

CHANGELOG

Summary

Changes By Version

Version 4.0.0 : 13.03.2018

- Move `Template.class` into folder `Template` for composer autoloading PSR-0.
- Fix internal function declaration which generate error due to missing namespace.

Version 3.6.0 : 24.08.2017

- Redesign Commentaries style to respect PSR-2 standard.

Version 3.5.0 : 03.05.2017

- Ajout de la capacité à définir le modèle de composition des balises du moteur pour fonctionner avec différent frameworks (En l'occurrence, Pug ou `<!--` rompe la colorisation syntaxique)
 - Modèle par défaut :
- Correction de la méthode `set_var_delims` qui générée une erreur `E_NOTICE`

Version 3.4.0 : 16.04.2017

- Ajout d'un test d'existence pour ne pas générer d'erreur `E_NOTICE` lorsqu'on tente d'utiliser un jeu de donnée inexistant.
- Dans la phase de bufferisation avant rendu, dans le cas où un block est détecté, une instruction était inscrite, même pour la déclaration de block. Or la déclaration sert à bufferiser uniquement et non pas à être inclus.

Version 3.3.0 : 30.12.2016

- Mise à jour de la méthode `__construct` :
 - `__construct` : Suppression du test `gettype($_SESSION)` générant une erreur de niveau NOTICE.
 - Ajout d'un paramètre facultatif définissant le mode debug ou non.
- Mise à jour de la méthode `get_if_block_name` :
 - Suppression de la fonction `die` au profit de `throw_error`
- Sécurisation de la saisie des arguments :
 - `__construct`

- set_mail_recipients
- set_mail_sender
- set_mail_sender_name
- set_mail_subject
- set_output_name
- set_render_type
- set_utf8_read_treatment
- set_utf8_write_treatment
- Mise à jour de la methode add_history :
 - Simplification sur lenregistrement et l'émission de message d'erreur
 - On ajoute toujours une entrée dans l'historique
 - Si le mode debug est activé, envoie l'erreur par la suite
- Ajout d'une méthode permettant de modifier le mode debug :
 - Boolean set_debug_mode(Boolean \$debug_mode)
- Ajout d'une méthode pour définir une variable précisément :
 - Boolean set_var(String \$var_name, Mixed \$var_value)
- Ajout d'une méthode pour mettre à jour une masse de variable :
 - Boolean update_vars(Array \$vars)
- Ajout d'une méthode pour désallouer une ou plusieurs variables :
 - Boolean unset_vars(Mixed \$vars) (String | Array)
- Ajout d'une méthode pour inverser les valeurs boolean des variables spécifiées :
 - Boolean xor_vars(Mixed \$vars) (String | Array)
- Ajout d'une méthode statiques permettant de supprimer les blancs (CRLF & Tabulation) :
 - String strip_blank(String \$text)
- Ajout de méthodes statiques pour supprimer les commentaires du langage donnée :
 - cleanse_sql
 - cleanse_js

Version 3.2.0 : 29.05.2016

- Ajout d'un nouveau type de block déclarable : TEMPLATE
 - Permet la création d'un ensemble (simple et block) et de l'interpréter à la facon d'un modèle inclus mais interne
 - Instruction : BEGIN_DECLARE (TEMPLATE) AS NAME
 - Callable à l'aide de
- Introduction des operateurs WITH et EXTEND WITH
 - Permet l'utilisation combiné (WITH @Fusion recursive) avec un jeu de donnée spécifié
 - Permet l'extension de donnée (EXTEND WITH @Join recursive) avec le jeu de donnée spécifié
 - Permet l'utilisation de block anonyme
 - Si utilisé, l'operateur (EXTEND)? WITH est obligatoire, sinon facultatifs
 - Permet la retrocompatibilité

- Permet d'optimiser les jeux de donnée à envoyer au moteur
- Introduction de la méthode : `array_merge_index_recursive`
 - méthode de fusion de tableau récursive en faisant abstraction des index
- Syntaxe de WITH dans l'instruction WITH :
- Syntaxe de EXTEND WITH dans l'instruction WITH :
- Simplification de la balise de fermeture où le nom n'est plus nécessaire et devient optionnel
- Ajout d'une méthode de génération d'erreur "throw_error"
 - Gestion des erreurs plus explicite et soignée
- Renommage de `show_warnings` en `show_historic`
- Ajout de la méthode public `add_history`
- Suppression de `get_block_name` au profit de `read_block_instruction`

Version 3.1.0 : 04.12.2015

- Permettre l'utilisation des parenthèses, des crochets et accolades en tant que délimiteurs
 - Introduction d'un délimiteur de début et un délimiteur de fin

