

Classe User

Table des matières

1. Contenu	4
2. Définition.....	4
3. Constantes	4
4. Méthodes publiques	5
Getters.....	5
getId.....	5
getNom	5
getPrenom.....	5
getNomPrenom	6
getPrenomNom	6
getEmail	6
getPassword	6
getLangue	7
getProfil	7
getTesteur.....	7
getDateCreation.....	8
getDernierAcces.....	8
getActionDemandee	8
getCodeValidation	9
getActive	9
getAutolog.....	10
getIp.....	10
getNotesPrivees	10
Setters	12
setId	12
setNom	12
setPrenom	12
setEmail	13
setPassword	13
setLangue.....	13
setProfil.....	14
setTesteur	14
setDateCreation	14
setDernierAcces	15
setActionDemandee	15
setCodeValidation	15
setActive	15

Classe User

setAutolog	16
setIp	16
setNotesPrivees	16
Méthodes statiques	17
isValidUser	17
isGrantedUser	17
isAdmin	18
getPrenomNomOrId	18
Méthodes publiques	19
majDernierAcces	19
chargeUser	19

1. Contenu

Ce document a pour but d'expliquer et de définir l'usage de la classe **User** de l'outil **UniversalWeb**.

2. Définition

Toute application a besoin à un moment ou à un autre de faire référence à un ou plusieurs utilisateurs. Ce peut être des clients de votre site web, de simples visiteurs ou encore les administrateurs du site. Pour réaliser cette fonction, UniversalWeb s'appuie sur la classe `User` qui elle-même s'appuie sur la table `_users` de la base de données dans laquelle sont stockées les informations de vos utilisateurs¹. On trouve son code dans le fichier `User.class.php`.

La classe `User` est bien entendu – et si besoin – dérivable pour votre propre application. Dans sa forme actuelle elle intègre un certain nombre d'informations sur l'utilisateur. Voir le document « Description de la base de données » qui décrit le MCD en place et chaque table de la base de données.

3. Constantes

Dans son code le développeur pourra – selon la méthode appelée –, faire références aux constantes suivante :

Nom de la constante	Valeur
ERREUR_AUCUNE	Aucune erreur
ERREUR_ANNUAIRE	Erreur annuaire LDAP
ERREUR_ANNUAIRE_USER_UNKNOWN	Utilisateur non trouvé dans l'annuaire LDAP
ERREUR_USER_UNKNOWN	Utilisateur non trouvé dans la base
ERREUR_USER_BADPWD	Mauvais mot de passe
ERREUR_USER_INACTIVE	Utilisateur désactivé

¹ **Attention** : Ne pas confondre « utilisateur de votre application » et « utilisateur connecté à votre application ». Dans ce deuxième cas, il ne faut pas utiliser la classe `User` mais la classe `Login` de UniversalWeb pour y faire référence. Le détail de cette classe fait l'objet d'un autre document. Merci de vous y reporter.

4. Méthodes

Getters

getId

Getter. Obtenir l'identifiant unique de l'utilisateur.

Description

```
string getId();
```

Liste des paramètres

Aucun

Valeurs de retour

Identifiant de l'utilisateur (100 caractères maximum).

Exemple

```
$user_id = $user->getId();
```

getNom

Getter. Obtenir le nom de l'utilisateur.

Description

```
string getNom();
```

Liste des paramètres

Aucun

Valeurs de retour

Nom de l'utilisateur (100 caractères maximum).

Exemple

```
$user_name = $user->getNom();
```

getPrenom

Getter. Obtenir le prénom de l'utilisateur.

Description

```
string getPrenom();
```

Liste des paramètres

Aucun

Valeurs de retour

Prénom de l'utilisateur (100 caractères maximum).

Exemple

Classe User

```
$user_firstname = $user->getPrenom();
```

getNomPrenom

Getter. Obtenir le nom et le prénom de l'utilisateur.

Description

```
string getNomPrenom();
```

Liste des paramètres

Aucun

Valeurs de retour

Nom et prénom de l'utilisateur (200 caractères maximum).

