

AIGLe Manuel de référence
v4.1.1

Généré par Doxygen 1.4.7

Fri Aug 22 16:45:16 2008

Table des matières

Chapitre 1

AIGLe Index hiérarchique

1.1 AIGLe Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

BD	??
BDMYSQL	??
Bibtex	??
ENDNOTE	??
ENDNOTECREATORPARSE	??
ENDNOTEDATEPARSE	??
ENDNOTEMAP	??
Form	??
Entete	??
Formulaire	??
Resultats	??
OAI	??
Publication	??
Session	??
SoapClientAigle	??
nuSoapClient	??
TypeDoc	??
User	??
WaitImg	??

Chapitre 2

AIGLe Index des classes

2.1 AIGLe Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

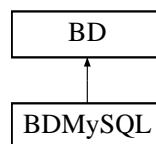
BD	??
BMySQL	??
Bibtex	??
ENDNOTE	??
ENDNOTECREATORPARSE	??
ENDNOTEDATEPARSE	??
ENDNOTEMAP	??
Entete	??
Form	??
Formulaire	??
nuSoapClient	??
OAI	??
Publication	??
Resultats	??
Session	??
SoapClientAigle	??
TypeDoc	??
User	??
WaitImg	??

Chapitre 3

AIGLe Documentation des classes

3.1 Référence de la classe BD

Graphe d'héritage de BD :



Fonctions membres publiques

- BD (\$login, \$mot_de_passe, \$base, \$serveur)
- free_result (\$result)
- execRequete (\$requete)
- objetSuivant (\$resultat)
- ligneSuivante (\$resultat)
- tableauSuivant (\$resultat)
- prepareChaine (\$chaine)
- genereID (\$nom_sequence)
- nbAttributs (\$res)
- nomAttribut (\$res, \$position)
- nbResultats (\$res)
- getId (\$res)
- messageSGBD ()
- getSGBD ()
- setSGBD (\$sgbd)

3.1.1 Description détaillée

Classe abstraite définissant une interface générique d'accès à une base de données. Version complète

3.1.2 Documentation des fonctions membres

3.1.2.1 BD : :BD (\$ login, \$ mot_de_passe, \$ base, \$ serveur)

Constructeur de la classe BD (p. ??)

Le constructeur établit la connexion à la base par l'appel à la méthode privée connect()

Paramètres :

\$login string : username ayant les droit d'admin sur la base \$base

\$mot_de_passe string : mot de passe du user \$login

\$base string : Base de donnée à utiliser.

\$serveur string : nom du serveur de la base de donnée.

3.1.2.2 BD : :execRequete (\$requete)

Méthode d'exécution d'une requête.

La méthode appelle aux méthodes privées exec() pour exécuter la requête et messageSGBD() (p. ??) pour envoyer le message d'erreur éventuel

Paramètres :

\$requete string : Requete SQL

3.1.2.3 BD : :free_result (\$result)

Methode free_result(\$result). Libere la ressource (memory out)

Paramètres :

\$result ressource MySQL

Réimplémentée dans **BDMYSQL p.** (classBDMYSQL_{f03c77b65267f2226e22e9ff87b9a2d??})

3.1.2.4 BD : :genereID (\$nom_sequence)

Génération d'un identifiant.

Paramètres :

\$nom_sequence string????

Réimplémentée dans **BDMYSQL p.** (classBDMYSQL_{91e4286b134ab991095c673c9ee8ab40??})

3.1.2.5 BD : :getId (\$res)

Methode qui renvoie l'id (auto_incremente) d'une insertion.

Paramètres :

\$res ressource MySQL

3.1.2.6 BD : :getSGBD ()

retourne le Nom du SGBD

3.1.2.7 BD : :ligneSuivante (\$ resultat)

Methode ligneSuivante (\$resultat) Accès à la ligne suivante, sous forme de tableau associatif

Paramètres :

\$resultat pointeur MySQL

Réimplémentée dans **BDMySQL p.** (classBDMySQL₁ab706ad8cdcc61f2e403f9355f5b735 ??)

3.1.2.8 BD : :messageSGBD ()

Retour du message d'erreur

Réimplémentée dans **BDMySQL p.** (classBDMySQL_a968fbbf4af4f517457a17864c9276b1 ??)

3.1.2.9 BD : :nbAttributs (\$ res)

Méthode indiquant le nombre d'attributs dans le resultat.

Paramètres :

\$res ressource MySQL

Réimplémentée dans **BDMySQL p.** (classBDMySQL₆33a06e5afc754d25d967713b54a0b3b ??)

3.1.2.10 BD : :nbResultats (\$ res)

Méthode donnant le nombre de résultats (nb de tableau ou d'objet) d'une requete.

Paramètres :

\$res ressource MySQL

Réimplémentée dans **BDMySQL p.** (classBDMySQL_a1296f72ad34b3f27f358e47d7daec1c ??)

3.1.2.11 BD : :nomAttribut (\$ res, \$ position)

Méthode donnant le nom d'un attribut dans un résultat.

Paramètres :

\$res ressource MySQL

\$position ???

Réimplémentée dans **BDMySQL p.** (classBDMySQL₅4f75481ee677d47dc1f57009c6e0734 ??)

3.1.2.12 BD : :objetSuivant (\$ resultat)

Methode objetSuivant (\$resultat) Accès à la ligne suivante, sous forme d'objet

Paramètres :

\$resultat pointeur MySQL

Réimplémentée dans **BDMySQL p.** (classBDMySQL₇6b117ce11d4572849614deeb0ec90c9 ??)

3.1.2.13 BD : :prepareChaine (\$ chaine)

Echappement des apostrophes et autres préparations à l'insertion.

Paramètres :

\$chaine string : Chaine a preparer

Réimplémentée dans **BMySQL p.** (classBMySQL_f68fbee0de7a7d5509081f5e28f11b ??)

3.1.2.14 BD : :setSGBD (\$ sgbd)

set le Nom du SGBD.

Paramètres :

\$sgbd string : type de SGBD

3.1.2.15 BD : :tableauSuivant (\$ resultat)

Methode tableauSuivant (\$resultat) Accès à la ligne suivante, sous forme de tableau indice.

Paramètres :

\$resultat pointeur MySQL

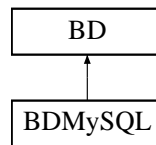
Réimplémentée dans **BMySQL p.** (classBMySQL_1a0ee4b219345b7265603292e8d73a8c ??)

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/BD.class.php

3.2 Référence de la classe BDMYSQL

Graphe d'héritage de BDMYSQL :



Fonctions membres publiques

- free_result (\$result)
- objetSuivant (\$resultat)
- ligneSuivante (\$resultat)
- tableauSuivant (\$resultat)
- getId ()
- prepareChaine (\$chaine)
- genereID (\$nom_sequence)
- messageSGBD ()
- schemaTable (\$nom_table)
- nbAttributs (\$resultat)
- nomAttribut (\$resultat, \$pos)
- nbResultats (\$resultat)

3.2.1 Description détaillée

Sous-classe de la classe abstraite **BD** (p. ??), implantant l'accès e MySQL

Pas de propriétés : elles sont héritées de la classe **BD** (p. ??).

Pas de constructeur : lui aussi est hérité.

3.2.2 Documentation des fonctions membres

3.2.2.1 BDMYSQL : :free_result (\$ result)

Méthode free_result(\$result). Libère la ressource (memory out)

Paramètres :

\$result ressource MySQL

Réimplémentée à partir de **BD p.** (classBD_{1874a3c678db931d3f90967fc3efb398} ??)

3.2.2.2 BDMYSQL : :genereID (\$ nom_sequence)

Génération d'un identifiant (pour les primary de type autoincrémenté)

\$nom_sequence = nom de la table pour laquelle on veut générer un nouvel identifiant.

