PHP Avancé

Olivier Martineau
SpreadButton
olivier@sb.am

Alexandre Pauchet
INSA Rouen Normandie
alexandre.pauchet@insa-rouen.fr

Framework ?

- «Un cadre»
- Couche supplémentaire à PHP
- Arrêter de re-inventer la roue

Les plus connus

- CodeIngiter
- CakePHP
- Symfony
- Zend Framework
- Jelix
- Laravel

Services apportés

- Internationalisation
- Template
- Routage
- ORM
- Cache
- Profiling
- Gestion des erreurs
- Sécurité

Internationalisation

- Pas de texte dans le code
- Gestion des langues par le framework
- Fonction __(‘Translate me’)
- tout en UTF-8
- + formatage dates, devises, nombres...

Template

- Les vues : génération du HTML
- Modèle HTML sans code métier
- Syntaxe courte PHP
  `<?=$var?>` équivalent à `<?php echo $var; ?>`

```
<html><body>
Hello <?=$tpl->nom?>
</body></html>
```

- Librairies existantes : Twig, Smarty, Mustache, RAIN Tpl

Template

- Organisation de son code HTML pour une réutilisation

```
<html>
 <head>
  <?php include('_head.html'); ?>
 </head>
 <body>
  <?php include('_top.html'); ?>
   Contenu de la page
 <?php include('_footer.html'); ?>
 </body>
</html>
```

Exemple template

- template.php

```php
<?php
$tpl = new ArrayObject();
$tpl->title = "Titre de la page";
$tpl->nom_page = "Nom de la page";
$tpl->liste = array();
$tpl->liste[] = "Element 1";
$tpl->liste[] = "Element 2";
$tpl->liste[] = "Element 3";
$tpl->liste[] = "Element 4";
include('template.html.php');
```

- template.html.php

```php
<html>
<head>
<title><?=$tpl->title?></title>
</head>
<body>
<h1><?=$tpl->nom_page?></h1>
<ul>
<?php foreach ($tpl->liste as $elem) { ?>
    <li><?=$elem?></li>
<?php } ?>
</ul>
</body>
</html>
```

Routage d'URL

- Le code PHP n'est plus directement lié aux URL
  - Sans : http://monsite.com/page.php?type=categorie&id=1
  - Avec : http://monsite.com/categorie/1
- Tout est géré par un seul index.php

Fichier .htaccess

- Configuration du serveur web Apache
- Local au dossier et sous-dossiers
- Usage :
  - redirections (temporaire, permanente...)
  - contrôle d’accès
  - re-écritures d’URL

Ré-écriture d'URL

- .htaccess

```
RewriteEngine On
# RewriteBase /URL_JUSQUA_L_URL_DU_DOSSIER_DU_HTACCESS
RewriteBase /~bleponge/TW
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule (.*) index.php?arg=$1 [QSA,L]
```

1. Activation du moteur de ré-écriture d'URL
2. Configuration de l'URL correspondant au répertoire qui contient le .htaccess (obligatoire pour les dossiers /~homedir)
3. Exclusion des URL correspondante à un fichier
4. Exclusion des URL correspondante à un dossier
5. Transformation de toutes les URL en paramètre de index.php

Routage d'URL

- Si l'URL demandée ne correspond pas à un fichier, `index.php` est appelé avec l'URL passé en parametre $_GET['arg']

- Gestion manuelle de la page d'erreur 404

```
header("HTTP/1.0 404 Not Found");
```

Exemple routage d'URL

- index.php

```php
<?php
if (!empty($_GET['arg'])) {
    include($_GET['arg'].".php");
} else {
    header("HTTP/1.0 404 Not Found");
    include("route404.php");
}
```

- route404.php

```php
<html>
<head>
    <title>ROUTE 404</title>
</head>
<body>
<h1>ROUTE 404</h1>
<a href="./route1">Route 1</a><br/>
<a href="./route2">Route 2</a><br/>
<a href="./route3">Route 3</a><br/>
</body>
</html>
```

Sécurité

- Authentification / Autorisation
- Protections contre les injections SQL, XSS (Cross-site scripting), CSRF (Cross-site request forgery)...

