

Ezpublish

Création d'un design

- Transfert de compétence
17,18,19 octobre 2011

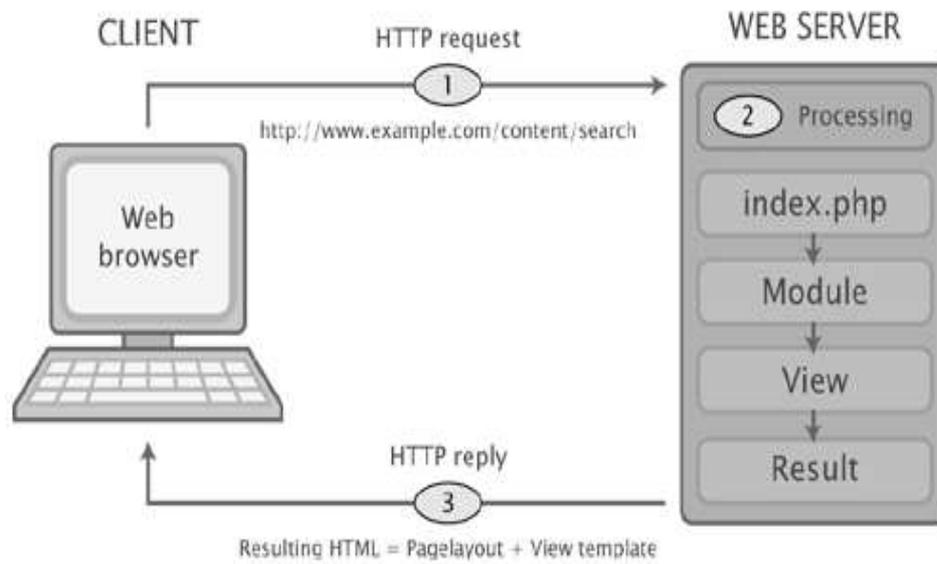
Rappel des objectifs

- Pouvoir créer un nouveau design pour un site ezpublish.
- Comprendre la structure d'une extension de design.
- Comprendre la combinaison des designs et l'ordre de prise en compte des templates.
- Avoir les bases du langage de template.

Terminologie

- Module
- Vue
- Template

Modules, vues et templates.



Modules

- Un module fournit une interface HTTP qui permet des interactions avec eZ Publish.
- Les modules servent d'interface avec les fonctionnalités du noyau; certains modules sont plus ou moins indépendants du noyau.
- Le système est livré avec des modules standard (Content, User, Role, etc..).
- Il est possible de créer des modules sur mesure
- Un module apporte des vues (views), des fonctions de fetch et des fonctions d'accès.

Vues

- Une vue est une interface du module et permet l'accès à ses fonctions
- Le nom de la vue apparaît dans l'URL après le nom du module, séparés par un slash
 - content/edit
 - user/login
- Appelée par le module, la vue exécute du code (php) et génère le résultat du module (via template)

Méthodologie de développement.

- Ne pas modifier le noyau de ezpublish.
- Ne pas modifier les extensions fournies avec ezpublish.
- Ne pas modifier les extensions tiers.
- Créer sa propre extension qui comprendra toutes les spécificités.
- Désactiver les caches.
- Ne pas réinventer ce qui existe déjà.
- Utiliser le mode debug.

Créer son extension de design

- Dans le répertoire "extension" (sans 's')
- Créer dans l'extension, les répertoires suivants :
 - design qui va contenir votre nouveau design
 - settings qui va contenir les paramètres
 - translations qui va contenir les traductions
- Créer dans le répertoire "design" le répertoire correspondant au nouveau design
- Déclarer l'extension dans le fichier `site.ini.append.php`

Design.

- Qu'est-ce qu'un design ?
- Structure d'un répertoire de design.
- Les designs préintégréés dans ezpublish
- La combinaison de designs et leur ordre d'exécution.

Qu'est-ce qu'un design ?