Réimplémentée à partir de **BD p.** (classBD_{147ab88dcb47c5a0d09b731449f72ece} ??)

3.2.2.3 BDMYSQL : :getId ()

Méthode qui renvoie l'id (auto_incrémenté) d'une insertion

3.2.2.4 BDMYSQL : :ligneSuivante (\$ resultat)

Accès à la ligne suivante, sous forme de tableau associatif

\$resultat = resultat de la requete ayant permis de selectionner le(s) tableau(x)

Réimplémentée à partir de **BD p.** (classBD₁1166929d9d4b3faa41823f80b3151b0 ??)

3.2.2.5 BDMYSQL : :messageSGBD ()

Retour du message d'erreur

Réimplémentée à partir de **BD p.** (classBD₀8f692d5f13f41c71f0a794bec939903 ??)

3.2.2.6 BDMYSQL : :nbAttributs (\$ resultat)

Methode renvoyant le nombre d'attributs des résultats d'une requête.

\$resultat : resultat d'une requete SQL.

Réimplémentée à partir de **BD p.** (classBD_e407a01eaf05cb5445ba3b0daad08d5 ??)

3.2.2.7 BDMYSQL : :nbResultats (\$ resultat)

Methode renvoyant le nombre de résultats d'une requête.

\$resultat : resultat d'une requete SQL.

Réimplémentée à partir de **BD p.** (classBD_f2d5ca3049b973aafea9a40b9a5ee2d9 ??)

3.2.2.8 BDMYSQL : :nomAttribut (\$ resultat, \$ pos)

Methode renvoyant le nom des attributs des résultats d'une requête.

\$resultat : resultat d'une requete SQL.

Réimplémentée à partir de **BD p.** (classBD₁dee75605cc8ea6656e9376c310a9b64 ??)

3.2.2.9 BDMYSQL : :objetSuivant (\$ resultat)

Accès à la ligne suivante, sous forme d'objet

\$resultat = resultat de la requete ayant permis de selectionner le(s) objet(s)

Réimplémentée à partir de **BD p.** (classBD₈231e0cadaafaf6fc01439b0390862116 ??)

3.2.2.10 BDMYSQL : :prepareChaine (\$ chaine)

Echappement des apostrophes et autres préparation à l'insertion.

\$chaine : chaine de caracteres à preparer.

Réimplémentée à partir de **BD p.** (classBD₃a6d589cf2bd1cceb7f7181f9c503b867 ??)

3.2.2.11 BMySQL : :schemaTable (\$ nom_table)

Renvoie le schéma d'une table.

\$nom_table : nom de la table dont on desire le schema.

retourne un tableau (des entrées de la table) de tableau (des attributs de chaque entrée)

3.2.2.12 BMySQL : :tableauSuivant (\$ resultat)

Accès à la ligne suivante, sous forme de tableau indicé

resultat = resultat de la requete ayant permis de selectionner le(s) tableau(x)

Réimplémentée à partir de **BD p.** (classBD_{d4aaecbb00a8e2a784fa656e8b9a5770} ??)

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/BMySQL.class.php

3.3 Référence de la classe Bibtex

Fonctions membres publiques

- **Bibtex** (*\$bd*, *\$btex_typedoc*)
- **GetTypeDoc** (*\$aigle_typedoc*)
- **GetCode** (*\$pfile=""*)
- **SetAttribsVals** (&*\$AIGLe_Publi*)

3.3.1 Description détaillée

class **Bibtex** (p. ??)

3.3.2 Documentation des fonctions membres

3.3.2.1 **Bibtex** : :**Bibtex** (*\$ bd*, *\$ btex_typedoc*)

Constructeur de Notice BibTex

Paramètres :

\$btex_typedoc string : Type de document **Bibtex** (p. ??).

\$bd object : BdD.

3.3.2.2 **Bibtex** : :**GetCode** (*\$ pfile = ""*)

!Ecrit le code **Bibtex** (p. ??) dans le fichier .

Paramètres :

\$pfile pointer : Pointeur du fichier dans lequel on écrit le code **Bibtex** (p. ??) .

3.3.2.3 **Bibtex** : :**GetTypeDoc** (*\$ aigle_typedoc*)

! Return le type de document **Bibtex** (p. ??) correspondant au type de Document Aigle Passe en paramètre

Paramètres :

\$aigle_typedoc string Type de document AIGLe

Renvoi :

string

3.3.2.4 **Bibtex** : :**SetAttribsVals** (&*\$ AIGLe_Publi*)

!Affect les valeurs des Attributs **Bibtex** (p. ??) (en fonction de ceux de la publi AIGLe)

Paramètres :

\$AIGLe_Publi Objet AIGLe **Publication** (p. ??)

Renvoi :

int

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/Bibtex.class.php

3.4 Référence de la classe ENDNOTE

Fonctions membres publiques

- ENDNOTE (\$bd, \$file, \$tab_types)
- EN_MontreTableau ()
- GetGoodPublis (\$typedoc2show=Array())
- EN_GetListRecords ()
- EN_GetPubli (\$identifiant, \$show_warn=1, \$in_session=1)

Attributs publics

- \$Good_Publis = Array()
- \$List_TypeDoc = Array()

3.4.1 Description détaillée

Classe OAI (p. ??)

3.4.2 Documentation des fonctions membres

3.4.2.1 ENDNOTE : :EN_GetListRecords ()

Affiche un tableau contenant une liste d'identifiants de documents EndNote. et leur type

3.4.2.2 ENDNOTE : :EN_GetPubli (\$ *identifiant*, \$ *show_warn* = 1, \$ *in_session* = 1)

Renvoie un tableau contenant :

- l' Objet Publi avec les caracteristiques du documents HAL dont l'identifiant est passe en parametre ;
 - le tableau des URLs HAL des textes inte22graux du document.
- @ params \$identifiant = identifiant EndNote REC-NUMBER d'un document .

3.4.2.3 ENDNOTE : :EN_MontreTableau ()

Affiche le tableau apres parsage ,pour debug only

3.4.2.4 ENDNOTE : :ENDNOTE (\$ *bd*, \$ *file*, \$ *tab_types*)

Constructeur de la classe.

char \$file : Fichier XML a parser.

object \$bd : Objet Base de donnee.

array \$tab_types = tableau des types doc EN inconnus et leur correspondance avec AIGLe

3.4.2.5 ENDNOTE : :GetGoodPublis (\$ *typedoc2show* = Array())

Initialize le tableau des publis reconnues (\$this->Good_Publis).

ainsi que la liste des types de documents (\$this->List_TypeDoc)

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/publications/EN_Classes/ENDNOTE.class.php

3.5 Référence de la classe ENDNOTECREATORPARSE

Fonctions membres publiques

- ENDNOTECREATORPARSE ()
- parse (\$input)

3.5.1 Description détaillée

Id

ENDNOTECREATORPARSE.class.php,v 1.3 2008/07/31 22 :47 :35 caron Exp

\$Aigle : publications/EN_Classes/ENDNOTECREATORPARSE.class.php

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://aigle.sf.org/copyleft/gpl.html>.

Released through <http://bibliophile.sourceforge.net> under the GPL licence. Do whatever you like with this - some credit to the author(s) would be appreciated.

A collection of PHP classes to manipulate bibtex files.

If you make improvements, please consider contacting the administrators at bibliophile.sourceforge.net so that your improvements can be added to the release package.

Mark Grimshaw 2006 <http://bibliophile.sourceforge.net> Modified 17/March/2005 Mark Grimshaw for Endnote XML author formats...