Espace de nommage

- Garantie de l'interopérabilité
- Nouveauté PHP 5.3
  - `namespace ASI (définition d'un espace de nommage)`
  - `use ASI (cration d'alias)`
  - `new ASI\maClasse (pour une classe, interface, etc.)`
- Attention, dans votre namespace appeler les classes native avec un \, par exemple lors d'un

```
throw new \Exception(« erreur »);
```

Sécurité

- Never Trust Input !
- Une fonction pour le filtrage ou la validation.

```
$var = filter_var($var, FILTER);
```

FILTER_VALIDATE_EMAIL	Valide une adresse email
FILTER_VALIDATE_INT	Valide un entier
FILTER_VALIDATE_URL	Valider une URL
FILTER_VALIDATE_FLOAT	Valide un nombre décimal

FILTER_SANITIZE_EMAIL	Nettoie une adresse email
FILTER_SANITIZE_STRING	Nettoie un chaine de tag
FILTER_VALIDATE_URL	Nettoie une URL
FILTER_VALIDATE_FLOAT	Nettoie un nombre décimal
FILTER_SANITIZE_SPECIAL_CHARS	Transforme en HTML les ‘« <>&

Sécurité

- Requêtes SQL « préparées »

```
$db = new \PDO(…);

$prep = $db->prepare('INSERT INTO matable (champs1, champs2')
VALUE ( :valeur1, :valeur2);

$prep->execute(array(':valeur1'=>'Pierre', ':valeur2'=>'Dubois');
```

- Évite tous problèmes d'injection SQL.

Abstraction BD

- Indépendance par rapport au serveur de BD
- Structure :
  - Classe objet métier
  - Classe de gestion de la persistance pour chaque classe métier
  - Classe d’abstraction base de données

classe DB

```php
<?php
class db {
    private static $conString; // chaine de connexion PDO
    private static $conActive = false; // connexion DB active
    private static $db; // objet retourné par nw PDO
    private static $preparequery = false; // identifiant requete préparée

    public static function Connect ($constring = false) {
        if ($constring) {
            db::$conString = $constring;
        }
        try {
          self::$db = new \PDO(db::$conString);
          self::$db->setAttribute(\PDO::ATTR_ERRMODE, \PDO::ERRMODE_EXCEPTION);
        } catch(Exception $e) {
            self::$conActive = false;
            throw $e;
        }
        self::$conActive = true;
    }

    public static function OnlyConActive () {
        if (!self::$conActive) {
            throw new \Exception("DB non connectée");
        }
    }

    public static function Query ($sql) {
        self::OnlyConActive();
        return self::$db->query($sql);
    }
}
```

Fichier : abstraction.db.php

classe DB

```php
public static function Prepare ($sql) {
    self::OnlyConActive();
    try {
        self::$preparequery = self::$db->prepare($sql);
    } catch(Exception $e) {
        throw $e;
    }
}
public static function Execute ($params) {
    self::OnlyConActive();
    if (!self::$preparequery) {
        throw new \Exception("On ne peut pas executer avant d'avoir préparé sa query");
    }
    try {
        self::$preparequery->execute($params);
        return self::$preparequery->fetchAll();
    } catch(Exception $e) {
        throw $e;
    }
    self::$preparequery = false;
}
```

Fichier : abstraction.db.php

Utilisation classe DB

```php
db::Connect('sqlite:db.sqlite');
db::Query ("CREATE TABLE IF NOT EXISTS matable (
matable_id INTEGER PRIMARY KEY,
matable_date TEXT NOT NULL,
matable_texte TEXT NOT NULL
)");
$result = db::Query('SELECT name FROM sqlite_master WHERE type="table"');
foreach($result as $row) {
    echo $row['name'];
}
```

Fichier : abstraction.demo.php

Classe de persistance Objet (ORM)

- Interface base de données <->objet
- Hérite de votre classe d'abstraction db
- Retour des objets, ou des tableaux d'objets
- Gestion CRUD (create, read, update et delete)
- Gestion des listes
- Gestion de l'initialisation (create table)

