

Classe SilentMail

V2.1.0

Table des matières

1. Contenu	3
2. Dépendances	3
3. Définition	3
4. Constantes	3
5. Méthodes	4
Méthodes statiques	4
getMessage	4
Méthodes classiques	4
__construct	4
setFrom	5
addTo	5
addCC	6
addBCC	6
clearAllRecipients	6
setSubject	7
setBody	7
setMode	7
setCharSet	8
send	8
getErreurNum	9
getErreurMessage	9

Classe SilentMail

1. Contenu

Ce document à pour but d'expliquer le fonctionnement de la classe **SilentMail** de l'outil **UniversalWeb**.

Pour obtenir la version de cette classe, tapez **echo SilentMail::VERSION**.

2. Dépendances

La classe est indépendante de l'outil UniversalWeb. Cependant, afin d'internationaliser les libellés de comptes-rendus d'erreurs, on pourra reproduire le contenu de la méthode publique interne `getLib` dans les fichiers de langue d'UniversalWeb.

3. Définition

La classe **SilentMail** est destinée à l'envoi de courrier électronique grâce a des méthodes simples. Voici un exemple de l'utilisation de la classe. Ici une fonction permet d'envoyer un mail à l'administrateur de application chaque fois qu'un client envoi un message.

```
function email_contactClient($infos)
{
    $mail = new SilentMail();
    $mail->setMode('plain');
    $mail->setFrom($infos['email']);
    $mail->addTo('serviceclient@domaine.fr');
    $mail->setSubject('Nouveau message client');
    $message = array();
    $message[] = ucfirst($infos['prenom']).' '.ucfirst($infos['nom']).' vient de déposer un message :';
    $message[] = '';
    $message[] = 'Date: '.$infos['date'];
    $message[] = 'eMail : '.$infos['email'];
    $message[] = 'Référence : '.$infos['reference'];
    $message[] = 'Message : '.$infos['message'];
    $mail->setBody($message);
    return $mail->send();
}

$donnees['prenom'] = 'Pierre';
$donnees['nom'] = 'Durand';
$donnees['email'] = 'pierre.durand@domaine.fr';
$donnees['reference'] = '31A2112';
$donnees['message'] = 'Bonjour. Pourriez-vous s'il vous plait me dire si cet article existe en mauve ?';
$donnees['date'] = date();
$res = email_contactClient($donnees);
if ($res != SilentMail::UWSM_SUCCESS) {
    echo 'Erreur n°'.$res.' Rencontrée. Aucun mail n'a été envoyé';
}
```

4. Constantes

Liste des constantes proposées par la classe. Il s'agit des constantes correspondantes aux codes d'erreur en retour de la méthode **send**.

Nom de la constante	Valeur
UWSM_TEST	Message de test : aucun envoi (sauf dans fichier email.txt)

Classe SilentMail

UWSM_SUCCESS	Aucune erreur
UWSM_NO_SENDER	Pas d'expéditeur déclaré
UWSM_BAD_SENDER	Adresse email de l'expéditeur erronée (ou mal formée)
UWSM_NO_OBJECT	Aucun objet déclaré
UWSM_NO_RECIPIENT	Pas de destinataire déclaré
UWSM_TOO_MANY_RECIPIENTS	Trop de destinataires déclarés
UWSM_BAD_RECIPIENT	Adresse email destinataire erronée (ou mal formée)
UWSM_ERROR	Autre erreur

5. Méthodes

Méthodes statiques

getMessage

V2.0.0. Permet de récupérer un message d'erreur en clair selon le code d'erreur passé au paramètre \$code.

Description

```
string getMessage(integer $code);
```

Liste des paramètres

code : code d'erreur à étudier.

Valeurs de retour

Le texte en clair du message. Si la classe est utilisée en association avec Universalweb et ses fichiers de langues, alors le message apparaîtra dans la langue sélectionnée en cours. Dans le cas contraire le message sera affiché en français.

Exemple

```
$leMessage = SilentMail::getMessage(1);
```

Méthodes classiques

__construct

Constructeur de la classe.

Classe SilentMail

Liste des paramètres

Aucun.

Exemple

```
$leMail = new SilentMail();
```

setFrom

Positionne l'email de l'expéditeur. Il doit s'agir d'une email valide au sens syntaxique.

