Framework Joomla! 1.5

Joomla! Day.fr

Joomla! Development

Développeurs

« Framework J!1.5 »

par Garstud

Afuj! Joomla!fr le portail francophone JoomGroupe

Paris, le 24 Mai 2009

Objectifs

- Présentation orientée développeurs PHP

1. Vulgariser la philosophie du framework Joomla
2. Appréhender les technologies de ce framework
3. Fournir les bases pour initier un développement J!1.5

Sommaire

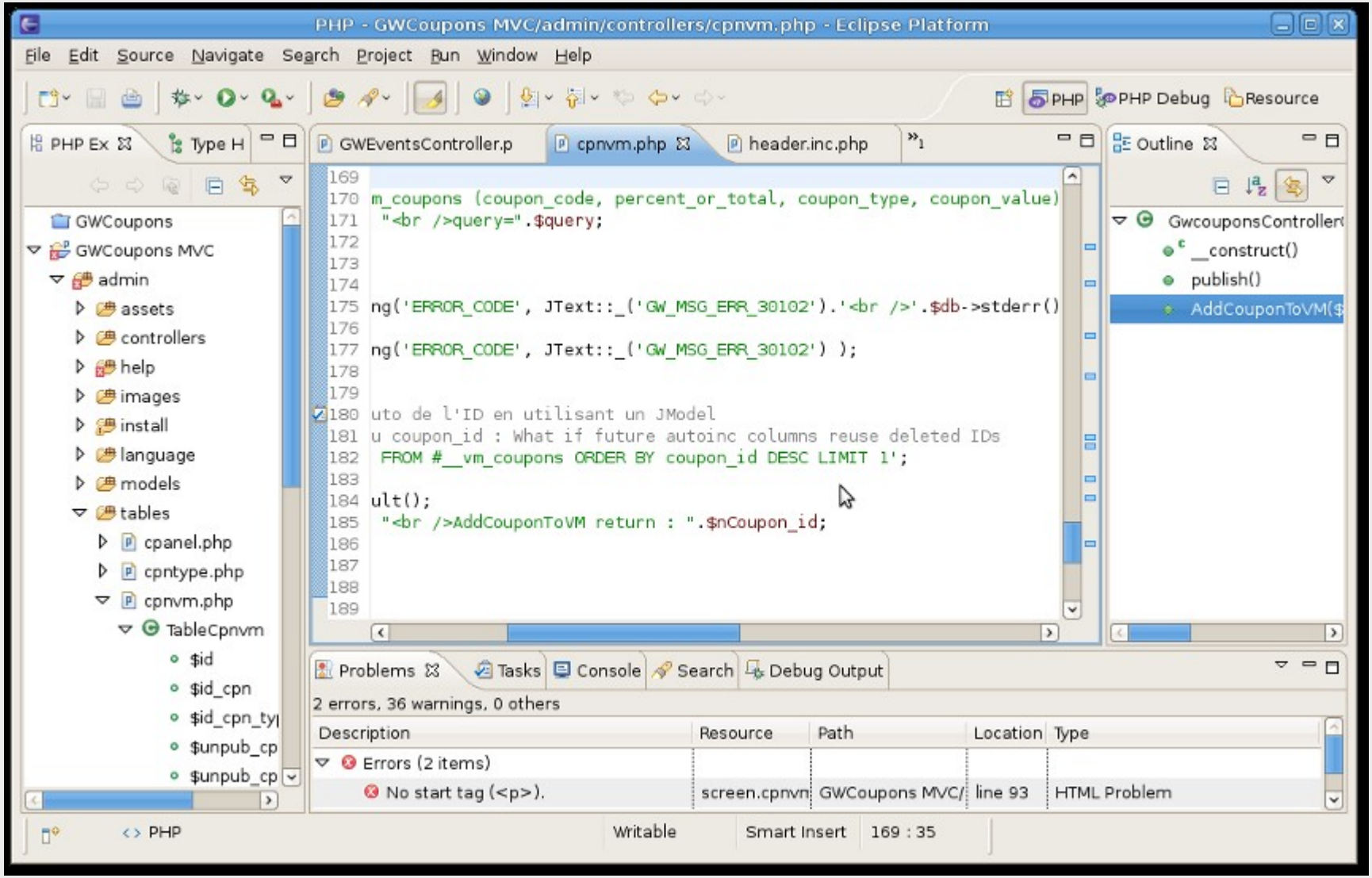
Historique et technologies

- Adaptation de Mambo
- « Framework » Joomla!1.0
  - API d'ergonomie (HTMLTools)
  - Objets de persistance (MosDbTable)
  - Plugin (Mambots)
  - Mode « Legacy »
  - …
- Technologies inclusent dans le framework 1.5
  - Standard web (CSS, XHTML , javascript) Joomla 1.0
  - API AJAX, Mootools, LDAP, Webservices ...

IDE & Outils

- **Eclipse** (Ganymède 3.4)
  - Colour-coding, auto-indentation, bookmark ...
  - Compilation syntaxique à la volée
  - Library Joomla pour l'auto-completion par introspection
  - Navigation hypertexte entre les fonctions et classes
- Un Eclipse « prémodé » PHP
  - Eclipse PDT (PHP Development Tools)
    http://www.eclipse.org/pdt
  - Sortie du plugin PDT en v2.1 (juin 2009)
- Une communauté, … des plugins !