3.5.2 Documentation des fonctions membres

3.5.2.1 ENDNOTECREATORPARSE : :ENDNOTECREATORPARSE ()

Constructeur **ENDNOTECREATORPARSE ()** (p.??)

3.5.2.2 ENDNOTECREATORPARSE : :parse (\$input)

Create writer arrays from endnote input.

Paramètres :

\$input string : author field can be : Grimshaw, Mark N. Smith, Alfred, Jr. Fisher, J.O. J O Fisher Mark M N G Grimshaw Mark N. Grimshaw The delimiter WIKINDX inserts for XML multiple authors is ' and '

La documentation de cette classe a été générée à partir du fichier suivant :

- /mnt/sshfs/AIGLE-DEVEL/aigle-main/publications/EN_Classes/ENDNOTECREATORPARSE.class.php

3.6 Référence de la classe ENDNOTEDATEPARSE

Fonctions membres publiques

- ENDNOTEDATEPARSE ()
- init (\$dateField)

3.6.1 Description détaillée

Id

```
ENDNOTEDATEPARSE.class.php,v 1.4 2008/07/31 22 :47 :35 caron Exp
```

```
$Aigle : publications/EN_Classes/ENDNOTEDATEPARSE.class.php
```

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://aigle.sf.org/copyleft/gpl.html>.

FROM WIKINDEX : Bibliographic Management system. Copyright (C)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

The WIKINDEX Team 2004 sirfragalot@users.sourceforge.net

3.6.2 Documentation des fonctions membres

3.6.2.1 ENDNOTEDATEPARSE : :ENDNOTEDATEPARSE ()

Constructor

3.6.2.2 ENDNOTEDATEPARSE : :init (\$ dateField)

```
init($dateField).
```

Paramètres :

\$dateField string : date a analyser.

La documentation de cette classe a été générée à partir du fichier suivant :

```
- /mnt/sshfs/AIGLE-DEVEL/aigle-main/publications/EN_-  
Classes/ENDNOTEDATEPARSE.class.php
```

3.7 Référence de la classe ENDNOTEMAP

Fonctions membres publiques

- ENDNOTEMAP (&\$EN_Publi_Array, \$list_types)
- GetElecResNum (\$resource)

3.7.1 Description détaillée

Id

ENDNOTEMAP.class.php,v 1.6 2008/07/31 22 :47 :35 caron Exp

\$Aigle : publications/EN_Classes/ENDNOTEMAP.class.php

Copyright 1999-2008 Dominique Caron <d.caron_AT_LPTA.univ-montp2.fr>

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://aigle.sf.org/copyleft/gpl.html>. inspired from Wikindx ENDNOTEMAP.php

3.7.2 Documentation des fonctions membres

3.7.2.1 ENDNOTEMAP::ENDNOTEMAP (&\$ EN_Publi_Array, \$ list_types)

Constructor

Paramètres :

&\$EN_Publi_Array pointer : Pointeur vers le tableau des attributs => valeur d'un document **ENDNOTE** (p.??).

\$list_types Array : Tableau des type de documents AIGLe (ART_ACL,ART_SCL etc..)

Version :

1

3.7.2.2 ENDNOTEMAP::GetElecResNum (\$ resource)

Fonction qui tente de retourner l'url de la publi (si possible)

Paramètres :

\$resource = identifiant a recherche : DOI, arxiv....

Renvoi :

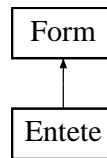
string : l'URL trouve

La documentation de cette classe a été générée à partir du fichier suivant :

- /mnt/sshfs/AIGLE-DEVEL/aigle-main/publications/EN_Classes/ENDNOTEMAP.class.php

3.8 Référence de la classe Entete

Graphe d'héritage de Entete : :



Fonctions membres publiques

- `champEntree ($libelle, $nom, $valeur, $icon="", $classe="", $css_width="", $libelle_width="", $br=1)`
- `genereHTML ()`

3.8.1 Description détaillée

Class **Entete** (p.??) Classe qui permet de generer le code HTML d'un entete (menu general) de module

3.8.2 Documentation des fonctions membres

3.8.2.1 Entete : :`champEntree ($ libelle, $ nom, $ valeur, $ icon = "", $ classe = "", $ css_width = "", $ libelle_width = "", $ br = 1)`

champEntree () (p.??) cree une nouvelle entree du menu general du module.

Il ne s'agit en fait que d'une methode simplifiant la generation d'une ligne de menu.

elle appelle les methodes generales **champImage ()** (p.??) et **ajoutTexte ()** (p.??) avec des parametres par default. Elle se charge de creer le champCache de nom \$nom si celui-ci n'est pas deja...

public \$nom : le nom de l'entree hidden affectee par le clique sur l'image (par default = 'todo')

public \$icon : URL de l'icone presente devant cette entree de menu (par default icons/right.png).

NB : Si icon = 'selected' alors l'icone par default devient icons/selected.png.

en plus les fontes du texte seront plus grosses.

\$valeur : Au click sur l'image on affecte la valeur \$valeur a l'entree hidden dont le nom est \$nom

On verifie que parmi les composants deja entres il y a deja un champ de type hidden ayant pour nom \$nom

3.8.2.2 Entete : :`genereHTML ()`

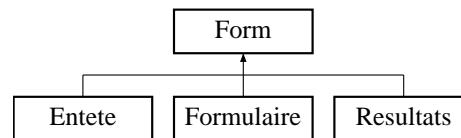
Methode qui genere le code HTML des entetes (menu) des modules

Réimplémentée à partir de **Form p.** (classForm ??)

La documentation de cette classe a été générée à partir du fichier suivant :
– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/Entete.class.php

3.9 Référence de la classe Form

Graphe d'héritage de Form :



Fonctions membres publiques

- Form (\$methode="post", \$action="", \$centre=TRUE, \$classeCSS="Form", \$nom="Form", \$OnSubmit="")
- getChamp (\$idComposant)
- champTexte (\$libelle, \$nom, \$valeur, \$taille, \$tailleMax=0, \$classe="", \$libelle_width="", \$css_width="", \$br=1, \$readonly=0, \$onchange="")
- champDownload (\$libelle, \$icon, \$url, \$classe="", \$libelle_width="", \$css_width="", \$br=1, \$onclick="")
- champCalendrier (\$libelle, \$nom, \$valeur, \$reset=0, \$libelle_width="", \$css_width="", \$br=1, \$readonly=0, \$onchange="")
- champMotDePasse (\$pLibelle, \$pNom, \$pVal, \$pTaille, \$pTailleMax=0, \$classe="", \$libelle_width="", \$css_width="", \$br=1, \$readonly=0)
- champRadio (\$libelle, \$nom, \$valeur, \$liste, \$classe="", \$libelle_width="", \$css_width="", \$br=1, \$readonly=0, \$onChange="")
- champBouton (\$valeur, \$nom, \$onClick, \$classe="", \$css_width="", \$br=1)
- champCheckBox (\$pLibelle, \$pNom, \$pVal, \$pListe, \$length=-1, \$classe="", \$libelle_width="", \$css_width="", \$br=1, \$readonly=0, \$onChange="")
- champListe (\$pLibelle, \$pNom, \$pVal, \$pTaille, \$pListe, \$multi=0, \$classe="", \$libelle_width="", \$css_width="", \$br=1, \$readonly=0, \$onChange="")
- champImage (\$libelle, \$nom, \$valeur, \$src, \$largeur, \$classe="", \$libelle_width="", \$css_width="", \$br=1, \$onClick="")
- champFenetre (\$libelle, \$nom, \$valeur, \$lig, \$col, \$classe="", \$libelle_width="", \$css_width="", \$br=1, \$readonly=0)
- champValider (\$pLibelle, \$pNom, \$classe="", \$css_width="", \$br=1, \$onClick="")
- champAnnuler (\$pLibelle, \$pNom, \$classe="", \$css_width="", \$br=1, \$onClick="")
- champFichier (\$pLibelle, \$pNom, \$pTaille, \$classe="", \$libelle_width="", \$css_width="", \$br=1)
- champCache (\$nom, \$valeur, \$identifiant=0)
- ajoutTexte (\$texte, \$classe="", \$css_width="", \$br=1)
- debutDiv (\$id_div, \$classe, \$style)
- finDiv ()
- changeComposant (\$idComposant, \$valeur, \$libelle, \$nom, \$liste, \$params)
- ajoutTitre (\$texte, \$classe, \$css_width, \$br)
- ajoutSelectAll (\$select, \$overwrite)
- setIdentifiant (\$valeur, \$numero_ligne)

Attributs publics

- \$help_by = "tooltips"

3.9.1 Description détaillée

Class Abstraite Form (p. ??).