Exemple classe de persistance objet

```php
<?php

class MonObjetDb extends db {
    public static function CreateTable () {
        self::Query ("CREATE TABLE IF NOT EXISTS matable (
        matable_id INTEGER PRIMARY KEY,
        matable_date TEXT NOT NULL,
        matable_texte TEXT NOT NULL
        )");
}

public static function LoadSample() {
    $allm = array(
        array('texte'=>"Symfony, l'un des framework les plus utilisé",'url'=>"http://symfony.com/"),
        array('texte'=>"L'un des premiers frameworks PHP, inspiré de RAIL",'url'=>"http://cakephp.org/"),
        array('texte'=>"Code Igniter, l'une des références",'url'=>"http://ellislab.com/codeigniter"),
        array('texte'=>"FuelPHP un framework qui monte en ce moment",'url'=>"http://www.fuelphp.com/")
    );
    foreach ($allm as $m) {
        $obj = new MonObjet();
        $obj->setTexte($m['texte']);
        self::Create($obj);
    }
}
```

Fichier : monobjet.db.php

Exemple classe de persistance objet

```php
public static function Create($obj) {
    self::Prepare("INSERT INTO matable (matable_date, matable_texte ) VALUES ( :date, :texte)");
    self::Execute(array(':date'=>$obj->getDate(),
                        ':texte'=>$obj->getTexte()
                        ));
}

public static function Read($id) {
    self::Prepare("SELECT * FROM matable WHERE matable_id = :id");
    $liste_messages = self::Execute(array(':id'=>(int)$id ));
    $row = $liste_messages[0];
    $obj = new MonObjet();
    $obj->setId($row['message_id']);
    $obj->setDate($row['message_date']);
    $obj->setTexte($row['message_texte']);
    return $obj;
}

public static function Update($obj) {
    self::Prepare("UPDATE matable SET matable_date = :date, matable_texte = :texte WHERE matable_id = :id");
    self::Execute(array(':date'=>$obj->getDate(),
                        ':texte'=>$obj->getTexte(),
                        ':id'=>$obj->getId()
                        ));
}

public static function Delete($id) {
    self::Prepare("DELETE FROM matable WHERE matable_id = :id");
    self::Execute(array(':id'=>$id ));
}
```

Fichier : monobjet.db.php

Exemple classe de persistance objet

```php
public static function ListObjet() {
  $objets = array();
  $liste_messages = self::Query("SELECT * FROM matable ORDER BY matable_date DESC");
  foreach($liste_messages as $row) {
      $obj = new MonObjet();
      $obj->setId($row['matable_id']);
      $obj->setDate($row['matable_date']);
      $obj->setTexte($row['matable_texte']);
      $objets[] = $obj;
  }
  return $objets;
}
```

Fichier : monobjet.db.php

Organisation du code

- Lisibilité du MVC
- Facilité de gestion et de mise à jour

`index.php`	Charge le framework, et gère le routage	
`conf.php`	Paramètres spécifiques à l'installation (db…)	
`/classes`	Classes de gestion des objets (modèles)	message.class.php
`/templates`	Templates HTML (vue)	home.tpl.php
`/routes`	Contrôleurs	home.route.php
`/static`	Fichiers images, css et JS	
`/lib`	Librairies externes	

Cache

- Objectif : accélérer l’affichage des pages
- Stockage plus ou moins temporaire
- Plusieurs niveaux de cache :
  - Cache HTTP
  - Cache d’opcodes PHP
  - Cache applicatif
  - Cache MySQL
  - Reverse Proxy
  - CDN : content delivery network

Cache HTTP

- Gestion de la durée de stockage dans le cache du navigateur.
- Header http :
  - Expires
  - Cache-control + max-age

```
$offset = 60 * 15;
header('Expires: '.date('D, d M Y H:i:s', time()+$offset).' GMT’);
header('Cache-Control: max-age=$offset, must-revalidate');
```

Cache HTTP

- Attention : les cookies désactivent le cache local, idem pour les `session_start()`