Version 3.0.0 : 21.11.2015

- Révision totale de `rendering_line` : 98% d'optimisation
 - Utilisation sur une app : 164 variable envoyée, 267 variable à remplacer - 13200 traitement effectué
 - A chaque line envoyé, ils parcourait toutes les données envoyées
 - Maintenant il recherche les variables, les cache, cherche si elles existent et si oui les remplace
 - 164 variables envoyées, 267 variables à remplacer - 267 traitements
- Révision de la méthode `remove_folder` pour qu'elle soit récursive
- Révision totale de `render()` et `prepare_buffers`
 - Facilite l'implémentation de nouvelles instructions
 - `Render()` et `prepare_buffers()` ont chacun un vrai rôle distinct et un comportement propre
 - Auparavant, elle avait le même principe de fonctionnement, mais des rôles légèrement différents
 - Beaucoup de répétitions de code entre elles
 - Prise en charge globale des imbrications sans développement supplémentaire pour les futures instructions
 - Plus de fichiers temporaires, mais plus aucun contenu répété
 - Méthodes impactées :
 - Supprimées :
 - `get_blocks_vars` : Plus de distinction entre variable simple et blocks
 - `get_php_code` : Dupliquet de `get_template_path`
 - `set_blocks_vars` : Plus de distinction entre variable simple et blocks
 - `render_code` : Dupliquet de `rendering_line`, sauf qu'elle faisait un return
 - Ajoutées :

- get_template_file_name : Renvois le modèle défini par set_template_file()
- control_buffer : Programme d'identification d'instruction et triggerer
- path_file_to_name : Convertir un path (folder/name) en name (folder.name)
- rendering_if : Programme de traitement des blocks IF
- rendering_php : Programme de traitement des blocks PHP
- store_buffer : Ecris dans le fichier final la ligne demandée
- Mise à jour :
 - prepare_buffer : Programme de dispatch des blocks en fichier temporaire
 - render : Programme de renderisation du document
 - rendering_block : Programme de traitement des blocks normaux
- Mise à jour & Renommées :
 - get_template_path ==> get_input_param : Lit les valeurs entre parenthèses dans les instruct°
 - rendering_line ==> render_buffer : Renvois le buffer renderisé, mais n'ecrit plus
- Implémentation de l'instruction de déclaration de block (BEGIN_DECLARE)
- Implémentation de l'instruction d'appel d'un block déclaré (USE)
 - Implémentation de l'instruction de déclaration de block (BEGIN_DECLARE)
- Implémentation de l'instruction d'appel d'un block déclaré (USE)
- Implémentation des cas ELSEIF et ELSE
 - Nouvelles Instructions :
 - Private :
 - Public :
 - <!-- IF (cdn) AS NAME -->
 - <!-- ELSEIF (cdn) -->
 - <!-- ELSE -->
 - <!-- ENDIF NAME -->
 - <!-- USE (nom_block_previously_declare_or_used) -->
 - <!-- BEGIN_DECLARE (BLOCK|PHP|IF->(%DNC%)) AS NAME -->
 - <!-- END_DECLARE NAME -->

Version 2.9.0 : 07.11.2015

- Permettre d'envoyer du texte à la place d'un template (soit l'un soit l'autre) :
 - Renommage de la méthode set_template_source en set_template_file
 - Ajout de deux flags :
 - \$_template_file_used AND \$_template_text_used
 - Ajout de deux fonctions de désallocation pour passer du mode de modèle "FICHER" a "TEXT"
 - unset_template_file AND unset_template_text
 - Ajout d'une méthode de création de fichier template temporaire avec assimilation en tant que source
 - set_text_as_file()
 - Ajout d'une méthode de récupération du text défini : get_template_text()
- Mettre en place un système pour qu'un bloc puisse appeler une variable simple (optimisation)

- Utilisation : %%VARIABLE%%
- Révision de la méthode render_line() :
 - Une première partie du fonctionnement est fixe sur les variables simple
 - Une seconde partie du fonctionnement est dynamique sur l'ensemble de variable indiqué
- Implémentation des block conditionnel
 - Révision de render() et rendering_block() pour gérer les blocks conditionnel
 - Création des fonctions get_block_name() et get_if_block_name()
 - Comportement identique à une simple variable lorsqu'il est inclus ds un block normal
 - Imbrication possible de block conditionnel
- Corrections diverses :
 - Intégration du flag render_env_exist = true directement dans la méthode make_render_env()
 - Pour le nettoyage, utilisation de SCRIPT_FILENAME au lieu de DOCUMENT_ROOT, plus compatible :
 - Compatible mutualisé OVH avec sous-domaine
 - Compatible Apach et NGNIX
 - Correction du traitement des blocks Imbriqués. Un block parent interprétait la balise de fin d'un block enfant. De ce fait la suite des codes étaient interprétés et insérés au block parent alors qu'il faisait partis du block enfant

Version 2.8.1 : 31.08.2015

- Correction du système de détermination du nom de dossier temporaire :
 - \$_REQUEST['PHPSESSID'] >>> session_id()

version 2.8.0 : 27.07.2015

- Correction du comportement de la methode de nettoyage de l'environnement de rendering
 - Methode cleansing_render_env() appelée par __destruct fonctionnait à la racine serveur
 - Création de la méthode cleansing_render_env_root() pour travailler avec \$_SERVER['DOCUMENT_ROOT']