Classe mere des Classes Formulaire (p. ??), Resultats (p. ??), Entete (p. ??).

3.9.2 Documentation des fonctions membres

3.9.2.1 Form : :ajoutSelectAll (\$ Select, \$ overwrite)

ajoutSelectAll() (p. ??).

Paramètres :

\$Select array : Tableau valeur=>libelle du HTML Select.

\$overwrite boolean : 0 = , 1 = .

Réimplémentée dans **Resultats p.** (classResultats_{64d5c6abaa4a87b4327ec1703ac36c7d} ??)

3.9.2.2 Form : :ajoutTexte (\$ texte, \$ classe = "", \$ css_width = "", \$ br = 1)

insertion d'un texte : insertion des parametres dans le tableau des composants

Paramètres :

\$texte string : texte a ajouter.

\$classe string : classe de l'entree.

\$css_width int : largeur du CSS en %.

\$br boolean : 1 = ret-char, 0 = pas de ret-char.

3.9.2.3 Form : :ajoutTitre (\$ texte, \$ classe, \$ css_width, \$ br)

ajoutTitre() (p. ??) Ajoute un Titre a un tableau de resultat.

Paramètres :

\$texte string : texte a ajouter.

\$classe string : classe de l'entree.

\$css_width int : largeur du CSS en %.

\$br boolean : 1 = ret-char, 0 = pas de ret-char.

Réimplémentée dans **Resultats p.** (classResultats_{fe2967f031abcf85e72d95772ef58a96} ??)

3.9.2.4 Form : :champAnnuler (\$ pLibelle, \$ pNom, \$ classe = "", \$ css_width = "", \$ br = 1, \$ OnClick = "")

Création d'un bouton de type RESET : insertion des parametres dans le tableau des composants

3.9.2.5 Form : :champBouton (\$ valeur, \$ nom, \$ OnClick, \$ classe = "", \$ css_width = "", \$ br = 1)

Methode ajoutant un bouton de type button.

char * \$valeur : valeur de l'INPUT (ce qui sera affiché sur le bouton).

char * \$nom : nom de l'INPUT.

char * \$OnClick : Code JS a executer au clique.

char * \$class : classe CSS du bouton.

char * \$css_width : largeur du div en %.

bool \$br : 1=dernier element d'une ligne de resultat
 \$pLibelle : libelle du bouton
 \$Nom : nom de l'INPUT html
 \$OnClick : Code JS a executer au clique
 \$classe : classe CSS du bouton
 \$css_width : largeur du div en % par rapport au formulaire
 \$br : 1=dernier element d'une ligne de resultat

3.9.2.6 Form : :champCache (\$ nom, \$ valeur, \$ identifiant = 0)

Creation d'une entree hidden :

Paramètres :

\$nom string : nom de l'entree
\$valeur string : valeur de l'entree
\$identifiant boolean : 1 : cette entree a un identifiant (identique a son nom) ou 0 : non.
 insertion des parametres dans le tableau des composants

3.9.2.7 Form : :champCalendrier (\$ libelle, \$ nom, \$ valeur, \$ reset = 0, \$ libelle_width = "", \$ css_width = "", \$ br = 1, \$ readonly = 0, \$ onchange = "")

Creation d'un champ Calendrier et son libelle insertion des parametres dans le tableau des composants
 \$reset = si cette variable est a 1 il y aura un bouton permettant de remettre a zero le champ.

3.9.2.8 Form : :champCheckBox (\$ pLibelle, \$ pNom, \$ pVal, \$ pListe, \$ length = -1, \$ classe = "", \$ libelle_width = "", \$ css_width = "", \$ br = 1, \$ readonly = 0, \$ OnChange = "")

Création d'un champ checkbox et son libellé : insertion des parametres dans le tableau des composants

3.9.2.9 Form : :champDownload (\$ libelle, \$ icon, \$ url, \$ classe = "", \$ libelle_width = "", \$ css_width = "", \$ br = 1, \$ onclick = "")

Création d'un champ Download de de son libellé :
 insertion des parametres dans le tableau des composants

3.9.2.10 Form : :champFenetre (\$ libelle, \$ nom, \$ valeur, \$ lig, \$ col, \$ classe = "", \$ libelle_width = "", \$ css_width = "", \$ br = 1, \$ readonly = 0)

Création d'un champ textarea et son libellé : insertion des parametres dans le tableau des composants

3.9.2.11 Form : :champFichier (\$ pLibelle, \$ pNom, \$ pTaille, \$ classe = "", \$ libelle_width = "", \$ css_width = "", \$ br = 1)

Création d'une entree de type file (upload) : insertion des parametres dans le tableau des composants

3.9.2.12 Form : :champImage (\$ libelle, \$ nom, \$ valeur, \$ src, \$ largeur, \$ classe = "", \$ libelle_width = "", \$ css_width = "", \$ br = 1, \$ OnClick = "")

Création d'un champ Image et de son libellé

3.9.2.13 Form : :champListe (\$ pLibelle, \$ pNom, \$ pVal, \$ pTaille, \$ pListe, \$ multi = 0, \$ classe = "", \$ libelle_width = "", \$ css_width = "", \$ br = 1, \$ readonly = 0, \$ OnChange = "")

Création d'un champ SELECT et son libellé : insertion des parametres dans le tableau des composants.
parametres : \$pLibelle = libelle du champ.

\$pNom = nom de la variable SELECT.

\$pVal = valeur de la variable SELECT.

\$pTaille = 'hauteur' du SELECT (utile pour le multi) .

\$pListe = Array des valeurs proposees.

\$multi = 0 :SELECT simple, 1 :SELECT multiple.

\$classe = classe CSS du champ SELECT.

\$libelle_width =largeur en % du champ libelle.

\$css_width =largeur en % du champ SELECT.

\$br= 0 : pas de ret_char, 1 : ret_char.

\$readonly.

\$OnChange=code JS a executer onchange.

Particularités :

Si dans le tableau des valeurs (\$pListe) il existe une (et une seule !) entree ayant

comme libelle 'Autre' ou 'autre' ou 'AUTRE', alors automatiquement la selection de cette option permettra au visiteur

d'entrer une nouvelle valeur option dans un INPUT de type 'text', cette nouvelle option aura comme valeur ce qu'aura tape

le visiteur dans l'INPUT.

Pour eviter ce comportement (Option 'Autre' sans possibilité de nouvelle valeur), il suffit de mettre \$On-change="return false ;".

