

Gestor de Contenidos de Vídeo bajo Demanda Manual de referencia

Id

Doxyfile 78 2007-02-17 16:26:10Z alberto

Generado por Doxygen 1.5.1

Sun Apr 8 17:48:04 2007

Índice general

1. Gestor de Contenidos de Vídeo bajo Demanda Índice jerárquico	1
1.1. Gestor de Contenidos de Vídeo bajo Demanda Jerarquía de la clase	1
2. Gestor de Contenidos de Vídeo bajo Demanda Índice de clases	3
2.1. Gestor de Contenidos de Vídeo bajo Demanda Lista de componentes	3
3. Gestor de Contenidos de Vídeo bajo Demanda Documentación de clases	5
3.1. Referencia de la Clase ApfAdminPage	5
3.2. Referencia de la Clase ApfAjax	7
3.3. Referencia de la Clase ApfAuth	8
3.4. Referencia de la Clase ApfAuthBase	10
3.5. Referencia de la Clase ApfBaseDocument	13
3.6. Referencia de la Clase ApfBaseVoDHandler	17
3.7. Referencia de la Clase ApfButton	19
3.8. Referencia de la Clase ApfDocument	20
3.9. Referencia de la Clase ApfEditPage	25
3.10. Referencia de la Clase ApfFileSystem	27
3.11. Referencia de la Clase ApfFileSystemRPC	28
3.12. Referencia de la Clase ApfFolder	29
3.13. Referencia de la Clase ApfHttpVoDHandler	31
3.14. Referencia de la Clase ApfLocal	32
3.15. Referencia de la Clase ApfLoginPage	34
3.16. Referencia de la Clase ApfMainPage	35
3.17. Referencia de la Clase ApfManager	36
3.18. Referencia de la Clase ApfMediaPage	39
3.19. Referencia de la Clase ApfMysqlIDB	40
3.20. Referencia de la Clase ApfNoteBook	42
3.21. Referencia de la Clase ApfNoteBookTab	43
3.22. Referencia de la Clase ApfRPCServer	44

3.23. Referencia de la Clase ApfSimplePage	46
3.24. Referencia de la Clase ApfTestPage	47
3.25. Referencia de la Clase ApfTree	48
3.26. Referencia de la Clase ApfUploadPage	50
3.27. Referencia de la Clase ApfVideo	51
3.28. Referencia de la Clase ApfVideoLanVoDHandler	52
3.29. Referencia de la Clase ApfVideoPage	53
3.30. Referencia de la Clase ApfVoDFile	54
3.31. Referencia de la Clase Builder	55
3.32. Referencia de la Interfaz iAuth	56
3.33. Referencia de la Interfaz iDB	59
3.34. Referencia de la Interfaz iDocument	60
3.35. Referencia de la Interfaz iFileController	61
3.36. Referencia de la Interfaz iLogger	62
3.37. Referencia de la Clase IncorrectPasswordException	63
3.38. Referencia de la Clase InvalidFileException	64
3.39. Referencia de la Interfaz iPlayer	65
3.40. Referencia de la Interfaz iVodFile	66
3.41. Referencia de la Interfaz iVoDHandler	67
3.42. Referencia de la Clase MediaMGR	69
3.43. Referencia de la Clase ModuleNotFoundException	74
3.44. Referencia de la Clase NotImplementedException	75
3.45. Referencia de la Clase PHPError	76
3.46. Referencia de la Clase Request	77
3.47. Referencia de la Clase SimpleTcpClient	78
3.48. Referencia de la Clase UnknownResourceException	79
3.49. Referencia de la Clase UploadCtrl	80
3.50. Referencia de la Clase VideoLanTelnetClient	81
3.51. Referencia de la Clase VoDFactory	83

Capítulo 1

Gestor de Contenidos de Vídeo bajo Demanda Índice jerárquico

1.1. Gestor de Contenidos de Vídeo bajo Demanda Jerarquía de la clase

Esta lista de herencias esta ordenada rigurosamente, pero no completa, por orden alfabético:

ApfAjax	7
ApfButton	19
ApfFileSystem	27
ApfFolder	29
ApfVideo	51
ApfLocal	32
ApfNoteBook	42
ApfNoteBookTab	43
ApfTree	48
ApfVoDFile	54
Builder	55
iAuth	56
ApfAuth	8
ApfAuthBase	10
ApfAuth	8
iDB	59
ApfMysqlDB	40
iDocument	60
ApfBaseDocument	13
ApfDocument	20
ApfManager	36
ApfAdminPage	5
ApfEditPage	25
ApfLoginPage	34
ApfMainPage	35
ApfMediaPage	39
ApfVideoPage	53
ApfSimplePage	46

ApfFileSystemRPC	28
ApfRPCServer	44
ApfTestPage	47
ApfUploadPage	50
ApfDocument	20
ApfEditPage	25
ApfFileSystemRPC	28
ApfLoginPage	34
ApfMainPage	35
ApfManager	36
ApfMediaPage	39
ApfRPCServer	44
ApfSimplePage	46
ApfTestPage	47
ApfUploadPage	50
ApfVideoPage	53
iFileController	61
iLogger	62
IncorrectPasswordException	63
InvalidFileException	64
iPlayer	65
iVodFile	66
iVodHandler	67
ApfBaseVoDHandler	17
ApfHttpVoDHandler	31
ApfVideoLanVoDHandler	52
ApfHttpVoDHandler	31
ApfVideoLanVoDHandler	52
MediaMGR	69
ModuleNotFoundException	74
NotImplementedException	75
PHPErrror	76
Request	77
SimpleTcpClient	78
VideoLanTelnetClient	81
UnknownResourceException	79
UploadCtrl	80
VoDFactory	83

Capítulo 2

Gestor de Contenidos de Vídeo bajo Demanda

Índice de clases

2.1. Gestor de Contenidos de Vídeo bajo Demanda Lista de componentes

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

ApfAdminPage	5
ApfAjax (Clase base para AJAX (Asynchronous JavaScript And XML))	7
ApfAuth (Modulo de autenticación)	8
ApfAuthBase (Clase base abstracta del modulo de autenticación)	10
ApfBaseDocument (Clase del documento base)	13
ApfBaseVoDHandler	17
ApfButton (Un botón)	19
ApfDocument (Documento base, con acceso a la base de datos y control de sesiones)	20
ApfEditPage (Página de edición de medios/categorías)	25
ApfFileSystem (Clase de gestion de ficheros)	27
ApfFileSystemRPC (Clase página del gestor)	28
ApfFolder (Carpeta o directorio (Nodo) del sistema virtual de ficheros)	29
ApfHttpVoDHandler (Implementa los manejadores necesarios para un servidor VoD sobre HTTP)	31
ApfLocal	32
ApfLoginPage (Página de autenticación)	34
ApfMainPage (Página principal)	35
ApfManager (Clase página del gestor)	36
ApfMediaPage (Navegador de la biblioteca multimedia. (Sistema virtual de ficheros))	39
ApfMysqlDB (Classe base de datos (MySQL))	40
ApfNoteBook (Clase NoteBook (Un conjunto de páginas agrupadas))	42
ApfNoteBookTab (Clase página de un libro)	43
ApfRPCServer (Clase página del gestor)	44
ApfSimplePage (Clase base para páginas simples)	46
ApfTestPage (Test class)	47
ApfTree (Crea un arbol del sistema virtual de ficheros)	48
ApfUploadPage (Clase página del gestor)	50
ApfVideo (Clase representativa de un video)	51

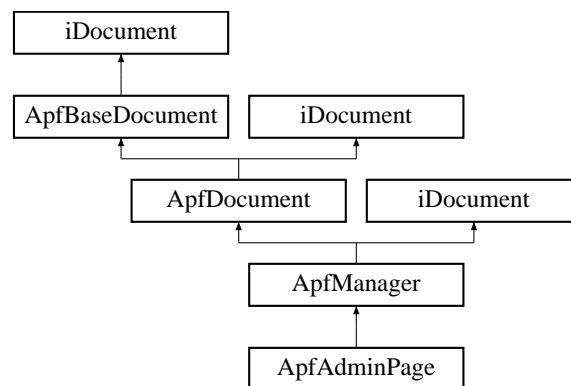
ApfVideoLanVoDHandler (Implementa los manejadores necesarios para un servidor VoD VideoLan)	52
ApfVideoPage (Página del vídeo)	53
ApfVoDFile (Clase que define un video, con sus respectiva propiedades)	54
Builder (Los constructores de cosas)	55
iAuth (Interfaz de autenticación)	56
iDB (Interfaz de Base de Datos)	59
iDocument (Interfaz Documento)	60
iFileController (Interfaz para trabajar sobre sistemas de ficheros, reales, virtuales, etc..)	61
iLogger (Interface que define un fichero Log)	62
IncorrectPasswordException (Excepción de contraseña incorrecta)	63
InvalidFileException (Fichero no válido)	64
iPlayer (Inerfaz reproductor)	65
iVodFile (Define un video, con sus respectiva propiedades)	66
iVoDHandler	67
MediaMGR (Clase controladora de los recursos guardados en la biblioteca)	69
ModuleNotFoundException (Excpección de modulo no encontrado)	74
NotImplementedException (Excepción de funcionalidad no implementada)	75
PHPError (Lanzar errores como si fueran excepciones)	76
Request (Petición)	77
SimpleTcpClient (Cliente Tcp sencillo)	78
UnknownResourceException (Excepción de recurso desconocido)	79
UploadCtrl (Clase control de subidas)	80
VideoLanTelnetClient (Cliente VideoLAN)	81
VoDFactory (Fabrica de manejadores para diferentes servidores VoD)	83

Capítulo 3

Gestor de Contenidos de Vídeo bajo Demanda Documentación de clases

3.1. Referencia de la Clase ApfAdminPage

Diagrama de herencias de ApfAdminPage:



Métodos públicos

- **ApfAdminPage ()**
Constructor.
- **body ()**

Atributos públicos

- **\$status = 0**
Estado (0=lectura datos, 1=datos salvados).
- **\$server_type = "http"**

3.1.1. Descripción detallada

Página de administración.

3.1.2. Documentación de las funciones miembro

3.1.2.1. `ApfAdminPage::body ()`

Método cuerpo, redefine el método de la clase padre.

