Classe abstraite

« Log »

METHODES DE LOG
log.php

Version de la doc : 1.1

METHODES STATIQUES

f ()

Crée un fichier nommé « $fileName.html » et y place le résultat de krumo(...$values)
krumo() est en résumé un var_dump() amélioré. D'autant plus que sur certains environnements, var_dump() peut afficher un résultat pas très human-readable.

```
$val = 'valeur_enorme';
 Log::f('mon_premier_log', $val);
// crée 'mon_premier_log.html' qui contient 'valeur_enorme'
```

append_f ()

Idem que f () mais conserve le contenu initial du fichier et ajoute à la fin : la date et l'heure du log ainsi que le nouveau contenu

```
Log::append_f('mon_premier_log' , 'nouveau_contenu_a_ajouter');
// ajoute 'nouveau_contenu_a_ajouter' à la fin de 'mon_premier_log.html'
// Si le fichier n'existe pas, il est créé
```

f1 ()

Idem que f () mais avec un fichier nommé 'f1.html'.

f2 ()

Idem que f1 () mais avec un fichier nommé 'f2.html'.

append_f1 ()

Idem que append_f () mais avec un fichier nommé 'append_f1.html'.

append_f2 ()

Idem que append_f1 () mais avec un fichier nommé 'append_f2.html'.

var_dump_f ()

Crée un fichier d'un nom donné et y place le résultat de var_dump(...$values)

$val = 'valeur_enorme';
Log::var_dump_f('mon_premier_log', $val);
// crée 'mon_premier_log.html' qui contient 'valeur_enorme'

var_dump_f1 ()

Crée un fichier nommé 'var_dump_f1.html' et y place le résultat de var_dump(...$values) en conservant le contenu initial du fichier

$val = 'toto';
Log::var_dump_f1($val);
// crée 'vdf1.html' qui contient 'toto'

var_dump_append_f ()

Idem que var_dump_f () mais conserve le contenu initial du fichier et ajoute à la fin la date et l'heure du log ainsi que le nouveau contenu

Log::var_dump_append_f('mon_premier_log' , 'nouveau_contenu_a_ajouter');
// ajoute 'nouveau_contenu_a_ajouter' à la fin de 'mon_premier_log.html'
// Si le fichier n'existait pas, il aurait été créé

printr_append_f ()

Idem que var_dump_append_f () mais insère l'équivalent de print_r () au lieu de var_dump ()

hacking_attempt ()

A utiliser pour logger une tentative de piratage qui aurait été détectée. Le log est mémorisé à la fin du fichier '$$___HACKING_ATTEMPTS___.html'.
La date et l'heure y sont enregistrés avec $_SESSION, $_POST, $_GET, debug_backtrace. Ce dernier contient également les arguments transmis à hacking_attempt().

Log:: hacking_attempt (« tentative d'insertion d'un script js »);

INSTALLATION SUR LE FRAMEWORK
développé Jean-Jacques Pagan @Jijou

1 / Editer le fichier de config .ini

Il s'agit du fichier nommé « config_$nom_du_projet.ini »
Ajouter la ligne 'PATH_LOGS' et 'PATH_VENDOR' sous 'PATH_FILES'.

En local, ce fichier est situé dans le dossier $wamp (ou autre plateforme) suivi de « /files/ »

```
...
PATH_FILES      = files/ap/HTML/
PATH_LOGS       = files/ap/logs/      ⇐  ligne à ajouter
PATH_VENDOR     = C:/ap/vendor/       ⇐  ligne à ajouter
...
```

« ap » peut être différent selon le nom du projet et la personne, tout comme 'C:/ap/' qui correspond au chemin du dossier du projet.

Dans l'exemple ci-dessus, les logs seront enregistrés dans « $dossier_du_projet/files/ap/logs/ »

2 / Profiter

Sur certains projets comme afpanier_v3, la classe est déjà intégrée au framework, vous pouvez l'utiliser dès maintenant.

Made in France

with ♥

#happy_coders