

phpPoA2



Contents

AutoPoA.php	0
definitions.php	0
Define AS_T	0
Define AUTHN_FAILED	0
Define AUTHN_SUCCESS	0
Define AUTHZ_FAILED	0
Define AUTHZ_SUCCESS	0
Define CONFIG_ERR	0
Define GPOA_T	0
Define INVITE_ERR	0
Define NOAUTH_ERR	0
Define SYSTEM_ERR	0
Define USER_ERR	0
AuthenticationEngine.php	0
crypt.php	0
Function decrypt_3DES	0
Function decrypt_AES	0
Function encrypt_3DES	0
Function encrypt_AES	0
Function openssl_decrypt	0
Function openssl_encrypt	0
Define PADDINGSIZE	0
Function papi_openssl_decrypt	0
Function papi_openssl_encrypt	0
GenericConfigurator.php	0
GenericEngine.php	0
Hook.php	0
LitePoA.php	0
PoA.php	0
PoAConfigurator.php	0
PoAEventHandler.php	0
PoAException.php	0
PoALog.php	0
PoAUtils.php	0
papi1.5-slo-extra.php	0
papi1.5-slo.php	0
authz_convert.php	0
Define ROOT	0
invite_convert.php	0
Class AuthenticationEngine	0
Var \$engine_type	0
Method authenticate	0
Method getAttribute	0
Method getAttributes	0

Method isAuthenticated	0
Method logout	0
Class AuthorizationConfigurator	0
Method getDefaultBehaviour	0
Class AuthorizationEngine	0
Var \$engine_type	0
Method authorize	0
Method getAuthorizedList	0
Method isAuthenticated	0
Method revoke	0
Class AutoPoA	0
Method authenticate	0
Method isAuthenticated	0
Class GenericConfigurator	0
Var \$cfg	0
Var \$mandatory_options	0
Constructor construct	0
Method configure	0
Method validate	0
Class GenericEngine	0
Var \$cfg	0
Var \$engine_type	0
Var \$handler	0
Var \$hooks	0
Var \$valid_hooks	0
Constructor construct	0
Method addHook	0
Method clean	0
Method configure	0
Method registerHandler	0
Method removeHook	0
Method runHooks	0
Method setHandler	0
Class Hook	0
Var \$ callable_name	0
Var \$ hook	0
Var \$ name	0
Constructor construct	0
Method getName	0
Method run	0
Class LitePoA	0
Var \$handler	0
Method checkAccess	0
Method setIDBuilder	0
Class PoA	0
Var \$attributes	0
Var \$authn_engine	0
Var \$authz_engines	0
Var \$autoload	0

Var \$cfg	0
Var \$db_manager	0
Var \$handler	0
Var \$local_site	0
Var \$log	0
Constructor construct	0
Method addHook	0
Method authenticate	0
Method authorize	0
Method clean	0
Method getAttribute	0
Method getAttributes	0
Method getAuthorizationEngines	0
Method getAuthorizationLevels	0
Method isAuthenticated	0
Method isAuthorized	0
Method logout	0
Method removeHook	0
Method revoke	0
Class PoAConfigurator	0
Var \$mandatory_options	0
Constructor construct	0
Method getAuthnEngine	0
Method getAuthnEngineConfFile	0
Method getAuthzEngineConfFile	0
Method getAuthzEngines	0
Method getAuthzLevels	0
Method getInviteErrorURL	0
Method getLanguage	0
Method getLogFile	0
Method getLogLevel	0
Method getNoAuthErrorURL	0
Method getSystemErrorURL	0
Method isDebug	0
Class PoAEventHandler	0
Constructor construct	0
Method abort	0
Method autoloadHandler	0
Method errorHandler	0
Method exceptionHandler	0
Method setDebug	0
Method setLogger	0
Class PoAException	0
Constructor construct	0
Method toString	0
Class PoALog	0
Constructor construct	0
Method getLogLevel	0
Method write	0

Class PoAUtils	0
Method lang_code	0
Method msg	0
AttributeFilterAuthzEngine.php	0
Define ATTRIBUTE_BEFORE_FILTERS	0
AttributeFilterConfigurator.php	0
papi1.5-attr-authz.php	0
Function attrHook	0
Class AttributeFilterAuthzEngine	0
Var \$valid_hooks	0
Method authorize	0
Method getAuthorizedList	0
Method isAuthorized	0
Method revoke	0
Class AttributeFilterConfigurator	0
Var \$mandatory_options	0
Method getAllowedAttributes	0
Method getDeniedAttributes	0
DBADB.php	0
GenericDB.php	0
GenericMySQLDB.php	0
SessionDB.php	0
Class DBADB	0
Var \$mandatory_options	0
Method check	0
Method close	0
Method configure	0
Method delete	0
Method fetch	0
Method fetch_all	0
Method open	0
Method replace	0
Class GenericDB	0
Var \$cfg	0
Var \$db	0
Var \$error	0
Var \$mandatory_options	0
Constructor construct	0
Method check	0
Method close	0
Method configure	0
Method delete	0
Method fetch	0
Method fetch_all	0
Method open	0
Method replace	0
Class GenericMySQLDB	0
Var \$create_sql	0
Var \$delete_sql	0

Var \$insert_sql	0
Var \$mandatory_options	0
Var \$prefix	0
Var \$search_all_sql	0
Var \$search_sql	0
Var \$update_sql	0
Method check	0
Method close	0
Method configure	0
Method delete	0
Method fetch	0
Method fetch_all	0
Method getError	0
Method open	0
Method replace	0
Class SessionDB	0
Var \$session_name	0
Method check	0
Method close	0
Method delete	0
Method fetch	0
Method fetch_all	0
Method open	0
Method replace	0
messages-en_EN.php	0
Global Variable \$messages	0
messages-es_ES.php	0
AuthorizedDB.php	0
AuthorizedDBADB.php	0
InviteAuthzEngine.php	0
Define AUTHORIZED_DBA	0
Define AUTHORIZED_MYSQL	0
Define AUTHORIZED_SESSION	0
Define INVITES_BEFORE_AUTHORIZATION	0
Define INVITES_DBA	0
Define INVITES_MYSQL	0
Define INVITES_SESSION	0
InviteConfigurator.php	0
InviteDB.php	0
InviteDBADB.php	0
papi1.5-invite-authz.php	0
Class AuthorizedDB	0
Method replace_authorization	0
Class AuthorizedDBADB	0
Var \$authorized	0
Var \$mandatory_options	0
Method fetch	0
Method fetch_all	0
Method open	0

Method replace_authorization	0
Class InviteAuthzEngine	0
Var \$authz_db	0
Var \$invites_db	0
Var \$valid_hooks	0
Method authorize	0
Method configure	0
Method emailVerify	0
Method finish	0
Method getAuthorizedList	0
Method getPendingInvites	0
Method invite	0
Method isAuthorized	0
Method removeInvite	0
Method revoke	0
Class InviteConfigurator	0
Var \$mandatory_options	0
Method doEmailVerify	0
Method getAdminEmail	0
Method getAuthorizedDBFile	0
Method getAuthorizedDBHost	0
Method getAuthorizedDBName	0
Method getAuthorizedDBPassword	0
Method getAuthorizedDBType	0
Method getAuthorizedDBUser	0
Method getEmailVerifyAlgRegEx	0
Method getEmailVerifyAttribute	0
Method getEmailVerifyRegEx	0
Method getInvitesDBFile	0
Method getInvitesDBHost	0
Method getInvitesDBName	0
Method getInvitesDBPassword	0
Method getInvitesDBType	0
Method getInvitesDBUser	0
Method getInviteSubject	0
Method getInviteText	0
Method getInviteURL	0
Method getUniqueAttributes	0
Class InviteDB	0
Method replace_invite	0
Class InviteDBADB	0
Var \$invite	0
Var \$mandatory_options	0
Method fetch	0
Method fetch_all	0
Method open	0
Method replace_invite	0
PAPIAuthnEngine.php	0
Define ATTR_SEPARATOR	0

