Module development tutorial:

pants.module

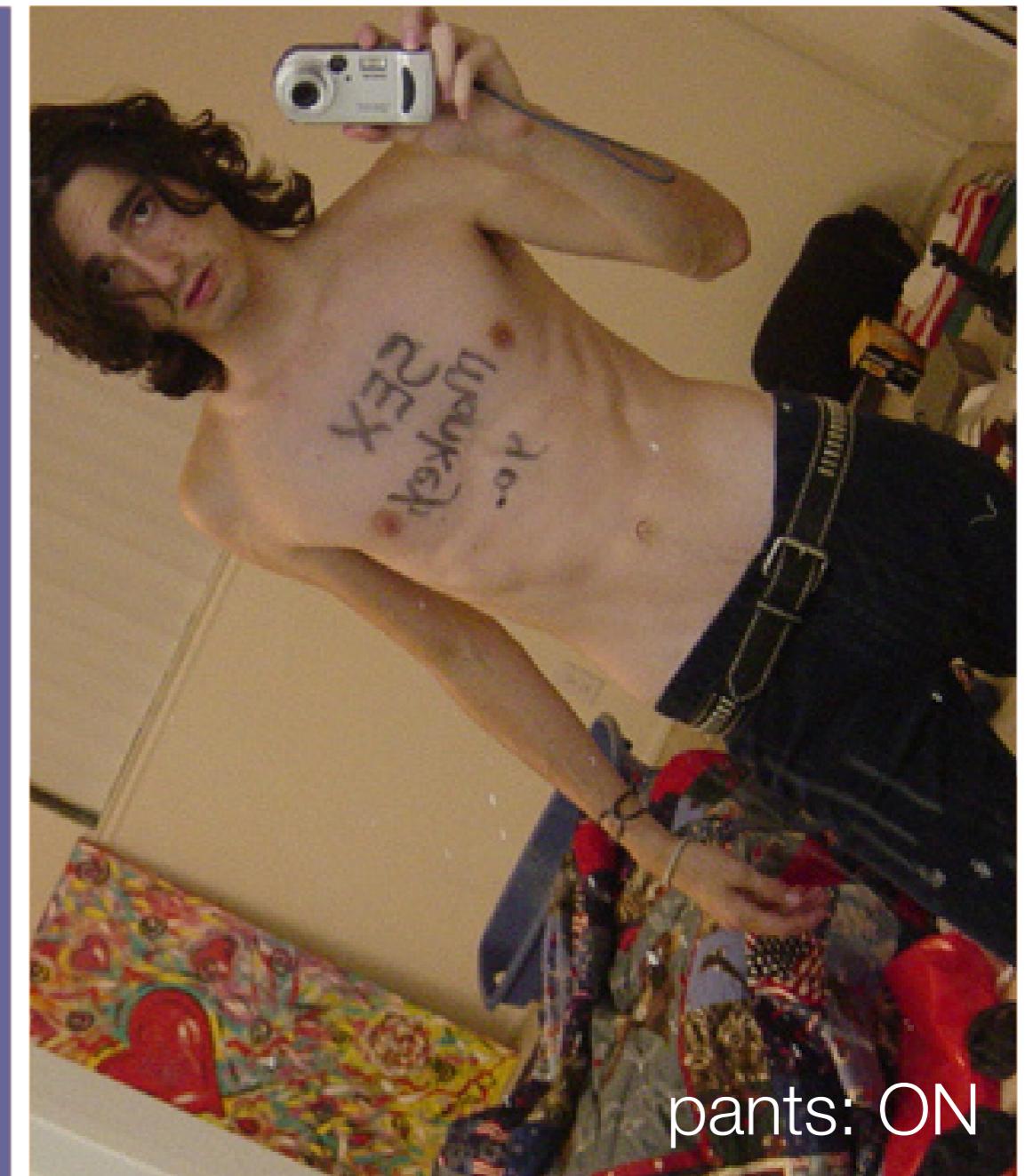DrupalCon Brussels
September 22, 2006

Pants? What? Why?

- We're gonna learn module development by example

- We all own pants

- We all wear pants (sometimes)

pants: off

Pants module history

- ▶ This is Andy Smith.