Reimplementado de `ApfManager` (p. 36).

3.1.3. Documentación de los datos miembro

3.1.3.1. `ApfAdminPage::$server_type = "http"`

Tipo de servidor. Posibles valores: http,rtsp,...

La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/pages/common/AdminPage.php`

3.2. Referencia de la Clase ApfAjax

Clase base para AJAX (Asynchronous JavaScript And XML).

Métodos públicos estáticos

- **static write ()**
Escribe el código base necesario para cualquier aplicación Ajax.
- **static get ()**
Escribe el código base necesario para cualquier aplicación Ajax.

3.2.1. Descripción detallada

Clase base para AJAX (Asynchronous JavaScript And XML).

3.2.2. Documentación de las funciones miembro

3.2.2.1. **static ApfAjax::get ()** [static]

Escribe el código base necesario para cualquier aplicación Ajax.

Devuelve el object XMLHttpRequest según el navegador devuelve falso si no se puede crear el objeto.

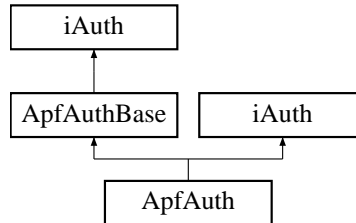
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/widgets/ajax.php

3.3. Referencia de la Clase ApfAuth

Modulo de autenticación.

Diagrama de herencias de ApfAuth:



Métodos públicos

- **__construct** (&\$database)
- **verify** (\$uid, \$hash)
- **getPassword** (\$login, \$enc="md5")
- **authSucesfull** ()

Hook llamado solo cuando la autenticación se considera exitosa.

Atributos públicos

- **\$DB**

Base de datos.

3.3.1. Descripción detallada

Modulo de autenticación.

3.3.2. Documentación del constructor y destructor

3.3.2.1. ApfAuth::__construct (&\$ database)

Constructor

Parámetros:

database Referencia a un objeto base de datos, que contenga la tabla de usuarios

3.3.3. Documentación de las funciones miembro

3.3.3.1. ApfAuth::verify (\$ uid, \$ hash)

Verifica que el usuario esta debidamente autenticado.

Parámetros:

uid Identificador del usuario.

hash Hash de comprobación.

Devuelve:

Verdadero si los datos fueron correctos, falso en caso de fallar

Implementa **iAuth** (p. 57).

3.3.3.2. ApfAuth::getPassword (\$ login, \$ enc = "md5")

Obtener el password asociado a ese Login

Parámetros:

enc Codificación (plain: plana, md5 hash md5 del password, etc...)

Implementa **iAuth** (p. 57).

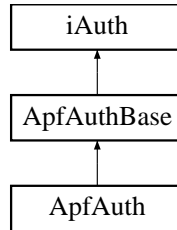
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/auth/SimpleAuth.php

3.4. Referencia de la Clase ApfAuthBase

Clase base abstracta del modulo de autenticación.

Diagrama de herencias de ApfAuthBase:



Métodos públicos

- **getLogin ()**
- **getLevel ()**
- **getUID ()**
- **getAuthHash ()**
- **getHash (\$what=“”)**
- **authenticate (\$login, \$pass, \$enc=‘plain’, \$sid=0)**
- **challenge ()**
Obtiene un desafio (CHAP).

- **hashme (\$login, \$pass, \$challenge, \$alg=‘md5’)**
- **validate (\$login, \$challenge, \$client_hash, \$alg=‘md5’)**

Atributos protegidos

- **\$login = “”**
Nombre del usuario.

- **\$hash = 0**
Hash del usuario.

- **\$uid = 0**
Identificador del usuario.

- **\$level = 0**
Nivel de acceso del usuario.

- **\$authed = False**
Indica autenticación satisfactoria.

3.4.1. Descripción detallada

Clase base abstracta del modulo de autenticación.

3.4.2. Documentación de las funciones miembro

3.4.2.1. ApfAuthBase::getHash (\$ what = " ")

Genera un hash de una información aleatoria

Parámetros:

what Entrada de datos diversos (nombre, ip, etc..)

Devuelve:

Devuelve un identificador único Es posible cambiar la función, lo importante es que el valor devuelto no pueda ser predecido por un posible atacante. Esto se utiliza principalmente para recuperar una sesión antigua, ya que las sesiones PHP no son persistentes, y expiran al cabo de un tiempo o al cerrar el navegador.

Nota:

Es necesario reflejar los cambios realizados en esta función en el las funciones de validación del script upload.py, en caso contrario el script devolverá error de validación del hash de autenticación.

Implementa **iAuth** (p. 57).

3.4.2.2. ApfAuthBase::authenticate (\$ login, \$ pass, \$ enc = 'plain', \$ sid = 0)

Autentica el usuario

Parámetros:

login Login del usuario

pass Password del usuario a comprobar

enc Codificación del password (plain: Password en texto plano, md5 hash MD5 del password)

sid Session del usuario

Devuelve:

Verdadero si los datos fueron correctos, falso en caso de fallar

Implementa **iAuth** (p. 56).

3.4.2.3. ApfAuthBase::hashme (\$ login, \$ pass, \$ challenge, \$ alg = 'md5')

Realizar el MD5(username + hashed_pass + challenge)

Parámetros:

login Login del usuario

pass Password o hash del password del usuario a comprobar

challenge Desafío

alg Algoritmo de Hash

Implementa **iAuth** (p. 58).

3.4.2.4. ApfAuthBase::validate (*\$login*, *\$challenge*, *\$client_hash*, *\$alg = 'md5'*)

Valida el usuario con el desafío fijado

Parámetros:

\$login Login del usuario

\$challenge Desafío que hemos enviado previamente al cliente

\$client_hash Respuesta al desafío por el cliente

\$alg Algoritmo de Hash utilizado

Implementa **iAuth** (p. 58).

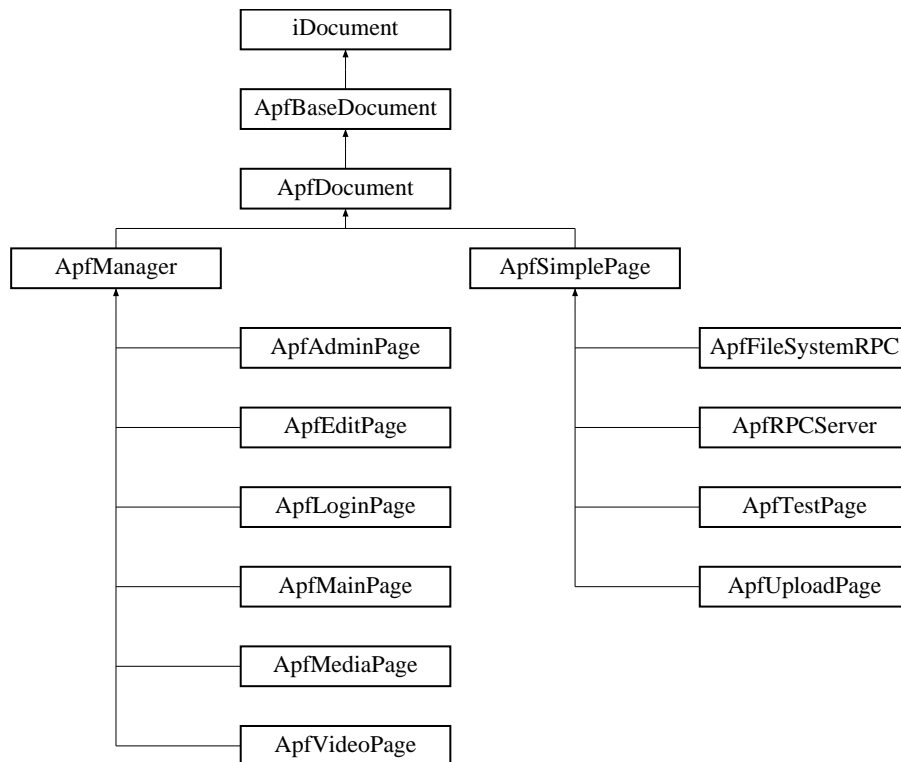
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/auth/AuthBase.php

3.5. Referencia de la Clase ApfBaseDocument

Clase del documento base.

Diagrama de herencias de ApfBaseDocument:



Métodos públicos

- **__construct** (\$title="Untitled")
- **setTitle** (\$title)
- **setBodyClass** (\$class)
- **addStyle** (\$name, \$path)
- **getTime** (\$show="")
- **head** ()
- **registerUnloadHook** (\$hook)
- **body** ()
- **foot** ()
- **print_exception** (\$e, \$die=True)
- **error** (\$msg, \$title="", \$die=False)
- **error_die** (\$msg, \$title="")
- **show** ()
 - *Genera y envía la página al cliente.*
- **buildBaseURI** (\$path="", \$relative=true)
- **getLastMod** ()
- **redirect** (\$to)

3.5.1. Descripción detallada

Clase del documento base.

3.5.2. Documentación del constructor y destructor

3.5.2.1. `ApfBaseDocument::__construct ($ title = "Untitled")`

Constructor

Parámetros:

\$title Título del documento

Reimplementado en `ApfLoginPage` (p. 34).

3.5.3. Documentación de las funciones miembro

3.5.3.1. `ApfBaseDocument::setTitle ($ title)`

Fija el título del documento

Parámetros:

\$title Título

Reimplementado en `ApfManager` (p. 37).

3.5.3.2. `ApfBaseDocument::setBodyClass ($ class)`

Fija la clase de Body

Parámetros:

class Classe

3.5.3.3. `ApfBaseDocument::addStyle ($ name, $ path)`

Añade un nuevo estilo en cascada

Parámetros:

\$name Nombre del estilo

\$path Ruta al estilo

3.5.3.4. `ApfBaseDocument::getTime ($ how = "")`

Obtener el tiempo del script actual

Parámetros:

\$show Especifica la medida (s,seconds,ms,milliseconds,us,microseconds)

3.5.3.5. `ApfBaseDocument::head ()`

Genera e imprime la cabecera del documento

Implementa `iDocument` (p. 60).

Reimplementado en `ApfManager` (p. 36), y `ApfMediaPage` (p. 39).