  - Aptana, XDebug, Subversion(SVN), SQLExplorer, Ant ...

IDE & Outils

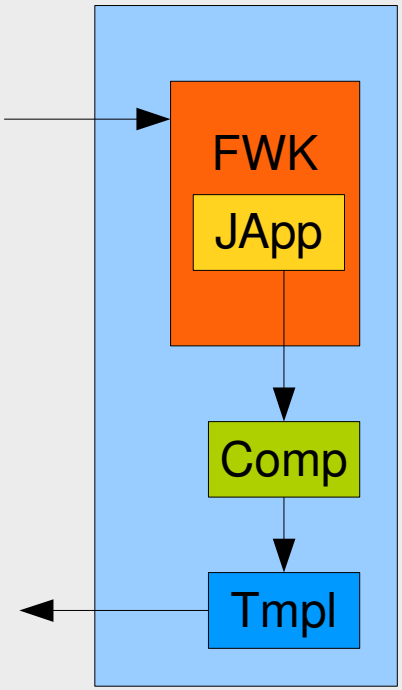

PHP - GWCoupons MVC/admin/controllers/cpnvm.php - Eclipse Platform
File Edit Source Navigate Search Project Run Window Help
PHP
PHP Debug
Resource
PHP Ex
Type H
GWEventsController.p
cpnvm.php
header.inc.php
Outline
GWCoupons
GWCoupons MVC
admin
assets
controllers
help
images
install
language
models
tables
cpanel.php
cpntype.php
cpnvm.php
TableCpnvm
$id
$id_cpn
$id_cpn_ty
$unpub_cp
$unpub_cp
GwcouponsController
__construct()
publish()
AddCouponToVM($
m_coupons (coupon_code, percent_or_total, coupon_type, coupon_value)
"
query=".$query;
ng('ERROR_CODE', JText::_('GW_MSG_ERR_30102').'
'.$db->stderr()
ng('ERROR_CODE', JText::_('GW_MSG_ERR_30102'));
uto de l'ID en utilisant un JModel
u coupon_id : What if future autoinc columns reuse deleted IDs
FROM #__vm_coupons ORDER BY coupon_id DESC LIMIT 1';
ult();
"
AddCouponToVM return : ".$nCoupon_id;
Problems
Tasks
Console
Search
Debug Output
2 errors, 36 warnings, 0 others
Description
Resource
Path
Location
Type
Errors (2 items)
No start tag (<p>).
screen.cpnvn
GWCoupons MVC/
line 93
HTML Problem
PHP
Writable
Smart Insert
169 : 35

