

Classe PageNavigator

V2.3.0

Classe PageNavigator

Table des matières

1.	Classe PageNavigator	3
2.	Constantes	3
3.	Méthodes publiques	4
	__construct	4
	draw	5
4.	Getters	6
	getItemDebut	6
	getItemFin.....	6
	getSmall	7
	getDisplayPage	7
5.	Setters	7
	setTotalData	8
	setNbLignesParPage.....	8
	setNbItemMenu	8
	setPageEncours	8
	setSchema.....	9
	setAncre	9
	setIndicePage	10
	setLangue.....	10
	setMinimaliste	11
	setSmallButtons	11
	setLageButtons	11
	setPageOn.....	11
	setPageOff.....	12
	setPrec	12
	setSuiv	13
	setSeparator	13
	setColor.....	13

Classe PageNavigator

Ce document a pour but de décrire la classe **PageNavigator** de l'outil **UniversalWeb** dédié à la pagination de données en tout genre.

Le terme « navigateur » ci-dessous employé fait référence à une instantiation objet de cette classe et ne signifie pas « navigateur internet ».

Pour obtenir la version de cette classe, tapez `echo PageNavigator::VERSION`.

1. Classe PageNavigator

UniversalWeb, via les classes **SimpleListingHelper** et **UniversalList**, fournit des outils pour présenter les données sous forme de tables au sens HTML avec de nombreuses options. La classe **PageNavigator** s'avère une classe complémentaire qui va pouvoir être utilisée conjointement pour gérer la pagination de ces tables. Entièrement paramétrable, Voici un exemple de mise en œuvre :



Pour un exemple concret de sa mise en œuvre, vous pouvez vous reporter au chapitre qui décrit la classe **UniversalList**.

2. Constantes

PageNavigator ne stocke aucune information en base de données ni en mémoire, l'utilisation de la classe est directe et la barre de navigation construite est automatiquement adaptée au contexte en cours. Parmi les informations de contexte se trouve le numéro de page en cours. Cette information doit se trouver impérativement sur la ligne d'url.

La classe met à disposition 3 constantes destinées à définir le type d'url employée dans votre application. En effet, la classe a la particularité de s'adapter à votre modèle d'url.

Nom de la constante	Type d'url
SCHEMA_STANDARD	Modèle d'url standard : les paramètres, dont le numéro de page en cours, sont posés en GET sur l'url dans une forme classique <code>photos.php?data=xxx&page=numPage</code>
SCHEMA_REWRITTE	Modèle d'url passé par une processus rewrite : les paramètres, dont le numéro de page en cours, sont intégrés dans un schéma syntaxique défini par le développeur sur l'url. <code>photos-xxx-numPage-blabla.htm</code>
SCHEMA_REPERTOIRE	Modèle d'url passé sous forme d'une hiérarchie de répertoires : les paramètres, dont le numéro de page en cours, sont intégrés dans un modèle virtuel de répertoires.

3. Méthodes publiques

__construct

Constructeur de la classe.

Description

```
__construct(int $total_data, int $nb_lignes_par_page, int $nb_item_menu,  
int $page_encours[, int $schema=self::SCHEMA_STANDARD] [, int  
$indicePage=null] [, string $lg='fr'))
```

Liste des paramètres

total_data :

nombre total de lignes de données à représenter.

nb_lignes_par_page :

nombre maximal de lignes à représenter dans une page.

nb_item_menu :

nombre d'item à afficher entre l'item de gauche et l'item de droite. Si le nombre est 0, alors l'affichage est destiné à une navigation minimale (comme par exemple pour une application destinée aux téléphones portables)

page_encours :

numéro de la page encours affichée.

schema :

type d'url utilisée par le site. Choisir parmi l'une des constantes définie plus haut. Ce paramètre est optionnel. Par défaut c'est le format standard (constante **SCHEMA_STANDARD**) qui est appliqué.

indicePage :

paramètre optionnel seulement utilisé si le paramètre **schema** est **SCHEMA_REWRITTE** ou **SCHEMA_REPERTOIRE**. Dans ce cas là le paramètre indique l'indice de la particule dans l'uri à laquelle se trouve l'information de page en cours.