3.5.3.6. `ApfBaseDocument::registerUnloadHook ($ hook)`

Registra una función js a ejecutar cuando se cierre la página

Parámetros:

hook Funcion a ejecutar

3.5.3.7. `ApfBaseDocument::foot ()`

Genera e imprime el pie del documento

Implementa `iDocument` (p. 60).

Reimplementado en `ApfManager` (p. 36).

3.5.3.8. `ApfBaseDocument::error ($ msg, $ title = "", $ die = False)`

Muestra un mensaje de error

Parámetros:

msg Mensaje

title Título

die Matar aplicación?

3.5.3.9. `ApfBaseDocument::error_die ($ msg, $ title = "")`

Mostrar mensaje de error y morir

Parámetros:

msg Mensaje

title Título

3.5.3.10. `ApfBaseDocument::buildBaseURI ($ path = "", $ relative = true)`

Construye una ruta relativa al directorio de la aplicación

Parámetros:

path Url a concatenar

relative Indica si queremos la ruta relativa o absoluta

Devuelve:

Si `relative paths` es falso, devuelve la dirección completa URL `protocol://base_install` al directorio base de la instalación, sino siempre devolverá el path relativo

3.5.3.11. `ApfBaseDocument::getLastMod ()`

Devuelve la fecha de la última modificación del script en ejecución.

3.5.3.12. `ApfBaseDocument::redirect ($ to)`

Genera una redirección.

Parámetros:

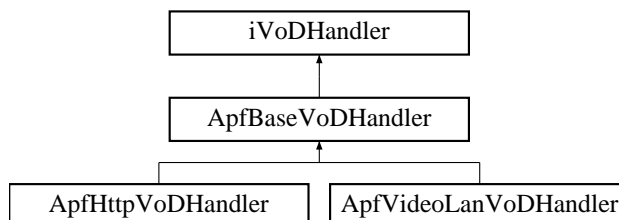
to Dirección destino

La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/pages/base/BaseDocument.php`

3.6. Referencia de la Clase ApfBaseVoDHandler

Diagrama de herencias de ApfBaseVoDHandler:



Métodos públicos

- **fileCleanner** (\$file)
Constructor.
- **getVideoFolder** ()
- **getImgFolder** ()
- **getDestinationFileName** (\$path, \$file)
- **UploadVideoFile** (\$path, \$file)
- **DeleteVideoFile** (\$file)
- **GeneratePreview** (\$path, \$file)
- **DeletePreview** (\$file)
- **doPreviewCmd** (\$path, \$dest)
- **CheckVideoFileBeforeUpload** (\$path)

3.6.1. Descripción detallada

Clase virtual que define la interfaz base para cada tipo de servidor VoD. Esta clase contiene los hooks que relacionan el servidor VoD con la aplicación.

3.6.2. Documentación de las funciones miembro

3.6.2.1. ApfBaseVoDHandler::UploadVideoFile (\$ path, \$ file)

Indica al VoD, que un nuevo fichero de video a sido subido al servidor

Parámetros:

path Ruta al fichero en el sistema local

file Nombre del fichero especificado por el usuario

Devuelve:

Devuelve el nombre del fichero subido al VoD server

Implementa **iVoDHandler** (p. 67).

3.6.2.2. ApfBaseVoDHandler::DeleteVideoFile (\$ file)

Borra el vídeo

Parámetros:

name vídeo a borrar

Implementa **iVoDHandler** (p. 68).

3.6.2.3. ApfBaseVoDHandler::GeneratePreview (\$ path, \$ file)

Se peticiona la generación de una previsualización

Parámetros:

path Ruta al fichero en el sistema local

name Nombre del fichero especificado por el usuario

Devuelve:

Devuelve la ruta a la prev

Implementa **iVoDHandler** (p. 68).

3.6.2.4. ApfBaseVoDHandler::CheckVideoFileBeforeUpload (\$ path)

Pasa un nombre de fichero al VoD para validar antes de ser subido

Parámetros:

path Nombre fichero

Devuelve:

Devuelve verdadero si el fichero es aceptable

Implementa **iVoDHandler** (p. 67).

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/vod/baseVoDhandler.php

3.7. Referencia de la Clase ApfButton

Un botón.

Métodos públicos

- `__construct` (\$parent, \$value, \$img, \$target)
- `write` ()

3.7.1. Descripción detallada

Un botón.

3.7.2. Documentación del constructor y destructor

3.7.2.1. ApfButton::__construct (\$parent, \$value, \$img, \$target)

Constructor

Parámetros:

- value* Valor
- img* Imágen
- target* Destino

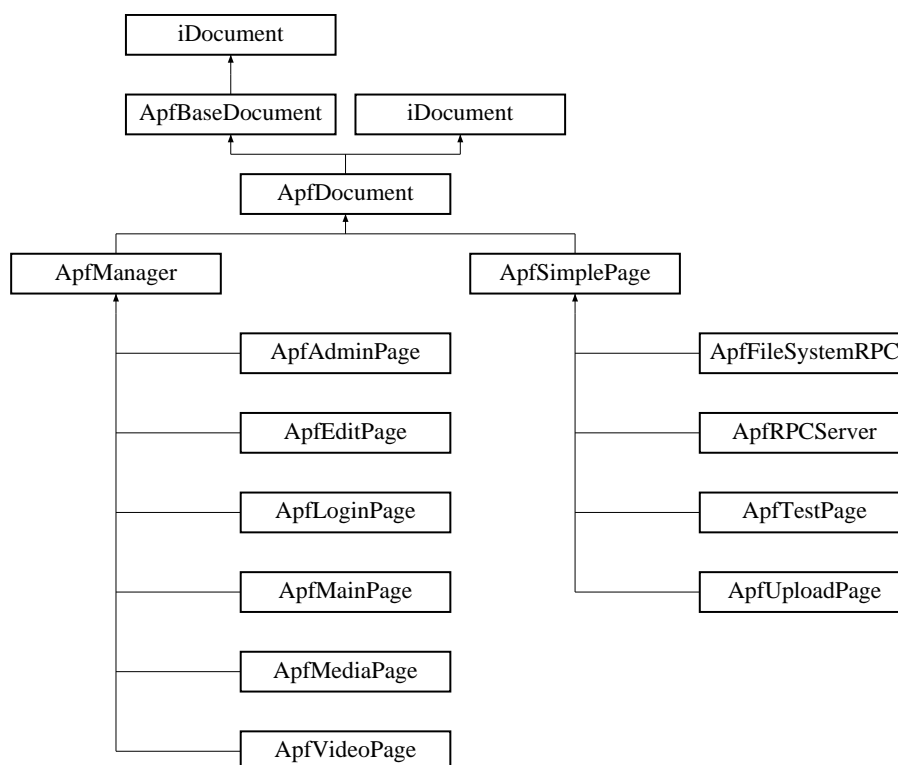
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/widgets/button.php

3.8. Referencia de la Clase ApfDocument

Documento base, con acceso a la base de datos y control de sesiones.

Diagrama de herencias de ApfDocument:



Métodos públicos

- **__construct** (\$title="", \$release_session=True)
- **startSession** (\$force=False)
- **CookieCryptAndSign** (\$data)
- **CookieDecryptAndCheck** (\$data)
- **getAuthChallenge** ()
 - Devuelve el Desafío para el método de autenticación Desafío-Respuesta.*
- **checkAuthData** (\$data, \$uidx)
- **checkLoggedUser** (\$release_session=True)
- **authenticate** (\$login, \$pass, \$enc='plain', \$sid=0)
- **authValidate** (\$login, \$challenge, \$client_hash, \$alg="md5")
- **release_session** ()
 - Libera la sesión.*
- **endSession** ()
 - Finaliza la sesión.*
- **checkPageId** ()

Obtener id de la página.

- **IAdmin ()**

Comprobar si tenemos permisos administrativos.

- **IAuthenticated ()**

Comprobar si estamos autenticados.

- **getLogin ()**

- **getId ()**

- **getUID ()**

- **setId (\$id)**

- **setParam (\$key, \$val)**

- **getParam (\$key)**

- **getArgs (\$override="", \$mask="", \$imask="", \$encode=True)**

- **getArgsHidden (\$override="", \$mask="", \$imask="")**

- **checkConnection ()**

Comprueba la conexión con la base de datos.

- **query (\$what)**

- **fetchArray ()**

Obtiene un array de los datos devueltos de la base de datos desde la última petición que devolvía datos.

- **escape_string (\$what)**

Escapa caracteres especiales como ""

- **insertId ()**

Devuelve el identificador de la ltima petición de inserción realizada a la base de datos.

- **redirect2page (\$page, \$redirect=False)**

- **getQueryCount ()**

Obtiene el número de peticiones.

3.8.1. Descripción detallada

Documento base, con acceso a la base de datos y control de sesiones.

3.8.2. Documentación del constructor y destructor

3.8.2.1. ApfDocument::__construct (\$ title = "", \$ release_session = True)

Constructor

Parámetros:

\$title Título del documento

\$release_session Indica si liberamos la sesión

Reimplementado en **ApfManager** (p. 36), y **ApfSimplePage** (p. 46).

3.8.3. Documentación de las funciones miembro

3.8.3.1. `ApfDocument::startSession ($force = False)`

Empieza la sesión, solo si existe la cookie de usuario

Parámetros:

\$force Fuerza el inicio de sesión

3.8.3.2. `ApfDocument::CookieCryptAndSign ($data)`

Firma y cifra la Cookie

Parámetros:

\$data Datos de entrada

3.8.3.3. `ApfDocument::CookieDecryptAndCheck ($data)`

Descifra y verifica la Cookie

Parámetros:

\$data Datos de entrada

3.8.3.4. `ApfDocument::checkAuthData ($data, $uidx)`

Comprueba el hash de autenticación

Parámetros:

data Datos de entrada a comprobar

Devuelve:

Verdadero si válido

3.8.3.5. `ApfDocument::checkLoggedUser ($release_session = True)`

Comprueba si el usuario ya tienen una sesión existente y valida

Parámetros:

release_session Indica si debemos liberar la sesión, para que pueda ser utilizada

3.8.3.6. ApfDocument::setParam (\$ key, \$ val)

Fija un parametro

Parámetros:

\$key Nombre del parametro

\$val Valor del parametro

3.8.3.7. ApfDocument::getParam (\$ key)

Obtiene un parametro

Parámetros:

\$key Nombre del parametro

3.8.3.8. ApfDocument::getArgs (\$ override = "", \$ mask = "", \$ imask = "", \$ encode = True)

Obtener vector de argumentos del documento. (Para construir enlaces)

Parámetros:

override Marca valores a substituir

mask Indicar que parámetros seran incluidos, array vacio implica todos.

encode Si es verdadero, codificará & como &

3.8.3.9. ApfDocument::getArgsHidden (\$ override = "", \$ mask = "", \$ imask = "")

Obtener vector de argumentos del documento. (Para uso en campos ocultos de un formulario)

Parámetros:

override Marca valores a substituir

mask Indicar que parámetros seran incluidos, array vacio implica todos.

imask Mascar inversa

3.8.3.10. ApfDocument::query (\$ what)

Realiza una petición a la base de datos.

