



Portage d'application de Moodle 1.9 vers Moodle 2.x



Outils et recommandations à l'usage des développeurs.

Jean FRUITET
Jean.fruitet@univ-nantes.fr





Plan

2

- **Plan**
 - Introduction
 - Les principaux changements entre M1.9 et M2.0 pour les développeurs d'application
 - Data Base API
 - Interface
 - Système de fichier
 - Todo List
 - Outils et documentation

Introduction

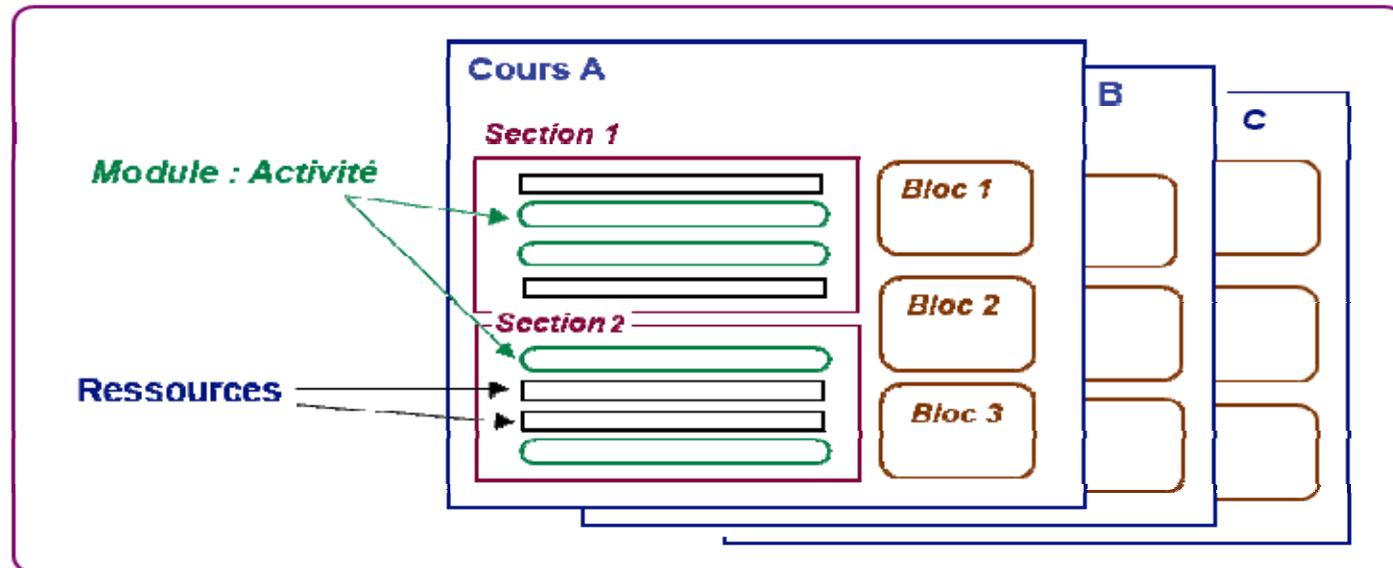
3

- **Objet de l'exposé**
 - Migrer ou porter des applications ?
 - ✦ **Migrer** c'est faire passer un serveur sous Moodle 1.9 vers une version sous Moodle 2.0
 - ✦ **Porter** c'est modifier le code source d'un module pour qu'il puisse s'exécuter sous moodle 2.0. *C'est l'objet de cet exposé.*
 - Migration impossible sans portage préalable
 - ✦ Le portage doit être complet pour chaque module, bloc, filtre sinon certaines fonctions / activités seront perdues. Il doit prendre en compte les fichiers utilisateurs et les backups sous M1.9 lors de la restauration sous M2.0, ce qui est très problématique.