Par exemple, si le schéma choisi est **SCHEMA_REWRITTE**, dans le modèle d'url suivante, **photos-xxx-numPage-blabla.htm**, **indicePage** vaut 2 car le numéro de la page se trouve en 3^{ème} position dans l'url (la première particule est la particule 0).

lg :

langue dans laquelle il faut afficher le navigateur. Seule l'anglais et le français sont disponibles. Si **lg** vaut 'fr', le navigateur sera construit en français. Dans tout autre cas il sera construit en anglais.

Valeurs de retour

Aucune

Classe PageNavigator

Exemple 1

Pour l'url suivante : `photos.php?data=xxx&page=7`, le code :

```
$navigateur = new PageNavigator($totalLignes, 25, 7, pageEncours);
```

affichera ce navigateur, possédant 7 choix de pages possibles entre les extrémités (première et dernière page)



Exemple 2

Pour l'url suivante : `photos-xxx-8-blabla.htm`, le code :

```
$navigateur = new PageNavigator($totalLignes, 25, 1, 8, PageNavigator::SCHEMA_REWRITTE, 2, 'fr');
```

affichera ce navigateur, possédant 1 choix de pages possibles entre les extrémités



Exemple 2

Pour l'url suivante : `photos/xxx/1/blabla.htm`, le code :

```
$navigateur = new PageNavigator($totalLignes, 25, 0, 1, PageNavigator::SCHEMA_REPERTOIRE, 2, 'fr');
```

affichera ce navigateur très différent qui n'affiche aucun choix de page intermédiaire (troisième paramètre = 0). Cela peut être intéressant pour le développement pour téléphone portable où la place disponible est limitée.



draw

Retourne le code HTML qui dessine le navigateur.

Description

`draw()`

Liste des paramètres

Aucun

Valeurs de retour

Aucune

Exemple

```
$pn = new PageNavigator($totalLignes, $_SESSION[_ID_LISTING_]->getNbLinesPerPage(), 15,
$_SESSION[_ID_LISTING_]->getPageEncours());
$navigation = $pn->draw();
```

4. Getters

getItemDebut

Retourne l'indice du jeu de données qui correspond à la première ligne affichées pour la page en cours. Ceci permet de repérer l'information affichées dans le flux de données.

Description

int getItemDebut ()

Liste des paramètres

Aucun

Valeurs de retour

Indice du jeu de données correspondant à la première ligne d'information affichée sur la page en cours. Le premier indice du jeu de données est toujours 0.

Exemple

```
$page = 1;  
$pn = new PageNavigator(100, 10, 4, $page);  
$navigation = $pn->draw();  
$premiereLigne = $pn->getItemDebut();
```

renvoie 0 (c'est-à-dire la toute première ligne d'information) car on demande d'afficher la première page

```
$page = 2;  
$pn = new PageNavigator(100, 10, 4, $page);  
$navigation = $pn->draw();  
$premiereLigne = $pn->getItemDebut();
```

renvoie 10 (c'est-à-dire la 11^{ème} ligne d'information) car on demande d'afficher la deuxième page et chaque page est composée de 10 lignes du flux de données.

getItemFin

Retourne l'indice du jeu de données qui correspond à la dernière ligne affichées pour la page en cours. Ceci permet de repérer l'information affichées dans le flux de données.

Description

int getItemFin ()

Liste des paramètres

Aucun

Valeurs de retour

Indice du jeu de données correspondant à la dernière ligne d'information affichée sur la page en cours.

Exemple

```
$page = 1;  
$pn = new PageNavigator(100, 10, 4, $page);  
$navigation = $pn->draw();  
$derniereLigne = $pn->getItemFin();
```

renvoie 9 (c'est-à-dire la 10^{ème} ligne d'information) car on demande d'afficher la première page

Classe PageNavigator

```
$page = 2;  
$pn = new PageNavigator(100, 10, 4, $page);  
$navigation = $pn->draw();  
$premiereLigne = $pn->getItemFin();
```

renvoie 19 (c'est-à-dire la 20^{ème} ligne) car on demande d'afficher la deuxième page et chaque page est composée de 10 lignes du flux de données.

getSmall

Indique si l'affichage en cours est composé de petits boutons.

Description

boolean getSmall()

Liste des paramètres

Aucun

Valeurs de retour

true : le navigateur en cours affiche des petits boutons.

false : le navigateur en cours affiche des boutons de taille standard.