Parámetros:

what Petición SQL.

3.8.3.11. ApfDocument::redirect2page (\$ *page*, \$ *redirect* = False)

Genera una redirección.

Parámetros:

page Página de destino

redirect Pasar pagina actual como redirect

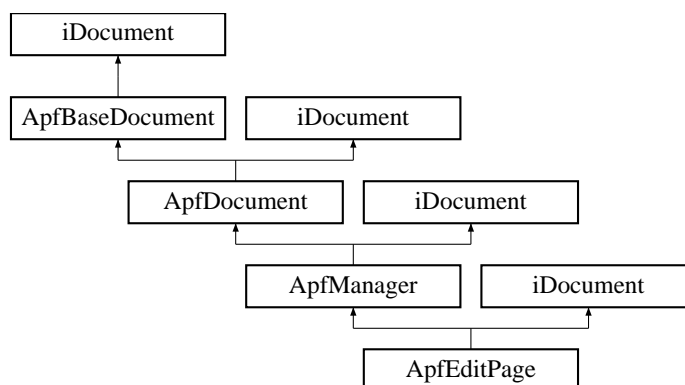
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/pages/base/Document.php

3.9. Referencia de la Clase ApfEditPage

Página de edición de medios/categorías.

Diagrama de herencias de ApfEditPage:



Métodos públicos

- **expandAction** (\$rv)
- **checkActionConsistency** (\$rv)
- **updateTitle** (\$action)
- **__construct** ()

Constructor.

- **preprocess** ()
- **process** ()
- **getVodServer** ()
- **body** ()

Método body.

3.9.1. Descripción detallada

Página de edición de medios/categorías.

3.9.2. Documentación de las funciones miembro

3.9.2.1. ApfEditPage::expandAction (\$ rv)

Expandir acción

Parámetros:

rv Array de props

Devuelve:

Array

3.9.2.2. ApfEditPage::checkActionConsistency (\$ rv)

Comprovar consistencia de la acción

Parámetros:

rv Array de props

Devuelve:

Array

3.9.2.3. ApfEditPage::updateTitle (\$ action)

Actualiza el titulo de la página en funcion de la acción

Parámetros:

action Acción

3.9.2.4. ApfEditPage::getVodServer ()

Obtener Manejador VoD

Devuelve:

El manejador VoD solicitado

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/pages/common/EditPage.php

3.10. Referencia de la Clase ApfFileSystem

Clase de gestion de ficheros.

Métodos públicos

- **__construct** (\$parent, \$name="", \$type='video')
Constructor.
- **get** ()

3.10.1. Descripción detallada

Clase de gestion de ficheros.

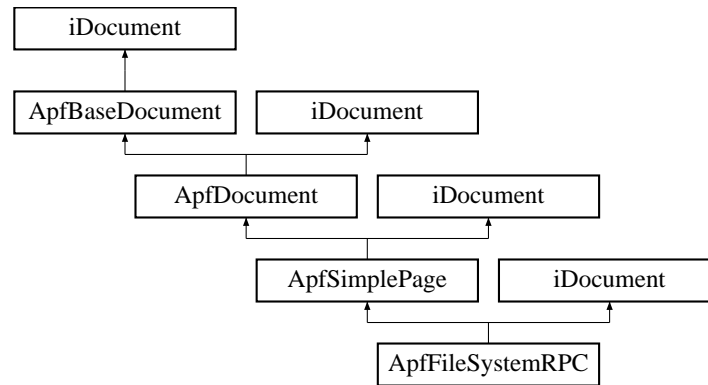
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/widgets/filesystem.php

3.11. Referencia de la Clase ApfFileSystemRPC

Clase página del gestor.

Diagrama de herencias de ApfFileSystemRPC:



Métodos públicos

- **show ()**
- **process ()**

3.11.1. Descripción detallada

Clase página del gestor.

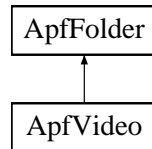
La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/pages/simple/FileSystemRPC.php`

3.12. Referencia de la Clase ApfFolder

Carpeta o directorio (Nodo) del sistema virtual de ficheros.

Diagrama de herencias de ApfFolder:



Métodos públicos

- **__construct** (&\$parent, \$vals)
- **getFinalUrl** ()
Obtener URL carpeta.
- **getShortName** (\$input, \$max)
- **show** ()
Mostrar la carpeta.
- **details** ()
Mostrar detalles.

Atributos protegidos

- **\$id**
- **\$name**
- **\$desc**
- **\$count**
- **\$parent**
- **\$image**

3.12.1. Descripción detallada

Carpeta o directorio (Nodo) del sistema virtual de ficheros.

3.12.2. Documentación del constructor y destructor

3.12.2.1. ApfFolder::__construct (&\$parent, \$vals)

Constructor

Parámetros:

parent Página padre

vals Array con las claves id, name, desc, count, etc...

Reimplementado en **ApfVideo** (p. 51).

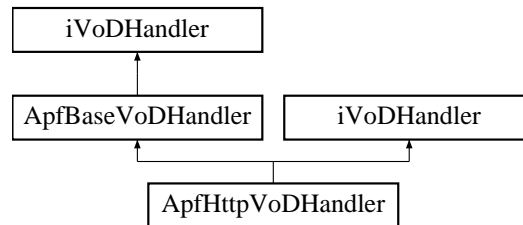
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/widgets/folder.php

3.13. Referencia de la Clase ApfHttpVoDHandler

Implementa los manejadores necesarios para un servidor VoD sobre HTTP.

Diagrama de herencias de ApfHttpVoDHandler:



3.13.1. Descripción detallada

Implementa los manejadores necesarios para un servidor VoD sobre HTTP.

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/vod/HttpVoDhandler.php

3.14. Referencia de la Clase ApfLocal

Métodos públicos

- `__construct ()`

Métodos públicos estáticos

- static `init ($lan="")`
- static `setLanguageVector ($lan)`
- static `getLanguageVector ()`
- static `getDefaultLanguage ()`
- static `get ($key)`

3.14.1. Descripción detallada

Gestor de idiomas, para cadenas localizadas

3.14.2. Documentación del constructor y destructor

3.14.2.1. `ApfLocal::__construct ()`

Constructor

Parámetros:

lan Vector de idiomas

3.14.3. Documentación de las funciones miembro

3.14.3.1. `static ApfLocal::init ($ lan = " ") [static]`

Inicialización del control de localizaciones

Parámetros:

lan Idioma forzado

3.14.3.2. `static ApfLocal::setLanguageVector ($ lan) [static]`

Fijar el vector

Parámetros:

lan Vector de idiomas

3.14.3.3. static ApfLocal::getLanguageVector () [static]

Devolver el vector

Devuelve:

Vector de idomas

3.14.3.4. static ApfLocal::getDefaultLanguage () [static]

Obtener el idioma por defecto

Devuelve:

Idioma por defecto

3.14.3.5. static ApfLocal::get (\$ key) [static]

Obtener una cadena traducida.

Parámetros:

key Clave.

Devuelve:

Cadena traducida.

Nota:

Si la cadena no existe, buscara una alternativa si esta disponible en el vector.

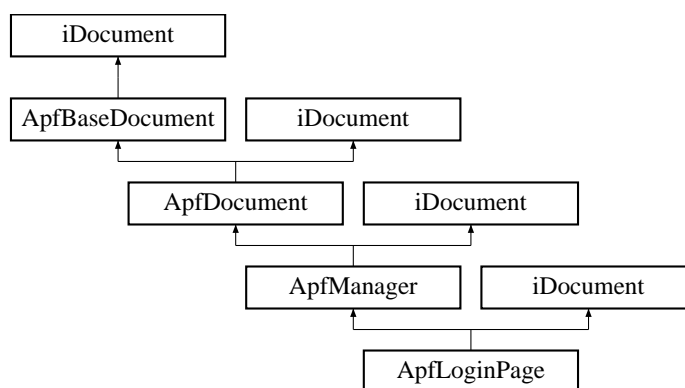
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/lan/strings.php

3.15. Referencia de la Clase ApfLoginPage

Página de autenticación.

Diagrama de herencias de ApfLoginPage:



Métodos públicos

- **__construct** (\$logout=False)
 - **body** ()
- Método cuerpo.*

3.15.1. Descripción detallada

Página de autenticación.

3.15.2. Documentación del constructor y destructor

3.15.2.1. ApfLoginPage::__construct (\$logout = False)

Constructor

Parámetros:

logout Si es verdadero cierra la sesión, en caso contrario muestra la página de entrada.

Reimplementado de **ApfBaseDocument** (p. 14).

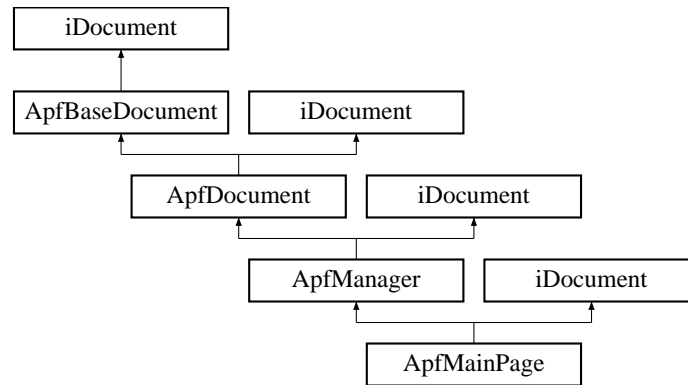
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/pages/common/LoginPage.php

3.16. Referencia de la Clase ApfMainPage

Página principal.