IDE & Outils

- J!Code
  - Projet JoomlaCode basé sur Eclipse
  - Spécifiquement pour développer des extensions Joomla!
- J!Dump
  - Un Dumper, … pas aussi pratique qu'un debuggeur
  - Composant/plugin Joomla :
    http://joomlacode.org/gf/project/jdump
  - Afficheur de contenu de variable après ajout de code PHP

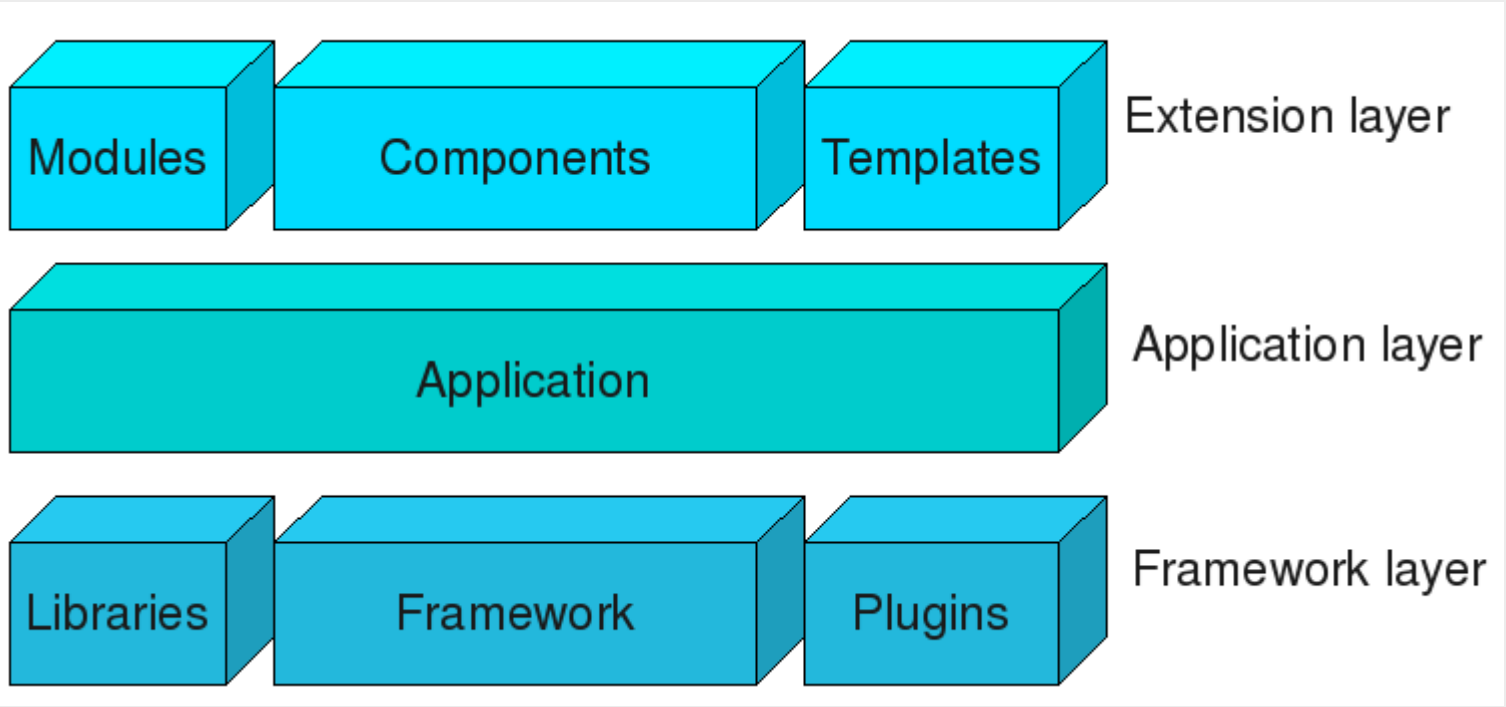
Concepts Joomla

- La requete HTTP Joomla
  - http://www.monsite.fr/index.php?option=xxx&id=xxx
  - index.php comme porte principale
    - sécurité et homogénéité
  - Cinématique Joomla
    1. Le serveur web reçoit une requête HTTP
    2. Le noyau de Joomla est chargé (framework et classes)
    3. instanciation de l'objet **JApplication**
    4. Initialisation de l'objet **JApplication**
    5. calcul le chemin **URI** d'appel
    6. **exécution** de l'appel de l'URI
    7. interprète le template et les documents à charger