Note

Pour changer la taille des boutons, faites appel aux méthodes **setSmallButtons** et **setLargeButtons**.

getDisplayPage

Indique si l'affichage en cours affiche le mot 'Page : ' précède l'affichage du navigateur.

Description

boolean getDisplayPage()

Liste des paramètres

Aucun

Valeurs de retour

true : le navigateur affiche le mot 'Page : '.

false : le navigateur n'affiche pas le mot 'Page : '.

Note

Pour modifier cette fonctionnalité, faites appel aux méthodes **setPageOn** et **setPageOff**.

5. Setters

Les setters sont mis à disposition du développeur. Cependant il y a peu de chance qu'il soit nécessaire d'y faire appel puisque tous les paramètres du navigateur sont fournis lors de l'instanciation de la classe (new).

Classe PageNavigator

setTotalData

Positionne le nombre total de données que le navigateur a à gérer.

Description

`setTotalData(int $valeur)`

Liste des paramètres

valeur : nombre total de données à gérer pour le navigateur.

Valeurs de retour

Aucune

setNbLignesParPage

Positionne le nombre total de données maxi à afficher par page.

Description

`setNbLignesParPages(int $valeur)`

Liste des paramètres

valeur : nombre total de données à afficher par page.

Valeurs de retour

Aucune

setNbItemMenu

Défini le nombre d'item (choix de pages pour l'utilisateur) maximum à afficher entre l'item de gauche et l'item de droite.

Description

`setNbItemMenu(int $valeur)`

Liste des paramètres

valeur : nombre maximum d'item affichable entre les items extérieurs. Si 0, le navigateur aura l'apparence suivante :



Valeurs de retour

Aucune

Note

La méthode a le même effet que l'appel à la méthode `setMinimaliste(true)`.

setPageEncours

Définit le numéro de la page en cours d'affichage.

Description

`setPageEncours(int $valeur)`

Classe PageNavigator

Liste des paramètres

valeur : numéro de la page à afficher.

Valeurs de retour

Aucune

setSchema

Définit le schéma d'url employé par l'application afin que la classe s'adapte à sa gestion.

Description

setSchema(**int** \$valeur)

Liste des paramètres

valeur : constante de schéma sélectionné parmi les valeurs ci-dessous.

SCHEMA_STANDARD

Modèle d'url standard : les paramètres, dont le numéro de page en cours, sont posés en GET sur l'url dans une forme classique

`photos.php?data=xxx&page=numPage`

SCHEMA_REWRITTE

Modèle d'url passé par une processus rewrite : les paramètres, dont le numéro de page en cours, sont intégrés dans un schéma syntaxique défini par le développeur sur l'url.

`photos-xxx-numPage-blabla.htm`

SCHEMA_REPERTOIRE

Modèle d'url passé sous forme d'une hiérarchie de répertoires : les paramètres, dont le numéro de page en cours, sont intégrés dans un modèle virtuel de répertoires.

`photos/xxx/numPage/blabla.htm`

Valeurs de retour

Aucune

Exemple

```
$pageEncours = 1;
$pn = new PageNavigator(100, 10, 4, $pageEncours);
$pn->setSchema(PageNavigator::SCHEMA_REWRITTE);
$navigation = $pn->draw();
```

setAncre

Définit une ancre html optionnelle qui sera déposée à la fin de chaque url. Cette méthode est seulement disponible pour le schéma **SCHEMA_STANDARD**.

Description

setAncre(**string** \$valeur)

Liste des paramètres

valeur : ancre à écrire en fin d'url.

Valeurs de retour

Aucune

Exemple

Pour une url ainsi définie: `https://serveur/?page=1#articles`

```
$pageEncours = 1;
$pn = new PageNavigator(100, 10, 4, $pageEncours);
$pn->setSchema(PageNavigator::SCHEMA_STANDARD);
$pn->setAncre('#articles');
$navigation = $pn->draw();
```

setIndicePage

Définit l'indice de la particule dans l'uri où se trouve l'information de numéro de page à afficher. Cette méthode ne concerne que les schémas définis par **SCHEMA_REWRITTE** et **SCHEMA_REPERTOIRE**.

Description

setIndicePage(**int** \$valeur)

Liste des paramètres

valeur : indice de la particule dans l'uri.