- Le design est la manière dont le contenu est balisé, stylisé et présenté
- eZ Publish est capable de gérer virtuellement un nombre illimité de designs
- Un design eZ Publish contient:
des templates, des css, des images non liées au contenu, des flashs, des fichiers de police...
- La présentation d'une page correspond généralement à la combinaison de fichiers en provenance de plusieurs designs

Design : emplacement et structure

- Les fichiers de design sont stockés dans le répertoire *design/*:
 - à la racine d'eZ Publish (pour les designs natifs)
 - dans une extension (pour les designs complémentaires)
- La structure d'un design
 - images
 - javascript
 - override
 - stylesheets
 - templates

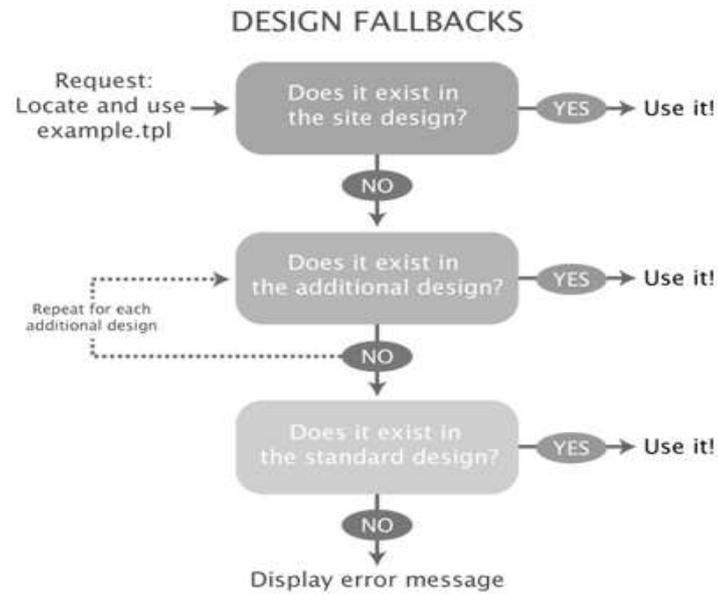
Les designs de base de ezpublish

- Toute installation d'eZ Publish contient des designs pré-intégrés
 - admin (Interface d'administration)
 - admin2 (Interface d'administration)
 - base
 - standard (Templates par défaut)
- Il ne faut jamais modifier ou supprimer ces designs

La combinaison des designs.

- La configuration d'un siteaccess précise le ou les designs utilisés
[*DesignSettings*] de *site.ini*
- Un design principal (obligatoire)
- Un ou plusieurs design(s) additionnels
- Un design par défaut (standard)

Le mécanisme de fallback.



Les fichiers de configuration.

- Emplacements.
- Structure.
- Ordre de priorité.

Hiérarchie de la surcharge de configuration

1. Configuration par défaut
settings/*.ini (à ne jamais modifier)
2. Configuration de l'Extension avec Siteaccess
extension/<XXX>/settings/siteaccess/<NAME>/*.ini.append.php
3. Configuration du Siteaccess
settings/siteaccess/<NAME>/*.ini.append.php
4. Configuration de l'Extension
extension/<XXX>/settings/*.ini.append.php
5. Surcharge globale
settings/override/*.ini.append.php

Les templates.

- Définition
- Contenu
- Composition d'une page
- Le template principal
- La surcharge de template
- Les balises personnalisées
- Le langage de template

Template - Définition.

- Séparation du contenu et de la forme
- Unité fondamentale d'un design détermine comment un contenu spécifique doit être représenté
- Fichier HTML "étendu"
Avec du code spécifique eZ Publish (mais pas de code PHP)
- Intégré au processus d'exécution module-view

La composition d'une page

- Une page est composée de plusieurs templates.
- eZ Publish utilise au minimum le template principal (pagelayout)

Le template principal (pagelayout.tpl).

- C'est le template principal
- Détermine la représentation globale du site
- Contient les balises HTML, HEAD and BODY
- Un site utilise à la fois des templates standard et des templates personnalisées

Les templates de vue par défaut.

- Une distribution d'eZ Publish contient les templates par défaut pour toutes les vues
- Les templates par défaut sont disponibles dans le design standard

Les templates de vue personnalisés.

- Créer un design qui intègre des templates pour tous les scénarii possible serait un travail trop important
- Le mécanisme de fallback rend possible d'utiliser une combinaison de templates personnalisées et standard
- Les templates personnalisées sont généralement des copies modifiées des templates standards.