Architecture d'un cours Moodle

4

Serveur Moodle



Modèle de base de données relationnelle



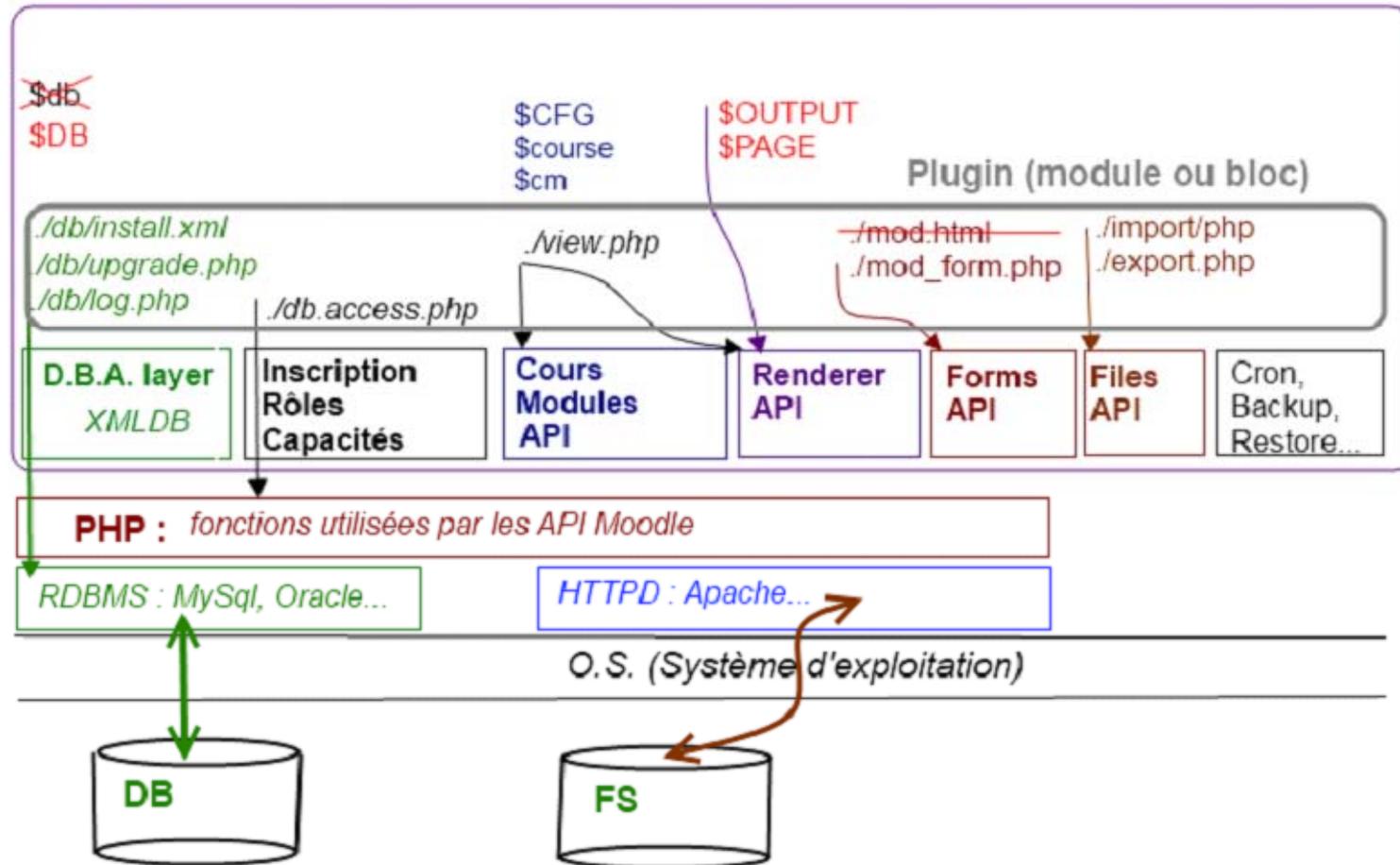
Variables de l'Application Programming Interface