Define NAMEVALUE_SEPARATOR	0
Define NS_PAPI_ATTRIBUTES	0
Define NS_PAPI_PROTOCOL	0
Define PAPI_ATTRIBUTE_PARSER	0
Define PAPI_CLEAN_URL	0
Define PAPI_DBA	0
Define PAPI_MYSQL	0
Define PAPI_REDIRECT_URL_FINISH	0
Define PAPI_RESTORE_ORIGINAL_REQUEST	0
Define PAPI_SESSION	0
Define PHP_VERSION_REQUIRED	0
Define PROTO_ATTR_AS_ID	0
Define PROTO_ATTR_EXPIRE_TIME	0
Define PROTO_ATTR_KEY	0
Define PROTO_ATTR_PREFIX	0
Define REQUEST_LIFETIME	0
Define VALUE_SEPARATOR	0
PAPIConfigurator.php	0
PAPIDB.php	0
PAPIDBADB.php	0
PAPIMySQLDB.php	0
Define PAPI_DB_TABLE	0
PAPISessionDB.php	0
papi1.5-authn-attr.php	0
papi1.5-customattr.php	0
Function handler	0
papi1.5-papiopoa-hook.php	0
Function hook	0
Class AttributeHandler	0
Method handle	0
Class Example	0
Method hook	0
Class PAPIAuthnEngine	0
Var \$assertion	0
Var \$as_id	0
Var \$attributes	0
Var \$cfg	0
Var \$clean_url	0
Var \$cookie_name	0
Var \$db	0
Var \$enforcing	0
Var \$expiration_time	0
Var \$global_expire_time	0
Var \$id	0
Var \$key	0
Var \$lkey	0
Var \$opoa	0
Var \$pkey	0
Var \$skip_redirection	0

Var \$status	0
Var \$valid_hooks	0
Method attributeQuery	0
Method authenticate	0
Method configure	0
Method deleteCookie	0
Method deleteRequest	0
Method getAttribute	0
Method getAttributes	0
Method getNewCookie	0
Method getRedirectLocation	0
Method getSingleLogoutLocation	0
Method getSingleLogoutResponseLocation	0
Method isAuthenticated	0
Method isSafe	0
Method loadRequest	0
Method logout	0
Method redirect	0
Method saveRequest	0
Method testCookie	0
Method testResponse	0
Class PAPIConfigurator	0
Var \$mandatory_options	0
Method getCookieDomain	0
Method getCookieTimeout	0
Method getDBFile	0
Method getDBHost	0
Method getDBName	0
Method getDBPassword	0
Method getDBPrefix	0
Method getDBType	0
Method getDBUser	0
Method getFriendlyName	0
Method getHomeLocatorID	0
Method getID	0
Method getLKey	0
Method getLocation	0
Method getLogoutURL	0
Method getPubKeyFile	0
Method getRedirectType	0
Method getRedirectURL	0
Method getRequestLifetime	0
Class PAPIDB	0
Method purge	0
Method replaceContents	0
Class PAPIDBADB	0
Var \$context	0
Method fetch	0
Method purge	0

Method replaceContents	0
Class PAPIMySQLDB	0
Var \$context	0
Var \$prefix	0
Constructor construct	0
Method fetch	0
Method replaceContents	0
Class PAPISessionDB	0
Var \$context	0
Method fetch	0
Method replaceContents	0
QueryFilterAuthzEngine.php	0
Define QUERY_BEFORE_FILTERS	0
QueryFilterConfigurator.php	0
papi1.5-query-authz.php	0
Function queryHook	0
Class QueryFilterAuthzEngine	0
Var \$valid_hooks	0
Method authorize	0
Method getAuthorizedList	0
Method isAuthorized	0
Method revoke	0
Class QueryFilterConfigurator	0
Var \$mandatory_options	0
Method getAllowedPatterns	0
Method getDeniedPatterns	0
SimpleSAMLphpAuthnEngine.php	0
Define NS_SAML2_PROTOCOL	0
Class SimpleSAMLphpAuthnEngine	0
Var \$attributes	0
Var \$simplesaml	0
Var \$status	0
Method authenticate	0
Method configure	0
Method getAttribute	0
Method getAttributes	0
Method isAuthenticated	0
Method logout	0
SourceIPAddrAuthzEngine.php	0
Define SOURCEADDR_BEFORE_FILTERS	0
SourceIPAddrConfigurator.php	0
papi1.5-ip-authz.php	0
Function ipHook	0
Class SourceIPAddrAuthzEngine	0
Var \$valid_hooks	0
Method authorize	0
Method getAuthorizedList	0
Method isAuthorized	0
Method revoke	0

Class SourceIPAddrConfigurator	0
Var \$mandatory_options	0
Method getAllowed	0
Method getDenied	0
Appendix A - Class Trees	0
phpPoA2	0
INSTALL	0
README	0

AutoPoA.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

definitions.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

AS_T = 1 [*line 80*]

'AS' type redirection in the PAPI 1.5 protocol.

AUTHN_FAILED = false [*line 35*]

Authentication failed.

AUTHN_SUCCESS = true [*line 30*]

Authentication succeeded.

AUTHZ_FAILED = false [*line 45*]

Authorization failed.

AUTHZ_SUCCESS = true [*line 40*]

Authorization succeeded.

CONFIG_ERR = 2 [*line 60*]

A configuration related error.

GPOA_T = 0 [*line 75*]

'GPoA' type redirection in the PAPI 1.5 protocol.

INVITE_ERR = 3 [*line 65*]

An invitation related error.

NOAUTH_ERR = 0 [*line 50*]

Authentication failed error.

SYSTEM_ERR = 1 [*line 55*]

A system related error.

USER_ERR = 4 [*line 70*]

An error triggered by the user.

AuthenticationEngine.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

crypt.php

- **Package** phpPoA2
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

void function decrypt_3DES(\$input, \$key) [line [114](#)]

Function Parameters:

- **\$input**
- **\$key**

void function decrypt_AES(\$input, \$key) [line [67](#)]

Function Parameters:

- **\$input**
- **\$key**

void function encrypt_3DES(\$input, \$key) [line [92](#)]

Function Parameters:

- **\$input**
- **\$key**

void function encrypt_AES(\$input, \$key) [line [46](#)]

Function Parameters:

- **\$input**
- **\$key**

void function openssl_decrypt(\$in, \$key, \$key_bits) [line [41](#)]

Function Parameters:

- **\$in**
- **\$key**
- **\$key_bits**

void function openssl_encrypt(\$in, \$key, \$key_bits) [line [36](#)]

Function Parameters:

- **\$in**
- **\$key**
- **\$key_bits**

PADDINGSIZE = 11 [line [29](#)]

- **Version 2.0**
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

void function papi_openssl_decrypt(\$in, \$key, [\$key_bits = 0]) [*line* [187](#)]

Function Parameters:

- **\$in**
- **\$key**
- **\$key_bits**

void function papi_openssl_encrypt(\$in, \$key, [\$key_bits = 0]) [*line* [144](#)]

Function Parameters:

- **\$in**
- **\$key**
- **\$key_bits**

GenericConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

GenericEngine.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

Hook.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

LitePoA.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

PoA.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

include_once ["LitePoA.php"](#)*[line 40]*

include_once ["AutoPoA.php"](#)*[line 39]*

require_once ["PoAEventHandler.php"](#)*[line 38]*

require_once ["PoAUtils.php"](#)*[line 37]*

PoAConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

PoAEventHandler.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

PoAException.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

PoALog.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

PoAUtills.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

papi1.5-slo-extra.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

include ["./PoA.php"](#)*[line 9]*

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

papi1.5-slo.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

include ["./PoA.php"](#)*[line 9]*

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

authz_convert.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

ROOT = 'PATH_TO_YOUR_DB_DIR' [*line 30*]

Authorized DB converter tool from phpPoA1 to phpPoA2.

invite_convert.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

Class AuthenticationEngine

[line [31](#)]

Abstract authentication engine.