Exemple

```
$user_name = $user->getNomPrenom();
```

getPrenomNom

Getter. Obtenir le prénom et le nom de l'utilisateur.

Description

```
string getPrenomNom();
```

Liste des paramètres

Aucun

Valeurs de retour

Prénom et nom de l'utilisateur (200 caractères maximum).

Exemple

```
$user_name = $user->getPrenomNom();
```

getEmail

Getter. Obtenir l'adresse eMail de l'utilisateur

Description

```
string getEmail();
```

Liste des paramètres

Aucun

Valeurs de retour

Email de l'utilisateur (255 caractères max)

Exemple

```
$user_email = $user->getEmail();
```

getPassword

Classe User

Getter. Obtenir le mot de passe crypté SHA1 de l'utilisateur.

Description

`string getPassword()` ;

Liste des paramètres

Aucun.

Valeurs de retour

Mot de passe crypté SHA1 de l'utilisateur.

Exemple

```
$user_pass = $user->getPassword();
```

getLangue

Getter. Obtenir la langue d'usage de l'utilisateur.

Description

`string getLangue()` ;

Liste des paramètres

Aucun.

Valeurs de retour

Code ISO 3166 alpha2 du pays correspondant à la langue utilisée. Par défaut, la langue est 'fr'.

Exemple

```
$user_langue = $user->getLangue();
```

getProfil

Getter. Obtenir le profil de l'utilisateur.

Description

`integer getProfil()` ;

Liste des paramètres

Aucun.

Valeurs de retour

Identifiant du profil de l'utilisateur.

Exemple

```
$user_profil = $user->getProfil();
```

getTesteur

Getter. Obtenir le statut « testeur » de l'utilisateur.

Description

`integer getTesteur()` ;

Classe User

Liste des paramètres

Aucun.

Valeurs de retour

1 l'utilisateur est un testeur, 0 dans le cas contraire.

Exemple

```
$user_testeur = $user->getTesteur();
```

Notes

Le retour de ce getter peut être comparé via le booléen true / false. **Attention** toutefois à l'opérateur de comparaison, employé :

```
if ($user->getTesteur() == true) renvoie true.  
if ($user->getTesteur() === true) renvoie false.
```

getDateCreation

Getter. Obtenir la date de création de l'utilisateur.

Description

```
datetime getDateCreation();
```

Liste des paramètres

Aucun.

Valeurs de retour

Datetime au format MySQL (Y-m-d H:i:s) de la date de création du compte de l'utilisateur.

Exemple

```
$user_creation = $user->getDateTime();
```

getDernierAcces

Getter. Obtenir le datetime du dernier accès de l'utilisateur à l'application.

Description

```
datetime getDernierAcces();
```

Liste des paramètres

Aucun.

Valeurs de retour

Datetime de dernier accès au format MySQL (Y-m-d H:i:s).

Exemple

```
$user_dernierAcces = $user->getDernierAcces();
```

getActionDemandee

Getter. Obtenir l'action demandées par l'utilisateur.

Description

Classe User

```
integer getActionDemandee();
```

Liste des paramètres

Aucun.

Valeurs de retour

Un entier qui correspond à une action particulière à définir par l'utilisateur. L'information peut être par exemple utilisée pour les demandes « mot de passe oublié ».

Exemple

```
$user_action = $user->getActionDemande();
```

getCodeValidation

Getter. Obtenir le code de validation.

Description

```
string getCodeValidation();
```

Liste des paramètres

Aucun.

Valeurs de retour

Une chaîne de caractères (30 max) qui peut être utilisée conjointement avec la méthode `getActionDemandee()` pour gérer des actions utilisateurs (par exemple pour les demandes « mot de passe oublié »).

Exemple

```
$user_code = $user->getCodeValidation();
```

getActive

Getter. Obtenir le statut « activé » de l'utilisateur.

Description

```
integer getActive();
```

Liste des paramètres

Aucun.