Diagrama de herencias de ApfMainPage:



Métodos públicos

- **__construct ()**
Constructor.
- **body ()**
Método cuerpo.

3.16.1. Descripción detallada

Página principal.

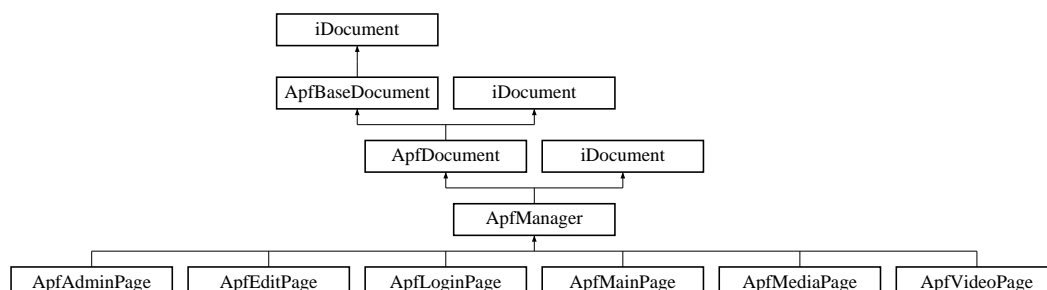
La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/pages/common/MainPage.php`

3.17. Referencia de la Clase ApfManager

Clase página del gestor.

Diagrama de herencias de ApfManager:



Métodos públicos

- **__construct** (\$title="", \$release_session=True)
Constructor.
- **setTitle** (\$title)
- **add2Menu** (\$title, \$page, \$link="", \$image="", \$hidden=False, \$redirect=False)
- **head** ()
Cabezera.
- **menu** ()
Crea el menú de navegación.
- **foot** ()
Pie de la página.
- **body** ()
Modo cuerpo, Genera el contenido de la página.
- **debug** ()
Muestra información de debug.
- **show** ()
Genera y envía la página al cliente.
- **getMediaMGR** ()
Obtener Gestor del catálogo.
- **getMediaTree** (\$force_update=False)
Genera el árbol de categorías.
- **writeCategoryListControl** (\$id=-1)
Escribe el árbol de categorías.

- `printFolders` (\$id)
- `printMedia` (\$id)
- `printNewMedia` (\$cut=4)
- `printTopMedia` (\$cut=4)

3.17.1. Descripción detallada

Clase página del gestor.

3.17.2. Documentación de las funciones miembro

3.17.2.1. `ApfManager::setTitle` (\$ *title*)

Fija el título del documento.

Parámetros:

title El título.

Reimplementado de `ApfBaseDocument` (p. 14).

3.17.2.2. `ApfManager::add2Menu` (\$ *title*, \$ *page*, \$ *link* = "", \$ *image* = "", \$ *hidden* = False, \$ *redirect* = False)

Añadir una nueva entrada al menú.

Parámetros:

title título de la página.

page dirección de la página.

link Valor diferente a "" indica que page es un documento externo

image Valor diferente a "" indica ruta a un icono

hidden Indica si la entrada estara oculta

redirect Indica si se debe redireccionar a la página anterior

3.17.2.3. `ApfManager::printFolders` (\$ *id*)

Escribe todas las carpetas

Parámetros:

id Identificador

3.17.2.4. `ApfManager::printMedia` (\$ *id*)

Escribe recursos multimedia

Parámetros:

id Identificador

3.17.2.5. ApfManager::printNewMedia (\$ cut = 4)

Escribe las ultimas novedades multimedia

Parámetros:

cut Limite

3.17.2.6. ApfManager::printTopMedia (\$ cut = 4)

Escribe los recursos más visitados

Parámetros:

cut Limite

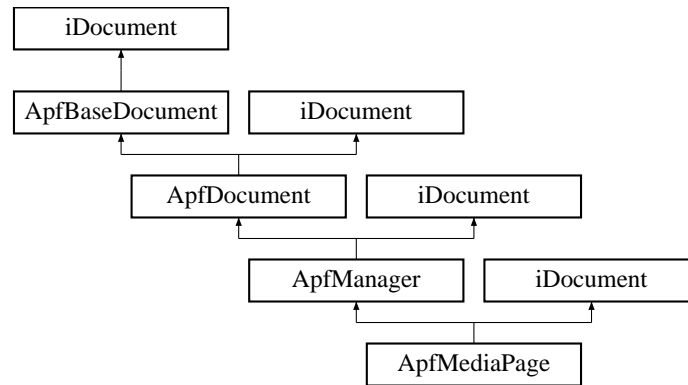
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/pages/base/manager.php

3.18. Referencia de la Clase ApfMediaPage

Navegador de la biblioteca multimedia. (Sistema virtual de ficheros).

Diagrama de herencias de ApfMediaPage:



Métodos públicos

- **__construct ()**
Constructor.
- **head ()**
Cabezera.
- **getFamily ()**
Imprime la familia.
- **printTopBar ()**
Imprime el area superior del navegador de categorias.
- **body ()**
Método cuerpo.
- **showAdminButtons ()**
Muestra los botones de administración.

3.18.1. Descripción detallada

Navegador de la biblioteca multimedia. (Sistema virtual de ficheros).

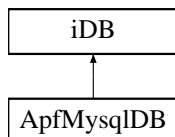
La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/pages/common/MediaPage.php`

3.19. Referencia de la Clase ApfMysqlDB

Clase base de datos (MySQL).

Diagrama de herencias de ApfMysqlDB:



Métodos públicos

- **__construct** (\$user="", \$password="", \$database="", \$host="")
Constructor.
- **connect** ()
- **disconnect** ()
Desconectar de la base de datos.
- **query** (\$query)
- **getError** ()
Obtener el último error producido en la última consulta.
- **fetchArray** ()
Obtener el vector de datos devueltos después de la última petición.
- **escape_string** (\$what)
Escapa la cadena.
- **insertId** ()
Devuelve el identificador de la última petición de inserción realizada a la base de datos.
- **getQueryCount** ()
Obtener número de peticiones.

3.19.1. Descripción detallada

Clase base de datos (MySQL).

3.19.2. Documentación de las funciones miembro

3.19.2.1. ApfMysqlDB::connect ()

Conectar a la base de datos

Implementa **iDB** (p. 59).

3.19.2.2. ApfMySQLDB::query (\$ *query*)

Enviar petición sql a la base de datos.

Parámetros:

query Petición

Implementa **iDB** (p. 59).

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/db/MySQL.php

3.20. Referencia de la Clase ApfNoteBook

Clase NoteBook (Un conjunto de páginas agrupadas).

Métodos públicos

- **__construct** (\$name="")
- **AddPage** (\$name, \$content)
- **Write** ()
Escribe el notebook con todos las páginas especificadas.

- **Get** ()
Obtiene el notebook.

3.20.1. Descripción detallada

Clase NoteBook (Un conjunto de páginas agrupadas).

3.20.2. Documentación del constructor y destructor

3.20.2.1. ApfNoteBook::__construct (\$ name = "")

Constructor

Parámetros:

name Nombre del NoteBook

3.20.3. Documentación de las funciones miembro

3.20.3.1. ApfNoteBook::AddPage (\$ name, \$ content)

Añade una nueva página

Parámetros:

name Nombre de la página

content Contenido de la página

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/widgets/notebook.php

3.21. Referencia de la Clase ApfNoteBookTab

Clase página de un libro.

Métodos públicos

- `__construct` (\$name, \$content)

Atributos públicos

- `$name`
- `$content`

3.21.1. Descripción detallada

Clase página de un libro.

3.21.2. Documentación del constructor y destructor

3.21.2.1. `ApfNoteBookTab::__construct` (\$ *name*, \$ *content*)

Constructor

Parámetros:

- name* Nombre de la página
- content* Contenido de la página

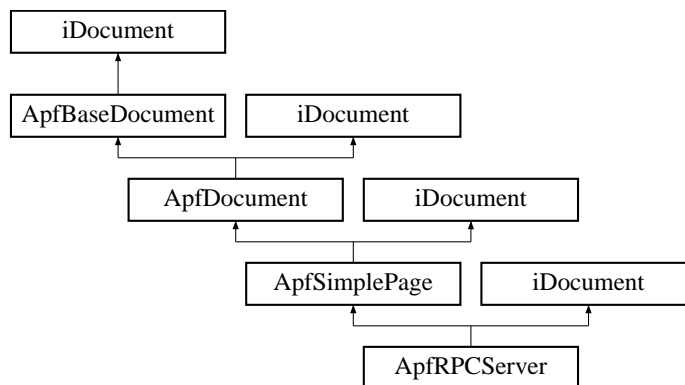
La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/widgets/notebook.php`

3.22. Referencia de la Clase ApfRPCServer

Clase página del gestor.

Diagrama de herencias de ApfRPCServer:



Métodos públicos

- **show ()**
Mostrar página.
- **AuthCheck (\$xsid_check=True, \$file_check=False)**
- **getVodServer ()**
- **copyImage (\$path, \$file)**
- **process ()**
Processar.

3.22.1. Descripción detallada

Clase página del gestor.

3.22.2. Documentación de las funciones miembro

3.22.2.1. ApfRPCServer::AuthCheck (\$xsid_check = True, \$file_check = False)

Verificar autenticación

Parámetros:

- xsid_check* VERificar xsid
- file_check* verificar nombre fichero

3.22.2.2. ApfRPCServer::getVodServer ()

Obtener Manejador VoD

Devuelve:

El manejador VoD solicitado

3.22.2.3. ApfRPCServer::copyImage (\$ path, \$ file)

Copiar una imagen

Parámetros:

path ruta del fichero temporal

file nombre definido por el usuario

Devuelve:

Nombre del fichero final

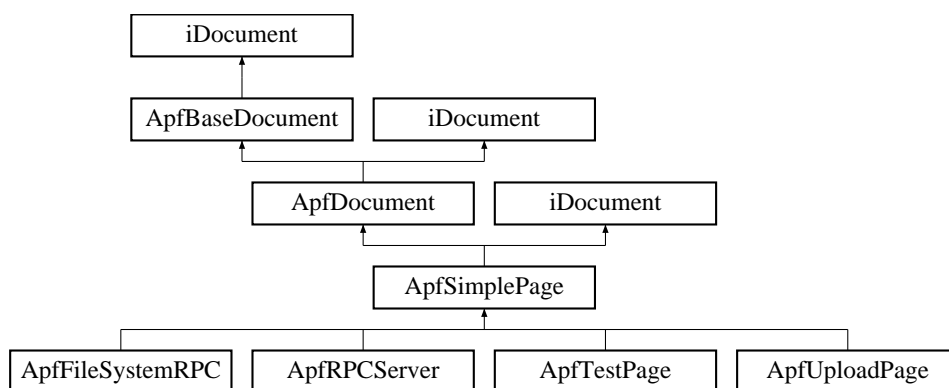
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/pages/simple/rpcserver.php

3.23. Referencia de la Clase ApfSimplePage

Clase base para páginas simples.