- **Package** phpPoA2
- **Abstract Element**

AuthenticationEngine::\$engine_type

mixed = "Authn" [[line 33](#)]

- **Access** protected

boolean function AuthenticationEngine::authenticate() [[line 41](#)]

Trigger the authentication of the user in the current context. May perform HTTP redirections or any other procedure to gather the authentication status if the user, so don't expect it to return control once called.

- **Abstract Element**
- **Access** public

string|array function AuthenticationEngine::getAttribute(\$name, [\$namespace = ""]) [[line 65](#)]
Function Parameters:

- *name* **\$name** The name of the attribute.
- *namespace* **\$namespace** The namespace of the attribute.

Get an specific attribute for the current user, by specifying its name and namespace. May trigger some attribute query procedure.

- **Abstract Element**
- **Access** public

array function AuthenticationEngine::getAttributes() [[line 55](#)]

Get an associative array with the set of common attributes for the current user. May trigger some attribute recollection.

- **Abstract Element**
- **Access** public

boolean function AuthenticationEngine::isAuthenticated() [*line* [48](#)]

Check the current authentication status withing this context. Does not trigger any authentication procedure.

- **Abstract Element**
- **Access** public

void function AuthenticationEngine::logout([\$slo = false]) [*line* [72](#)]

Function Parameters:

- *slo* **\$slo** Whether to perform the logout globally (Single Log Out) or not. Defaults to local.

Perform a logout (locally or global).

- **Abstract Element**
- **Access** public

Class AuthorizationConfigurator

[*line* [26](#)]

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Abstract Element**
- **License** [GNU General Public License](#)

boolean function AuthorizationConfigurator::getDefaultBehaviour() [*line 33*]

Determines the default behaviour of the authorization engine, whether it should authorize or not if no pattern (nor allowed nor denied) matches.

- **Access** public

Class AuthorizationEngine

[*line 26*]

- **Package** phpPoA2

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Abstract Element**
- **License** [GNU General Public License](#)

AuthorizationEngine::\$engine_type

mixed = "Authz" [*line 28*]

- **Access** protected

boolean function AuthorizationEngine::authorize(\$user, \$attrs, \$ref, [\$expires = 0]) [*line 47*]

Function Parameters:

- **\$user**
- **\$attrs**
- **\$ref**
- **\$expires**

- **Abstract Element**
- **Access** public

array function AuthorizationEngine::getAuthorizedList() [*line 42*]

- **Abstract Element**
- **Access** public

boolean function AuthorizationEngine::isAuthorized(\$user, \$attrs) [*line 37*]

Function Parameters:

- *user* **\$user** The string that identifies the user.
- *attrs* **\$attrs** All attributes related to the user.

Check authorization for the specified user.

- **Abstract Element**
- **Access** public

boolean function AuthorizationEngine::revoke(\$mail) [*line 52*]

Function Parameters:

- **\$mail**

- **Abstract Element**
- **Access** public

Class AutoPoA

[[line 30](#)]

Class that automatically redirects to error pages if any error is detected.

- **Package** phpPoA2

void function AutoPoA::authenticate() [[line 32](#)]

- **Access** public

void function AutoPoA::isAuthorized(\$user, \$attrs, [\$engine = null]) [[line 40](#)]

Function Parameters:

- **\$user**
- **\$attrs**
- **\$engine**

- **Access** public

Class GenericConfigurator

[[line 31](#)]

Generic configuration class.

- **Package** phpPoA2
- **Abstract Element**

GenericConfigurator::\$cfg

mixed = array() [[line 33](#)]

- **Access** protected

GenericConfigurator::\$mandatory_options

mixed = array() [[line 34](#)]

- **Access** protected

Constructor *void* function GenericConfigurator::__construct(\$file, \$site) [[line 42](#)]

Function Parameters:

- *file* **\$file** The file which stores the configuration.
- *site* **\$site** The section of the configuration that applies for the current site.

Main constructor.

- **Throws** PoAException if any error occurs.

- **Access** public

void function GenericConfigurator::configure(\$file, \$site) [*line* [59](#)]

Function Parameters:

- *file* **\$file** The file which stores the configuration.
- *site* **\$site** The section of the configuration that applies for the current site.

Read the configuration from the specified file and section.

- **Throws** PoAException if any error occurs.
- **Access** protected

boolean function GenericConfigurator::validate([\$mandatory = null]) [*line* [84](#)]

Function Parameters:

- *mandatory* **\$mandatory** Optional. An array of mandatory attributes that should be searched within the current configuration.

Check all mandatory attributes are set.

- **Throws** PoAException if any error occurs.
- **Access** public

Class GenericEngine

[[line 31](#)]

A generic engine class.

- **Package** phpPoA2
- **Abstract Element**

GenericEngine::\$cfg

mixed = [[line 33](#)]

- **Access** protected

GenericEngine::\$engine_type

mixed = [[line 36](#)]

- **Access** protected

GenericEngine::\$handler

mixed = [[line 37](#)]

- **Access** protected

GenericEngine::\$hooks

mixed = array() [[line 34](#)]

- **Access** protected

GenericEngine::\$valid_hooks

mixed = array() [*line* [35](#)]

- **Access** protected

Constructor *void* function GenericEngine::__construct(\$file, \$site) [*line* [44](#)]

Function Parameters:

- *file* **\$file** The path to the configuration file. Can be in the include path.
- *section* **\$site** The section of the configuration file, if any.

Main constructor for the engine.

- **Access** public

boolean function GenericEngine::addHook(\$name, \$hook) [*line* [71](#)]

Function Parameters:

- *name* **\$name** The name of the hook.
- *hook* **\$hook** A mixed object. Can be the name of a function (string) or an array with two elements: the former, the name of a class or an object, and the latter the name of the method.

Adds a function to the specified hook, which will be executed at some point of the code.

- **Access** public

void function GenericEngine::clean() [*line* [150](#)]

Unregister error and exception handlers. Use it only for methods not declared in the interface that previously called registerHandler() method.

- **Access** protected

void function GenericEngine::configure(\$file, \$section) [*line* [54](#)]

Function Parameters:

- *file* **\$file** The path to the configuration file. Can be in the include path.
- *section* **\$section** The section of the configuration file, if any.

Configure the engine.

- **Access** public

void function GenericEngine::registerHandler() [*line* [140](#)]

Register error and exception handlers for logging. Use it only for methods not declared in the interface that could trigger errors.

