Desarrolladores: Bienvenidos a Drupal 7

Drupalcamp Spain 2011, Sevilla 1-2 Octubre

commerce guys

Sobre mí

- Desarrollador Drupal en Commerce Guys.
- Miembro hiperactivo de la comunidad drupalera de habla hispana.

Pedro Cambra
@pcambra

commerce guys

Drupal 7

Drupal 7 incorpora gran cantidad de novedades a nivel de diseño, usabilidad pero quizás el mayor cambio que se ha producido entre las versiones 6x y 7x es a nivel **desarrollo**.

commerce guys

Entidades en Core

Las entidades en Drupal añaden una **nueva capa de abstracción** basada en objetos sobre los datos que permite que todo el contenido comparta API y workflow.

Nodos, usuarios, términos de taxonomía, vocabularios, comentarios, ficheros... son ahora entidades

commerce guys

Entidades en Core

"~~Everything is a node~~"

vs

"**Nodes are entities too**"

commerce guys

Ejemplo #1: Entity Controllers

```
class CommerceProductEntityController extends
DrupalCommerceEntityController {
(..)
  public function create(array $values = array()) {
    return (object) ($values + array(
      'product_id' => '',
      'is_new' => TRUE,
      'sku' => '',
      'title' => '',
      'uid' => '',
      'status' => 1,
      'created' => '',
      'changed' => '',
    ));
  }
(..)
}
```

Ejemplo #2: EntityFieldQuery

```
function
commerce_product_reference_commerce_product_can_delete($product) {
  // Use EntityFieldQuery to look for line items referencing this
  // product and do not allow the delete to occur if one exists.
  $query = new EntityFieldQuery();

  $query
    ->entityCondition('entity_type', 'commerce_line_item', '=')
    ->entityCondition('bundle', 'product', '=')
    ->fieldCondition('product', 'product_id',
        $product->product_id, '=')
    ->count();

  return $query->execute() > 0 ? FALSE : TRUE;
}
```

commerce guys

Entity API

- El módulo **Entity API** se crea para facilitar el acceso a las entidades y para rellenar los huecos que le faltan al core de Drupal.
- Añade elementos muy interesantes como propiedades, exportables, interfaz de administración o los *metadata wrappers*
- También proporciona un controlador de CRUD estándar para la mayoría de entidades.

Ejemplo#1: Entity Metadata Wrapper

```
function commerce_line_items_quantity($line_items, $type = '') {
  // Sum up the quantity of all matching line items.
  $quantity = 0;

  foreach ($line_items as $line_item) {
    if (!is_a($line_item, 'EntityMetadataWrapper')) {
      $line_item = entity_metadata_wrapper('commerce_line_item',
          $line_item);
    }

    if (empty($type) || $line_item->type->value() == $type) {
      $quantity += $line_item->quantity->value();
    }
  }

  return $quantity;
}
```

commerce guys

Ejemplo#2: Entity Properties

```
/**
 * Implements hook_entity_property_info().
 */
function commerce_product_entity_property_info() {
  $info = array();

  // Add meta-data about the basic commerce_product properties.
  $properties = &$info['commerce_product']['properties'];
(..)
  $properties['sku'] = array(
    'label' => t('SKU'),
    'description' => t('The human readable product SKU.'),
    'type' => 'text',
    'setter callback' => 'entity_property_verbatim_set',
    'required' => TRUE,
    'schema field' => 'sku',
  );
```

Field API

- Parte del módulo **CCK** se ha refactorizado y es ahora el Field API de **Drupal 7**.
- Los nuevos campos generados por Field API se pueden adjuntar a cualquier entidad que tenga la propiedad *fieldable*.
- Se ha facilitado en gran medida la forma de crear campos personalizados y se han añadido y estandarizado gran cantidad de nuevas opciones.

Extendiendo Field API

Field API incorpora gran cantidad de hooks, ejemplos curiosos:

- hook_field_attach_* - controlan los formularios y acciones cuando un campo se adjunta a una entidad (y operaciones CRUD).
- hook_field_storage_* - controlan la forma en la que se almacena el campo (NoSQL?)
- hook_field_extra_fields* - Permiten exponer “pseudo campos” en las entidades.