3.9.2.14 Form : :champMotDePasse (\$ pLibelle, \$ pNom, \$ pVal, \$ pTaille, \$ pTailleMax = 0, \$ classe = "", \$ libelle_width = "", \$ css_width = "", \$ br = 1, \$ readonly = 0)

Création d'un champ Password et son libellé : insertion des parametres dans le tableau des composants

3.9.2.15 Form : :champRadio (\$ libelle, \$ nom, \$ valeur, \$ liste, \$ classe = "", \$ libelle_width = "", \$ css_width = "", \$ br = 1, \$ readonly = 0, \$ OnChange = "")

Création d'un champ radio et son libellé : insertion des parametres dans le tableau des composants

3.9.2.16 Form : `:champTexte ($ libelle, $ nom, $ valeur, $ taille, $ tailleMax = 0, $ classe = "", $ libelle_width = "", $ css_width = "", $ br = 1, $ readonly = 0, $ onchange = "")`

Création d'un champ INPUT de type text et de son libellé :

`$libelle` = Libellé de ce champ.

`$nom` = nom de ce champ (nom de la variable HTML correspondant à ce champ).

`$valeur` = valeur de la variable HTML `$nom` `$taille` = valeur du parametre HTML 'SIZE'.

`$tailleMax` = valeur du parametre HTML 'MAXLENGTH'.

`$classe` = classe de l'INPUT.

`$libelle_width` = Largeur en pourcentage du block parent du libellé.

`$css_width` = Largeur en pourcentage du block parent de l'INPUT.

`$br` = fin de ligne.

`$readonly` = INPUT readonly.

`$onchange` = Code Javascript à executer si l'INPUT Change de valeur.

Insertion des parametres dans le tableau des composants

3.9.2.17 Form : `:champValider ($ pLibelle, $ pNom, $ classe = "", $ css_width = "", $ br = 1, $ OnClick = "")`

Création d'un bouton de type submit : insertion des parametres dans le tableau des composants

3.9.2.18 Form : `:changeComposant ($ idComposant, $ valeur, $ libelle, $ nom, $ liste, $ params)`

Change un(des) parametre(s) d'un composant.

Paramètres :

`$idComposant` int : identifiant du Composant.

`$valeur` string : nouvelle valeur.

`$libelle` string : nouveau libelle.

`$nom` string : nouveau nom.

`$liste` array : nouvelle liste.

`$params` array : parametre additionnels

3.9.2.19 Form : `:debutDiv ($ id_div, $ classe, $ style)`

Ajout d'un div quelconque a partir d'un identifiant.

Paramètres :

`$id_div` string : identifiant du Div.

`$classe` string : classe du Div.

`$style` string : style du Div.

3.9.2.20 Form : :finDiv ()

fin du div precedemment ajoute

3.9.2.21 Form : :Form (\$ methode = "post", \$ action = "", \$ centre = TRUE, \$ classeCSS = "Form", \$ nom = "Form", \$ OnSubmit = "")

!Constructeur de la classe

3.9.2.22 Form : :getChamp (\$ idComposant)

Méthode permettant de récupérer un champ par son identifiant.

\$idComposant = identifiant du composant (renvoyé à sa création)

3.9.2.23 Form : :SetIdentifiant (\$ valeur, \$ numero_ligne)

SetIdentifiant() (p. ??) ajoute un checkbox pour identifier le resultat selectionne.

Paramètres :

\$valeur string valeur de la checkbox.

\$numero_ligne int : numero de la ligne de resultat (permet d'identifier la checkbox.

Réimplémentée dans **Resultats p.** (classResultats_{825633ae5d7c06cee31de0dfe8817b19} ??)

3.9.3 Documentation des données membres

3.9.3.1 Form : :\$help_by = "tooltips"

variable qui indique si il faut ou non afficher le help et sous quel form (tooltips,bouton,null).

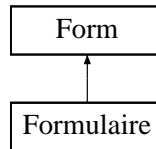
par default le help s'affiحه par tooltips.

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/Form.class.php

3.10 Référence de la classe Formulaire

Graphe d'héritage de Formulaire :



Fonctions membres publiques

- `genereHTML ()`

3.10.1 Description détaillée

Class **Formulaire** (p. ??) Classe qui permet de generer le code HTML d'un formulaire

3.10.2 Documentation des fonctions membres

3.10.2.1 Formulaire : `:genereHTML ()`

Génère le code HTML d'un objet **Formulaire** (p. ??)

Le formulaire a un attribut ENCTYPE si on transfère un fichier (la méthode **champFichier()** (p. ??) a été appelée)

on parcourt les composants pour créer le HTML

On agit selon le type du composant

On retourne la chaîne de caractères contenant le formulaire

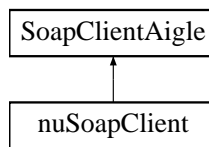
Réimplémentée à partir de **Form p.** (classForm ??)

La documentation de cette classe a été générée à partir du fichier suivant :

- `/mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/Formulaire.class.php`

3.11 Référence de la classe nuSoapClient

Graphe d'héritage de nuSoapClient :



Fonctions membres publiques

- addSoapParam (\$name, \$value, \$type="")
- cleanSoapParams ()
- encodeChaine (\$chaine)

3.11.1 Description détaillée

Sous-classe de la classe abstraite SoapClient, implantant l'accès au serveur SOAP

Pas de propriétés : elles sont héritées de la classe SoapClient.

Pas de constructeur : lui aussi est hérité.

3.11.2 Documentation des fonctions membres

3.11.2.1 nuSoapClient : :addSoapParam (\$ name, \$ value, \$ type = ")

Ajout d'un paramètre (avant de faire le callMethod).

Paramètres :

\$name string : nom du paramètre.

\$value string : valeur du paramètre.

\$type string : type du paramètre

Réimplémentée à partir de SoapClientAigle p. (classSoapClientAigle_{6d583ac7f8bdb66d8c17c04753288020} ??)

3.11.2.2 nuSoapClient : :cleanSoapParams ()

Efface tous les paramètres , permet de reconstruire les paramètres pour un autre appel

Réimplémentée à partir de SoapClientAigle p. (classSoapClientAigle_{64e632008349c87beaada46e4cf33411} ??)

3.11.2.3 nuSoapClient : :encodeChaine (\$ chaine)

encodage au format du serveur soap (utf8 pour HAL).

Paramètres :

\$chaine string : chaîne à encoder.

Renvoi :

string

Réimplémentée à partir de **SoapClientAigle p.** (classSoapClientAigle_{06c5b4f1a73e3043c336e2282773e5f5}??)

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/SOAPnusoap.class.php

3.12 Référence de la classe OAI

Fonctions membres publiques

- OAI (\$bd, \$set)
- GetListIdentifiers (\$from="", \$wait_img=0)
- GetInfoPubli (\$identifiant)
- GetPubli (\$identifiant, \$show_warn=1, \$in_session=1)

3.12.1 Description détaillée

Classe OAI (p. ??)

3.12.2 Documentation des fonctions membres

3.12.2.1 OAI : :GetInfoPubli (\$ *identifiant*)

GetInfoPubli : Renvoie un tableau contenant le typedoc et le titre du document HAL dont l'identifiant est passe en parametre ;

Paramètres :

\$identifiant string : identifiant HAL OAI (p. ??) d'un document (hal-123456_v1) .

3.12.2.2 OAI : :GetListIdentifiers (\$ *from* = " ", \$ *wait_img* = 0)

Retourne un tableau contenant une liste d'identifiants de documents HAL.

Paramètres :

\$from string : date a partir de laquelle on doit rechercher les identifiants (format SQL YYYY-MM-DD).

\$wait_img boolean : 1=afficher le div d'attente de reponse de HAL, 0 ne rien afficher.

3.12.2.3 OAI : :GetPubli (\$ *identifiant*, \$ *show_warn* = 1, \$ *in_session* = 1)

Renvoie un tableau contenant :

- l' Objet Publi avec les caracteristiques du documents HAL dont l'identifiant est passe en parametre ;
- le tableau des URLs HAL des textes integraux du document.