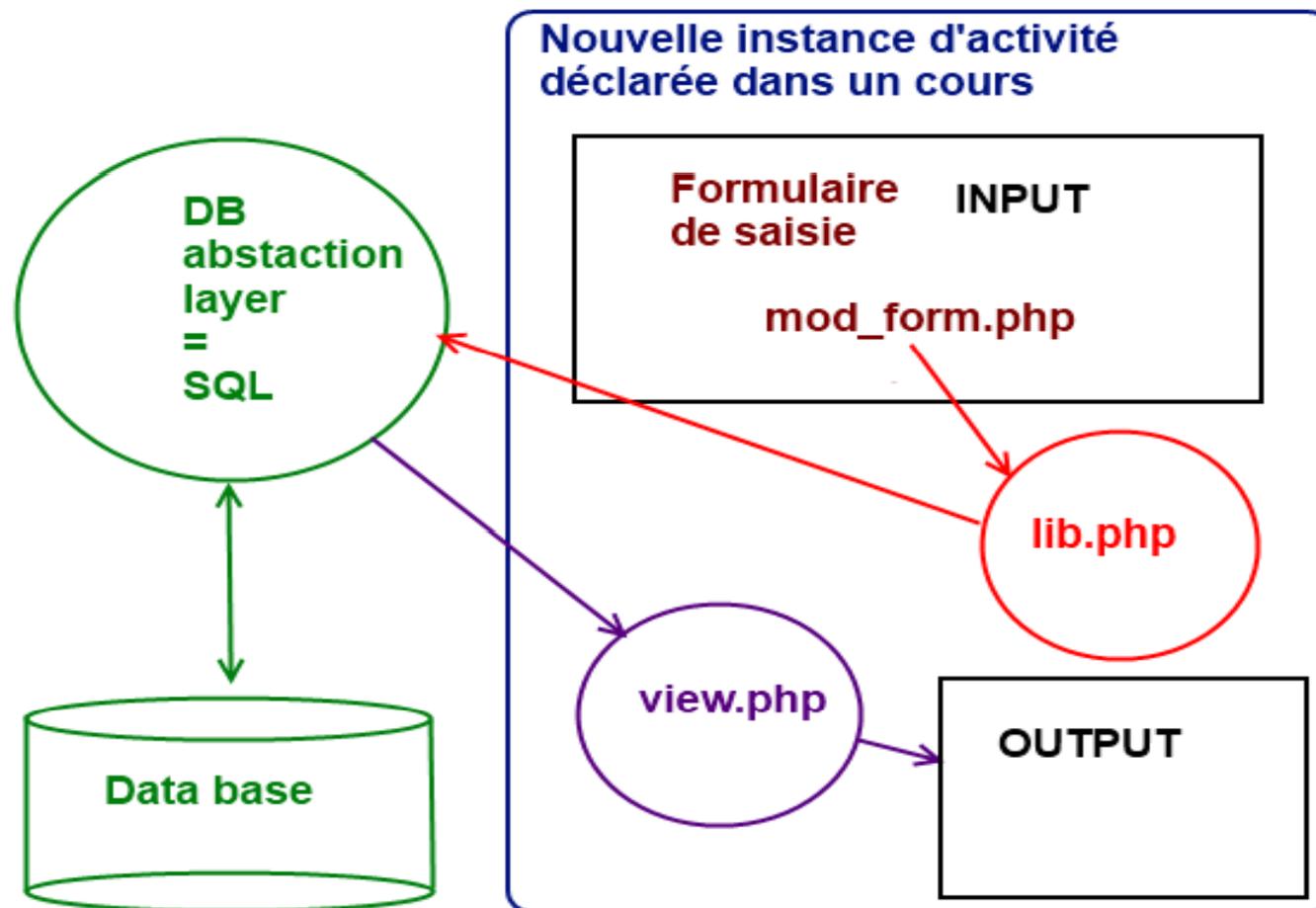


Architecture d'un plugin Moodle

5

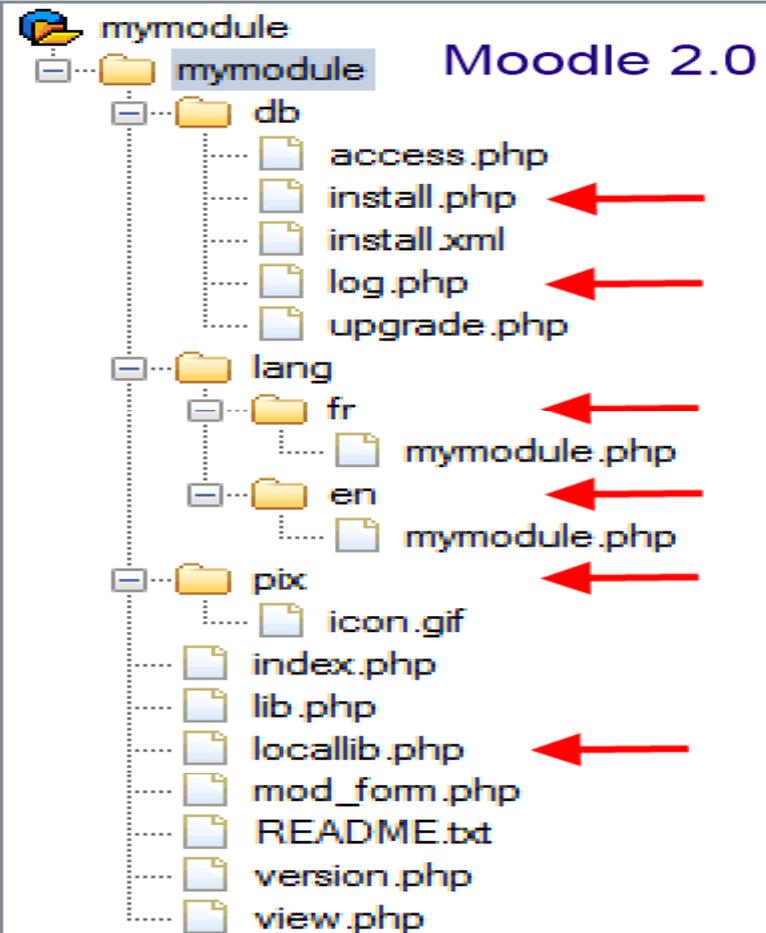
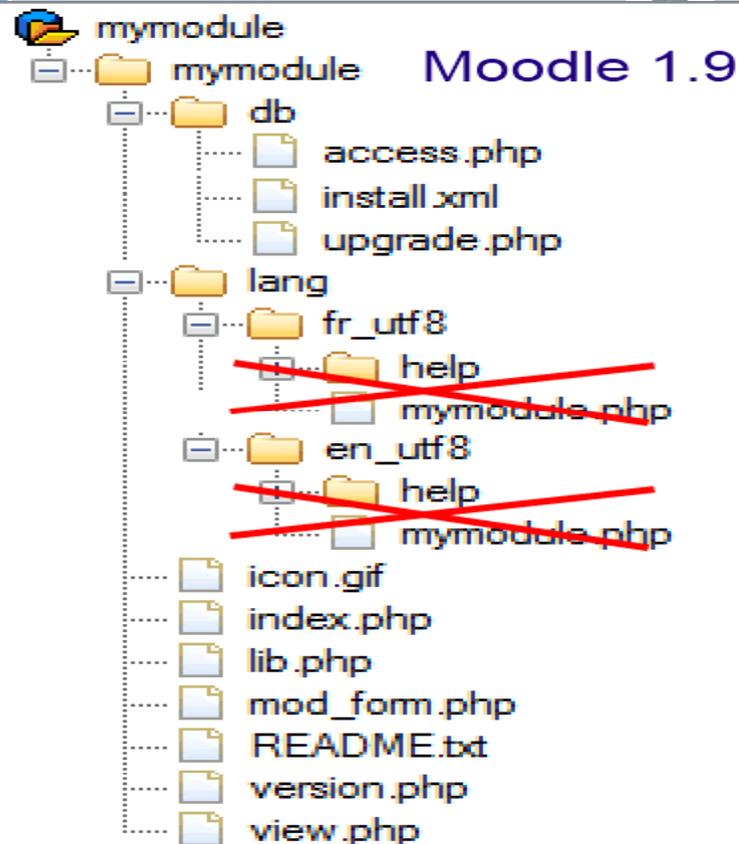
Moodle





Fichiers d'un module

7





Ce qui change avec Moodle 2.0 : BD

8

- L'interface de base de données
 - Data Base API : Généralisation du modèle abstrait de base de données permettant de développer indépendamment des SGBD sous-jacents (MySQL, Postgres, oracle, SQL Server...)
 - Du coup il faut réécrire pas mal de scripts
 - ✦ ./db/INSTALL.XML
 - ✦ ./db/upgrade.php
 - ✦ ./db/access.php
 - Il faut corriger les appels aux fonctions de gestion de données :
 - ✦ `$DB->get_record('mymodule_mytable', array(« id » => $id));`

Et reprendre les requêtes SQL

- ✦ `$params=array("code" => $code, "id" => $id);`
- ✦ `$sql="SELECT * FROM {mymodule_mytable} WHERE code=:code AND ref_referentiel=:id ";`
- ✦ `$record=$DB->get_record_sql($sql, $params);`