Diagrama de herencias de ApfSimplePage:



Métodos públicos

- **__construct** (\$title="", \$release_session=True)
- **body** ()
Método cuerpo.
- **show** ()
Muestra el documento.

3.23.1. Descripción detallada

Clase base para páginas simples.

3.23.2. Documentación del constructor y destructor

3.23.2.1. ApfSimplePage::__construct (\$ title = "", \$ release_session = True)

Constructor

Parámetros:

title Título de la página

release_session Indica si debemos liberar la sesión

Reimplementado de **ApfDocument** (p. 21).

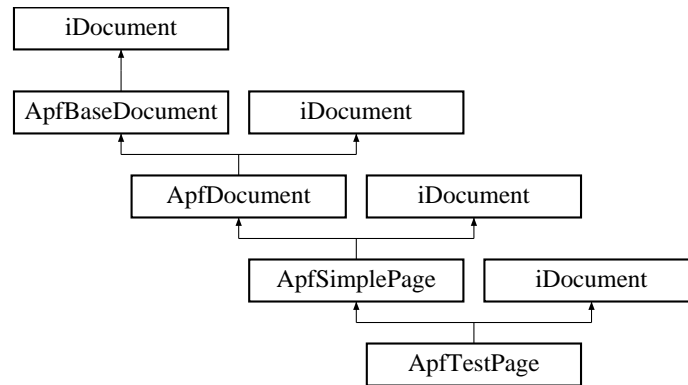
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/pages/base/simplepage.php

3.24. Referencia de la Clase ApfTestPage

Test class.

Diagrama de herencias de ApfTestPage:



Métodos públicos

- `body ()`

3.24.1. Descripción detallada

Test class.

La documentación para esta clase fué generada a partir del siguiente archivo:

- `test/test.php`

3.25. Referencia de la Clase ApfTree

Crea un arbol del sistema virtual de ficheros.

Métodos públicos

- `__construct` (\$vals)
- `writeOptions` (\$id)
- `getOptions` (\$id)
- `findNode` (\$id)
- `writeNode` (\$id, \$name, \$level, \$sel_id, \$close)
- `getNode` (\$id, \$name, \$level, \$sel_id, \$close)
- `isChildOf` (\$child, \$parent)

3.25.1. Descripción detallada

Crea un arbol del sistema virtual de ficheros.

3.25.2. Documentación del constructor y destructor

3.25.2.1. `ApfTree::__construct` (\$ vals)

Constructor

Parámetros:

vals Array de valores (id,parent,name)

3.25.3. Documentación de las funciones miembro

3.25.3.1. `ApfTree::writeOptions` (\$ id)

Escribe las opciones del nodo seleccionado.

Parámetros:

id El identificador del nodo.

3.25.3.2. `ApfTree::getOptions` (\$ id)

Obtiene las opciones del nodo seleccionado.

Parámetros:

id El identificador del nodo.

Devuelve:

Opciones

3.25.3.3. ApfTree::findNode (\$ id)

Busca un nodo

Parámetros:

id Identificador

Devuelve:

Devuelve una referencia al nodo (o null)

3.25.3.4. ApfTree::writeNode (\$ id, \$ name, \$ level, \$ sel_id, \$ close)

Escribe un Nodo.

Parámetros:

id Identificador

name Nombre del nodo

level Nivel dentro del arbol

sel_id Identificador del nodo actualmente seleccionado

close_table Tabla que identifica que niveles del arbol deben cerrarse

3.25.3.5. ApfTree::getNode (\$ id, \$ name, \$ level, \$ sel_id, \$ close)

Escribe un Nodo.

Parámetros:

id Identificador

name Nombre del nodo

level Nivel dentro del arbol

sel_id Identificador del nodo actualmente seleccionado

close_table Tabla que identifica que niveles del arbol deben cerrarse

Devuelve:

Contenido generado

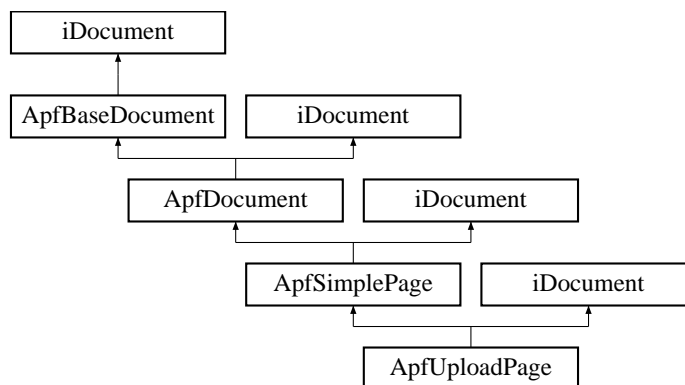
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/widgets/tree.php

3.26. Referencia de la Clase ApfUploadPage

Clase página del gestor.

Diagrama de herencias de ApfUploadPage:



Métodos públicos

- **ApfUploadPage** (\$title="", \$resource_type='video')
- **dump_js** ()
- **body** ()

3.26.1. Descripción detallada

Clase página del gestor.

3.26.2. Documentación de las funciones miembro

3.26.2.1. ApfUploadPage::ApfUploadPage (\$ title = "", \$ resource_type = ' video ')

Constructor

Parámetros:

title Título de la página

resource_type Tipo de recurso a subir (video,imagen,etc...)

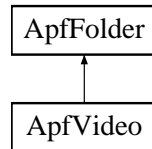
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/pages/simple/UploadPage.php

3.27. Referencia de la Clase ApfVideo

Clase representativa de un video.

Diagrama de herencias de ApfVideo:



Métodos públicos

- **__construct** (&\$parent, \$vals)
 - **getFinalUrl** ()
Obtener URL.
- **details** ()
Mostrar detalles.

Atributos protegidos

- **\$dur** = 0

3.27.1. Descripción detallada

Clase representativa de un video.

3.27.2. Documentación del constructor y destructor

3.27.2.1. ApfVideo::__construct (&\$parent, \$vals)

Constructor

Parámetros:

- parent* Página padre
- vals* Array con las claves id, name, desc, count, etc...

Reimplementado de **ApfFolder** (p. 29).

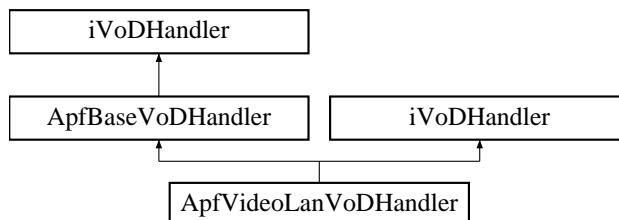
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/widgets/folder.php

3.28. Referencia de la Clase ApfVideoLanVoDHandler

Implementa los manejadores necesarios para un servidor VoD VideoLan.

Diagrama de herencias de ApfVideoLanVoDHandler:



Métodos públicos

- **notifyNewUploadedVideo** (\$name, \$path)
- **notifyVideoDeleted** (\$name, \$path)

3.28.1. Descripción detallada

Implementa los manejadores necesarios para un servidor VoD VideoLan.

3.28.2. Documentación de las funciones miembro

3.28.2.1. ApfVideoLanVoDHandler::notifyNewUploadedVideo (\$ name, \$ path)

Notifica al VoD que un fichero nuevo a sido subido

Parámetros:

name Nombre del fichero

path Ruta al fichero

Implementa **iVoDHandler** (p. 68).

3.28.2.2. ApfVideoLanVoDHandler::notifyVideoDeleted (\$ name, \$ path)

Notifica al vod el borrado de un fichero

Parámetros:

name Nombre

path Ruta absoluta

Implementa **iVoDHandler** (p. 68).

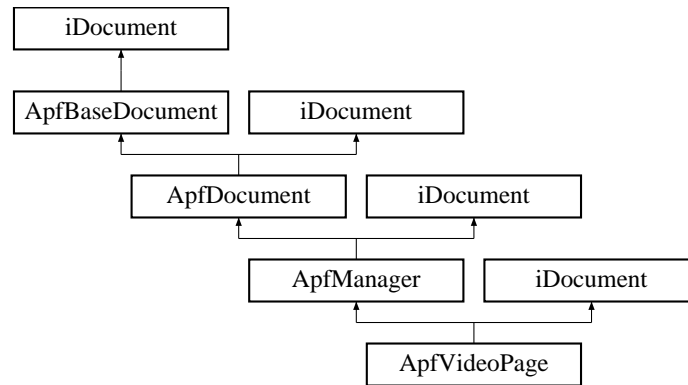
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/vod/VideoLanVoDhandler.php

3.29. Referencia de la Clase ApfVideoPage

Página del vídeo.

Diagrama de herencias de ApfVideoPage:



Métodos públicos

- **__construct ()**
Constructor.
- **body ()**
Método cuerpo.
- **showAdminButtons ()**
Muestra los botones de administración.

3.29.1. Descripción detallada

Página del vídeo.

La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/pages/common/VideoPage.php`

3.30. Referencia de la Clase ApfVoDFile

Clase que define un video, con sus respectiva propiedades.

3.30.1. Descripción detallada

Clase que define un video, con sus respectiva propiedades.

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/vod/baseVoDhandler.php

3.31. Referencia de la Clase Builder

Los constructores de cosas.

Métodos públicos estáticos

- static **build** (\$what)
Intenta construir el objeto solicitado.

3.31.1. Descripción detallada

Los constructores de cosas.