Paramètres :

\$identifiant string : identifiant HAL OAI (p. ??) d'un document (hal-123456_v1)..

\$show_warn boolean : affiche (1) ou non (0) les messages.

\$in_session boolean : met en (ou restore de la) session le document trouve.

Renvoie :

Array(Object, Array)

3.12.2.4 OAI : :OAI (\$ *bd*, \$ *set*)

Constructeur de la classe.

l'url **OAI** (p. ??) est "http ://hal.archives-ouvertes.fr/oai/oai.php".

Paramètres :

\$set string : sigle du Laboratoire pour la recherche sur HAL.

\$bd object : Objet Base de donnée.

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/OAI.class.php

3.13 Référence de la classe Publication

Fonctions membres publiques

- Publication (\$bd, \$typedoc="", \$ref_preprint="")
- SetAttribVal (\$attrib_name, \$attrib_value, \$magic_quotes=0)
- SetAttribsVals (\$param_tbl, \$magic_quotes=0)
- GetRequiredAttribs ()
- SetRequiredAttrib (\$nom, \$require=1)
- Details (\$HAL_files, \$xml_id="", \$just_details=0)
- searchPubli ()
- compare_flou (\$hal_string, \$aigle_string)
- InsertPubli (&\$HAL_files, \$verbose=1)
- GetChampsDate ()
- GetLibelle (\$name, \$req_icon=1)

3.13.1 Description détaillée

\$Aigle : Classes/Publication.class.php , Dernière Modification : 17 Jul 2008 : 17 :16

Id

Publication.class.php,v 1.32 2008/08/21 12 :01 :48 caron Exp

Copyright 1999-2007 Dominique Caron <d.caron_AT_LPTA.univ-montp2.fr>

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://www.fsf.org/licenses/licenses/lgpl.html>.

3.13.2 Documentation des fonctions membres

3.13.2.1 Publication : :compare_flou (\$ hal_string, \$ aigle_string)

Fonction de comparaison flou de titre de document.

Paramètres :

\$hal_string : titre du premier document.

\$aigle_string : titre du second document.

Renvoie :

int

3.13.2.2 Publication : :Details (\$ HAL_files, \$ xml_id = "", \$ just_details = 0)

Affiche le details d'une publi avant son insertion .

Paramètres :

\$HAL_files Array : tableau des URL des texte integraux des publi (utilise pour hal_import).

\$xml_id int : identifiant XML si il s'agit d'un publication issue d'un fichier XML EndNote.

\$just_details boolean : 0 : recherche de doc semblable et + , 1 afficher juste le details de la publi

3.13.2.3 Publication : :GetChampsDate ()

Methode qui retourne la liste des champs date par ordre de pertinence

3.13.2.4 Publication : :GetLibelle (\$ name, \$ req_icon = 1)

!Methode retournant le libelle d'un attribut.

Paramètres :

\$name string : nom de l'attribut.

\$req_icon boolean : 1=Ajouter l'icone indiquant que le champ est obligatoire.

Renvoie :

char : le libelle de l'attribut.

3.13.2.5 Publication : :GetRequiredAttribs ()

GetRequiredAttribs () (p.??) retourne un Array contenant tous les attributs OBLIGATOIRE de la publi

Renvoie :

Array

3.13.2.6 Publication : :InsertPubli (&\$ HAL_files, \$ verbose = 1)

Methode inserant une Publi a partir des informations recueillies sur le XML d'OAI-HAL

Paramètres :

\$HAL_files array : le tableau des url des fichiers sur HAL (si il y en a).

\$verbose boolean : affiche des messages , sinon genere une chaine de \$msg.

\$HAL_files est le tableau des url des fichiers sur HAL (si il y en a).

\$verbose = 1; affiche des messages , sinon genere une chaine de \$msg.

Renvoie :

string ref_preprint si tout est OK.

0 en cas de Pb.

3.13.2.7 Publication : :Publication (\$ bd, \$ typedoc = "", \$ ref_preprint = "")

!Constructeur de la classe.

Paramètres :

\$bd object : BdD.

\$typedoc string : Type de Document.

\$ref_preprint string : Reference interne du document

3.13.2.8 Publication : :searchPubli ()

Verifie l'existence d'un document ressemblant a la publication

Retourne un tableau (article=>objet(\$article),poids=>'veracite de la detection').

objet (\$article) correspondant au document trouve.

Les champs ont des poids qui vont nous permettre d'estimer la veracite de la comparaison : Titre strictement ou presque identique avec des auteurs_inlab identique = 6 points.

Titre strictement ou presque identique = 4 points.

hal_ref identique = 2 points (sans hal_ref cote AIGLe = 0 points) ,
hal_ref differente = -4 points.

typedoc identique = 1 point , typedoc differen=-1 point doi identique = 2 points (sans doi HAL ou AIGLe = 0 points) , doi different=-1 point.

On peut considerer que la veracite est bonne si le poids >= 6

3.13.2.9 Publication : :SetAttribsVals (\$ param_tbl, \$ magic_quotes = 0)

Methode pour affecter les valeurs de plusieurs attributs.

Paramètres :

\$param_tbl array : Tableau des attributs => valeurs.

\$magic_quotes boolean : \$magic_quotes.

3.13.2.10 Publication : :SetAttribVal (\$ attrib_name, \$ attrib_value, \$ magic_quotes = 0)

Methode pour affecter une valeur a un attribut.

Paramètres :

\$attrib_name string : nom de l'attribut.

\$attrib_value string : valeur de l'attribut.

\$magic_quotes boolean : \$magic_quotes

3.13.2.11 Publication : :SetRequiredAttrib (\$ nom, \$ require = 1)

Change la valeur 'required' d'un attribut .

Paramètres :

\$nom string : nom de l'attribut.

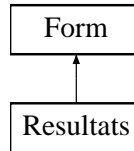
\$require boolean : 0 ou 1.

La documentation de cette classe a été générée à partir du fichier suivant :

- /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/Publication.class.php

3.14 Référence de la classe Resultats

Graphe d'héritage de Resultats : :



Fonctions membres publiques

- `GetListLimit ()`
- `ajoutSelectAll ($Select, $overwrite=0)`
- `SetIdentifiant ($valeur, $numero_ligne=?)`
- `ajoutTitre ($texte, $classe, $css_width, $br)`
- `genereHTML ($nbLignesTot=0)`

Attributs publics

- `$showprint = 1`

3.14.1 Description détaillée

Classe gerant les resultats de recherche SQL

Cette classes genere un formulaire (pour faire des requetes eventuelles sur un ou des resultats particuliers). char \$method : methode du formulaire (POST ou GET)

char \$action : script d'action à la soumission du formulaire

bool \$centre : formulaire centree ou non

char \$classeCSS : la classe du formulaire

char \$nom : nom du formulaire

char \$OnSubmit : code JS a executer au submity du formulaire

Elle inclut un numeroteur. Le numeroteur envoie dans le formulaire la variable 'rech_class_offset'

Il existe une methode (**ajoutSelectAll ()** (p.??)) permettant d'ajouter une checkbox a chaque entrée de résultat et un select (name='rech_class_todo') permettant de choisir l'action a executer sur les entrees selectionnées. la liste des entrees selectionnées est passée dans le tableau : 'rech_class_identifiant[]'. L'identifiant de la ligne (la valeur de la checkbox associée) est passée par la méthode Set-Identifiant (\$valeur, \$numero_ligne=").

