



W1 - PHP

W-PHP-501

My_h5ai-bootstrap

Indexeur de fichiers pour serveurs web



My_h5ai-bootstrap

repository name: my_h5ai_bootstrap
repository rights: ramassage-tek
language: PHP



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

INTRODUCTION

L'objectif de ce bootstrap est de récupérer et d'organiser toute l'arborescence de dossiers/fichiers suivant un chemin donnée sous la forme de tableaux imbriqués.

Vous créerez donc un script quiinstanciera une classe puis lancera la récupération des fichiers du chemin passé en paramètre au constructeur de la classe mère. Enfin, une fois ces fichiers récupérés, ceux-ci seront affichés dans la console ou au sein d'une page HTML sous la forme de listes imbriqués.



PROJET

+ CLASSE MÈRE

Ce projet se concentrera autour d'une classe unique: **H5AI**.
Cette dernière possèdera deux variables privées:

- **_tree**: Le tableau de tableaux imbriqués contenant la liste de vos dossiers/fichiers. Il sera initialisé en tant que tableau vide.
- **_path**: Le chemin initial sur lequel votre programme se basera.

Ces deux variables étant privés, vous penserez à leur créer des getters sous cette forme:

- `getPath();`
- `getTree();`

De plus vous créerez un constructeur à cet classe qui initialisera le tableau **_tree** à `[]` et le **_path** à la variable passée en paramètre à votre constructeur.
`__construct($path);`



+ CRÉATION DE L'ARBRE

Vous rajouterez alors à cette classe une fonction *getFiles(fileList, parent)*.

Cette fonction aura pour but de remplir le tableau membre **_tree**.

Les deux arguments seront la liste de fichier que vous traitez actuellement et l'endroit du tableau où les raccrocher.

Un **parent** sera toujours un tableau.

La **fileList** sera la liste de fichiers/dossiers que nous devons rajouter au sein du parent.

Cette fonction est une fonction récursive afin de pouvoir gérer autant de dossiers imbriqués que possible.

Celui-ci sera organisé suivant quelques règles très simples:

- Si le fichier sur lequel nous sommes est un dossier, celui-ci deviendra un tableau qui contiendra les noms de tous les fichiers contenus dans ce dossier ainsi que des tableaux représentant les autres dossiers avec en clé le nom du dossier.
- Si le fichier sur lequel nous sommes est un fichier, nous assignons sa valeur au tableau parent passé en paramètre.
- Sinon un message d'erreur indique que le fichier n'existe pas/est invalide.

Exemple d'un tableau créé après l'appel de *getFiles()*:

```
array() => [
  pathPasséEnArgument => [
    NomDuFichier1,
    NomDuFichier2,
    NomDuDossier1 => [
      NomDuFichier1-1,
      NomDuDossier1-1 => [
        NomDuFichier1-1-1,
        NomDuFichier1-1-2
      ],
      NomDuFichier1-2
    ],
    NomDuDossier2 => [
      ],
      NomDuFichier3
    ]
  ]
]
```



+ AFFICHAGE DE L'ARBRE

Nous en venons enfin à la dernière étape de ce bootstrap. Le but, ici, va être d'afficher le tableau `_tree` dans le terminal ou la page html de votre choix.

Pour cela vous allez créer la fonction `printTree(tree);`

Cette fonction utilisera elle aussi de la récursivité.

Reprenons l'exemple du haut. Dans ce cas, notre fonction affichera:

```
Terminal
~/W-PHP-501> $ - ./my_h5ai_bootstrap pathPasséEnArgument
pathPasséEnArgument
- NomDuFichier1
- NomDuFichier2
- NomDuDossier1
  - NomDuFichier1-1
  - NomDuDossier1-1
    - NomDuFichier1-1-1
    - NomDuFichier1-1-2
  - NomDuFichier1-2
- NomDuDossier2
- NomDuFichier3
$ -
```