

# Panels : construire autrement

Pascal Morin

bellesmanieres @ d.o.

Senior Dev @ Code Enigma



DRUPALCAMP  
PARIS 2013



Acquia®

adyax  
Open Source Experts

commerce  
guys

OPEN WEB SOLUTIONS  
OWS

Windows Azure

1. Définition, utilisation, concepts de base >site

builders

2. Panels et le theme >themers & frontend devs

3. Développement et concepts avancés >devs

1. Définition, utilisation, concepts de base

# Qu'est-ce que c'est ?

La page du projet :

<https://drupal.org/project/panels>

# Qu'est-ce que c'est ?

- At its core it is a drag and drop content manager that lets you visually design a layout and place content within that layout.
- **À la base, c'est un gestionnaire de contenu drag and drop qui vous permet de concevoir visuellement une mise en page et d'y placer des éléments de contenu.**

## Qu'est-ce que c'est ?

- At its core it is a drag and drop content manager that lets you visually design a layout and place content within that layout.
- **À la base, c'est un gestionnaire de contenu drag and drop qui vous permet de concevoir visuellement une mise en page et d'y place des éléments de contenu.**

# Qu'est-ce que c'est ?

Il faut le voir comme une manière différente de faire interagir ensemble les éléments et concepts habituels de Drupal:

- Système de menu
- Entités (noeuds, utilisateurs, etc)
- Système de thème
- ...

# Pour quoi ?

Pour aller au delà de ce que Drupal permet de faire, de manière simple et “propre”.

La preuve par l'exemple ...

# Article

Titre

Image principale

Corps du texte

Bloc téléchargement

Les 5 derniers noeuds de la même catégorie

# Drupal

- hook\_node\_preprocess
- node--article.php
- page.tpl.php
- hook\_page\_preprocess
- hook\_block\_info
- module\_invoke('block', 'block\_view', XXX);
- arg(1)
- node\_load/entity\_load
- field\_view\_field
- global \$user
- user\_is\_logged\_in()
- If/else/switch

# Page

- C'est l'élément de base, qui correspond à un chemin :  
node/%node, mapage, autre/page.
- Ce chemin peut contenir des arguments, et sert de « support »  
aux variantes et aux autres éléments.
- Créer une 'page', c'est donc l'équivalent d'implémenter un  
hook\_menu, dont le callback, les arguments, les droits d'accès,  
etc sont gérés pour nous par 'Panels'\*

\* En fait, c'est pas vrai !

# Contexte

- L'ensemble des informations disponibles pour tous les éléments de la page.
- Un contexte est souvent une entité ou un objet « chargé » (nœud, utilisateur, terme de taxonomie), mais pas forcément : langue pour un site multilingue, chaîne de texte, ...
- Un contexte peut découler d'un argument, du contexte global, d'une relation.

# Variante

- C'est la partie responsable de générer le contenu, à partir des éléments préparés par la 'page'
- Une page peut contenir plusieurs variantes, dont l'une ou l'autre sera choisie en fonction de critères prédéfinis, souvent liés au(x) contextes.
- C'est ce qui permet de moduler la mise en page et les éléments de contenu pour une même page.

# Layout

- C'est un modèle de mise en page qui représente la disposition dans laquelle les différents éléments seront placés (colonnes, lignes).
- C'est l'équivalent d'un template de page dans un thème drupal, avec des **régions** (gauche, droite, bas, etc)
- Panels fourni un layout 'flexible', qui permet de créer par l'interface quasiment n'importe quel layout, et de le réutiliser.

# Pane (panneau)

- C'est l'élément de contenu lui même, semblable au blocs de Drupal
- Un pane peut contenir n'importe quel type d'élément : un bloc, un menu, un élément d'entité, une vue, ...

## 2. Panels et le theme

# En guise d'introduction

- L'interface d'administration permet d'ajouter des classes et des ids aux différents éléments (body, régions, panes).
- Suffisant dans certains cas simples.

# Concevoir sa page

- Panels conserve le système de theming standard drupal, mais les éléments de bases sont différents.
- Par défaut, Panels agit au niveau du « contenu » de la page : `$page['content']` .
- Le niveau de « Panelization » est donc principalement une question de choix, de besoins et de familiarité avec le module.

# Choix du thème

- Certains thèmes sont plus adaptés que d'autres à l'utilisation avec Panels. En vrac :  
Tao, Zen, AT Panels Everywhere, Aurora, Panels 960gs, Precision, Clarity, Omega, ...
- Le contenu d'un page.tpl.php idéal :  
`<?php print render($page['content']); ?>`

# Templates

- Par défaut, il y a peu de templates qu'on peut directement override dans le theme. En gros un seul : `panels-pane.tpl.php`
- Les suggestions de templates sont basées sur le type de pane : (voir <https://drupal.org/node/1678810>)
  - `panels-pane--node-title`
  - `panels-pane--views-panes`
- Il est bien sûr possible d'implémenter d'autres templates (`HOOK_preprocess_panels_pane`) pour des besoins précis.

# Layouts

- Panels fournit quelques layouts les plus courants : 2 et 3 colonnes, colonnes + haut et bas, etc
- Il existe aussi un layout flexible, qui permet de construire et de réutiliser presque tout type de layout.
- [https://drupal.org/project/panels\\_extra\\_layouts](https://drupal.org/project/panels_extra_layouts)
- Dans la plupart des cas, on a besoin de plus de contrôle sur le markup généré. On doit donc créer son propre layout.