Valeurs de retour

1 : l'utilisateur est activé.

0 : l'utilisateur est désactivé.

Exemple

```
$user_active = $user->getActive();
```

Notes

Le retour de ce getter peut être comparé via le booléen `true` / `false`. **Attention** toutefois à l'opérateur de comparaison, employé :

```
if ($user->getActive() == true) renvoie true.
```

```
if ($user->getActive() === true) renvoie false.
```

getAutolog

Getter. Obtenir le statut « autolog » de l'utilisateur.

Description

`integer` `getAutolog()` ;

Liste des paramètres

Aucun.

Valeurs de retour

1 : l'auto-login est activé.

0 : l'auto-login n'est pas activé.

Exemple

```
$user_autolog = $user->getAutolog();
```

Notes

L'auto-login est un mécanisme d'auto reconnexion pour les utilisateurs. Il n'est pas mis en œuvre dans le squelette UniversalWeb mais le développeur pourra utiliser cette méthode pour développer la fonctionnalité à loisirs.

getIp

Getter. Obtenir l'adresse IP de l'utilisateur lors de sa dernière connexion.

Description

`string` `getIp()` ;

Liste des paramètres

Aucun.

Valeurs de retour

IP (V4 ou V6) de l'utilisateur connecté. 39 caractères max.

Exemple

```
$user_ip = $user->getIp();
```

getNotesPrivees

Getter. Obtenir les annotations ou commentaires à propos de l'utilisateur et réservées à l'administrateur.

Description

`string` `getNotesPrivees()` ;

Liste des paramètres

Aucun.

Valeurs de retour

Chaine de caractères.

Exemple

Classe User

```
$user_notes = $user->getNotesPrivees();
```

Setters

Attention, les setters modifient seulement le contenu de l'objet et n'inscrivent aucune information en base de données.

setId

Setter. Positionne l'identifiant de l'utilisateur.

Description

```
setId(string $id);
```

Liste des paramètres

id : identifiant à positionner (100 caractères max).

Valeurs de retour

Aucun

Exemple

```
$user->setId('thx1138');
```

setNom

Setter. Positionne le nom l'utilisateur.

Description

```
setNom(string $nom);
```

Liste des paramètres

nom : chaîne de caractères à positionner (100 caractères max).

Valeurs de retour

Aucun

Exemple

```
$user->setNom('Lucas');
```

setPrenom

Setter. Positionne le prénom de l'utilisateur.

Description

```
setPrenom(string $prenom);
```

Liste des paramètres

prenom : chaîne de caractères à positionner (100 caractères max).

Valeurs de retour

Aucun

Exemple

```
$user->setPrenom('George');
```

setEmail

Setter. Positionne l'adresse email de l'utilisateur.

Description

```
setEmail(string $email);
```

Liste des paramètres

email : chaîne de caractères à positionner (255 caractères max).

Valeurs de retour

Aucun

Exemple

```
$user->setEmail('george.lucas@lucasfilm.com');
```

setPassword

Setter. Positionne le mot de passe de l'utilisateur.

Description

```
setPassword(string $mdp);
```

Liste des paramètres

mdp : le mot de passe (40 caractères max) à positionner pour l'utilisateur.

Valeurs de retour

Aucun

Exemple

```
$user->setPassword('jarjar');
```

Notes

Le mot de passe est à saisir sans cryptage (SHA1), tel que fourni par l'utilisateur.

setLangue

Setter. Positionne la langue de l'utilisateur.

Description

```
setLangue(string $langue);
```

Liste des paramètres

langue : Code ISO 3166-1 alpha2.

Valeurs de retour

Aucun

Exemple

```
$user->setLangue('fr');
```

setProfil

Setter. Positionne le profil utilisateur de l'utilisateur.

Description

```
setProfil(integer $profil);
```

Liste des paramètres

profil : Identifiant du profil utilisateur.

Valeurs de retour

Aucun

Exemple

```
$user->setProfil(2);
```

setTesteur

Setter. Positionne le statut « testeur » de l'utilisateur.