Form API

- #states
- #ajax
- #attached
- hook_form_alter() desde plantilla
- #title_display
- #limit_validation_errors

- Vertical Tabs
- machine_name
- tableselect
- managed_file
- Elementos HTML5 a través del módulo Elements

Ejemplo: #attached & Vertical Tabs

```
  $form['additional_settings'] = array(
    '#type' => 'vertical_tabs',
    '#weight' => 99,
  );
(..)
  // Add a section to update the status and leave a log message.
  $form['order_status'] = array(
    '#type' => 'fieldset',
    '#title' => t('Order status'),
    '#collapsible' => TRUE,
    '#collapsed' => FALSE,
    '#group' => 'additional_settings',
    '#attached' => array(
      'js' => array(
        drupal_get_path('module', 'commerce_order') . '/commerce_order.js',
        array(
          'type' => 'setting',
          'data' => array('status_titles' => commerce_order_status_get_title()),
        ),
      ),
    ),
    '#weight' => 20,
  );
```

commerce guys

DBTNG

El proyecto **DBTNG** ha reformado la capa de abstracción de Drupal para acercarla a un modelo OOP.

```
$result = db_query("SELECT n.nid, u.name
                    FROM {node} n
                    WHERE n.type = '%s'
                    AND n.status = %d",
          array('page', 1));
```

Drupal 6

```
$product_count = db_select('commerce_product', 'cp')
  ->fields('cp', array('product_id'))
  ->countQuery()
  ->execute()
  ->fetchField();
```

Drupal 7

commerce guys

Render arrays

Los **render arrays** son los elementos utilizados a partir de Drupal 7 para construir las páginas.

Estos elementos proporcionan una flexibilidad enorme y dan la posibilidad de alterar cómo se generan las páginas con la misma facilidad que alteramos un formulario.

Novedad: *hook_page_alter()*.

Ejemplo: render arrays

```
$note = array(
  '#title' => t('Render Array Example'),
  '#items' => $items,
  // The functions in #pre_render get to alter the actual data before it
  // gets rendered by the various theme functions.
  '#pre_render' => array('render_example_change_to_ol'),
  // The functions in #post_render get both the element and the rendered
  // data and can add to the rendered data.
  '#post_render' => array('render_example_add_hr'),
  // The #theme theme operation gets the first chance at rendering the
  // element and its children.
  '#theme' => 'item_list',
  // Then the theme operations in #theme_wrappers can wrap more around
  // what #theme left in #chilren.
  '#theme_wrappers' => array('render_example_add_div',
                             'render_example_add_notes'),
  '#weight' => -9999,
);
$page['sidebar_first']['render_array_note'] = $note;
$page['sidebar_first']['#sorted'] = FALSE;
```

Cache

- Se introduce drupal_static() que reemplaza la llamada PHP 'static'.
- Integración con los render arrays:

```
    t('cache demonstration') => array(
      '#markup' => t('The current time was %time when this was cached. Updated every
%interval seconds', array('%time' => date('r'), '%interval' => $interval)),
      '#cache' => array(
        'keys' => array('render_example', 'cache', 'demonstration'),
        'bin' => 'cache',
        'expire' => time() + $interval,
        'granularity' => DRUPAL_CACHE_PER_PAGE | DRUPAL_CACHE_PER_ROLE,
      ),
    ),
```

http://www.lullabot.com/articles/beginners-guide-caching-data-drupal-7

commerce guys

Novedades Js

Drupal 7 incorpora muchas novedades técnicas para fomentar la **flexibilidad y control del javascript** en nuestros proyectos.

- drupal_add_js() permite peso y ficheros externos.
- hook_js_alter()
- Jquery 1.4.4 y jQuery UI 1.8
- Mejoras en los behaviors.

commerce guys

Ejemplo js

```
$form['attributes'][$field_name] = array(
    '#type' => 'select',
    '#title' => check_plain($data['instance']['label']),
    '#options' => array_intersect_key($data['options'],
          drupal_map_assoc($used_options[$field_name])),
    '#default_value' => $default_product_wrapper->{$field_name}
          ->value(),
    '#weight' => $data['instance']['widget']['weight'],
    '#ajax' => array(
       'callback' =>
          'commerce_cart_add_to_cart_form_attributes_refresh',
              ),
);
```

commerce guys

File API

El nuevo interfaz de ficheros de Drupal 7 permite acceder a cualquier recurso como si fuera un fichero.

- Introducción de **streams**: public:// private:// temporary://
- file_unmanaged_* (copy, move, delete) permite tratar ficheros sin grabarlos en base de datos.