# Layouts

- Un layout est constitué de :
  - Un fichier de définition des caractéristiques de notre layout (nom, catégorie, régions disponibles, etc)
  - Un fichier tpl.php
  - (Un fichier CSS)
  - (Un fichier PNG)
- C'est un peu comme un page.tpl.php, ou un mini-theme qui ne s'appliquerait qu'au contenu que l'on choisi.

# Fichier paris2013.inc

```
$plugin = array(  
    'title' => t('2 colonnes'),  
    'category' => t('Paris 2013'),  
    'icon' => 'paris2013.png',  
    'theme' => 'paris2013', // Les underscores doivent être remplacés par des tirets.  
    'css' => 'paris2013.css',  
    'regions' => array(  
        'section1' => t('Section 1'),  
        'section2' => t('Section 2'),  
    ),  
);
```

# Fichier paris2013.tpl.php

```
<div class="panel-display panel-2col clearfix" <?php if (!empty($css_id)) { print "id=\""$css_id\""; } ?  
>>  
  <div class="panel-panel panel-col-first">  
    <div class="inside"><?php print $content['section1']; ?></div>  
  </div>  
  
  <div class="panel-panel panel-col-last">  
    <div class="inside"><?php print $content['section2']; ?></div>  
  </div>  
</div>
```

# Styles

- Un style correspond à un ensemble de mise en forme.
- Un style est réutilisable et s'applique au niveau de la région, du pane ou de la variante.
- Un sous-module de Panels (stylizer) permet de créer et réutiliser des styles par l'interface.

# Styles

- Exemple tiré de : [https://drupal.org/project/panels\\_tabs](https://drupal.org/project/panels_tabs)

```
$plugin = array(  
  'title' => t('Tabs'),  
  'description' => t('Show panel panes in a region as tabs.'),  
  'render region' => 'panels_tabs_style_render_region',  
);
```

- Pour résumer, la démarche est à l'opposée du theming Drupal habituel :  
au lieu de subir du contenu formaté et de chercher comment supplanter le markup, on définit le markup séparément (layout, style) et on choisit à quel élément l'appliquer.
- C'est toujours la même démarche de séparation de la logique et de la génération de contenu.

# Pour aller plus loin

- [https://drupal.org/project/semantic\\_panels](https://drupal.org/project/semantic_panels)
- <https://drupal.org/project/ds>
- [https://drupal.org/project/panels\\_everywhere](https://drupal.org/project/panels_everywhere)
  
- <http://www.codeenigma.com/en/blog/drupal-theming-death-of-a-themer>

### 3. Développement et concepts avancés

# Rendre à César ...

- Panels lui-même ne fournit en fait que certaines des fonctionnalités que l'on a vu jusqu'à présent.
- C'est en fait la combinaison de 3 modules :
  - CTools
  - Page Manager
  - Panels

# Ctools

- C'est le module sur lequel tout repose. Il fournit les composants de base :
  - Contextes et arguments
  - Droits d'accès
  - Cache\*
  - Relations
  - « Content type »

# Page Manager

- Il joue un rôle similaire au menu router de Drupal
- Implémente/supplante les entrées de menu
- Détermine quels handlers doivent être appelés
- Fournit aussi l'interface d'administration des pages

# Panels

- Il étend (beaucoup!) les fonctionnalités du page manager et de Ctools et sert de glue entre les deux
- Définit des 'renderers' (ce qui effectivement génère le HTML)
- Ajoute de nouveaux types de plugins Ctools : style, layout, cache
- ...
- Et bien sûr, fournit l'interface d'administration qui permet de créer tout notre contenu

# Plugins

- Taches
- Contexte
- Cache
- Access
- Argument
- Content type
- Relationship
- (Layout)
- (Style)
- ...

# Plugins

- `HOOK_ctools_plugin_directory()`
- `HOOK_ctools_plugin_api()`

# plugin\_name.inc

```
$plugin = array(  
    'title' => t('Mon plugin'),  
    'category' => t('Ma catégorie'),  
    'callback' => 'ma_function_de_callback',  
);
```

```
MODULENAME_PLUGINNAME_PLUGINTYPE_CALLBACKNAME()
```

Task handler : json\_output.inc

# Content type : paris\_2013.inc

```
$plugin = array(
  'single' => TRUE,
  'title' => t('Infos utilisateurs'),
  'icon' => 'icon_test.png',
  'category' => t('Paris 2013'),
  'defaults' => array(),
  'render callback' => 'panelstalk_paris_2013_content_type_render',
  'required context' => array(
    new ctools_context_optional(t('User ID'), 'user') //@see argument uid.inc @see node.inc
  ),
);

function panelstalk_paris_2013_content_type_render($subtype, $conf, $panel_args, $context)
function panelstalk_paris_2013_content_type_admin_title($subtype, $conf)
function panelstalk_paris_2013_content_type_edit_form($form, &$form_state)
function panelstalk_paris_2013_content_type_edit_form_submit($form, &$form_state)
```

# Accès

- Garder à l'esprit que les contextes ne sont pas encore disponibles
- Variante + 403
- Un exemple de plugin d'accès :  
[https://drupal.org/project/domain\\_ctools](https://drupal.org/project/domain_ctools)

# Contexte

- Ctools fournit déjà un contexte pour les entités, même custom.

# Renderer

- Permet de contrôler entièrement le rendu d'un display de Panels.
- C'est, par exemple, la méthode utilisée par les modules IPE et ESI.

# En vrac

- [https://drupal.org/project/pm\\_existing\\_pages](https://drupal.org/project/pm_existing_pages)
- [http://drupal.org/project/fieldable\\_panels\\_panes](http://drupal.org/project/fieldable_panels_panes)
- <https://drupal.org/project/panelizer>
- <https://drupal.org/project/panopoly>