Ce qui change : API

9

- **L'interface utilisateur**

- Elle repose exclusivement sur l'interaction entre l'API `formlib.php` et la librairie d'affichage sur la sortie standard.
- Finies les fonctions à 10 paramètres, on n'utilise plus que deux objets :

```
$PAGE = new moodle_page();
```

```
$OUTPUT = new core_renderer(null, null);
```

Qui ont toutes les méthodes nécessaires pour créer une page Web et y placer du texte et des formulaires. Magique 😊

- ✦ `$PAGE->requires->css('/mod/mymodule/mystyle.css');`
- ✦ `$PAGE->requires->js('/lib/overlib/overlib.js');`
- ✦ `$PAGE->navbar->add(get_string('list', 'mymodule');`
- ✦ `$PAGE->set_title(get_string('modulename', 'mymodule');`
- ✦ `$PAGE->set_heading($course->fullname);`

Formslib.php

10

- Cette API est améliorée par rapport à Moodle 1.9
 - ✧ `<?php`
 - ✧ `if (!defined('MOODLE_INTERNAL'))`
 - ✧ `die('Direct access to this script is forbidden.');`
 - ✧ `require_once ($CFG->dirroot.'/course/moodleform_mod.php');`
 - ✧ `class mod_mymodule_mod_form extends moodleform_mod {`
 - ✧ `protected $_mymoduleinstance = null;`
 - ✧ `function definition() {`
 - ✧ `global $CFG, $DB, $cm, $context;`
 - ✧ `$mform =& $this->_form;`
 - ✧ `(...)`
 - ✧ `$mform->addElement('header', 'general',`
`get_string('creer_instance_mymodule', 'mymodule'));`
 - ✧ `$mform->addElement('text', 'name', get_string('name_instance',`
`'mymodule'), array('size'=>'64'));`
 - ✧ `$mform->setType('name', PARAM_TEXT);`
 - ✧ `.../...`

Formslib.php (suite)

11

- Suite...

- ✦ `$mform->addRule('name', get_string('name', 'mymodule'), 'required', null, 'client');`
- ✦ `$mform->addHelpButton('name', 'name_instanceh', 'mymodule');`
- ✦ `$this->add_intro_editor(true, get_string('intro', 'mymodule'));`
- ✦ (...)
- ✦ `// hidden params`
- ✦ `$mform->addElement('hidden', 'id', $cm->id);`
- ✦ `$mform->setType('id', PARAM_INT);`
- ✦ `$mform->addElement('hidden', 'contextid', $context->id);`
- ✦ `$mform->setType('contextid', PARAM_INT);`
- ✦ (...)
- ✦ `// buttons`
- ✦ `$this->add_action_buttons(true, get_string('savechanges', 'mymodule'));`
- ✦ (...)
- ✦ `}`
- ✦ `}`



Formulaire de saisie correspondant au code

12

Créer une instance de référentiel

Titre de l'instance *



Texte d'introduction*

Police Taille police Format

B *I* U ABC x_2 x^2

Chemin:

Format HTML

Ce qui change : FILE API

13

- **Le système de fichiers**
 - C'est le gros morceau. Les fichiers utilisateurs ne sont plus liés à un cours. Le même fichier peut être **déposé une fois** et **cité n fois**.
 - L'interface de dépôt de fichier accepte des « Réservoirs de données » distants. Il faut réécrire le code d'importation / exportation de données
 - ✦ 1. Un utilisateur démarre la création / modification d'un item existant (i.e. un message de forum, une ressource, une rentrée de glossaire, etc.)
 - ✦ 2. Il clique un bouton pour sélectionner un nouveau fichier ;
 - ✦ 3. Il accède à une boîte de dialogue "Choisir fichier..."
 - ✦ 4. Il sélectionne un fichier, l'API Repository prend en charge le copie du fichier dans la zone "draft file area" (brouillon) de Moodle
 - ✦ 5. Le fichier est lié au formulaire.
 - ✦ 6. Quand l'utilisateur clique sur "sauvegarder", la File API est invoquée pour déplacer le fichier vers un emplacement permanent.