Description

```
setTesteur(integer $statut);
```

Liste des paramètres

statut : Entier représentant le statut (1 testeur / 0 non testeur).

Valeurs de retour

Aucun

Exemple

```
$user->setTesteur(0);
```

setDateCreation

Setter. Positionne la date de création de l'utilisateur.

Description

```
setDateCreation(datetime $date);
```

Liste des paramètres

date : datetime de création de l'utilisateur.

Valeurs de retour

Aucun

Exemple

```
$user->setDateCreation(date());
```

Notes

Cette méthode utilisée en interne par l'objet n'a que peu d'utilité pour le développeur puisque c'est SQL qui auto-enregistre la date de création de l'utilisateur dans la base de données.

setDernierAcces

Setter. Positionne la date de dernier accès de l'utilisateur à l'application.

Description

```
setDernierAcces(datetime $date);
```

Liste des paramètres

date : datetime de dernier accès à l'application.

Valeurs de retour

Aucun

Exemple

```
$user->setDateCreation(date());
```

setActionDemandee

Setter. Positionne l'action demandée par l'utilisateur.

Description

```
setActionDemandee(integer $action);
```

Liste des paramètres

action : entier définissant l'action à considérer.

Valeurs de retour

Aucun

Exemple

```
$user->setActionDemandee(_MOT_DE_PASSE_OUBLIE_);
```

setCodeValidation

Setter. Positionne le code de validation en relation avec l'action demandée par l'utilisateur.

Description

```
setCodevalidation(string $code);
```

Liste des paramètres

code : Chaîne de caractères (30 maximum).

Valeurs de retour

Aucun

Exemple

```
$user->setCodeValidation('af14e56dd1sq6zxs63sdddddz662');
```

setActive

Setter. Positionne le statut « activé » de l'utilisateur.

Classe User

Description

```
setActive(integer $statut);
```

Liste des paramètres

statut : 1 utilisateur activé / 0 utilisateur non activé.

Valeurs de retour

Aucun

Exemple

```
$user->setActive(1);
```

setAutolog

Setter. Positionne le flag « autolog » pour l'utilisateur.

Description

```
setAutolog(integer $flag);
```

Liste des paramètres

flag : 1 utilisateur être logué automatiquement / 0 utilisateur non activé.

Valeurs de retour

Aucun

Exemple

```
$user->setActive(1);
```

setIp

Setter. Positionne l'adresse IP pour l'utilisateur.

Description

```
setIP(string $ip);
```

Liste des paramètres

ip : adresse IP de l'utilisateur (39 caractères max).

Valeurs de retour

Aucune

Exemple

```
$user->setIP('10.20.30.20');
```

Notes

Prend en compte les adresse IPV4 et IPV6.

setNotesPrivees

Setter. Positionne les notes privées pour l'utilisateur. Pour rappel le concept de notes privées sont les notes de l'administrateur à propos de l'utilisateur (le client) que seul lui peut voir.

Description

Classe User

```
setNotesPrivees(string $notes);
```

Liste des paramètres

notes : chaîne de caractères sans limite.

Valeurs de retour

Aucune

Exemple

```
$user->setNotesPrivees('Ce client paye souvent en retard ...');
```

Méthodes statiques

isValidUser

Méthode statique : vérifie l'existence d'un utilisateur dans la base de données. Retourne le profil de l'utilisateur si celui-ci existe ou un code d'erreur dans le cas contraire.

Description

```
integer isValidUser(string $id);
```

Liste des paramètres

id : identifiant de l'utilisateur à interroger.

Valeurs de retour

Retourne le profil de l'utilisateur si celui-ci est connu de la base de données ou le code d'erreur **User::ERREUR_USER_UNKNOWN** sinon.