Version 2.7.0 : 30.03.2015

- Implémentation d'une variable superglobale personnalisée pour enoyer des données dans les codes PHP des templates : \$_PHP
- Création d'une fonction de traitement des instructions du moteur en cas d'utilisation de variables
 - Methode créée pour traiter les instruction : render_code(\$buffer)
 - Permettre d'utiliser des variables pour les INCLUDE_TEMPLATE
 - Permettre d'utiliser des variables dans les block PHP BEGIN_PHP
- Optimisation de render_line : compter le nombre de variable et une fois remplacée :
 - Interrompre la boucle à l'aide d'un break;

Version 2.6.1 : 29.03.2015

- Modification des sortie de la méthode render pour cascader avec display et get_render_content sous la forme de \$moteur->render()->display()

Version 2.6.0 : 27.03.2015

- Permettre l'utilisation de code PHP dans les templates
- Permet l'utilisation de template inclus pour du BEGIN_BLOCK
 - Création de la méthode get_template_path
- Gérer le multi repository sur output_directory dans un cas render_type perma
 - Récupération par argument des repository
 - Dépot se fait par copy à l'aide d'une boucle sur l'array stockant les repository
 - Suppression de tout ce qui est attrait à _output_file_name (obsolete)
 - Suppression de tout ce qui est attrait à _output_directory > > > _output_directories

Version 2.5.4. : 14.03.2015

- Correction de l'environnement de travail temporaire lorsqu'on change
 - De modèle
 - De dossier de dépôt
 - De dossier temporaire
- Correction des methodes get_render_content() et display() - Peut importe le render_type, le constructeur PHP __destruct déclenche la purge de l'environnement de travail.
 - Pas besoin de déclencher un nettoyage apres l'execution de ces deux methodes
 - Permet d'executer display(), get_render_content() et sendMail() à la suite meme en render_type temporary.

Version 2.5.3 : 07.03.2015

- Création d'une methode pour demander de conserver les fichiers temporaire : set_keep_temp_file
- La définition du délimiteur n'est plus obligatoire. La valeur par défaut est "%"
 - Reduit le nombre d'instruction de configuration du moteur à deux ligne seulement

Version 2.5.2 : 05.3.2015

- Implémentation de la gestion d'imbrication de template (recursive)
 - Modification des methodes open_template_file() et close_template_file()
 - Modification des methodes render(), prepare_buffer()
 - En paramètre, est spécifié le modele sur lequel on travail
 - *Permet l'imbrication de template à x niveau, mais l'inclusion de template ne fonction pas dans un block

version 2.5.1 : 05.03.2015

- Révision complète de la mécanique des fichiers temporaires et du mode de rendu
 - Modification de la methode close_output_file() en close_temporary_render_file

- Modification de la methode open_output_file() en open_temporary_render_file
- Modification de la methode make_temporary_directory() en make_render_env()

Version 2.4.0

- La propriété _output_directory à la valeur par défaut : . faisant référence au dossier executant la classe
- Ajout d'une méthode pour obtenir le contenu du rendu : get_render_content()
 - get_render_content() retourne une chaine alors que display() l'affiche directement
- Permettre de faire des rendu temporaire et permanent
 - Ajout d'une methode pour définir le type de rendu attendu
 - Ajout d'une methode pour purger et supprimer un dossier : remove_folder (non recursif)
 - Si Permanent, utiliser le render repository spécifié uniquement
 - Si Temporaire, créer un dossier temp dans le render repository spécifié
 - Mise à jour de display() : a la fin, appel de remove_folder si render_type = temporary
 - Remplacement de remove_temporary_folder par remove_folder qui est globale car path spécifié

version 2.3.0

- Ajout de la méthode get_blocks_vars -> retourne le tableau pour manipulation si besoin
- Edit de la méthode get_vars -> retourne le tableau pour manipulation si besoin
- Intégration de la notion de chemin absolue pour les fichiers utilisé par la methode (help)
- Ajouter un boolean de sortie pour permet des tests de succes lors de l'utilisation de la classe
- Initialisation de \$_vars et \$_blocks_vars pour prevenir des erreurs dans la fonction foreach
- Correction de la methode __construct qui n'utilisait pas \$_REQUEST[PHPSESSID]
- Correction de la méthode remove_temporary_directory qui ne pouvait supprimer le dossier
- Ajout de la méthode help
- Amélioration de show_warnings

Version 2.2.0 : 30.10.2014

- Specification de l'encodage d'écriture du rendu (encode, decode, none(default)) {facultatif}
- Specification de l'encodage de lecture du rendu (encode, decode, none(default)) {facultatif}
- Specification du chemin du dossier temporaire
- Creation de warnings et un afficheur des erreurs rencontrer lors de la génération du rapport

Version 2.1.0 :

- Suppression des warnings et die des methodes make_ et remove_ temporary_directory

Version 2.0.0 :

- Implémentation de l'imbrication des blocs
- Intégration de la fonction externe sendMail

Version 1.0.0

- Première release