Ce qui change : Formlib file API

14

- **Les formulaires de sélection de fichiers**
 - **Trois formulaires de sélection disponibles :**
 - ✦ 1. filemanager - attacher un ou plusieurs fichiers à un objet
 - ✦ 2. editor - spécifier un champ textarea avec éditeur HTML intégré, ainsi que la gestion des images et des vidéos encapsulés dans du code HTML.
 - ✦ 3. filepicker - désigner un fichier à la fois pour les situations où vous voulez traiter le fichier sans l'enregistrer dans l'espace utilisateur
 - **Par rapport à Moodle 1.9 les types de champs de formulaires suivants sont devenus obsolètes :**
 - ✦ 1. file - utilisé pour un chargement standard depuis le poste de travail.
 - ✦ 2. htmleditor - ancienne méthode pour encapsuler un éditeur HTML dans un champ textarea et qui ne permet pas le support des réservoir de données externes...



Exemple : Sélectionner un fichier (1)

15

- `<?php`
- `// upload_form.php for Moodle 2.0`
- `// This file is part of Moodle -
http://moodle.org/`
- `require_once($CFG->libdir.'/formslib.php');`
`//putting this is as a safety as i got a class
not found error.`
- `class referentiel_upload_simple_form extends
moodleform {`
- `function definition() {`
- `$mform = & $this->_form;`
- `$instance = $this->_customdata;`
- `// visible elements`
- `$mform->addElement('header', 'general',
$instance['msg']);`
- `(...)`



Exemple : Sélectionner un fichier (2)

16

```
• // sélection fichier pour importation
•     $mform->addElement('filepicker', 'mymodule_file',
get_string('uploadafile'), null, $instance['options']);
•     // hidden params
•     $mform->addElement('hidden', 'id', $instance['id']);
•     $mform->setType('id', PARAM_INT);
•     $mform->addElement('hidden', 'contextid',
$instance['contextid']);
•     $mform->setType('contextid', PARAM_INT);
•
•     $mform->addElement('hidden', 'filearea',
$instance['filearea']);
•     $mform->setType('filearea', PARAM_ALPHA);
•
•     $mform->addElement('hidden', 'action', 'uploadfile');
•     $mform->setType('action', PARAM_ALPHA);
•     // buttons
•     $this->add_action_buttons(true,
get_string('savechanges', 'admin'));
•     }
• }
```



Sélectionner un fichier (3)

17

- `<?php`
- `// script de sélection de fichier à télécharger Moodle 2.0`
- `require_once("../../config.php");`
- `require_once('upload_form.php');`
- `require_once("$CFG->dirroot/repository/lib.php");`
- `(...) // autres traitements`
- `$PAGE->set_heading($title);`

- `$options = array('subdirs'=>0, 'maxbytes'=>get_max_upload_file_size($CFG->maxbytes, $course->maxbytes, 0), 'maxfiles'=>1, 'accepted_types'=>'*', 'return_types'=>FILE_INTERNAL);`
- `$mform = new mod_mymodule_upload_form(null, array('id'=>$cm->id, 'contextid'=>$context->id, 'filearea'=>'document', 'options'=>$options));`



Sélectionner un fichier (4)

18

```
• if ($mform->is_cancelled()) {
•     redirect(new
moodle_url('/mod/mymodule/view.php', array('id'=>$cm-
>id)));
• } else if ($mform->get_data()) {
•     mymodule_upload_document($mform, $mymodule->id);
•     die();
• }

• echo $OUTPUT->header();
• echo $OUTPUT->box_start('generalbox');
• $mform->display();
• echo $OUTPUT->box_end();
• echo $OUTPUT->footer();
• die();

• ?>
```

Formulaire File Picker

19

- Exemple de sélecteur de fichier

Titre : Déclarez vos compétences C2i2e - version 2011

Connecte sous le nom «

Section 1 ► Déclarez vos compétences C2i2e - version

Sélecteur de fichiers

Ressources(s) associée(s) à l'activité

Type de document
Description
Titre du document
URL ?
Déposer un fichier Taille

Aucun fichier joint

Annexe: Aucun fichier choisi

Enregistrer sous:

Auteur:

Choisir une licence:

Formulaire File Picker

20

- Le fichier téléchargé est disponible

ID	Auteur	Cours	Type	Suivi	Validation	Date auteur	Date suivi	Menu																				
Compétences déclarées dans les activités par Alpha ARAIGNE																												
	A.1-1	A.1-2	A.1-3	A.1-4	A.1-5	A.2-1	A.2-2	A.2-3	A.3-1	A.3-2	A.3-3	A.3-4	B.1-1	B.1-2	B.1-3	B.2-1	B.2-2	B.2-3	B.2-4	B.2-5	B.3-1	B.3-2	B.3-3	B.3-4	B.3-5	B.4-1	B.4-2	B.4-3
9	Alpha ARAIGNE	Ref	Déposer un document Consignes	Admin Utilisateur	Validé	lundi 20 juin 2011, 13:44	lundi 20 juin 2011, 17:43																					
B.2-3 B.2-4 J'ai sélectionné le référentiel C2i2e						Tâche approuvée par Admin Utilisateur (20/06/2011 13:56)																						
			Document	1	referentiel-c2i2e_etudiant-20110531.csv																							

Exporter des données

21

- Soit un module générant un fichier XML (du texte)
 - Le fichier est stocké dans le système de fichiers de Moodle
 - ✦ `<?php // Moodle 2.0`
 - ✦ `$id = required_param('id', 0, PARAM_INT); // course module`
 - ✦ `$cm = get_coursemodule_from_id('mymodule', $id, 0, false, MUST_EXIST);`
 - ✦ `$course = $DB->get_record("course", array("id" => "$cm->course"), '*', MUST_EXIST);`
 - ✦ `$mymodule = $DB->get_record("mymodule", array("id" => "$cm->instance"), '*', MUST_EXIST);`
 - ✦ `(...)`
 - ✦ `$context = get_context_instance(CONTEXT_MODULE, $cm->id);`
 - ✦ `$export_dir = '/dossier1/dossier2/'; // commençant et finissant par '/'`
 - ✦ `$filename = 'mon_fichier_20100614'; // nom de fichier (daté)`
 - ✦ `$ext = '.xml'; // extension selon type de fichier`

Exporter des données

22

- Suite...

```
✧ if ($fs = get_file_storage()){
✧     // Prepare file record object
✧     $fileinfo = array(
✧         'contextid' => $context->id, // ID du context
✧         'component' => 'mod_mymodule', // habituellement =
composant
✧         'filearea' => 'mymodule', // habituellement = nom de
table
✧         'itemid' => 0, // ID de la ligne de la table ou 0
✧         'filepath' => $export_dir, // beginning and ending in /
✧         'filename' => $filename.'.'.$ext // any filename
✧     );
✧     // Créer le fichier contenant le texte (les données sont
dans $contenu_expout)
✧     $fs->create_file_from_string($fileinfo,
$contenu_expout);
✧ } ?>
```

Exporter des données (suite)

23

- **Seconde étape : afficher l'URL du fichier exporté**
 - ✧ `<?php // Moodle 2.0`
 - ✧ `(...)`
 - ✧ `$id = required_param('id', 0, PARAM_INT); // course module id`
 - ✧ `$cm = get_coursemodule_from_id('mymodule', $id, 0, false, MUST_EXIST);`
 - ✧ `$course = $DB->get_record("course", array("id" => "$cm->course"), '*', MUST_EXIST);`
 - ✧ `$mymodule = $DB->get_record("mymodule", array("id" => "$cm->instance"), '*', MUST_EXIST);`
 - ✧ `(...) $context =`
`get_context_instance(CONTEXT_MODULE, $cm->id);`

Exporter des données (suite)

24

- Affichage de l'URL du fichier exporté (suite)
 - ✧ // Paramètres du chemin du fichier
 - ✧ `$filearea='mymodule';` // en général le nom de la table
 - ✧ `$item = 0;` // en général l'ID de la ligne dans la table, 0 par défaut
 - ✧ `$export_dir = '/dossier1/dossier2/';` // commençant et finissant par '/'
 - ✧ `$filename = 'mon_fichier_20100614';` // nom de fichier (daté)
 - ✧ `$ext = '.xml';` // extension en fonction du type de fichier
 - ✧ // Chemin relatif dans le système de fichiers de Moodle 2.0
 - ✧ **`$fullpath =`**
 - ✧ **`"/$context->id/mod_mymodule/$filearea/$item . $export_dir . $filename.$ext";`**
 - ✧ // Url pour télécharger le fichier enregistré
 - ✧ **`$url= new moodle_url($CFG->wwwroot.' /pluginfile.php' . $fullpath);`**
 - ✧ // Affichage de l'url
 - ✧ `echo "<div>$exportfilename</div>\n";`
 - ✧ `?>`