- **Access** protected

boolean function GenericEngine::removeHook(\$name, \$hook) [*line* [93](#)]

Function Parameters:

- *name* **\$name** The name of the hook.
- *hook* **\$hook** A mixed object. Can be the name of a function (string) or an array with two elements: the former, the name of a class or an object, and the latter the name of the method.

Removes a function from the specified hook.

- **Access** public

void function GenericEngine::runHooks(\$hook, &\$params) [[line 118](#)]

Function Parameters:

- *hook* **\$hook** The name of the hook.
- *params* **&\$params** An array with all params (in order) that must be passed to the function.

Run all hooks attached to an specific action.

- **Access** protected

void function GenericEngine::setHandler(\$handler) [[line 132](#)]

Function Parameters:

- *handler* **\$handler** The event handler to use.

Set the event handler to the one specified.

- **Access** public

Class Hook

[line [30](#)]

Hook class that allows to run a function or a method specified by their name.

- **Package** phpPoA2

Hook::\$_callable_name

mixed = [line [33](#)]

- **Access** protected

Hook::\$_hook

mixed = [line [32](#)]

- **Access** protected

Hook::\$_name

mixed = [line [34](#)]

- **Access** protected

Constructor *boolean* function Hook::__construct(\$hook) [[line 42](#)]

Function Parameters:

- *hook* **\$hook** The name of a function or an array specifying a class and its method.

Build a new hook.

- **Throws** PoAException if something goes wrong.
- **Access** public

string function Hook::getName() [[line 120](#)]

Get the name of the hook.

- **Access** public

boolean function Hook::run(&\$args) [[line 103](#)]

Function Parameters:

- *args* **&\$args** An array with all the params the function receives.

Run the hook with the specified params.

- **Access** public

Class LitePoA

[line 33]

Class that embeds all functionality (authentication, attribute collection and authorization) in one method, just like older versions of phpPoA.

- **Package** phpPoA2
- **Deprecated** This class is for backwards compatibility. Please avoid using it.

LitePoA::\$handler

mixed = [line 35]

- **Access** protected

A function LitePoA::checkAccess() [line 55]

A shortcut for the whole authentication and authorization process.

User will be authenticated, his attributes collected and then he will be checked for authorization with the identifier built with the ID builder function.

- **Deprecated** This method is for backwards compatibility. Please avoid using it.
- **Access** public

void function LitePoA::setIDBuilder(\$handler) [line 43]

Function Parameters:

- *handler* **\$handler** The name of the function.

Set an attribute handler to build identifiers for the users according to their attributes.
Method must receive a hash of attributes ("name" => "value").

- **Access** public

Class PoA

[line [46](#)]

Standard class that implements all the functionality of the phpPoA.

- **Package** phpPoA2

PoA::\$attributes

mixed = [line [51](#)]

- **Access** protected

PoA::\$authn_engine

mixed = [line [50](#)]

- **Access** protected

PoA::\$authz_engines

mixed = [[line 52](#)]

- **Access** protected

PoA::\$autoload

mixed = [[line 54](#)]

- **Access** protected

PoA::\$cfg

mixed = [[line 48](#)]

- **Access** protected

PoA::\$db_manager

mixed = [[line 53](#)]

- **Access** protected

PoA::\$handler

mixed = [[line 55](#)]

- **Access** protected

PoA::\$local_site

mixed = [line [47](#)]

- **Access** protected

PoA::\$log

mixed = [line [49](#)]

- **Access** protected

Constructor *void* function PoA::__construct(\$site, [\$file = null]) [line [62](#)]

Function Parameters:

- *site* **\$site** The identifier to determine which configuration to apply.
- *file* **\$file** The path to the configuration file.

Main constructor. Configures the PoA and performs initialization.

- **Access** public

true function PoA::addHook(\$name, \$hook) [line [127](#)]

Function Parameters:

- *name* **\$name** The name of the hook. Refer to each individual engine for a complete list of available hooks.
- *hook* **\$hook** A hook object with the function or method to attach.

Attach a hook object to the appropriate entry point of the available authentication or authorization engines.

- **Access** public

AUTHN_SUCCESS function PoA::authenticate() [*line* [178](#)]

Perform a federated login for the user.

- **Access** public

AUTHZ_SUCCESS function PoA::authorize(\$user, \$attrs, [\$reference = null], [\$expires = 0], [\$engine = null]) [*line* [375](#)]

Function Parameters:

- *user* **\$user** The subject of authorization.
- *attrs* **\$attrs** The attributes of the user.
- *reference* **\$reference** An internal reference that may be valuable for the engine, typically referring to a previous invitation or similar.
- *expires* **\$expires** The time (POSIX) when authorization will expire. Use 0 if authorization should never expire. Defaults to 0.
- *engine* **\$engine** The authorization engine(s) to use. All engines are used if none specified. If more than one engine should be checked then this must be an array.

Authorize a given subject with the data retrieved from federated login.

Multiple authorization engines are supported, so authorization will be done in all of them.

- **Access** public

void function PoA::clean() [*line* [113](#)]

- **Access** protected

the function PoA::getAttribute(\$name, \$namespace) [*line* [265](#)]

Function Parameters:

- *name* **\$name** The name of the attribute.
- *namespace* **\$namespace** The namespace of the attribute, if any.

Get the value (or values) of an attribute, if present.

- **Access** public

an function PoA::getAttributes() [*line* [242](#)]

Retrieve the attributes provided by the user when logged in.

- **Access** public

The function PoA::getAuthorizationEngines([\$engine = null]) [*line* [456](#)]

Function Parameters:

- *engine* **\$engine** The name of the authorization engine to retrieve. If more than one engine should be returned then this must be an array.

Returns the authorization engines configured for the current PoA, or the one specified.

- **Access** public

An function PoA::getAuthorizationLevels(\$user, \$attributes) [[line 490](#)]

Function Parameters:

- *user* **\$user** The user identifier.
- *attributes* **\$attributes** An array of attributes available for the user.

Get the authorization levels that match for the user specified, according to the configuration.

An array with the names of the levels matching the user is returned. An empty array is returned if no match is found.

- **Access** public

AUTHN_SUCCESS function PoA::isAuthenticated() [[line 213](#)]

Query the current status of the user in the federation.

- **Access** public

AUTHZ_SUCCESS function PoA::isAuthorized(\$user, \$attrs, [\$engine = null]) [[line 318](#)]

Function Parameters:

- *user* **\$user** The subject queried.
- *attrs* **\$attrs** The attributes of the user.
- *engine* **\$engine** The authorization engine(s) to use. All engines are used if none specified. If more than one engine should be checked then this must be an array.

Perform authorization for the a given subject.

Multiple authorization engines are supported, so authorization will succeed if any of these succeeds.

- **Access** public

true function PoA::logout(\$slo) [*line* [287](#)]

Function Parameters:

- *slo* **\$slo** Whether to perform a Single Log Out or a local logout.

Remove the user's session and trigger a logout for the specified authentication protocol.

- **Access** public

true function PoA::removeHook(\$name, \$hook) [*line* [152](#)]

Function Parameters:

- *name* **\$name** The name of the hook. Refer to each individual engine for a complete list of available hooks.
- *hook* **\$hook** The hook object which shall be removed.

Remove a hook from the specified entry point of the available authentication or authorization engines.

- **Access** public

true function PoA::revoke(\$mail, [\$engine = null]) [*line* [415](#)]

Function Parameters:

- *mail* **\$mail** The e-mail of the user.
- *engine* **\$engine** The authorization engine(s) to use. All engines are used if none specified. If more than one engine should be checked then this must be an array.