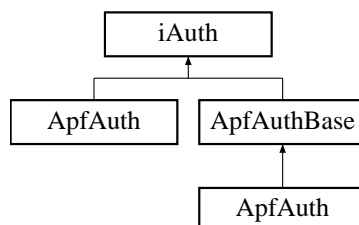
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/common/gatebuilder.php

3.32. Referencia de la Interfaz iAuth

Interfaz de autenticación.

Diagrama de herencias de iAuth:



Métodos públicos

- **authenticate** (\$login, \$pass, \$enc="plain", \$sid=0)
- **verify** (\$uid, \$hash)
- **getHash** (\$what="")
- **getPassword** (\$login, \$enc="md5")
- **challenge** ()

Obtiene un desafío (CHAP).

- **validate** (\$login, \$challenge, \$client_hash, \$alg="md5")
- **hashme** (\$login, \$pass, \$challenge, \$alg="md5")
- **authSucesfull** ()

Hook llamado solo cuando la autenticación se considera exitosa.

- **getLogin** ()
- **getLevel** ()
- **getUID** ()
- **getAuthHash** ()

3.32.1. Descripción detallada

Interfaz de autenticación.

3.32.2. Documentación de las funciones miembro

3.32.2.1. iAuth::authenticate (\$ login, \$ pass, \$ enc = "plain", \$ sid = 0)

Autentica el usuario

Parámetros:

login Login del usuario

pass Password o hash del password del usuario a comprobar

enc Codificación del password (plain: Password en texto plano, md5 hash MD5 del password)

sid Session del usuario

Devuelve:

Verdadero si los datos fueron correctos, falso en caso de fallar

Implementado en **ApfAuthBase** (p. 11).

3.32.2.2. iAuth::verify (\$ uid, \$ hash)

Verifica que el usuario esta debidamente autenticado.

Parámetros:

uid Identificador del usuario.

hash Hash de comprobación.

Devuelve:

Verdadero si los datos fueron correctos, falso en caso de fallar

Implementado en **ApfAuth** (p. 8).

3.32.2.3. iAuth::getHash (\$ what = "")

Genera un hash de una información aleatoria

Parámetros:

what Entrada de datos diversos (nombre, ip, etc..)

Devuelve:

Devuelve un identificador único Es posible cambiar la función, lo importante es que el valor devuelto no pueda ser predecido por un posible atacante. Esto se utiliza principalmente para recuperar una sesión antigua, ya que las sesiones PHP no son persistentes, y expiran al cabo de un tiempo o al cerrar el navegador.

Nota:

Es necesario reflejar los cambios realizados en esta función en el las funciones de validación del script upload.py, en caso contrario el script devolverá error de validación del hash de autenticación.

Implementado en **ApfAuthBase** (p. 11).

3.32.2.4. iAuth::getPassword (\$ login, \$ enc = "md5")

Obtener el password asociado a ese Login

Parámetros:

enc Codificación (plain: plana, md5 hash md5 del password, etc...)

Implementado en **ApfAuth** (p. 9).

3.32.2.5. iAuth::validate (\$ login, \$ challenge, \$ client_hash, \$ alg = "md5")

Valida el usuario con el desafío fijado

Parámetros:

\$login Login del usuario

\$challenge Desafío que hemos enviado previamente al cliente

\$client_hash Respuesta al desafío por el cliente

\$alg Algoritmo de Hash utilizado

Implementado en **ApfAuthBase** (p. 12).

3.32.2.6. iAuth::hashme (\$ login, \$ pass, \$ challenge, \$ alg = "md5")

Realizar el MD5(username + hashed_pass + challenge)

Parámetros:

login Login del usuario

pass Password o hash del password del usuario a comprobar

challenge Desafío

alg Algoritmo de Hash

Implementado en **ApfAuthBase** (p. 11).

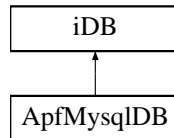
La documentación para esta interfaz fué generada a partir del siguiente archivo:

- lib/auth/AuthBase.php

3.33. Referencia de la Interfaz iDB

Interfaz de Base de Datos.

Diagrama de herencias de iDB:



Métodos públicos

- **connect ()**
Conectar a la base de datos.
- **disconnect ()**
Desconectar de la base de datos.
- **query (\$query)**
Realizar una petición a la base de datos.
- **getError ()**
Obtener último error producido por la última petición.
- **fetchArray ()**
Obtener array de resultados.
- **insertId ()**
Obtener id del último registro insertado.
- **escape_string (\$data)**
Escapar los datos.
- **getQueryCount ()**
Obtener número de peticiones.

3.33.1. Descripción detallada

Interfaz de Base de Datos.

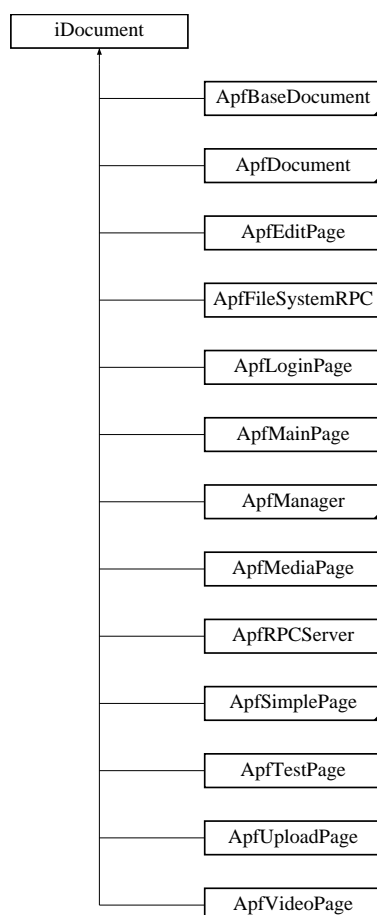
La documentación para esta interfaz fué generada a partir del siguiente archivo:

- lib/db/dbBase.php

3.34. Referencia de la Interfaz iDocument

Interfaz Documento.

Diagrama de herencias de iDocument:



Métodos públicos

- `show ()`
- `head ()`
- `body ()`
- `foot ()`

3.34.1. Descripción detallada

Interfaz Documento.

La documentación para esta interfaz fué generada a partir del siguiente archivo:

- `lib/pages/base/BaseDocument.php`

3.35. Referencia de la Interfaz iFileController

Interfaz para trabajar sobre sistemas de ficheros, reales, virtuales, etc...

3.35.1. Descripción detallada

Interfaz para trabajar sobre sistemas de ficheros, reales, virtuales, etc...

La documentación para esta interfaz fué generada a partir del siguiente archivo:

- lib/widgets/filesystem.php

3.36. Referencia de la Interfaz iLogger

Interface que define un fichero Log.

Métodos públicos

- **write** (\$msg)
- **close** ()

Cierra el fichero log.

3.36.1. Descripción detallada

Interface que define un fichero Log.

3.36.2. Documentación de las funciones miembro

3.36.2.1. iLogger::write (\$msg)

Escribe al fichero log

Parámetros:

msg Mensaje a escribir

La documentación para esta interfaz fué generada a partir del siguiente archivo:

- lib/log/logger.php

3.37. Referencia de la Clase IncorrectPasswordException

Excepción de contraseña incorrecta.

3.37.1. Descripción detallada

Excepción de contraseña incorrecta.

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/vod/VideoLanTelnetClient.php

3.38. Referencia de la Clase InvalidFileException

Fichero no válido.

3.38.1. Descripción detallada

Fichero no válido.

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/common/exceptions.php

3.39. Referencia de la Interfaz iPlayer

Interfaz reproductor.

Métodos públicos

- `write ()`

3.39.1. Descripción detallada

Interfaz reproductor.

La documentación para esta interfaz fué generada a partir del siguiente archivo:

- `lib/players/basePlayer.php`

3.40. Referencia de la Interfaz iVodFile

define un video, con sus respectiva propiedades

3.40.1. Descripción detallada

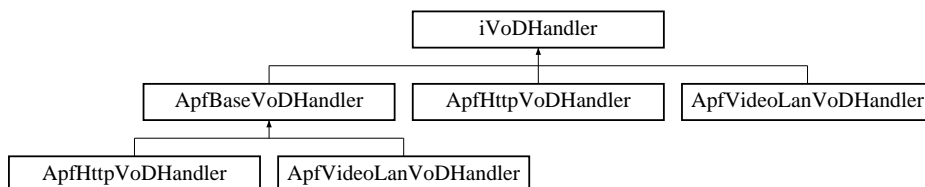
define un video, con sus respectiva propiedades

La documentación para esta interfaz fué generada a partir del siguiente archivo:

- lib/vod/baseVoDhandler.php

3.41. Referencia de la Interfaz iVoDHandler

Diagrama de herencias de iVoDHandler:



Métodos públicos

- **UploadVideoFile** (\$path, \$filename)
- **CheckVideoFileBeforeUpload** (\$path)
- **notifyNewUploadedVideo** (\$name, \$path)
- **DeleteVideoFile** (\$name)
- **notifyVideoDeleted** (\$name, \$path)
- **GeneratePreview** (\$path, \$name)

3.41.1. Descripción detallada

interfaz base para cada tipo de servidor VoD contiene los hooks que relacionan el servidor VoD con la aplicación

3.41.2. Documentación de las funciones miembro

3.41.2.1. iVoDHandler::UploadVideoFile (\$ path, \$ filename)

Indica al VoD, que un nuevo fichero de video a sido subido al servidor

Parámetros:

path Ruta al fichero en el sistema local

filename Nombre del fichero especificado por el usuario

Devuelve:

Devuelve verdadero en caso de éxito

Implementado en **ApfBaseVoDHandler** (p. 17).

3.41.2.2. iVoDHandler::CheckVideoFileBeforeUpload (\$ path)

Pasa un nombre de fichero al VoD para validar antes de ser subido

Parámetros:

path Nombre fichero

Devuelve:

Devuelve verdadero si el fichero es aceptable

Implementado en **ApfBaseVoDHandler** (p. 18).

3.41.2.3. iVoDHandler::notifyNewUploadedVideo (\$ name, \$ path)

Notifica al VoD que un fichero nuevo a sido subido

Parámetros:

name Nombre del fichero

path Ruta al fichero

Implementado en **ApfVideoLanVoDHandler** (p. 52).