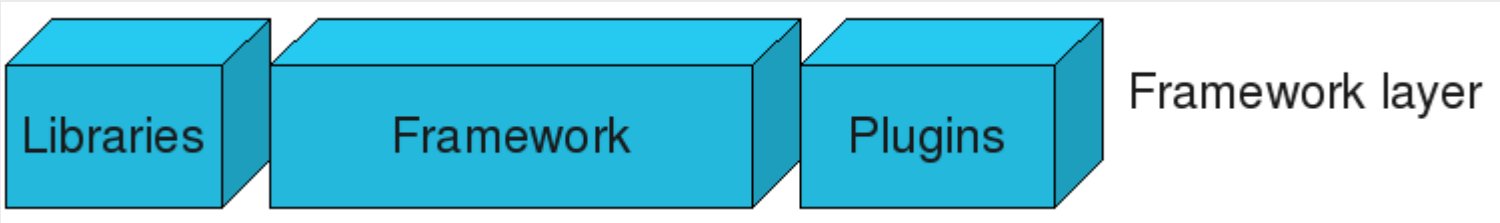


Architecture du Framework

- Les 3 couches du framework



Architecture du Framework



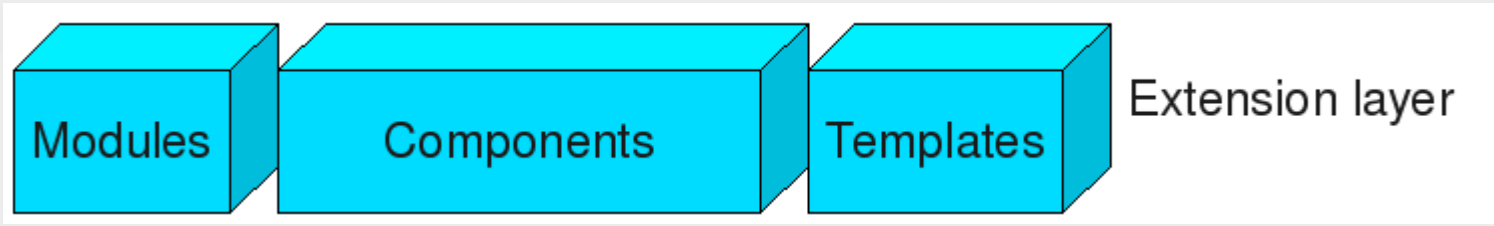
- Brique Framework
  - JFactory : Objet Fabrique donnant accès à de nombreuses infos du système
  - … JRoute, JText, JMenu, JView, JObject, JDocument
- Brique Libraries
  - Archive, PDF, PatError, PEAR, GACL, Mailer, UTF8 …
  - *jimport (Joomla.filesystem.*);*
- Brique Plugins
  - « **étendre** » les fonctionnalités du framework

Architecture du Framework

- Couche Application
  - Application qui implémente le framework :
    - JInstallation, JAdministrator, JSite, …
    - Connecté au Framework via JApplication

```
$app  =& JFactory::getApplication();
if ($app->isSite())  echo 'Client is site';
if ($app->isAdmin()) echo 'Client is administrator';
```