Revoke authorization for a given subject identified by an e-mail.

- **Access** public

Class PoAConfigurator

[*line* [30](#)]

Main configurator for the phpPoA.

- **Package** phpPoA2

PoAConfigurator::\$mandatory_options

mixed = array('LogFile',
 'LogLevel') [*line* [32](#)]

- **Access** protected

Constructor *void* function PoAConfigurator::__construct(\$site, [\$file = null]) [*line* [35](#)]

Function Parameters:

- **\$site**
- **\$file**

- **Access** public

void function PoAConfigurator::getAuthnEngine() [*line* [93](#)]

- **Access** public

void function PoAConfigurator::getAuthnEngineConfFile() [*line* [105](#)]

- **Access** public

void function PoAConfigurator::getAuthzEngineConfFile(\$engine) [*line* [109](#)]

Function Parameters:

- **\$engine**

- **Access** public

void function PoAConfigurator::getAuthzEngines() [*line* [97](#)]

- **Access** public

void function PoAConfigurator::getAuthzLevels() [*line* [113](#)]

- **Access** public

void function PoAConfigurator::getInviteErrorURL() [*line* [89](#)]

- **Access** public

void function PoAConfigurator::getLanguage() [*line* [73](#)]

- **Access** public

void function PoAConfigurator::getLogFile() [*line* [65](#)]

- **Access** public

void function PoAConfigurator::getLogLevel() [*line* [77](#)]

- **Access** public

void function PoAConfigurator::getNoAuthErrorURL() [*line* [81](#)]

- **Access** public

void function PoAConfigurator::getSystemErrorURL() [*line* [85](#)]

- **Access** public

void function PoAConfigurator::isDebug() [*line* [69](#)]

- **Access** public

Class PoAEventHandler

[*line* [30](#)]

Class to handle events inside the library.

- **Package** phpPoA2

Constructor *void* function PoAEventHandler::__construct(\$site, [\$log = null], [\$debug = false]) [*line* [39](#)]

Function Parameters:

- **\$site**
- **\$log**
- **\$debug**

Main constructor.

- **Access** public

`void function PoAEventHandler::abort($code, $message) [line 64]`

Function Parameters:

- **\$code**
- **\$message**

Finish execution and print an error message with the current backtrace.

- **Access** public

`void function PoAEventHandler::autoloadHandler($class_name) [line 83]`

Function Parameters:

- **\$class_name**

Override `__autoload` standard function to allow loading classes dinamically.

`void function PoAEventHandler::errorHandler($code, $message, $file, $line) [line 116]`

Function Parameters:

- *code* **\$code** The error code.
- *message* **\$message** The error message.
- *file* **\$file** The file that triggered the error.
- *line* **\$line** The line of the file where the error was triggered.

A generic error handler that logs the error message.

- **Access** public

void function PoAEventHandler::exceptionHandler(\$exception) [line [105](#)]

Function Parameters:

- *exception* **\$exception** The caught exception.

A generic exception handler that logs the exception message.

- **Access** public

void function PoAEventHandler::setDebug(\$debug) [line [57](#)]

Function Parameters:

- *debug* **\$debug** A boolean value telling whether to make debugging active or not.

Set debugging mode.

- **Access** public

void function PoAEventHandler::setLogger(\$log) [line [49](#)]

Function Parameters:

- *log* **\$log** The logger to use.

Set the logger.

- **Access** public

Class PoAException

[line [30](#)]

Exception handler class for the phpPoA.

- **Package** phpPoA2

Constructor *void* function PoAException::__construct(\$message, \$code, [\$args = null]) *[line [33](#)]*

Function Parameters:

- **\$message**
- **\$code**
- **\$args**

- **Access** public

void function PoAException::__toString() *[line [40](#)]*

- **Access** public

Class PoALog

[[line 30](#)]

A simple class to write log messages to a file.

- **Package** phpPoA2

Constructor *void* function PoALog::__construct(\$level, \$file) [[line 39](#)]

Function Parameters:

- *level* **\$level** The level of this log.
- *file* **\$file** The file where to write messages.

Build a new logger.

- **Access** public

integer function PoALog::getLogLevel() [[line 72](#)]

Get the current log level.

- **Access** public

void function PoALog::write(\$msg, [\$level = E_USER_NOTICE]) [[line 58](#)]

Function Parameters:

- *msg* **\$msg** The message to write.
- *level* **\$level** The level of the message.

Write a message to the log file if the current log level allows it.

- **Access** public

Class PoAUtils

[line [27](#)]

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es >
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

string function PoAUtils::lang_code(\$filename) [*line* [46](#)]

Function Parameters:

- *filename* **\$filename** The name of the file.

Return the language code identifying applicable to a messages file.

- **Static**
- **Access** public

string function PoAUtils::msg(\$s, [\$args = array()]) [*line* [35](#)]

Function Parameters:

- *s* **\$s** The identifier of the message.
- *args* **\$args** An array of arguments that the message expects.

Return the internationalized message identified by the code 's'.

- **Static**
- **Access** public

AttributeFilterAuthzEngine.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

ATTRIBUTE_BEFORE_FILTERS = "ATTRIBUTE_BEFORE_FILTERS" [*line 37*]

This hook is executed right after retrieving the arrays of allowed and denied attributes that will be checked immediately.

It can be used to configure the filters on runtime and modify the user's attributes. The hook receives the attributes, and the allowed and denied attribute arrays. Functions for this hook must be defined like this:

```
function attributeBeforeFilterHook(&$attrs, &$allowed, &$denied);
```

Please bear in mind that hooks must return TRUE or they'll keep other hooks from executing.

AttributeFilterConfigurator.php

- **Package** phpPoA2
- **Sub-Package** AttributeFilterAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es >
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

papi1.5-attr-authz.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Sub-Package** AttributeFilterAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

void function attrHook(&\$attrs, &\$allowed, &\$denied) [line 17]

Function Parameters:

- **&\$attrs**
- **&\$allowed**
- **&\$denied**

include ["./PoA.php"](#) [line 10]

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

Class AttributeFilterAuthzEngine

[line 46]

Authorization engine that works by checking the attributes of the user. The first match of an attribute against one of the filters will trigger the authorization result, no matter if it's positive or negative.

- **Package** phpPoA2
- **Sub-Package** AttributeFilterAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

AttributeFilterAuthzEngine::\$valid_hooks

mixed = array(ATTRIBUTE_BEFORE_FILTERS) [*line* [48](#)]

- **Access** protected

void function AttributeFilterAuthzEngine::authorize(\$user, \$attrs, \$ref, [\$expires = 0]) [*line* [93](#)]

Function Parameters:

- **\$user**
- **\$attrs**
- **\$ref**

- **\$expires**

- **Access** public

void function AttributeFilterAuthzEngine::getAuthorizedList() [*line* [86](#)]

- **Access** public

void function AttributeFilterAuthzEngine::isAuthorized(\$user, \$attrs) [*line* [50](#)]

Function Parameters:

- **\$user**
- **\$attrs**

- **Access** public

void function AttributeFilterAuthzEngine::revoke(\$mail) [*line* [97](#)]

Function Parameters:

- **\$mail**

- **Access** public

Class AttributeFilterConfigurator

[line 33]

Configurator for the Attribute Filter Authorization Engine.