Valeurs de retour

Aucune

Exemple

Pour une url ainsi définie : `photos-xxx-numPage-blabla.htm`

```
$pageEncours = 1;
$pn = new PageNavigator(100, 10, 4, $pageEncours);
$pn->setSchema(PageNavigator::SCHEMA_REWRITTE);
$pn->setIndicePage(2);
$navigation = $pn->draw();
```

setLangue

Positionne la langue d'affichage du navigateur.

Description

setLangue(**string** \$langue)

Liste des paramètres

langue : si 'fr' (valeur par défaut) alors l'affichage du navigateur sera en français. Toute autre entrée affichera le navigateur en anglais..

Valeurs de retour

Aucune

Exemple

```
$pageEncours = 1;
$pn = new PageNavigator(100, 10, 4, $pageEncours);
$pn->setLangue('fr');
$navigation = $pn->draw();
```

setMinimaliste

Affiche un navigateur épuré.

Description

`setMinimaliste` (**boolean** \$valeur)

Liste des paramètres

valeur :

si **true**, le navigateur sera affiché de la manière suivante :



Si **false**, le navigateur aura un affichage standard



Valeurs de retour

Aucune

Note

La méthode a le même effet que l'appel à la méthode `setNbItemMenu(0)`.

setSmallButtons

Affiche des boutons de navigation plus petit.

Description

`setSmallButtons` ()

Liste des paramètres

Aucun

Valeurs de retour

Aucune

setLargeButtons

Affiche des boutons de navigation normaux.

Description

`setLargeButtons` ()

Liste des paramètres

Aucun

Valeurs de retour

Aucune

setPageOn

Affiche le libelle « Page » sur le navigateur.

Classe PageNavigator

« Préc. | Page: 1 2 ...5 | Suiv. »

Description

`setPageOn()`

Liste des paramètres

Aucun

Valeurs de retour

Aucune

Note

Le navigateur affiche le libellé « Page » par défaut. Pour supprimer ce libellé appeler la méthode `setPageOff()`.

setPageOff

Annule l'affichage du libellé « Page » sur le navigateur.

Description

`setPageOff()`

Liste des paramètres

Aucun

Valeurs de retour

Aucune

setPrec

Définit le texte d'affichage pour signifier « Page précédente ».

Description

`setPrec(string $valeur)`

Liste des paramètres

valeur : chaîne de caractère à afficher sur l'item « Page précédente »

Valeurs de retour

Aucune

Exemple

```
$pn = new PageNavigator($totalLignes, 5, 2, $page);  
$debut = $pn->getItemDebut();  
$pn->setPrec('titi');
```

titi | Page: 1 2 ...5 | Suiv. »

```
$pn->setPrec('<span class="fas fa-home"></span>');
```

 | Page: 1 2 ...5 | Suiv. »

Classe PageNavigator

setSuiv

Définit le texte d'affichage pour signifier « Page suivante ».

Description

setSuiv(**string** \$valeur)

Liste des paramètres

valeur : chaîne de caractère à afficher sur l'item « Page suivante »

Valeurs de retour

Aucune

Exemple

Voir méthode **setPrec**.

setSeparator

Affiche des séparateurs entre les boutons.

Description

setSeparator(**boolean** \$valeur)

Liste des paramètres

valeur :

si **true**, le navigateur affiche des séparateurs (valeur par défaut)

« Préc. | 1 2 ...5 | Suiv. »

Si **false**, le navigateur n'affiche pas de séparateurs

« Préc. 1 2 ...5 Suiv. »

Valeurs de retour

Aucune

Exemple

```
$pn = new PageNavigator($totalLignes, 5, 2, $page);  
$debut = $pn->getItemDebut();  
$pn->setPageOff();  
$pn->setSeparator(false);
```

setColor

Change la couleur des boutons.

Description

setColor(**string** \$couleur)

Liste des paramètres

couleur :

primary (valeur par défaut)

« Préc. | 1 2 ...5 | Suiv. »

success

« Préc. | 1 2 ...5 | Suiv. »

Classe PageNavigator

warning



danger



Info



link



secondary



Valeurs de retour

Aucune

Exemple

```
$pn = new PageNavigator($totalLignes, 5, 2, $page);  
$debut = $pn->getItemDebut();  
$pn->setPageOff();  
$pn->setColor('warning');
```

Note

Les identifiants de couleurs correspondent aux identifiants de couleur Bootstrap.