Architecture du Framework



- Couche Extension
  - Toutes les extensions « *pluggable* »
  - Modules : extensions légères utilisées essentiellement « en mode boite »
  - Composants : extensions complexes, pouvant gérer à la fois le front et le backend d'un site
    - Implémentation MVC
    - Persistance des données
  - Templates : extensions de type design
  - Langages : la plus basique des extensions. Ils correspondent au pilotage de fichier de type « clé/valeur » (idem fichier .ini)

Objets contextuels

- JUser
  - Chaque requête Joomla! est associée à un utilisateur

```
$user =& Jfactory::getUser();
$language  = $user->getParam('language', 'french');
echo $user->name.", votre langue est {$language}";
```

  - Statut de connexion

```
$user  =& JFactory::getUser();
if  ($user->guest) {
        echo "<p>Vous devez vous connecter pour voir ...</p>";
} else {
        echo "<p>bienvenue ".$user->username;
}
```

Objets contextuels

- JRequest
  - Accéder aux paramètres HTTP

```
$reqParam1 =& Jrequest::getVar('param1', 'défaut', 'post');
```

    - Param 2 et 3 facultatif (valeur par défaut et méthode)
    - Param 4 pour « caster » le type
    - Param 5 pour filtrer les données (Raw, HTML ...)

Objets contextuels

- JSession
  - Cet objet remplace l'appel de la variable PHP $_SESSION.

```php
$userSession  =& JFactory::getSession();
$value = $userSession->get('maValeur', 1);
```

  - Modification des données de session

```php
$userSession->set('maValeur', 6);
```

Manifest d'installation

- « *Descripteur de déploiement* » ...XML
- Commun à toutes les extensions
  - Entête déclarative
  - Fichiers a installer
  - Paramètres (module et plugin uniquement)
- Paramètres simples … ou évolués :
  - Text, textarea, radio ...
  - Filelist, calendar, section, category, usergroup, sql ...