Code registry

En Drupal 7 se introduce el **registro de código**, para inventariar los ficheros y clases que se deben cargar en cada momento.

El único fichero que se carga automáticamente es el .module, el resto de ficheros, plugins, includes, tests, etc... deben declararse en el array files[] del fichero .info.

info files

- Se pueden añadir css y js usando las propiedades *stylesheets* y *scripts*.
- La propiedad *files* fuerza el auto load de los ficheros declarados.
- *dependencies* soporta versiones.
- La propiedad *configure* expone el path de configuración del módulo
- *required* fuerza que un módulo o theme sea obligatorio y no pueda ser deshabilitado

http://drupal.org/node/542202

Queue API

Drupal 7 incorpora una nueva API para gestionar colas basada en objetos.

Varios componentes del core como Aggregator, Batch API o Cron ya la implementan de base.

Permite guardar los elementos en memoria o en base de datos, y es totalmente configurable según el caso de uso que necesitemos.

http://www.slideshare.net/philipnorton42/drupal-7-queues

commerce guys

Ejemplo queue

```
$queue = DrupalQueue::get('tofu_sandwich');
$queue->createQueue(); // no-op.
$things = array('bread', 'tofu', 'provolone', 'sprouts');
foreach ($things as $item) {
  $queue->createItem($item);
}
```

```
$items = array();
while ($item = $queue->claimItem()) {
  $message .= $item->item_id . ':' . $item->data    . '; ';
  $items[] = $item;
}
drupal_set_message('Queue contains: ' . check_plain($message));
foreach ($items as $item) {
  $queue->releaseItem($item);
}
```

http://www.ent.iastate.edu/it/Batch_and_Queue.pdf

commerce guys

Cambios en Schema

A partir de **Drupal 7**, declarar *hook_schema* en el fichero *.install* es suficiente para instalar y desinstalar tablas en la base de datos, no hay que hacerlo explícitamente con *hook_install* y *hook_uninstall*.

hook_field_schema() se encarga de los esquemas de datos para campos.

http://api.drupal.org/api/drupal/modules--field--field.api.php/function/hook_field_schema/7