ATTENTION *****

Un résultat (une ligne de résultat) est déterminée (se termine) par la présence d'un composant dont le bool \$br est à 1. Le passage à la ligne dans la présentation (d'un résultat) est à la charge du développeur et s'effectue en complétant à 100% les \$libelle_width et \$css_width.

3.14.2 Documentation des fonctions membres

3.14.2.1 Resultats : :ajoutSelectAll (\$ Select, \$ overwrite = 0)

A l'appel de cette methode, chaque entree pourra avoir une checkbox associee (voir Methode setIdentifiant(\$valeur,\$numero_ligne=") , et un SELECT permettra de choisir l'action a executer sur les entrees cochees.

Le formulaire sera soumi (avec demande de confirmation) des le changement du SELECT.

Les parametres 'rech_class_todo' (option du select choisie) et 'rech_class_identifiant[]' (liste des entrees cochees) sont passes dans le formulaire.

Paramètres :

\$Select Array : Tableau associatif des actions possibles (rech_class_todo).

\$overwrite boolean : ecrase le un precedent \$Select si il en existe un

Si la methode a deja ete appelee et qu'on la rappelle sans \$overwrite=1 alors on 'return 0', sinon on ecrase l'ancien \$Select

\$Select = Array() :Tableau associatif des option=>valeur , decrivant le SELECT qui sera genere pour les actions a faire sur les entres cochees.

On ajoute TOUJOURS l'action : 0=>' ' au debut de la liste des valeur=>option

On creer le HTML du select en appelant la methode privee champSelect() Réimplémentée à partir de **Form p.** (classForm_{84dd9147ca546970c38c8c41d423490f}??)

3.14.2.2 Resultats : :ajoutTitre (\$ texte, \$ classe, \$ css_width, \$ br)

Methode qui ajoute un texte restant dans toutes les pages de resultats.

Paramètres :

\$texte string : texte a ajouter.

\$classe string : classe du div contenant le texte.

\$css_width int : largeur du div contenant le texte.

\$br boolean : 1 = fin de ligne, 0 par de fin de ligne.

Réimplémentée à partir de **Form p.** (classForm_{4bbddc1126c1102713a52b4bddb1df2d}??)

3.14.2.3 Resultats : :genereHTML (\$ nbLignesTot = 0)

Methode generant le code HTML : on peut faire une version qui effectue directement les 'echo', ce qui evite de transmettre une grosse chaine de caracteres en retour?).

La variable sert pour le numeroteur. \$nbLignesTot= nombre totale de resultats (On a passe a l'objet que les resultats a afficher - et non pas l'ensembles des resultats), si \$nbLignesTot<=Nbre de lignes passees a l'object alors il n'y aura pas de numeroteur.

rech_class_offset = nom du champ hidden pour la valeur de l'offset , la valeur sera attribue par le numeroteur

numeroteur : Uniquement si :

Le nombre total de ligne (\$nbLignesTot) a ete passe en argument de la methode genereHTML().

On parcourt les composants et on cree le HTML

Agissons selon le type de la ligne

retourne la chaîne de caracteres contenant les resultats

3.14.2.4 Resultats : :GetListLimit ()

Retourne le nombre de resultat affiche par Page.

3.14.2.5 Resultats : :SetIdentifiant (\$ valeur, \$ numero_ligne = "")

Methode qui attribut la valeur de la checkbox (**ajoutSelectAll**) (p. ??) d'une ligne de resultat.

\$valeur : valeur de la checkbox pour la ligne.

Si le numero de ligne (\$numero_ligne) n'est pas passe en parametre alors c'est la ligne courante

On enregistre la valeur \$valeur dans le tableau (prive) des identifiants a l'index \$numero_ligne

Réimplémentée à partir de **Form p.** (classForm_{377066bd3cbbbfe50fd3f22ebf1ad??})

3.14.3 Documentation des données membres

3.14.3.1 Resultats : :\$showprint = 1

Cette variable permet de definir si oui ou non l'objet Resultat a un (2) bouton(s) d'impression du tableau .

\$showprint=0 pas de bouton d'impression.

\$showprint=1 par default. un bouton d'impression sous le tableau.

\$showprint=2 un bouton d'impression au dessus du tableau et en dessous.

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/Resultats.class.php

3.15 Référence de la classe Session

Fonctions membres publiques

- Session (\$nom_session)
- SetInSession (\$attribut, \$valeur, \$nom_module="", \$nom_objet="")
- GetFromSession (\$attribut="", \$nom_module="", \$nom_objet="")
- DelFromSession (\$attribut="", \$nom_module="", \$nom_objet="")
- DelArrayFromSession (\$nom_objet, \$nom_module="")
- CloseSession ()

Attributs publics

- \$nom_session

3.15.1 Description détaillée

class Session (p. ??)

3.15.2 Documentation des fonctions membres

3.15.2.1 Session : :CloseSession ()

Fermeture de la Session (p. ??)

3.15.2.2 Session : :DelArrayFromSession (\$ nom_objet, \$ nom_module = "")

Efface un tableau enregistré en session

Paramètres :

\$nom_objet string : Nom de l'objet qui rappelle la session

\$nom_module string : Nom du module qui rappelle la session

3.15.2.3 Session : :DelFromSession (\$ attribut = "", \$ nom_module = "", \$ nom_objet = "")

DelFromSession() (p. ??) Efface une valeur enregistrée en session de

Paramètres :

\$attribut string : Nom de l'attribut pour lequel on efface la valeur

\$nom_module string : Nom du module qui rappelle la session

\$nom_objet string : Nom de l'objet qui rappelle la session (ou pour lequel on veut effacer les valeurs
- tableau -)

3.15.2.4 Session : :GetFromSession (\$ attribut = "", \$ nom_module = "", \$ nom_objet = "")

GetFromSession() (p. ??) renvoie la valeur enregistrée en session

Paramètres :

\$attribut string : Nom de l'attribut pour lequel on souhaite la valeur

\$nom_module string : Nom du module qui rappelle la session

\$nom_objet string : Nom de l'objet qui rappelle la session (ou pour lequel on souhaite recuperer le tableau des valeurs)

Renvoi :

mixed

3.15.2.5 Session : :Session (\$ nom_session)

Session (p. ??) Objet gerant les valeurs a enregistrer/lire dans la session.

Paramètres :

\$nom_session string : Nom de la session

3.15.2.6 Session : :SetInSession (\$ attribut, \$ valeur, \$ nom_module = "", \$ nom_objet = "")

SetInSession() (p. ??). Enregistre en session :

Paramètres :

\$attribut string : Nom de l'attribut a metre en **Session** (p. ??)

\$valeur mixed : Valeur a mettre en session

\$nom_module string : nom du module qui fait appelle a la session

\$nom_objet string : Nom de l'objet du module qui fait appelle a la session

3.15.3 Documentation des données membres

3.15.3.1 Session : :\$nom_session

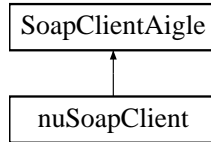
Nom de la session

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/Session.class.php

3.16 Référence de la classe SoapClientAigle

Graphe d'héritage de SoapClientAigle :



Fonctions membres publiques

- SoapClientAigle (\$code, \$url='http://hal.archives-ouvertes.fr/ws/')
- callMethod (\$methode, &\$params)
- addSoapParam (\$name, \$value, \$type)
- cleanSoapParams ()
- encodeChaine (\$chaine)
- messageSOAP ()

3.16.1 Description détaillée

Classe abstraite définissant une interface générique d'accès à une API SOAP

3.16.2 Documentation des fonctions membres

3.16.2.1 SoapClientAigle : :addSoapParam (\$ name, \$ value, \$ type)

Ajout d'un paramètre (avant de faire le callMethod).

Paramètres :

\$name string : nom du paramètre SOAP.