Manifest d'installation

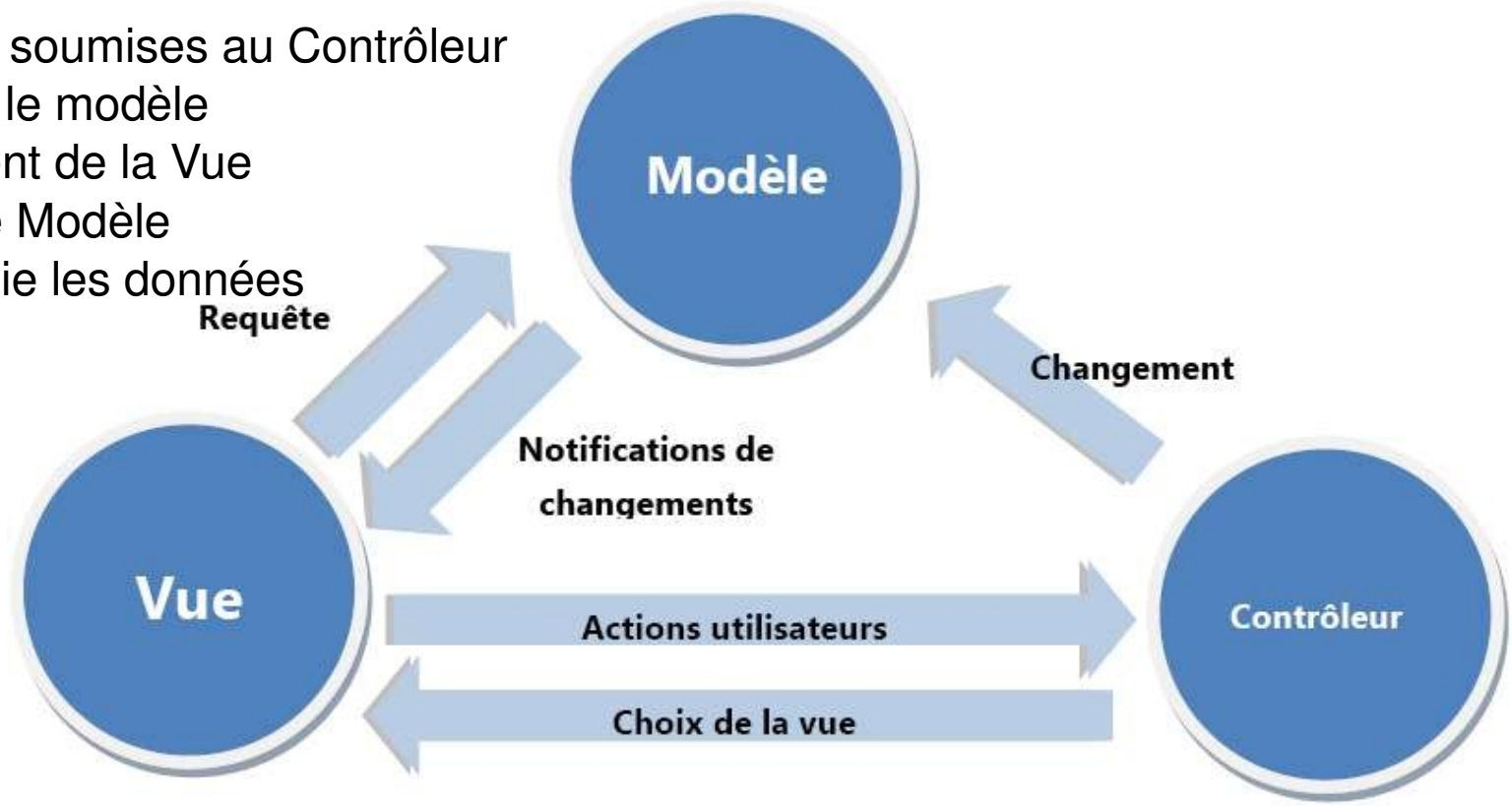
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE install SYSTEM "http://dev.joomla.org/xml/1.5/module-install.dtd">
<install type="module" version="1.5.0" client="site">
	<name>Hello World - Hello</name>
	<author>Marc STUDER</author>
	<creationDate>2008-09-23</creationDate>
	<copyright>All rights reserved by garstud workshop 2009.</copyright>
	<license>GPL 2.0</license>
	<authorEmail>mail@domain.tdl</authorEmail>
	<authorUrl>www.garstud.com</authorUrl>
	<version>1.0.0</version>
	<description>Provides a basic "Hello World" notice</description>
	<files>
		<filename module="mod_helloworld">mod_helloworld.php</filename>
		<filename>index.html</filename>
	</files>

	<params />
</install>
```

Design Pattern

- MVC Model-View-Controller … pour les composants

1.Actions utilisateurs soumises au Contrôleur
2.Changement dans le modèle
3.Choix et chargement de la Vue
4.La Vue interroge le Modèle
5.Le modèle lui envoie les données



Accès aux Données

- Exécution d'une requête SQL

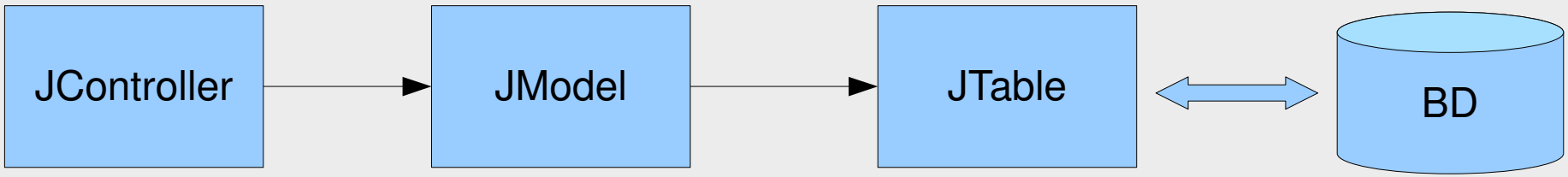
```php
$db =& JFactory::getDBO();
$db->setQuery("SELECT name FROM #__users");
if(!$result = $db->query()) {
    // erreur
}
```

- Différents mode de chargement des résultats
  - $db->LoadResult() : resultat simple (1 enreg, 1 seul champ)
  - $db->loadAssoc (1 enreg) et $db->loadAssocList (tableau)
  - … loadObject, LoadRow …

- Attention Sécurité : injection SQL
  - n'oubliez pas $db->nameQuote() et $db->Quote()

Le CRUD ?

- CRUD = Create Read Update Delete
- Mécanisme de gestion de la persistance
  - Plus de requêtes SQL de type INSERT, UPDATE, DELETE
    - Seul les SELECT sont a implémenter
- Utilisation du MVC pour manipuler les données
  - JController : lance les actions (edit, remove, save, cancel …)
  - JModel : prépare et exécute la gestion des données
  - JTable : (mapping et gestion de la persistance)
    - Reçoit la demande de manipulation des données
    - La répercute sur la base de données



Le CRUD ?

