

M.V.C., un T.P. long avec PHP et MySQL

Le but de ce T.P. est d'implémenter un mini-framework de développement en PHP avec MySQL. On nommera **minifp** ce mini-framework.

Voici les consignes générales : ce mini-framework doit réaliser un maximum de choses en ligne de commandes sans que l'utilisateur n'ait à écrire de PHP. Chaque commande qui génère du MySQL doit afficher systématiquement le code MySQL produit.

1. Aide et liste des options

Implémenter tout d'abord une structure générale de programme avec un rappel de l'aide si aucun paramètre n'est fourni. Les options principales se nomment

```
--aide  
--modele  
--vue  
--controle
```

On affichera un message d'erreur et on quittera le script si une option autre que celles listées est utilisée.

2. Option M comme MODELE

Les sous-options de `--modele` sont

```
table -def    DONNEES_CR
table -data   DONNEES
table -def    -yaml FICHIER_CR
table -data   -yaml FICHIER
```

Le but de ces options est soit de créer et de mettre à jour la définition des tables (`table -def` et `table -def -yaml`) soit de saisir des données (`table -data` et `table -data -yaml`). Les options `-yaml` réalisent la même chose que les autres options mais à l'aide de fichiers YAML et non pas à la suite des options en ligne de commande.

`table` désigne le nom d'une table MySQL. Si elle n'existe pas, on la crée silencieusement. Si elle existe déjà, on la met à jour. Un champ ID, auto-incrémenté sera automatiquement créé.

`DONNEES_CR` correspond à des couples (nomChamp,typeChamp) séparés par le symbole deux-points. On peut se restreindre dans un premier temps aux deux types **entier** et **chaîne**. `DONNEES` correspond à des couples (nomChamp,valeurChamp) séparés par le symbole deux-points, comme pour la création.

3. Option V comme VUE

Les sous-options de `--vue` sont

```
-liste
table -montre
table -uri
table -navigue
```

Ici, `table` correspond au nom d'une table créée avec l'option `--modele`.

L'option `-liste` affiche à l'écran la liste des tables disponibles.

L'option `-montre` affiche la table à l'écran alors que l'option `uri` montre l'URL associée et l'option `navigue` lance le navigateur et affiche les données associées à la table et à l'URL. On utilisera le port 1234.

4. Option C comme CONTROLEUR

Les sous-options de `--controle` sont

```
-sauve  
-restaure FICHIER  
-simule table NB  
-admin
```

`-sauve` sauvegarde de façon incrémentale les tables créées avec leurs données.

`-restaure` supprime toutes les tables courantes et reprend un `-FICHIER` créé par `-sauve`.

`-simule` permet de remplir pseudo-alaétoirement une `table` avec `NB` enregistrements.

`-admin` utilise le navigateur pour afficher un panneau login/password et, une fois l'authentification réalisée pour l'administrateur, permet de modifier les définitions des tables et/ou les valeurs des données.

5. Options et paramètres complémentaires

On ajoutera et on documentera toute option et tout paramètre complémentaire qui paraît intéressant. Par exemple on pourrait limiter les champs affichés via une liste de champs, on pourrait proposer un export CSV ou XML, la production d'un PDF...

6. Exemples de session

```
$gh>minifp
```

```
[minifp] Rappel des options :
```

```
--help      (cette aide)  
--modele    (pour créer/remplir les tables)  
--vue       (pour consulter les tables à l'écran ou dans un navigateur)  
--controle  (pour gérer l'application)
```

```

$gh>minifp --modele DEMO -def nom:chaine age:entier

  [minifp] Table DEMO créée.
           Voici le code MySQL utilisé :
...

$gh>minifp --modele DEMO -data nom:DUPONT age:25

  [minifp] Données insérées dans la table DEMO.
           Voici le code MySQL utilisé :
...

$gh>minifp --modele DEMO -def prenom:chaine

  [minifp] Table DEMO mise à jour.
           Voici le code MySQL utilisé :

$gh>minifp --modele EXEMPLE -def produit:chaine code:entier

  [minifp] Table EXEMPLE créée.
           Voici le code MySQL utilisé :
...

$gh>minifp --modele DEMO -data -yaml demo1.yaml

  [minifp] Données insérées dans la table DEMO
           à partir du fichier demo1.yaml.
           Voici le code-mysql utilisé :
...

$gh>minifp --modele DEMO -data -yaml demo2.yaml

  [minifp] Données insérées dans la table DEMO
           à partir du fichier demo1.yaml.
           Voici le code-mysql utilisé :
...

$gh>minifp --vue -liste

  [minifp] Voici la liste des 2 tables disponibles :
           1. DEMO          avec 1 enregistrement

```

```

                2. EXEMPLE                0 enregistrement
...

$gh>minifp --vue DEMO -montre

    [minifp] Voici la structure de la table DEMO
...
    [minifp] Voici les données de la table DEMO
...

$gh>minifp --vue EXEMPLE -montre

    [minifp] Voici la structure de la table EXEMPLE
...
    [minifp] Voici les données de la table EXEMPLE
...

$gh>minifp --vue EXEMPLE -uri

    [minifp] L'URL pour la table EXEMPLE est :

    http://localhost:1234/minifp/EXEMPLE

$gh>minifp --vue EXEMPLE -navigue

    [minifp] exécution de :

    firefox http://localhost:1234/minifp/EXEMPLE &

$gh>minifp --controle -sauve

    [minifp] Le 07 avril 2015 à 16 h22, sauvegarde des 2 tables :
            1. DEMO
            2. EXEMPLE
    dans le fichier minifp_sauve001_mysql.zip

```