Exemple

```
$profil = User::isValidUser('thx1138');  
if ($profil != User::ERREUR_USER_UNKNOWN) {  
    echo 'cet utilisateur existe';  
}
```

Notes

Cette méthode est statique, ce qui signifie qu'elle n'a pas besoin d'être instanciée sous forme d'objet pour être utilisée et permet d'interroger la base de données à la volée sur l'existence ou non d'un utilisateur.

isGrantedUser

Méthode statique : détermine l'autorisation d'un utilisateur à entrer dans l'application. Il faut pour ceci que le couple login / mot de passe soit valide et que l'utilisateur soit activé.

Description

```
integer isGrantedUser(string $id, string $pwd);
```

vérifie que l'utilisateur ayant l'identifiant `$id` ait bien le mot de passe `$pwd` et qu'il soit activé.

Liste des paramètres

id : identifiant de l'utilisateur qui souhaite entrer.

pwd : mot de passe saisi par l'utilisateur pour entrer.

Classe User

Valeurs de retour

Plusieurs valeurs possibles :

User::**ERREUR_USER_UNKNOWN** : utilisateur inconnu.

User::**ERREUR_USER_BADPWD** : mauvais mot de passe pour l'utilisateur \$id.

User::**ERREUR_USER_INACTIVE** : utilisateur connu et mot de passe ok mais compte désactivé.

User::**ERREUR_AUCUNE** : aucune erreur, l'utilisateur peut entrer.

Exemple

```
$retour = User::isGrantedUser('thx1138', 'monmotdepasse');
```

Notes

Cette méthode est statique, ce qui signifie qu'elle n'a pas besoin d'être instanciée sous forme d'objet pour être utilisée et permet d'interroger la base de données à la volée.

isAdmin

Méthode statique : détermine si un utilisateur possède un profil d'administration.

Description

```
boolean isAdmin(string $id);
```

Liste des paramètres

id : identifiant de l'utilisateur à tester.

Valeurs de retour

false : l'utilisateur est inconnu ou ne possède pas le profil administrateur.

true : l'utilisateur existe et possède le profil administrateur.

Exemple

```
$retour = User::isAdmin('thx1138');
```

Notes

Cette méthode est statique, ce qui signifie qu'elle n'a pas besoin d'être instanciée sous forme d'objet pour être utilisée et permet d'interroger la base de données à la volée.

getPrenomNomOrId

Méthode statique : renvoie le prénom et le nom d'un utilisateur particulier. A défaut renvoie l'identifiant passé en paramètre.

Description

```
string getPrenomNomOrId(string $id);
```

Liste des paramètres

id : identifiant de l'utilisateur recherché.

Valeurs de retour

Le prénom et le nom de l'utilisateur (200 caractères maximum) si trouvé. La valeur de id sinon.

Exemple

```
$user_name = $user->getPrenomNomOrId('thx1138');
```

Renvoie nom prénom de l'utilisateur 'thx1138' si celui-ci est trouvé dans la base de données

Renvoie `thx1138` si celui-ci n'est pas trouvé dans la base de données.

Méthodes publiques

majDernierAcces

Modifie dans la base de données la date de dernier accès de l'utilisateur (avec l'heure courante) ainsi que son adresse ip courante.

Description

`boolean` majDernierAcces();

Liste des paramètres

Aucun.

Valeurs de retour

`true` : la mise à jour a été effectuée et les nouvelles informations sont disponibles via les méthodes `getDernierAcces()` et `getIp()`;
`false` : la mise à jour n'a pas été effectuée.

Exemple

```
$user = new User() ;
$res = $user->majDernierAcces();
if ($res) {
    echo 'dernier accès le ' . $user->getDernierAcces() . ' avec l\'IP ' . $user->getIp();
}
```

chargeUser

Chargement des informations d'un utilisateur dans l'objet User.

Description

`Boolean|integer` chargeUser(`string` \$id);

Liste des paramètres

`id` : identifiant de l'utilisateur à charger.

Valeurs de retour

`true` : l'utilisateur `$id` a été chargé depuis la base de données dans l'objet
`User::ERREUR_USER_UNKNOWN` : l'utilisateur est inconnu de la base de données.

Exemple

```
$user = new User() ;
$user->chargeUser('thx1138');
```