```php
function save() {
    $model = $this->getModel('user');

    if ($model->store($post)) {
        $msg = JText::_( 'User enregistré !' );
    } else {
        $msg = JText::_( 'Erreur' );
    }
    $link = 'index.php?option=com_user';
    $this->setRedirect($link, $msg);
}
```

```php
function store() {
    $row =& $this->getTable();
    $data = JRequest::get( 'post' );

    if (!$row->bind($data)) {
        return false;
    }

    if (!$row->check()) {
        return false;
    }

    if (!$row->store()) {
        return false;
    }

    return true;
}
```

```php
class  TableUser extends JTable
{
    /** @var int Primary key */
    var $id                    = 0;
    var $nom                   = '';

    /**
     * Constructor
     * @param object Database connector object
     */
    function TableUser(& $db) {
        parent::__construct('#__user', 'id', $db);
    }
}
```

Module

- La plus simple des extensions

```php
<?php
//interdit a d'autres script de récupérer ou d'exécuter ce fichier
 defined('_JEXEC') or die('Accès interdit');
?>
<p>
    Hello World
</p>
```

- Front ou Backend (préciser dans le manifest)

- Pour se rapprocher du MVC
  - Helper : version simplifié du couple JModel+JTable
    - Classe standard pour manipuler les requetes SQL
  - Layout : similaire à la notion de Vue
    - Apparences multiples (getLayoutPath() +2eme param)

Module : les Layouts

```php
...

// récupère les données pour la vue
$items  = ModHelloWorldHelper::getItems($userCount);

// ajoute le template de la vue pour l'affichage
require(JModuleHelper::getLayoutPath('mod_helloworld'));
?>
```

```php
<?php
 defined('_JEXEC') or die('Accès interdit');
echo  JText::_('RANDOM USERS');
?>
<ul>
    <?php  foreach ($items as $item) { ?>
    <li>
        <?php  echo JText::sprintf('USER LABEL', $item->name); ?>
    </li>
    <?php  } ?>
</ul>
```

Composant

- Backend et/ou Frontend
- Normalisation de l'arborescence et des noms des classes
- Gestion simplifiée des paramètres de configuration
  - config.xml (même paramétrage que les manifests)