Description

```
setFrom(string $email [, string $name = null]);
```

Liste des paramètres

email : email de l'expéditeur.

name : paramètre optionnel. Nom de l'expéditeur correspondant à l'email tel qu'il doit apparaître dans le message. Si le paramètre optionnel **name** est renseigné, le mail sera envoyé précédé du nom de l'expéditeur sous la forme **nom <email>**. Si le champ **name** n'est pas renseigné, le mail sera envoyé seulement avec l'email de l'expéditeur.

Si aucun expéditeur n'a été spécifié, une erreur surviendra lors de l'envoi du mail.

Valeurs de retour

Aucune.

Exemple

```
$mail->setFrom('pierre.durand@domaine.fr', 'Pierre Durand');  
//ou  
$mail->setFrom('pierre.durand@domaine.fr');
```

addTo

Ajoute un destinataire. Il doit s'agir d'une email valide au sens syntaxique.

Description

```
addTo(string $email [, string $name = null]);
```

Liste des paramètres

email : email du destinataire.

name : paramètre optionnel. Nom du destinataire correspondant à l'email tel qu'il doit apparaître dans le message. Si le paramètre optionnel **name** est renseigné, le mail sera envoyé précédé du nom du destinataire sous la forme **nom <email>**. Si le champ **name** n'est pas renseigné, le mail sera envoyé seulement avec l'email du destinataire.

L'oubli du destinataire dans la construction du mail provoque une erreur lors de l'envoi du mail. Une email de destinataire mal formée provoque également une erreur. La méthode limite volontairement le nombre de destinataires à 3. Au-delà une erreur est levée.

Valeurs de retour

Aucune.

Exemple

```
$mail->addTo('pierre.durand@domaine.fr', 'Pierre Durand');  
//ou  
$mail->addTo('pierre.durand@domaine.fr');
```

addCC

Ajoute un destinataire en copie carbone. Il doit s'agir d'une email valide au sens syntaxique.

Description

```
addCC(string $email [, string $name = null]);
```

Liste des paramètres

email : email du destinataire.

name : paramètre optionnel. Nom du destinataire correspondant à l'email tel qu'il doit apparaître dans le message. Si le paramètre optionnel **name** est renseigné, le mail sera envoyé précédé du nom du destinataire sous la forme **nom <email>**. Si le champ **name** n'est pas renseigné, le mail sera envoyé seulement avec l'email du destinataire.

Valeurs de retour

Aucune.

Exemple

```
$mail->addCC('pierre.durand@domaine.fr', 'Pierre Durand');  
//ou  
$mail->addCC('pierre.durand@domaine.fr');
```

addBCC

Ajoute un destinataire en copie carbone cachée. Il doit s'agir d'une email valide au sens syntaxique.

Description

```
addBCC(string $email [, string $name = null]);
```

Liste des paramètres

email : email du destinataire.

name : paramètre optionnel. Nom du destinataire correspondant à l'email tel qu'il doit apparaître dans le message. Si le paramètre optionnel **name** est renseigné, le mail sera envoyé précédé du nom du destinataire sous la forme **nom <email>**. Si le champ **name** n'est pas renseigné, le mail sera envoyé seulement avec l'email du destinataire.

Valeurs de retour

Aucune.

Exemple

```
$mail->addBCC('pierre.durand@domaine.fr', 'Pierre Durand');  
//ou  
$mail->addBCC('pierre.durand@domaine.fr');
```

clearAllRecipients

Réinitialise (supprime) tous les destinataires du message : destinataires principaux (ajoutés avec **addTo**), en copie carbone (ajoutés avec **addCC**) et en copie carbone cachée (ajoutés avec **addBCC**).

Description

```
clearAllRecipients();
```

Liste des paramètres

Aucun.

Valeurs de retour

Aucune.

Exemple

```
$mail->clearAllRecipients();
```

setSubject

Ajoute un sujet au message.

Description

```
setSubject (string $subject);
```

Liste des paramètres

subject : chaîne de caractère contenant le sujet du message.

Valeurs de retour

Aucune.

Exemple

```
$mail->setSubject('A sujet de votre commande');
```

setBody

Ajoute le corps du message. Le contenu du message doit être présenté sous forme de tableau avec une entrée dans la tableau pour chaque ligne à écrire.