- **Package** phpPoA2
- **Sub-Package** AttributeFilterAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

AttributeFilterConfigurator::\$mandatory_options

mixed = array("Allowed", "Denied") [line 35]

- **Access** protected

array function AttributeFilterConfigurator::getAllowedAttributes() [line 42]

Returns an array of attributes that would be allowed if their values match any of the patterns for each attribute.

- **Access** public

array function AttributeFilterConfigurator::getDeniedAttributes() [*line 51*]

Returns an array of attributes that would be denied if their values match any of the patterns for each attribute.

- **Access** public

DBADB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

GenericDB.php

- **Package** phpPoA2
- **Author** Candido Rodriguez < candido.rodriguez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

GenericMySQLDB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

SessionDB.php

- **Package** phpPoA2
- **Author** Candido Rodriguez < candido.rodriguez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

Class DBADB

[*line* [31](#)]

Berkeley DB backend.

- **Package** phpPoA2
- **Sub-Package** GenericDatabaseHandlers

DBADB::\$mandatory_options

mixed = array("DBFile") [[line 33](#)]

- **Access** protected

void function DBADB::check(\$key) [[line 47](#)]

Function Parameters:

- **\$key**

- **Access** public

void function DBADB::close() [[line 78](#)]

- **Access** public

void function DBADB::configure() [[line 35](#)]

- **Access** protected

void function DBADB::delete(\$key) [[line 71](#)]

Function Parameters:

- **\$key**

- **Access** public

void function DBADB::fetch(\$key) [*line* [57](#)]

Function Parameters:

- **\$key**

- **Access** public

void function DBADB::fetch_all() [*line* [61](#)]

- **Access** public

void function DBADB::open() [*line* [42](#)]

- **Access** public

void function DBADB::replace(\$key, \$value) [*line* [51](#)]

Function Parameters:

- **\$key**
- **\$value**

- **Access** public

Class GenericDB

[line 31]

Provides the basic operations and defines the interface of any DB operating class.

- **Package** phpPoA2
- **Sub-Package** GenericDatabaseHandlers
- **Abstract Element**

GenericDB::\$cfg

mixed = [line 33]

- **Access** protected

GenericDB::\$db

mixed = [line 35]

- **Access** protected

GenericDB::\$error

mixed = null [line 34]

- **Access** protected

GenericDB::\$mandatory_options

mixed = array() [[line 36](#)]

- **Access** protected

Constructor *void* function GenericDB::__construct(\$cfg) [[line 42](#)]

Function Parameters:

- *cfg* **\$cfg** A *Configurator object with the current configuration.

Main constructor.

- **Access** public

boolean function GenericDB::check(\$key) [[line 76](#)]

Function Parameters:

- *key* **\$key** The key to look for.

Checks if the specified key exists in the database.

- **Abstract Element**
- **Access** public

boolean function GenericDB::close() [[line 69](#)]

Closes the database.

- **Abstract Element**
- **Access** public

boolean function GenericDB::configure() [[line 54](#)]

Configure the database handler.

- **Access** protected

boolean function GenericDB::delete(\$key) [[line 105](#)]

Function Parameters:

- *key* **\$key** The key to remove.

Removes the specified key from the database.

- **Abstract Element**
- **Access** public

mixed|boolean function GenericDB::fetch(\$key) [[line 92](#)]

Function Parameters:

- *key* **\$key** The key to look for.

Gets the value of the specified key.

- **Abstract Element**
- **Access** public

array function GenericDB::fetch_all() [[line 98](#)]

Gets all the contents stored in the database.

- **Abstract Element**
- **Access** public

boolean function GenericDB::open() [[line 63](#)]

Opens the database.

- **Abstract Element**
- **Access** public

boolean function GenericDB::replace(\$key, \$value) [[line 85](#)]

Function Parameters:

- *key* **\$key** The key to replace.
- *value* **\$value** The new value for the specified key.

Replaces the specified key with a new value. If the key does not exist previously, it will be created.

- **Abstract Element**
- **Access** public

Class GenericMySQLDB

[[line 32](#)]

Generic MySQL database class.

- **Package** phpPoA2
- **Sub-Package** GenericDatabaseHandlers
- **Abstract Element**

GenericMySQLDB::\$create_sql

mixed = [[line 38](#)]

- **Access** protected

GenericMySQLDB::\$delete_sql

mixed = [[line 38](#)]

- **Access** protected

GenericMySQLDB::\$insert_sql

mixed = [[line 38](#)]

- **Access** protected

GenericMySQLDB::\$mandatory_options

```
mixed = array("DBHost",  
              "DBUser",  
              "DBPassword",  
              "DBName") [line 34]
```

- **Access** protected

GenericMySQLDB::\$prefix

```
mixed = [line 39]
```

- **Access** protected

GenericMySQLDB::\$search_all_sql

```
mixed = [line 38]
```

- **Access** protected

GenericMySQLDB::\$search_sql

```
mixed = [line 38]
```

- **Access** protected

GenericMySQLDB::\$update_sql

```
mixed = [line 38]
```

- **Access** protected

void function GenericMySQLDB::check(\$key) [[line 62](#)]

Function Parameters:

- **\$key**

void function GenericMySQLDB::close() [[line 95](#)]

void function GenericMySQLDB::configure() [[line 41](#)]

- **Access** protected

void function GenericMySQLDB::delete(\$key) [[line 90](#)]

Function Parameters:

- **\$key**

void function GenericMySQLDB::fetch(\$key) [[line 78](#)]

Function Parameters:

- **\$key**

void function GenericMySQLDB::fetch_all() [[line 84](#)]

void function GenericMySQLDB::getError() [[line 58](#)]

void function GenericMySQLDB::open() [[line 48](#)]

void function GenericMySQLDB::replace(\$key, \$value) [[line 68](#)]

Function Parameters:

- **\$key**
- **\$value**

Class SessionDB

[*line* [33](#)]

Session database backend.

WARNING: please note that PHP only allows one session at a time, so using this backend will break any applications that make use of sessions beneath.

- **Package** phpPoA2
- **Sub-Package** GenericDatabaseHandlers

SessionDB::\$session_name

mixed = "session_db" [*line* [35](#)]

- **Access** protected

void function SessionDB::check(\$key) [*line* [41](#)]

Function Parameters:

- **\$key**

- **Access** public

void function SessionDB::close() [*line* [99](#)]

- **Access** public

void function SessionDB::delete(\$key) [*line* [83](#)]

Function Parameters:

- **\$key**

- **Access** public

void function SessionDB::fetch(\$key) [*line* [63](#)]

Function Parameters:

- **\$key**

- **Access** public

void function SessionDB::fetch_all() [*line* [73](#)]

- **Access** public

void function SessionDB::open() [*line* [37](#)]

- **Access** public

void function SessionDB::replace(\$key, \$value) [*line* [52](#)]

Function Parameters:

- **\$key**
- **\$value**

- **Access public**

messages-en_EN.php

- **Package** phpPoA2
- **Sub-Package** Internationalization
- **Author** Jaime Perez < jaime.perez@rediris.es >
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

