

    public function iCal() {
        $ical =<<<_ICAL_
BEGIN:VCALENDAR
PRODID:{$this->product}
VERSION:2.0
CALSCALE:GREGORIAN
_ICAL_;
        foreach ($this->events as $e) {
            $ical .= $e->iCal();
        }
        $ical .= "END:VCALENDAR\n";
        return $ical;
    }

    public function __toString() {
        return $this->iCal();
    }
```

```
}

class Event {

    private $id;
    private $summary;
    private $location;
    private $start;
    private $end;
    private $duration;
    private $tz;
    private $rrule;

    public function __construct($info) {
        $this->id = $info['id'];
        $this->summary = $info['summary'];
        /* If there's no time zone, treat start (and end)
         * as localtime. If time zone is set and is 'Z',
         * treat them as GMT. If time zone is set to something
         * else, treat them as local to that time zone
         */
        $this->start = self::ISODate($info['dt_start']);
        if ($info['end']) {
            $this->end = self::ISODate($info['dt_end']);
        }
        if ($info['location']) {
            $this->location = $info['location'];
        }
        if ($info['duration']) {
            $this->duration = $info['duration'];
        }
        if ($info['tz']) {
            $this->tz = $info['tz'];
        }
        if ($info['rrule']) {
            $this->rrule = $info['rrule'];
        }
    }

    public function __toString() { return $this->iCal(); }

    public function iCal() {
        $ical = "BEGIN:VEVENT\nSUMMARY:{$this->summary}\n";
        $ical .= "UID:{$this->id}\n";
        $tzprefix = is_null($this->tz) ? '' : ";TZID={$this->tz}";
        if (! is_null($this->location)) {
                $ical .= "LOCATION:{$this->location}\n";
        }
        $ical .= "DTSTART{$tzprefix}:{$this->start}\n";
        if (! is_null($this->duration)) {
            $ical .= "DURATION:{$this->duration}\n";
        }
        if (! is_null($this->end)) {
            $ical .= "DTEND{$tzprefix}:{$this->end}\n";
        }
        if (! is_null($this->rrule)) {
            $ical .= "RRULE:{$this->rrule}\n";
        }
        $ical .=  "END:VEVENT\n";

        return $ical;
    }


    private static function ISODate($dt) {
        $year = substr($dt,0,4);
        $mo   = substr($dt,5,2);
        $dy   = substr($dt,8,2);
        $hr   = substr($dt,11,2);
        $mn   = substr($dt,14,2);
        $sec  = substr($dt,17,2);
        return "$year$mo{$dy}T$hr$mn$sec";
    }
}

```

config.php

Basic app/DB configuration

```
<?php
require_once 'DB.php';
$dsn = 'mysql://user:password@host/database';

$db = DB::Connect($dsn);
if (DB::isError($db)) {
    die("Can't connect to $dsn: " . $db->getMessage());
 }
$db->setFetchMode(DB_FETCHMODE_ASSOC);
$db->setErrorHandling(PEAR_ERROR_CALLBACK,'dbError');

function dbError($e) {
    print '<pre>'; var_dump($e); print '</pre>';
}
```

calendar.php

Generate iCalendar output

```php
<?php

require_once 'config.php';
require_once 'Event.php';

$events = $db->getAll('SELECT * FROM events');

$calendar = new Calendar();
foreach ($events as $event) {
    $calendar->add(new Event($event));
}

header('Content-type: text/calendar');
print $calendar;
```

list.php

Display human-readable (but ugly) event summary for debugging.

```php
<?php

require_once 'config.php';

echo '<table border="1">';
foreach ($db->getAll('SELECT * FROM events') as $i => $event) {
    if ($i == 0) {
        echo '<tr><th>' . implode('</th><th>', array_keys($event)) . '</th></tr>';
    }
    echo '<tr><td>' . implode('</td><td>',$event).'</td></tr>';
}
```

add.php

Add a new event

```php
<?php
require 'config.php';
require_once 'HTML/QuickForm.php';

$form = new HTML_QuickForm();

$form->addElement('text','summary','Event summary:');
$form->addElement('text','location','Event location:');
```

```
$form->addElement('date','start_dt','Event start:',
                  array('format' => 'd M Y, h:i:a',
                        'optionIncrement' => array('i' => 5)));
$form->addElement('date','end_dt','Event end:',
                  array('format' => 'd M Y, h:i:a',
                        'optionIncrement' => array('i' => 5),
                        'addEmptyOption' => true,
                        'emptyOptionValue' => 0,
                        'emptyOptionText' => '--'));

$form->setDefaults(array('start_dt' => time(),
                         'end_dt' => array('d' => 0)));

$form->addElement('submit','submit','Save Event');

$form->addRule('summary','Event summary is required','required');
$form->addRule('start_dt','Event start is required','required');

// @todo check end date if supplied
if ($form->validate()) {
    $form->process('saveEvent');
 } else {
    $form->display();
 }


function saveEvent($formData) {
    global $db;

    $start = formToStamp($formData['start_dt']);
    $end = $formData['end_dt']['d'] ?
        formToStamp($formData['end_dt']) : null;

    $id = $db->nextID('events');
    $db->query('INSERT INTO events
(event_id,summary,location,start_dt,end_dt)
                     VALUES (?,?,?,?,?)',
               array($id,
                     $formData['summary'],
                     $formData['location'],
                     $start, $end));

    print '
Saved event <br/>
<a href="list.php">List all events</a>';
}

function formToStamp($d) {
    // Turn the hour into 24hr time
    if (($d['a'] == 'am') && ($d['h'] ==12)) {
        $d['h'] = 0;
    } elseif (($d['a'] == 'pm') && ($d['h'] != 12)) {
        $d['h'] += 12;
    }

    return date('Y-m-d H:i:s',
                mktime($d['h'],$d['i'],0,$d['M'],$d['d'],$d['Y']));
}
```