commerce guys

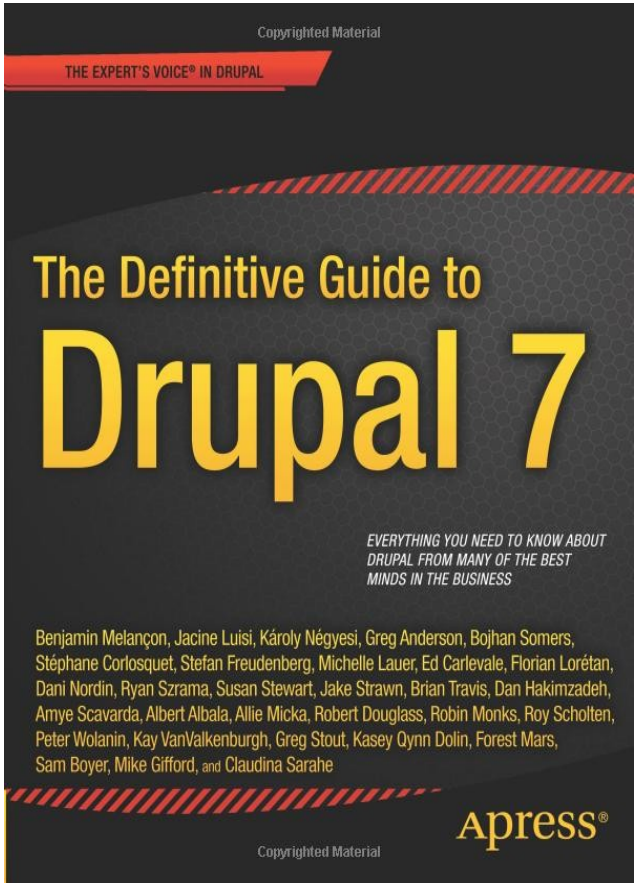
Simpletest en core

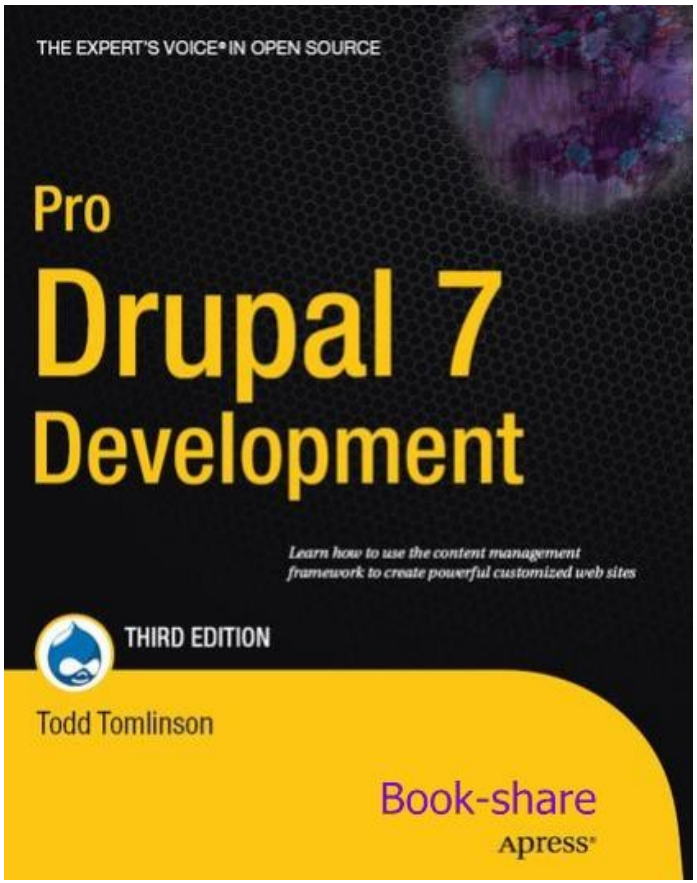
- Drupal 7 incorpora el módulo Simpletest en su núcleo y tiene **unit testing** e **integration testing** en todos los componentes por defecto.
- Tener los elementos verificados proporciona **seguridad** y **calidad** en los componentes.

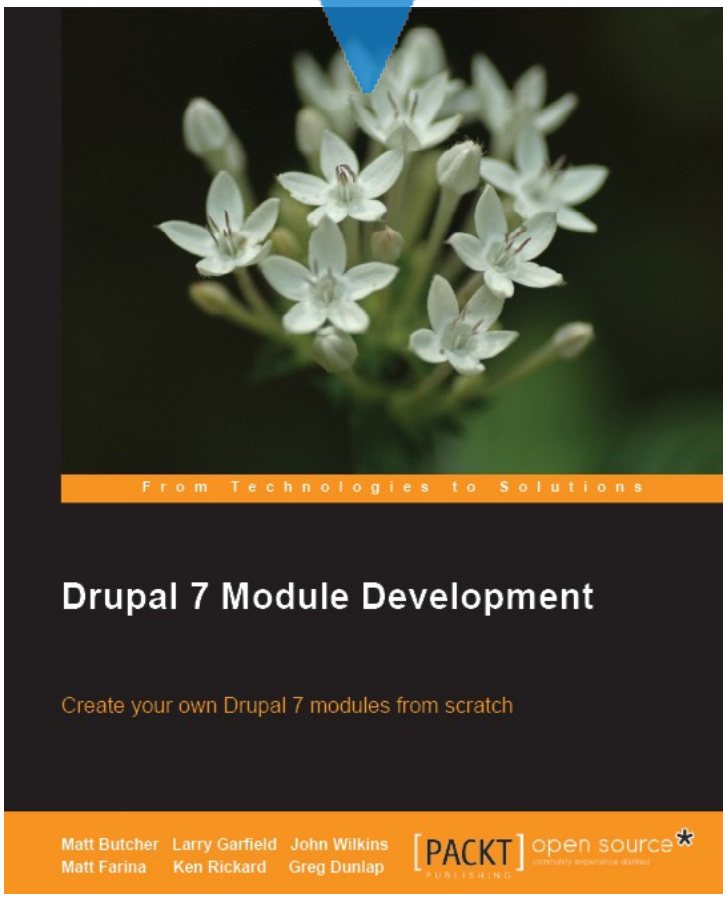
  http://drupal.org/simpletest-tutorial-drupal7

Libros

¡Recomendado!







commerce guys

¿Preguntas?

- @pcambra
- cambrico.net
- Perfil en Drupal.org

commerce guys

We're hiring!

commerce guys

¡Muchas gracias!

commerce guys