Description

```
setBody (array $message);
```

Liste des paramètres

message : tableau de chaîne de caractère contenant le corps du message.

Valeurs de retour

Aucune.

Exemple

```
$mail = new SilentMail();  
$message = array();  
$message[] = 'Merci pour votre message';  
$message[] = '';  
$message[] = 'Nous allons y répondre dans les meilleurs délais.';  
$message[] = 'En attendant vous pouvez nous retrouver sur le site.';  
$mail->setBody($message);
```

setMode

Définit le mode d'envoi du message, texte brut ou en HTML.

Description

```
setMode (string $mode);
```

Liste des paramètres

mode : deux valeurs possible :

Classe SilentMail

- **plain** : le message sera envoyé en mode texte (c'est le mode par défaut car offrant le plus de réussite)
- **html** : le message sera envoyé encapsulé dans du HTML (qu'il vous faudra écrire).

Valeurs de retour

Aucune.

Exemple

```
$mail = new SilentMail();  
$mail->setMode('plain');
```

setCharset

Spécifie le codage des caractères pour le message.

Description

```
setCharset(string $charset);
```

Liste des paramètres

charset : encodage des caractères (par défaut 'UTF-8'). Exemples d'autres valeurs possibles : ISO-8859-1, Windows-1252, CP1252, etc.

Valeurs de retour

Aucune.

Exemple

```
$mail = new SilentMail();  
$mail->setCharset('ISO-8859-1');
```

send

Envoie le message construit.

Description

```
integer send([string $trace]);
```

Liste des paramètres

trace : paramètre optionnel à choisir parmi **off** (par défaut), **on** et **test**.

- **test** : Mis en place depuis la version 2.0.0, si le paramètre trace vaut test, alors le mail sera généré à l'intérieur du fichier texte `email.txt` sans être envoyé par le serveur `smtp`. Idéal pour simuler vos messages avant mise en production.
- **on** : le mail est envoyé après en avoir enregistré le contenu dans le fichier texte `email.txt`. Ceci peut s'avérer utile dans le cas où l'envoi du mail ne fonctionnerait pas (par exemple suite à un mauvais paramétrage `smtp` de votre serveur) afin de vérifier la qualité de son contenu, ou bien à de simples fins de test.
- **off** : le mail est envoyé sans option (valeur par défaut).

Valeurs de retour

Code de retour de bonne exécution de l'envoi. Le code est à comparer aux constantes décrites au chapitre précédent. Si le paramètre `$trace` vaut 'test', le code de retour `UWSM_TEST` est systématiquement renvoyé.

Exemple

```
$retour = $mail->send();
if ($retour != SilentMail::UWSM_SUCCESS) {
    echo 'Erreur n°'.$retour.' Rencontrée. Aucun mail n'a été envoyé';
}
```

getErreurNum

Renvoie le dernier numéro d'erreur rencontré à la suite d'un **send**.

Description

integer getErreurNum();

Liste des paramètres

Aucun.

Valeurs de retour

Code de retour de bonne exécution de l'envoi. Le code est à comparer aux constantes décrites au chapitre précédent.

Attention si l'appel à la méthode est réalisé alors qu'un **send** n'a été effectué, la méthode **getErreurNum** renvoie **null**.

Exemple

```
$retour = $mail->send();
if ($mail->getErreurNum() != SilentMail::UWSM_SUCCESS) {
    echo 'Erreur n°'.$retour.' Rencontrée. Aucun mail n'a été envoyé';
}
```

getErreurMessage

Renvoie le message en clair de la dernière erreur rencontrée à la suite d'un **send**.

Description

string getErreurMessage();

Liste des paramètres

Aucun.

Valeurs de retour

Message en clair de l'erreur rencontré lors du dernier **send**.

Attention si l'appel à la méthode est réalisé alors qu'aucun **send** n'a été effectué, la méthode **getErreurMessage** renvoie **null**. Si aucune erreur n'a été rencontrée, la méthode renverra également la valeur **null**.

Exemple

```
$retour = $mail->send();
if ($mail->getErreurNum() != SilentMail::UWSM_SUCCESS) {
    echo 'Erreur n°'.$retour.' ('.$mail->getErreurMessage().')';
}
```