- ▶ Andy is wearing pants (but not a shirt)

- ▶ Andy wrote the first pants.module

pants: ON

Feature Requirements

▶ Track User Pants status

  ■ Display current status on profile page

  ■ Allow users to update their "pants status"

  ■ Show a block of recent pants updates

▶ Allow pants updates via XML-RPC

First Steps

- ▶ Check drupal.org for existing similar modules
  - If so, contact the module maintainer - collaborate!
- ▶ Check groups.drupal.org for similar areas of interest
- ▶ Read: http://drupal.org/contributors-guide

Let's get started!

▶ We need some files:
- pants.info
- pants.install
- pants.module

▶ We need some code

▶ No need for pants

pants.info

```
; $Id$
name = Pants
description = Tracks pants status for users.
```

pants.install

- Implement table creation per-type.

- Ideally support MySQL & Pgsql

- Include any updates by implementing pants_update_N

```php
<?php
function pants_install() {
  switch ($GLOBALS['db_type']) {
    case 'mysql':
    case 'mysqli':
      // Create the pants status table.
      db_query("CREATE TABLE IF NOT EXISTS {pants} (
        uid INTEGER UNSIGNED NOT NULL,
        status INTEGER NOT NULL DEFAULT 0,
        PRIMARY KEY (uid)
        )/*!40100 DEFAULT CHARACTER SET utf8 */;");
      // Create pants history table.
      db_query(
"CREATE TABLE IF NOT EXISTS {pants_history} (
        uid INTEGER UNSIGNED NOT NULL,
        status INTEGER NOT NULL,
        changed INTEGER NOT NULL,
        changed_by INTEGER UNSIGNED NOT NULL,
        PRIMARY KEY (uid)
        ) /*!40100 DEFAULT CHARACTER SET utf8 */;");
      break;
    case 'pgsql':
      // TODO
      break;
  }
}
```

pants.module

▶ Let's get ready for pants.

helper functions

- By convention "private" helper functions are prefixed by "_<module>"

- Modules with lots of helpers can use separate ".inc" files to reduce RAM usage.

- Higher granularity => Better reuse.

```php
<?php
function _pants_change($uid) {
  $status = 1 - _pants_get($uid);
  _pants_set($uid, $status);
}

function _pants_get($uid, $readable = FALSE) {
  $status = db_result(db_query("SELECT status FROM {pa
));
  return $status;
}

function _pants_set($uid, $status) {
  db_query('DELETE FROM {pants} WHERE uid = %d', $uid
);
  db_query('INSERT INTO {pants} (uid, status) VALUES (
);
  db_query('INSERT INTO {pants_history} (uid, status,
());
}

function _pants_status($status) {
  return ($status) ? t('On') : t('Off');
}
```

hook_menu

▶ Drupal's menu system is multipurpose. It:

- registers URL-based callbacks (REST)

- populates navigation menus

- serves as highlevel access control

```php
<?php
function pants_menu($may_cache) {
  $items = array();
  if ($may_cache) {
    $items[] =
      array('title' => t('Change Pants'),
            'path' => 'pants/change',
            'callback' => 'pants_change',
            'type' => MENU_CALLBACK,
            'access' =>
            user_access('change pants status')
      );
  }
  return $items;
}

/**
 * Implementation of hook_perm
 */
function pants_perm() {
  return array('change pants status');
}
```

hook_user

▶ Several options implemented:

- ■ "load": get pants status

- ■ "insert" / "update": set pants status

- ■ "view" : show pants status on profile page

```php
<?php
function pants_user($op, $edit, &$account, $category
) {
  switch ($op) {
    case 'form':
      $form['pants'] =
        array('#type' => 'radios',
              '#title' => t('Pants'),
              '#default_value' => $account->pants,
              '#options' => array(0 => t('Off'), 1 =>
)),
              '#description' => t('Your current pants
),
        );
      return $form;
    case 'load':
      $account->pants = _pants_get($account->uid);
      break;
    case 'insert':
    case 'update':
      _pants_set($account->uid, $edit['pants']);
      unset ($edit['pants']);
      break;
    case 'view' :
      $items[] = array('title' => t('Pants status'),
                       'value' => _pants_status(_pant
->uid)),
                       'class' => 'user');
      return array('' => $items);
  }
}
```