\$messages

```
array = array(
'fatal-error'      => 'Fatal error',
'error-desc'       => 'There was an unexpected error. Please contact the administrator and specify
the session identifier and backtrace as shown below.',
'error-message'    => 'Error message',
'session-id'       => 'Session identifier',
'backtrace'        => 'Backtrace',

'invalid-php-version' => 'Invalid PHP version. PAPI extension requires at least PHP %s.',
'extension-required' => 'Extension "%s" is required, but it is not loaded.',
'library-required'   => 'Library "%s" is required, but it cannot be found.',
'authenticating-via' => 'Authenticating via the "%s" engine.',
'authorize-user-via' => 'Authorizing user "%s" via the
"%s" engine.',
'query-authz'        => 'Querying for authorization with all available engines.',
'query-authz-via'    => 'Querying for authorization with "%s".',
'authz-engine-err'   => 'Cannot run unknown authorization engine "%s", check
your configuration file.',
'authz-levels-err'   => 'No authorization levels defined for this PoA.'
```



```

'authn-engine-err'    => 'No authentication engine found, check your configuration file.',
'check-authn-status' => 'Checking current authentication status via the %s engine.',
'invalid-config'     => 'Invalid configuration object.',
'config-not-found'   => 'Cannot configure PoA, configuration file "%s" not
found.',
'config-err-php'     => 'Cannot configure PoA, "%%1$s" or
"%%1$s[%2$s]" not set.',
'config-err-ini'     => 'Cannot configure PoA, "%s" section not found.',
'config-param-err'   => 'Cannot configure PoA, "%s" param not found.',
'cannot-open-log'    => 'Cannot open log file "%s".',
'cannot-write-log'   => 'Cannot write log file "%s".',
'class-not-found'     => 'Cannot find class "%s".',
'authn-success'      => 'Successful authentication with the "%s" engine.',
'authn-err'          => 'Authentication error.',
'authz-err'          => 'User "%s" not authorized.',
'authz-expired'      => 'Authorization has expired for user "%s".',
'authz-default-fallback'=> 'Cannot make any authorization decision, falling to default fallback.',
'valid-cookie'       => 'The cookie is valid',
'cookie-not-found'   => 'Cannot find cookie "%s".',
'cookie-location-err' => 'Invalid cookie location: "%s".',
'cookie-service-err' => 'Service ID "%s" does not match for cookie, should be
"%s".',
'cookie-expired-err' => 'Expired cookie.',
'cookie-rejected-err' => 'Assertion matches rejection cookie filter "%s".',
'cannot-set-cookie'  => 'Cannot set cookie. Please check no output was already sent to the
browser.',
'empty-cookie-err'   => 'Empty cookie "%s" found.',
'empty-response-err' => 'Empty AS/GPoA response.',
'pubkey-error'       => 'Cannot read public key from file "%s".',
'cannot-decrypt'     => 'Cannot decrypt response, please check AS/GPoA public key.',
'valid-response'     => 'AS/GPoA response authenticates assertion "%s".',
'expired-response'   => 'AS/GPoA response is expired.',
'unknown-request'    => 'AS/GPoA response received for an unknown request.',
'req-db-purged'      => 'Purged %d outdated requests from the database.',
'redirecting'        => 'Redirecting the user to "%s".',
'cannot-redirect'    => 'Cannot redirect the user.',
'cannot-save-request' => 'Cannot save the original request.',
'cannot-open-req-db'  => 'Cannot open request database.',
'cannot-open-authz-db' => 'Cannot open authorized database.',
'cannot-open-inv-db'  => 'Cannot open invitations database.',
'cannot-fetch-key'    => 'Cannot find database key "%s".',
'cannot-del-key'      => 'Cannot delete database key "%s".',
'as-id-error'         => 'Authentication Server identifier "%s" does not match with
expected "%s".',
'continue'           => 'Continue',
'source-ip-allowed'  => 'The source IP "%s" matches allowed filter
"%s".',
'source-ip-denied'   => 'The source IP "%s" matches denial filter
"%s".',
'invalid-hook'       => 'Trying to set hook "%s" with an invalid function or
method.',
'running-hook'       => 'Running hook "%s" for "%s".',
'add-hook'           => 'Adding hook "%s" for "%s".',
'remove-hook'        => 'Removing hook "%s" for "%s".',
'hook-error'         => 'Invalid hook "%s".',
'allowed-param-match' => 'The following request parameters are allowed: [%s].',
'denied-param-match'  => 'The following request parameters are denied: [%s].',
'allowed-attr-match'  => 'The following user attributes are allowed: [%s].',
'denied-attr-match'   => 'The following user attributes are denied: [%s].',

```

```

'invite-non-existent' => 'Cannot find invitation with reference "%.".',
'missing-attrs'      => 'None of the mandatory attributes were found.',
'missing-mail-attr'  => 'Cannot find "%." attribute, e-mail verification failed.',
'mail-attr-err'      => 'Email verification attribute unset or empty, verification failed.',
'mail-attr-alg-err'  => 'Cannot extract the algorithm from the attribute "%." with
value "%.".',
'mail-attr-val-err'  => 'Cannot extract the e-mail verification string from the attribute
"%." with value "%.".',
'mail-verify-err'    => 'Cannot verify e-mail "%.", cannot match any of the
received values.',
'mail-verify-ok'     => 'Verification success for e-mail "%." with algorithm
"%." and value "%.".',
'cannot-authorize'   => 'Cannot authorize user "%.".',
'cannot-del-invite'  => 'Cannot remove invite for user "%." with reference
"%.".',
'invite-sent-to'     => 'Invite sent to address "%.".',
'user-authorized'    => 'User "%." is now authorized.',
'user-authz-ok'      => 'User "%." is authorized.',
'user-authz-err'     => 'User "%." is not authorized.',
'user-already-authz' => 'User "%." is already authorized.',
'slo-conf-error'     => 'Cannot perform single logout, please review configuration.',
'already-logged-out' => 'User was not authenticated.',
'local-logout-success' => 'User was successfully logged out.',
'slo-logout'         => 'Single logout requested.',
'slo-requested'      => 'User requested single logout.',
) [line 42]

```

An array with all the internationalized messages used by phpPoA2.

- **Name** \$messages

messages-es_ES.php

- **Package** phpPoA2
- **Sub-Package** Internationalization
- **Author** Jaime Perez < jaime.perez@rediris.es >
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

AuthorizedDB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

AuthorizedDBADB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

InviteAuthzEngine.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

```
AUTHORIZED_DBA = 'AuthorizedDBADB' [line 32]
AUTHORIZED_MYSQL = 'AuthorizedMySQLDB' [line 33]
AUTHORIZED_SESSION = 'AuthorizedSessionDB' [line 34]
INVITES_BEFORE_AUTHORIZATION = "INVITES_BEFORE_AUTHORIZATION" [line 48]
```

This hook is executed right before it is checked whether the user was invited or not to continue with authorization.

It can be used to trick the engine to believe the user was previously invited (and therefore forcing authorization). The hook receives a boolean parameter representing whether the referenced invitation was found or not.

```
function invitesBeforeAuthorization(&$invite_exists);
```

Please bear in mind that hooks must return TRUE or they'll keep other hooks from executing.

```
INVITES_DBA = 'InviteDBADB' [line 29]
```

Supported database types.

```
INVITES_MYSQL = 'InviteMySQLDB' [line 30]  
INVITES_SESSION = 'InviteSessionDB' [line 31]
```

InviteConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

InviteDB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

InviteDBADB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

papi1.5-invite-authz.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Sub-Package** InviteAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

include ["./PoA.php"](#)*[line 10]*

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

Class AuthorizedDB

[line 31]

Generic interface for a PAPI request database.

- **Package** phpPoA2
- **Sub-Package** InviteAuthorizationEngine

void function AuthorizedDB::replace_authorization(\$alias, \$attributes, \$email, \$expires) [[line 33](#)]

Function Parameters:

- **\$alias**
- **\$attributes**
- **\$email**
- **\$expires**

- **Access** public

Class AuthorizedDBADB

[[line 31](#)]

Authorization lists binding to a Berkeley DB backend.