3.41.2.4. iVoDHandler::DeleteVideoFile (\$ name)

Borra el vídeo

Parámetros:

name video a borrar

Implementado en **ApfBaseVoDHandler** (p. 18).

3.41.2.5. iVoDHandler::notifyVideoDeleted (\$ name, \$ path)

Notifica al vod el borrado de un fichero

Parámetros:

name Nombre

path Ruta absoluta

Implementado en **ApfVideoLanVoDHandler** (p. 52).

3.41.2.6. iVoDHandler::GeneratePreview (\$ path, \$ name)

Se peticiona la generación de una previsualización

Parámetros:

path Ruta al fichero en el sistema local

name Nombre del fichero especificado por el usuario

Devuelve:

Devuelve la ruta a la prev

Implementado en **ApfBaseVoDHandler** (p. 18).

La documentación para esta interfaz fué generada a partir del siguiente archivo:

- lib/vod/baseVoDhandler.php

3.42. Referencia de la Clase MediaMGR

Clase controladora de los recursos guardados en la biblioteca.

Métodos públicos

- `__construct ($DB)`
- `query ($q)`
- `fetchArray ()`
- `insertId ()`
- `increaseCategoryCounter ($id)`
- `decreaseCategoryCounter ($id)`
- `findCategory ($id)`
- `saveCategory ($rv)`
- `deleteCategory ($id)`
- `createCategory ($rv)`
- `getMediaTree ($force_update=False)`
- `getFolders ($id)`
- `getMedia ($id)`
- `getNewMedia ($cut)`
- `getTopMedia ($cut)`
- `getVideo ($id)`
- `increaseVideoHitCount ($id)`
- `saveVideo ($rv)`
- `deleteVideo ($id)`
- `createVideo ($rv)`
- `filterVars ($rv)`

3.42.1. Descripción detallada

Clase controladora de los recursos guardados en la biblioteca.

3.42.2. Documentación del constructor y destructor

3.42.2.1. MediaMGR::__construct (\$ DB)

Constructor

Parámetros:

DB Base de datos a usar

3.42.3. Documentación de las funciones miembro

3.42.3.1. MediaMGR::increaseCategoryCounter (\$ id)

Incrementa el número de hits de un vídeo

Parámetros:

id Identificador

3.42.3.2. MediaMGR::decreaseCategoryCounter (\$ *id*)

Incrementa el número de hits de un vídeo

Parámetros:

id Identificador

3.42.3.3. MediaMGR::findCategory (\$ *id*)

Buscar datos de una categoria en especial

Parámetros:

id Identificador

Devuelve:

Un array con los datos pedidos

3.42.3.4. MediaMGR::saveCategory (\$ *rv*)

Salvar categoria

Parámetros:

rv Array de valores

3.42.3.5. MediaMGR::deleteCategory (\$ *id*)

Borrar categoria

Parámetros:

id ID

TODO - borrar nodos huérfanos

3.42.3.6. MediaMGR::createCategory (\$ *rv*)

Crear una nueva categoria

Parámetros:

rv Array datos

Devuelve:

id de la ctg

3.42.3.7. MediaMGR::getMediaTree (\$force_update = False)

Genera el árbol de categorías

Devuelve:

El arbol

3.42.3.8. MediaMGR::getFolders (\$ id)

Obtener todas las carpetas

Parámetros:

id Identificador

3.42.3.9. MediaMGR::getMedia (\$ id)

Obtener todas las vídeos

Parámetros:

id Identificador

3.42.3.10. MediaMGR::getNewMedia (\$ cut)

Obtener vídeos nuevos

Parámetros:

cut Corte overflow

Devuelve:

array

3.42.3.11. MediaMGR::getTopMedia (\$ cut)

Obtener vídeos más vistos

Parámetros:

cut Corte overflow

Devuelve:

array

3.42.3.12. MediaMGR::getVideo (\$ id)

Devuelve un array con datos del vídeo solicitado

Parámetros:

id Identificador

Devuelve:

array

3.42.3.13. MediaMGR::increaseVideoHitCount (\$ id)

Incrementa el número de hits de un vídeo

Parámetros:

id Identificador

3.42.3.14. MediaMGR::saveVideo (\$ rv)

Salva un video en la DB

Parámetros:

rv array

3.42.3.15. MediaMGR::deleteVideo (\$ id)

Borra un vídeo

Parámetros:

id Identificador

3.42.3.16. MediaMGR::createVideo (\$ rv)

Crea un vídeo

Parámetros:

rv Array

Devuelve:

id

3.42.3.17. MediaMGR::filterVars (\$ rv)

Filtrar variables

Parámetros:

rv Array de entrada

Devuelve:

Array filtrado y limpio

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/mediamgr/MediaMGR.php

3.43. Referencia de la Clase `ModuleNotFoundException`

Excepción de modulo no encontrado.

Métodos públicos

- `__construct` (\$name="None")

3.43.1. Descripción detallada

Excepción de modulo no encontrado.

3.43.2. Documentación del constructor y destructor

3.43.2.1. `ModuleNotFoundException::__construct` (\$ *name* = "None")

constructor

Parámetros:

name Nombre

La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/common/appcore.php`

3.44. Referencia de la Clase NotImplementedException

Excepción de funcionalidad no implementada.

3.44.1. Descripción detallada

Excepción de funcionalidad no implementada.

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/common/exceptions.php

3.45. Referencia de la Clase PHPError

Lanzar errores como si fueran excepciones.

Métodos públicos estáticos

- static `errorHandlerCallback` (\$code, \$string, \$file, \$line, \$context)

3.45.1. Descripción detallada

Lanzar errores como si fueran excepciones.

La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/common/exceptions.php`

3.46. Referencia de la Clase Request

Petición.

Métodos públicos estáticos

- static **getRemoteAddress** (\$show="")
- static **getProtocol** ()
Obtiene el protocolo.
- static **getPort** ()
Obtiene el puerto.
- static **buildRootURI** (\$path="")

3.46.1. Descripción detallada

Petición.

3.46.2. Documentación de las funciones miembro

3.46.2.1. static Request::getRemoteAddress (\$ how = "") [static]

Muestra la ip del cliente

Parámetros:

how short=solo ip, rshort=ip + x_forward, sino mostrará información completa ip+x_forward+client_
ip+via

3.46.2.2. static Request::buildRootURI (\$ path = "") [static]

Construye una ruta relativa a la raiz

Parámetros:

path Ruta

Devuelve:

Ruta completa

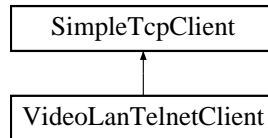
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/common/util.php

3.47. Referencia de la Clase SimpleTcpClient

Cliente Tcp sencillo.

Diagrama de herencias de SimpleTcpClient:



Métodos públicos

- `__construct` (\$destination, \$port)
- `expect` (\$what, \$throw=True)
- `write` (\$msg)
- `read` (\$size)

3.47.1. Descripción detallada

Cliente Tcp sencillo.

3.47.2. Documentación del constructor y destructor

3.47.2.1. SimpleTcpClient::__construct (\$ destination, \$ port)

Constructor

Parámetros:

destination Dirección de máquina

port Puerto

La documentación para esta clase fué generada a partir del siguiente archivo:

- `lib/net/simpletcpclient.php`

3.48. Referencia de la Clase UnknownResourceException

Excepción de recurso desconocido.

Métodos públicos

- `__construct` (\$name="None")

3.48.1. Descripción detallada

Excepción de recurso desconocido.

3.48.2. Documentación del constructor y destructor

3.48.2.1. UnknownResourceException::__construct (\$ name = "None")

constructor

Parámetros:

name Nombre

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/common/appcore.php

3.49. Referencia de la Clase UploadCtrl

Clase control de subidas.

Métodos públicos

- **__construct** (\$parent, \$resource_type='video', \$callback='')
- **write** ()
Escribe el control.
- **get** ()

3.49.1. Descripción detallada

Clase control de subidas.

3.49.2. Documentación del constructor y destructor

3.49.2.1. UploadCtrl::__construct (\$parent, \$resource_type = 'video', \$callback = "")

Constructor

Parámetros:

parent Página padre

3.49.3. Documentación de las funciones miembro

3.49.3.1. UploadCtrl::get ()

Obtiene el control

Devuelve:

Control generado

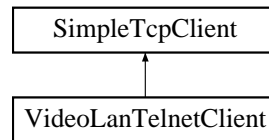
La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/widgets/upload.php

3.50. Referencia de la Clase VideoLanTelnetClient

Cliente VideoLAN.

Diagrama de herencias de VideoLanTelnetClient:



Métodos públicos

- **__construct** (\$addr, \$port, \$passwd)
- **addVodResource** (\$name, \$path)
- **removeVodResource** (\$name)
- **__destruct** ()

Destructor.

3.50.1. Descripción detallada

Cliente VideoLAN.

3.50.2. Documentación del constructor y destructor

3.50.2.1. VideoLanTelnetClient::__construct (\$ addr, \$ port, \$ passwd)

Constructor

Parámetros:

addr Dirección

port Puerto

passwd Contraseña

3.50.3. Documentación de las funciones miembro

3.50.3.1. VideoLanTelnetClient::addVodResource (\$ name, \$ path)

Añadir recurso

Parámetros:

name Nombre

path Ruta

3.50.3.2. VideoLanTelnetClient::removeVodResource (\$ *name*)

Eliminar recurso

Parámetros:

name Nombre

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/vod/VideoLanTelnetClient.php

3.51. Referencia de la Clase VoDFactory

Fabrica de manejadores para diferentes servidores VoD.

Métodos públicos estáticos

- static `getVoDHandler` (\$type)

Atributos públicos estáticos

- static `$handlers` = array()

3.51.1. Descripción detallada

Fabrica de manejadores para diferentes servidores VoD.

3.51.2. Documentación de las funciones miembro

3.51.2.1. static `VoDFactory::getVoDHandler` (\$ type) [static]

Obtiene una instancia del manejador VoD pedido

Parámetros:

type Tipo de manejador VoD

Devuelve:

Clase manejadora del servidor VoD solicitado

La documentación para esta clase fué generada a partir del siguiente archivo:

- lib/vod/VoDFactory.php