```
$component  = JComponentHelper::getComponent('com_helloworld');
$params = new JParameter( $component->params );
JToolBarHelper::preferences('com_helloworld', 700, 500, 'titre');
```

- Les Vues (JView)
  - Une vue / plusieurs layouts

Implémentation Composant

- MVC : JController, JModel, JView
- Implémentation du CRUD

- Objet spécifique pour l'Admin
  - La barre d'outils (JToolBar)
  - Les sous-menus (JMenu et JSubMenu)

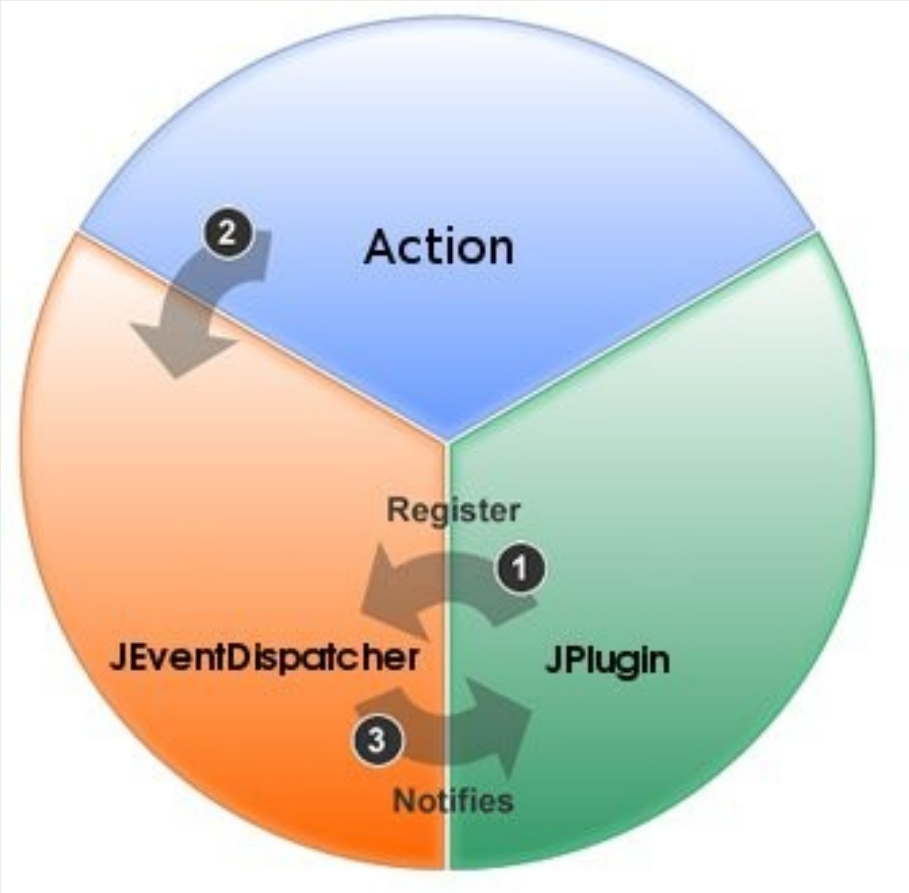
- Les Vues
  - Chaque vue déclare un form « adminForm »

Design Pattern

- Observer … pour les plugins
  - Publish & Subscribe asynchrone

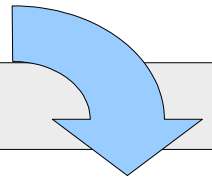
- La classe **JPlugin** est dérivée de la classe **JObserver**

- La classe **JEventDispatcher** est dérivée de la classe **JObservable**.



Plugin

```
...
$dispatcher  =& JDispatcher::getInstance();
$results = $dispatcher->trigger('onLoginHello', array(&$myData));
...
```



```
jimport('joomla.event.plugin');

$dispatcher =& JDispatcher::getInstance();
$dispatcher->register( 'onLoginHello', 'plgSystemHelloworld' );

class plgSystemHelloworld extends JPlugin {
...
    function onLoginHello(&$Tab) {
...
```

Pour aller plus loin ...

- La conformité du multi-lunguisme
  - La traduction via JText et JString
- Compatibilité SEF
  - Redirection via l'objet JRoute
- Optimisation cache
  - Contrôle de rentabilité : JProfiler
  - Stocker votre code récurrent en cache
- Sécurité :
  - Protégez vous des injections SQL
  - Protection CRSF via les Jetons Joomla (form.token)
  - Se protéger des attaques XSS (JRequest casté !)

Questions-réponses

- Avez-vous des questions ?
  - Quelles réponses complémentaires recherchez-vous ?

+ de questions : Espace **Développeurs** sur forum.joomla.fr

Forum Joomla.fr > Développeurs

+ site Développeurs francophone : prochainement sur le portail