Importer des données

25

- Importer un fichier XML
 - Le fichier importé **n'est pas stocké** dans le système de fichiers
 - ✦ `<?php // Moodle 2.0`
 - ✦ `// import.php`
 - ✦ `require_once('../../config.php');`
 - ✦ `require_once('lib.php');`
 - ✦ `require_once('import_form.php');` // formulaire
 - ✦ `require_once("$CFG->dirroot/repository/lib.php");`
 - ✦ `$id = required_param('id', 0, PARAM_INT); // course module id`
 - ✦ `$cm = get_coursemodule_from_id('mymodule', $id, 0, false, MUST_EXIST);`
 - ✦ `$course = $DB->get_record("course", array("id" => "$cm->course"), '*', MUST_EXIST);`
 - ✦ `$mymodule = $DB->get_record("mymodule", array("id" => "$cm->instance"), '*', MUST_EXIST);`
 - ✦ `(...)`
 - ✦ `$context = get_context_instance(CONTEXT_MODULE, $cm->id);`
 - ✦ `require_capability('mod/mymodule:import', $context);`

Importer des données (suite)

26

- .../...
 - ✧ // formulaire de saisie d'un fichier
 - ✧ `$options = array('subdirs'=>0, 'maxbytes'=>get_max_upload_file_size($CFG->maxbytes, $course->maxbytes, 0), 'maxfiles'=>1, 'accepted_types'=>('*', 'return_types'=>FILE_INTERNAL);`
 - ✧ // Formulaire de saisie du nom de fichier à importer
 - ✧ `$mform = new mymodule_import_form(null, array('id'=>$cm->id, 'contextid'=>$context->id, 'filearea'=>'mymodule', 'action' => 'importmymodule', 'msg' => get_string('import', 'mymodule'), 'options'=>$options));`
 - ✧ // recuperer le fichier charge
 - ✧ `if ($mform->is_cancelled()) {`
 - ✧ `redirect(new moodle_url('/course/view.php', array('id'=>$course->id)));`
 - ✧ `}`
 - ✧ `else if ($mform->get_data()) {`
 - ✧ `$returnlink = new moodle_url('/mod/mymodule/view.php', array('id'=>$cm->id));`

Importer des données (suite)

27

- ```
if ($formdata = $mform->get_data()) {
 * // Tester si le file area existe
 * if (empty($formdata->filearea)) return false;
 * $fs = get_file_storage(); // récupérer les données
 * if ($newfilename=
 $mform->get_new_filename('mymodule_file')) {
 * // Récupérer le contenu du fichier importé
 * if (!empty($contents = $mform->
 * >get_file_content('mymodule_file'))){
 * // Traiter les données téléchargées
 * (...)
 * die();
 * }
 * redirect($returnlink);
 * }
```
- Je ne détaille pas le code du script du formulaire de sélection ...



# Ce qui ne change (pratiquement) pas

28

- **L'authentification, les capacités, les rôles**
  - Quelques variables à changer dans le fichier `.mod/mymodule/db/access.php`
  - La table `moodle_role_assignment` a changé ce qui entraîne la réécriture de quelques requêtes sur les rôles...
- **Les cours**
  - La notion de méta-cours disparaît mais cela n'a pas d'influence sur les modules
- **Les groupes**
  - On peut conserver les fonctions de gestion des groupes
- **Les fichiers de langue**
  - Sauf ... les fichiers d'aide en ligne qui doivent être réécrits et intégrés dans les fichiers de langue.



# Documentation développeur

29

- **Outils de migration**
  - Deux scripts permettant de tester le code source avant et en cours de migration sont disponibles sur <http://moodlemoot2011.uvt.rnu.tn/course/view.php?id=17>
    - ✦ `check_db_syntax.php` (disponible sur Moodle.org)
    - ✦ `check_m2_syntax.php` (fait maison 😊)
- **Documentation**
  - Index des pages concernant des caractéristiques Moodle 2.0 [http://docs.moodle.org/dev/Category:Moodle\\_2.0](http://docs.moodle.org/dev/Category:Moodle_2.0)
  - API File [http://docs.moodle.org/en/Development:Repository\\_API](http://docs.moodle.org/en/Development:Repository_API)  
[http://docs.moodle.org/en/Development:Using\\_the\\_File\\_API\\_in\\_Moodle\\_forms](http://docs.moodle.org/en/Development:Using_the_File_API_in_Moodle_forms)