Le système de surcharge de template.

- Permet d'utiliser des templates personnalisées basées sur de multiples conditions
- Permet de créer des templates alternatives pour toutes les templates utilisées par eZ Publish
- Est particulièrement utile pour afficher différents types de noeuds de manière distincte
- Une surcharge de template de vue est activée par un ensemble de conditions. Si les conditions correspondent, la template alternative est utilisée

Où définir les surcharges ?

- Fichier INI spécifiant les surcharges (override.ini.append.php)
 - de manière globale, ou
 - pour un siteaccess, ou
 - pour une extension
- Constitué de blocs de conditions
- Ensemble de règles nommées précisant à eZ Publish la template alternative à utiliser pour un cas spécifique

Description d'un bloc de surcharge

- Un nom unique pour le bloc de surcharge
- La template devant être surcharger
- La template qui doit être utilisée à la place
- Le nom du dossier contenant la template de surcharge
- Un ensemble de conditions/règles contrôlant le cas d'utilisation de la surcharge

Les balises personnalisées

- Templates utilisées par l'éditeur ezoe.
- Situées dans un répertoire
design/xxx/templates/content/datatype/view/ezx
mltags/
- Définies dans le fichier content.ini

Contenu d'un fichier TPL.

- XHTML 1.0
- Code spécifique eZ Publish
permet d'extraire de l'information depuis eZ
permet d'effectuer des tâches classiques de
programmation (conditions, boucles, etc.)
- Doit être entre accolades

Le langage de templates (TPL)

- Langage riche comportant les instructions de base de tous les langages de programmation.
- Les catégories d'instruction
 - Opérateurs
 - Fonctions
 - Structures de contrôle
 - Fonctions fetch
- Peut être intégré dans une balise html
- Attention aux instructions "deprecated".

Types de variable.

- Nombres
- Chaines
- Booléens
- Tableaux
- Objets

Utilisation des variables.

- Notation \$var
- Définition def \$var=val
- Valorisation set \$var=val
- Affichage d'une variable {\$var}
- Inspection d'un tableau ou d'un objet
\$var|attribute(show)

Instructions courantes - 1

- Commentaires
`{* code *}`
- Inclure une autre template :
`{include uri='design:footer.tpl'}`
- Concatenation :
`{def $my_string='sausage'}
{concat('Liver ', $my_string, ' sandwich')}`
- Protéger un e-mail :
`{def $email_address='username@example.com'}
{ $email_address|wash('email')}`

Instructions courantes - 2

- Nettoyage de l'affichage (HTML) :
{def \$bogus_string='hello < world'}
{\$bogus_string|wash()}
- Protéger du code JavaScript
{literal}
<script language="Javascript" type="text/javascript">
<!--
var divA = document.getElementById('divA');
divA.InnerHTML +=
{/literal}
{\$bogus_string| wash('javascript')}
{literal}
-->
</script>
{/literal}

Instructions courantes - 3

- URL eZ Publish :
`Login`
- Lien vers un noeud :
`Link to a node`
- Images :
`<img src={'women.jpg'|ezimage()}
alt="This is my image." ... />`
- Design (css, javascript, flash...) :
`<style type="text/css">
@import url({'stylesheets/my_stuff.css'|ezdesign()});
</style>`

Instructions courantes - 4

- Récupérer un noeud :

```
{def $my_node=fetch( content, node, hash( node_id,13 ) )}  
... {undef}
```
- Récupérer une liste de noeuds :

```
{def $my_node=fetch( content, list,hash( parent_node_id,  
13 ) )}... {undef}
```
- Récupérer une liste de noeuds triés :

```
{def  
$my_node=fetch( content,List,hash( parent_node_id,13,Li  
mit, 10,class_filter_type,include,class_filter_array,  
array( 'article' ) ) )}...{undef}
```

Variables du pagelayout.

- \$access_type [array]
- \$current_user [object]
- \$module_result [array]
 - \$module_result.uri (string)
 - \$module_result.content (string)
- http://ez.no/doc/ez_publish/technical_manual/4.x/te