\$value string : valeur du paramètre SOAP.

\$type string : type du paramètre SOAP.

Réimplémentée dans **nuSoapClient p.** (classnuSoapClient_0ab71c89c7a4235dea31943942157 ??)

3.16.2.2 SoapClientAigle : :callMethod (\$ methode, &\$ params)

Méthode d'appel à une méthode.

La méthode appelle aux méthodes privées soap_call() pour exécuter l'appel et messageSOAP() (p. ??) pour envoyer le message d'erreur éventuel

Paramètres :

\$methode string : Méthode à appeler.

\$params array : tableau des paramètres à passer à la méthode

3.16.2.3 SoapClientAigle : :cleanSoapParams ()

Efface tous les parametres , permet de reconstruire les param pour u nautre call.

Réimplémentée dans **nuSoapClient p.** (classnuSoapClient_{4867157aa04d03f9332f7e718543eb42 ??})

3.16.2.4 SoapClientAigle : :encodeChaine (\$ chaine)

encodage au format du serveur soap (utf8 pour HAL).

Paramètres :

\$chaine string : chaine a encoder.

Renvoi :

string

Réimplémentée dans **nuSoapClient p.** (classnuSoapClient_{b788ece4dda1316777139a24b37ca67 ??})

3.16.2.5 SoapClientAigle : :messageSOAP ()

Retour du message d'erreur

3.16.2.6 SoapClientAigle : :SoapClientAigle (\$ code, \$ url = 'http://hal.archives-ouvertes.fr/ws/')

Constructeur de la classe SoapClient

Le constructeur initialise les variables du client soap

Paramètres :

\$code string : Url (Pour HAL : ref, ou search ou submit).

\$url string : Url du WebServer.

La documentation de cette classe a été générée à partir du fichier suivant :

– /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/SOAP.class.php

3.17 Référence de la classe TypeDoc

Fonctions membres publiques

- `Typedoc ($typedoc=)`
- `GetAttributs ($which='all')`
- `GetNeedFile ($typedoc=)`
- `GetTypeDoc ($show_warn=1, $hal=0)`
- `GetAttribLibelle ($attrib_nom, $typedoc=)`

Attributs publics

- `$Attributs`
- `$LaboAttributs = Array()`
- `$typedoc`

3.17.1 Description détaillée

\$Aigle : Classes/TypeDoc.class.php , Dernière Modification : 17 Jul 2008 : 17 :15

Id

TypeDoc.class.php,v 1.10 2008/08/21 12 :01 :49 caron Exp

Copyright 1999-2007 Dominique Caron <d.caron_AT_LPTA.univ-montp2.fr>

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://www.fsf.org/licenses/licenses/lgpl.html>.

3.17.2 Documentation des fonctions membres

3.17.2.1 TypeDoc : :GetAttribLibelle (\$ attrib_nom, \$ typedoc = ")

`GetAttribLibelle($attrib_nom)` : renvoie le libelle d'un attribut de publication.

Paramètres :

\$attrib_nom string :nom de l'attribut dont on veut le libelle.

\$typedoc string type du document.

Renvoie :

string : le libelle de l'attribut (ou son nom si le libelle n'a pas ete trouve dasn le tableau.

3.17.2.2 TypeDoc : :GetAttributs (\$ which = 'all')

!Methode publique **GetAttributs()** (p.??) . retourne un tableau des attributs du type de publication.

Paramètres :

\$which string

all = tous les attributs (par default) ,
 spec = specifiques aux type de document,
 hal = uniquement les attributs hal,
 aigle = uniquement les attributs aigle.

Renvoie :

```
array $attributs
```

3.17.2.3 TypeDoc : :GetNeedFile (\$ typedoc = "")

!Methode qui renvoie en entier dont la valeur indique le 'besoin' de fichier attache.

Paramètres :

\$typedoc string : type de document.

Renvoie :

```
int : 0 => pas de document attache.  
1 => document attache possible.  
2 => document attache ou 'ref arxiv' obligatoire .  
3 => document attache obligatoire.
```

3.17.2.4 TypeDoc : :GetTypeDoc (\$ show_warn = 1, \$ hal = 0)

!Methode qui controle si les type de documents ont changes sur HAL et retourne le tableau des type de document

3.17.2.5 TypeDoc : :Typedoc (\$ typedoc = "")

!Constructeur de la classe abstraite

3.17.3 Documentation des données membres**3.17.3.1 TypeDoc : :\$Attributs**

array Tableau des attribut du type de Document

3.17.3.2 TypeDoc : :\$typedoc

string : type du document

La documentation de cette classe a été générée à partir du fichier suivant :

```
- /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/TypeDoc.class.php
```

3.18 Référence de la classe User

Fonctions membres publiques

- `User ($bd, $id_user=0, $username=)`
- `Services ($year=)`
- `TachesAdmin ($year=)`
- `Responsables ()`
- `Distinctions ()`
- `OtherThemes ()`

3.18.1 Description détaillée

Classe `OAI` (p.??)

3.18.2 Documentation des fonctions membres

3.18.2.1 `User : :Distinctions ()`

`Distinctions ()` (p.??) : affect les distinctions du user (variable array `$this->distinctions`)

3.18.2.2 `User : :OtherThemes ()`

public `OtherThemes ()` (p.??) affect les themes (non principaux) du user (variable array `$this->other_themes`)

3.18.2.3 `User : :Responsables ()`

`Responsables ()` (p.??) : affecte les responsables du theme du user (variable array `$this->theme_responsables`). TODO : BUG Il faut des responsables de theme PAR ANNEE!!!

3.18.2.4 `User : :Services ($ year =)`

Affecte les l'etat de service du user pour l'annee `$year`.

Paramètres :

`$year` int : annee pour laquelle on demande l'etat de service.

3.18.2.5 `User : :TachesAdmin ($ year =)`

Affecte les taches administratives du user pour l'annee `$year`.

Paramètres :

`$year` int : annee pour laquelle on demande les Taches Administratives

3.18.2.6 User ::User (\$ *bd*, \$ *id_user* = 0, \$ *username* = ")

Constructeur de la classe.

Paramètres :

\$bd object : Objet Base de donnée. L'instanciation peut se faire sur l'*id_user* ou sur le *username*.

\$id_user int : id de l'utilisateur dans la base AIGLe.

\$username string : *username* de l'utilisateur dans la base AIGLe.

La documentation de cette classe a été générée à partir du fichier suivant :

- /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/User.class.php

3.19 Référence de la classe WaitImg

Fonctions membres publiques

- WaitImg (\$timeout, \$title, \$texte)
- ChgTexte (\$texte)
- ChgTitre (\$texte)
- End ()
- Start (\$timeout=0)

3.19.1 Description détaillée

Classe definissant un div contenant un 'roue tournante' pour attendre un evenement

3.19.2 Documentation des fonctions membres

3.19.2.1 WaitImg : :ChgTexte (\$ *texte*)

Methode de changement de texte.

3.19.2.2 WaitImg : :ChgTitre (\$ *texte*)

Methode de changement de titre.

3.19.2.3 WaitImg : :End ()

Methode pour cacher le div d'image

3.19.2.4 WaitImg : :Start (\$ *timeout* = 0)

Methode pour (re)montrer le div d'image

3.19.2.5 WaitImg : :WaitImg (\$ *timeout*, \$ *title*, \$ *texte*)

Constructeur de la classe **WaitImg** (p.??)

\$timeout : temps max d'attente

\$title : Titre de l'attente.

\$texte : texte en grisee.

La documentation de cette classe a été générée à partir du fichier suivant :

- /mnt/sshfs/AIGLE-DEVEL/aigle-main/Classes/WaitImg.class.php