User Edit

User View

admin

view	edit

History

Blog
> **view recent blog entries**

Member for
> 2 weeks 1 day

Pants status
> Off

hook_block

- ▶ Again: multiple "modes" for hook_block

  - ■ "list": used to show available blocks

  - ■ "view": called with $delta to display an individual block

```php
<?php
function pants_block($op = 'list', $delta = 0) {
  global $user;

  switch ($op) {
    case 'list':
      $blocks[0]['info'] = t('Your Pants');
      $blocks[1]['info'] = t('Recent Pants');
      return $blocks;
    case 'view':
      switch ($delta) {
        case 0:
          $output .= "<div>". _pants_status(_pants_get
->uid)) ."</div>\n";
          $output .= l(t('change'), 'pants/change');
          $block['subject'] = t('Your Pants are:');
          $block['content'] = $output;
          return $block;
        case 1:
          $block['subject'] = t('Recent Pants');
          $block['content'] = theme('pants_block');
          return $block;
      }
  }
}
```

Themeable functions

▶ Allow themers to override your markup

▶ Good idea anytime you're including raw HTML in your code

```php
<?php

function theme_pants_block() {
  $result = db_query_range(
          'SELECT u.uid, u.name, p.status, p.changed
           FROM {users} u INNER JOIN {pants_history} p
           WHERE u.status = 1 ORDER BY p.changed DESC',
);
  $items = array();
  while ($pants = db_fetch_object($result)) {
    $label = ($pants->status) ?
t('put pants on') :t('took pants off');
    $items[] = theme('username', $pants)
        .' '. $label .'<br />'
        . t('@time ago',
          array('@time' => format_interval(time() -
$pants->changed)));
  }
  return theme('item_list', $items);
}
```

Once more with AJAX!

▶ Drupal 5.0 features jQuery in core

▶ jQuery makes JS/AJAX fun!

```javascript
$(document).ready(
  function() {
    // Get Drupal basePath
    var path = Drupal.settings.pants.basePath;

    $("a#pants").click( function() {
      $.get(path + "pants/change", function(result) {
        $(".pants-status").html(result).fadeIn("slow");
      } );
      return false;
    });
  });
```

Pants block with AJAX

```php
<?php
function pants_block($op = 'list', $delta = 0) {
  global $user;

  switch ($op) {
    case 'list':
      $blocks[0]['info'] = t('Your Pants');
      $blocks[1]['info'] = t('Recent Pants');
      return $blocks;
    case 'view':
      switch ($delta) {
        case 0:
          $output .= '<div class="pants-status">';
          $output .= _pants_status(_pants_get($user->uid));
          $output .= "</div>\n";
          $output .= l(t('change'), 'pants/change', array('id' => 'pants'));
          drupal_add_js(array('pants' => array('basePath' => base_path())), 'setting');
          drupal_add_js(drupal_get_path('module', 'pants') . '/pants.js');
          $block['subject'] = t('Your Pants are:');
          $block['content'] = $output;
          return $block;
        case 1:
          $block['subject'] = t('Recent Pants');
          $block['content'] = theme('pants_block');
          return $block;
      }
  }
}
```

Pants Blocks

Your Pants are:

Off
change

Recent Pants

- **admin** took pants off 1 min 15 sec ago
- **admin** put pants on 1 min 17 sec ago
- **admin** took pants off 1 min 19 sec ago
- **admin** put pants on 1 min 20 sec ago
- **admin** took pants off 1 min 30 sec ago
- **admin** put pants on 1 min 31 sec ago
- **admin** took pants off 1 min 35 sec ago
- **admin** put pants on 1 min 53 sec ago
- **admin** took pants off 10 hours 5 min ago
- **admin** put pants on 10 hours 5 min ago

hook_xmlrpc

- ▶ Allows you to expose callbacks over XML-RPC

- ▶ Re-use core functionality

- ▶ Change pants status from:

  - ■ Desktop Applications

  - ■ Other Sites

```php
<?php
function pants_xmlrpc() {
  return array(
    array(
      'drupal.changePants',
      'pants_xmlrpc_change',
      array('array', 'string', 'string'),
      t('Changes pants status for the given user')),
    array(
      'drupal.getPants',
      'pants_xmlrpc_get',
      array('array', 'string', 'string'),
      t('Changes pants status for the given user')));
}

function pants_xmlrpc_change() {
  global $user;

  $user = user_authenticate($username, $password);
  if (!$user-
>uid || !user_access('change pants status', $user)) {
    return xmlrpc_error(1, t('Wrong username or passwo
));
  }

  return _pants_change($user->uid);
}
```
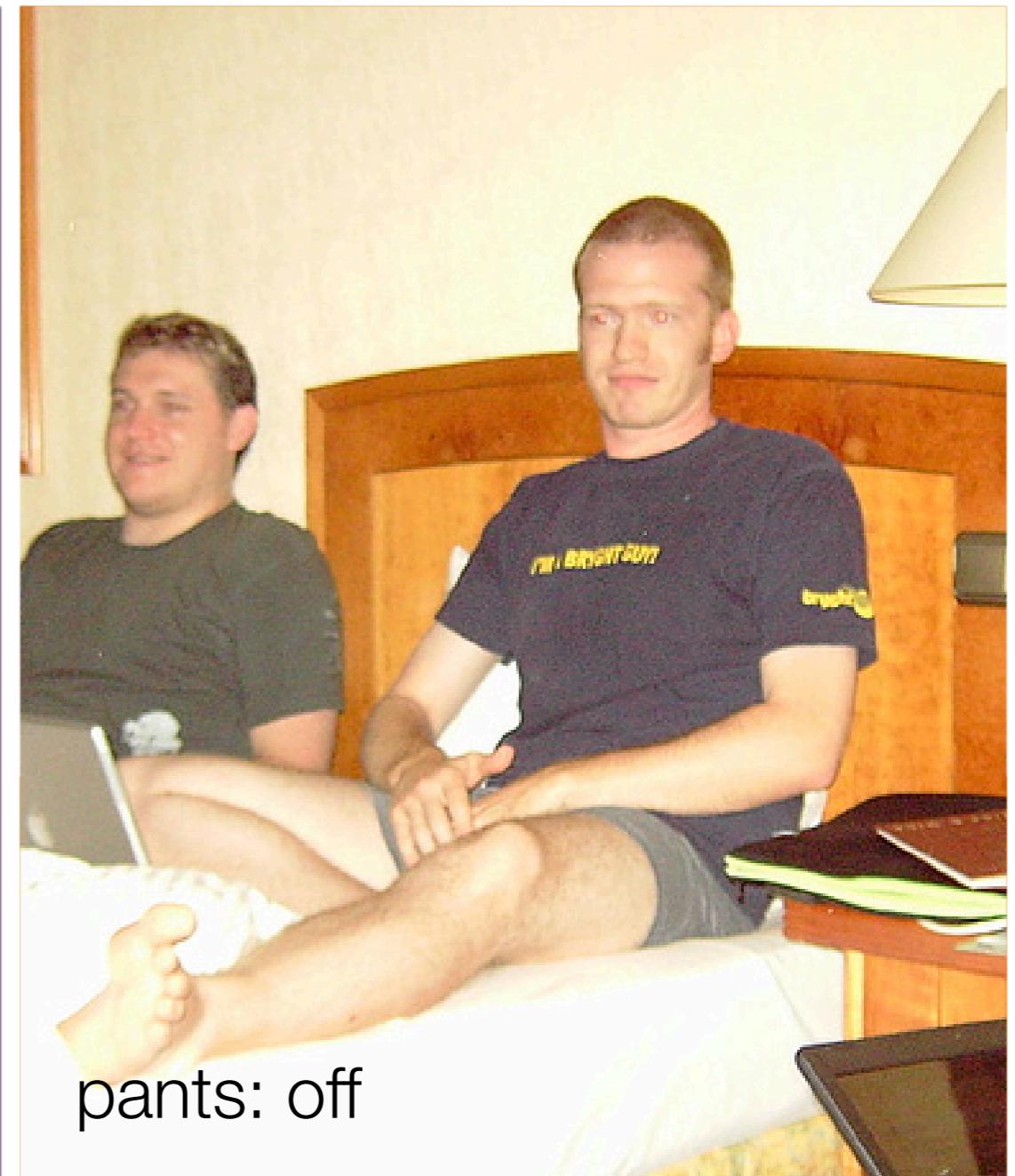
Pants References

- ▶ http://www.nopantsday.com/

- ▶ http://term.ie/p/browser/drupal_pants/

- ▶ http://en.wikipedia.org/wiki/Pants

Developer References

- http://api.drupal.org/

- http://groups.drupal.org/

- irc://irc.freenode.net/#drupal

- http://drupal.org/contributors-guide

- http://drupal.org/update/modules

Thanks!

- James Walker

- Bryght Guy

- mailto/xmpp :
  james@bryght.com

- http://walkah.net/

pants: off