- **Package** phpPoA2
- **Sub-Package** InviteAuthorizationEngine

AuthorizedDBADB::\$authorized

mixed = [[line 33](#)]

- **Access** protected

AuthorizedDBADB::\$mandatory_options

mixed = array("AuthorizedDBFile") [*line 34*]

- **Access** protected

void function AuthorizedDBADB::fetch(\$key) [*line 51*]

Function Parameters:

- **\$key**

- **Access** public

void function AuthorizedDBADB::fetch_all() [*line 61*]

- **Access** public

void function AuthorizedDBADB::open() [*line 36*]

- **Access** public

void function AuthorizedDBADB::replace_authorization(\$alias, \$attributes, \$email, \$expires) [*line 41*]

Function Parameters:

- **\$alias**
- **\$attributes**
- **\$email**
- **\$expires**

- **Access** public

Class InviteAuthzEngine

[line 55]

Invitation based authorization engine.

- **Package** phpPoA2
- **Sub-Package** InviteAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

InviteAuthzEngine::\$authz_db

mixed = [line 57]

- **Access** protected

InviteAuthzEngine::\$invites_db

mixed = [line [58](#)]

- **Access** protected

InviteAuthzEngine::\$valid_hooks

mixed = array(INVITES_BEFORE_AUTHORIZATION) [line [59](#)]

- **Access** protected

boolean function InviteAuthzEngine::authorize(\$user, \$attrs, \$ref, [\$expires = 0]) [line [172](#)]

Function Parameters:

- *user* **\$user**
- *attrs* **\$attrs**
- *ref* **\$ref**
- *expires* **\$expires**

Authorize the specified user.

- **Access** public

void function InviteAuthzEngine::configure(\$file, \$section) [[line 66](#)]

Function Parameters:

- *file* **\$file** The configuration file.
- *section* **\$section** The section of the configuration file to use.

Configure the authorization engine.

- **Access** public

boolean function InviteAuthzEngine::emailVerify(\$mail, \$attrs) [[line 370](#)]

Function Parameters:

- *mail* **\$mail** The e-mail of the user.
- *attrs* **\$attrs** The array of attributes of the user.

Perform e-mail verification for the current user according to the configuration for this site.

- **Access** protected

void function InviteAuthzEngine::finish(\$value) [[line 443](#)]

Function Parameters:

- *value* **\$value** The value that must be returned. @return mixed The first param.

Close databases and return the same value received as a parameter.

- **Access** protected

array function InviteAuthzEngine::getAuthorizedList() [[line 135](#)]

Get a list of all authorized users.

- **Access** public

array function InviteAuthzEngine::getPendingInvites() [[line 152](#)]

Get a list of all pending invitations.

- **Access** public

boolean function InviteAuthzEngine::invite(\$mail, [\$expires = 0]) [[line 309](#)]

Function Parameters:

- *mail* **\$mail** The e-mail of the user.
- *expires* **\$expires** The time (POSIX) when authorization will expire. Use 0 if authorization should never expire. Defaults to 0.

Send an invitation to an e-mail address (that is, send an e-mail to that address with instructions on how to get authorized and an URL to follow).

- **Access** public

boolean function InviteAuthzEngine::isAuthorized(\$user, \$attrs) [[line 88](#)]

Function Parameters:

- *user* **\$user** The name of the user to check authorization for.
- *attrs* **\$attrs** The attributes of the user to match his identity.

Check if the specified user is authorized with the given attributes.

- **Access** public

boolean function InviteAuthzEngine::removeInvite(\$ref) [*line* [349](#)]

Function Parameters:

- *ref* **\$ref** The reference to the invite to remove.

Remove an invitation from the database.

- **Access** public

boolean function InviteAuthzEngine::revoke(\$mail) [*line* [275](#)]

Function Parameters:

- **\$mail**

Revoke authorization for the specified user.

- **Access** public

Class InviteConfigurator

[line [31](#)]

Configurator for the Invitation based authorization engine.

- **Package** phpPoA2
- **Sub-Package** InviteAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

InviteConfigurator::\$mandatory_options

```
mixed = array('InvitesDBType',  
              'AuthorizedDBType',  
              'AdminEmail',  
              'InviteURL',  
              'InviteText',  
              'InviteSubject',  
              'UniqueAttributes') [line 33]
```

- **Access** protected

void function InviteConfigurator::doEmailVerify() [*line [113](#)*]

- **Access** public

string function InviteConfigurator::getAdminEmail() [*line [45](#)*]

Returns

- **Access** public

string function InviteConfigurator::getAuthorizedDBFile() [*line [188](#)*]

Returns

- **Access** public

string function InviteConfigurator::getAuthorizedDBHost() [*line [196](#)*]

Returns

- **Access** public

string function InviteConfigurator::getAuthorizedDBName() [*line [220](#)*]

Returns

- **Access** public

string function InviteConfigurator::getAuthorizedDBPassword() [*line* [212](#)]

Returns

- **Access** public

integer function InviteConfigurator::getAuthorizedDBType() [*line* [180](#)]

Returns

- **Access** public

string function InviteConfigurator::getAuthorizedDBUser() [*line* [204](#)]

Returns

- **Access** public

void function InviteConfigurator::getEmailVerifyAlgRegEx() [*line* [102](#)]

- **Access** public

void function InviteConfigurator::getEmailVerifyAttribute() [*line* [80](#)]

- **Access** public

void function InviteConfigurator::getEmailVerifyRegEx() [*line* [91](#)]

- **Access** public

string function InviteConfigurator::getInvitesDBFile() [*line* [140](#)]

Returns

- **Access** public

string function InviteConfigurator::getInvitesDBHost() [*line* [148](#)]

Returns

- **Access** public

string function InviteConfigurator::getInvitesDBName() [*line* [172](#)]

Returns

- **Access** public

string function InviteConfigurator::getInvitesDBPassword() [*line* [164](#)]

Returns

- **Access** public

integer function InviteConfigurator::getInvitesDBType() [*line [132](#)*]

Returns

- **Access** public

string function InviteConfigurator::getInvitesDBUser() [*line [156](#)*]

Returns

- **Access** public

string function InviteConfigurator::getInviteSubject() [*line [61](#)*]

Returns

- **Access** public

string function InviteConfigurator::getInviteText() [*line [53](#)*]

Returns

- **Access** public

string function InviteConfigurator::getInviteURL() [*line [124](#)*]

Returns

- **Access** public

array function InviteConfigurator::getUniqueAttributes() [*line* [69](#)]

Returns

- **Access** public

Class InviteDB

[*line* [31](#)]

Generic interface for a database of invites.

- **Package** phpPoA2
- **Sub-Package** InviteAuthorizationEngine

void function InviteDB::replace_invite(\$key, \$email, \$expires) [*line* [33](#)]

Function Parameters:

- **\$key**
- **\$email**
- **\$expires**

- **Access** public

Class InviteDBADB

[line [31](#)]

Invites binding to a Berkeley DB backend.

- **Package** phpPoA2
- **Sub-Package** InviteAuthorizationEngine

InviteDBADB::\$invite

mixed = [line [33](#)]

- **Access** protected

InviteDBADB::\$mandatory_options

mixed = array("InvitesDBFile") [line [34](#)]

- **Access** protected

void function InviteDBADB::fetch(\$key) [line [49](#)]

Function Parameters:

- **\$key**

- **Access** public

void function InviteDBADB::fetch_all() [[line 59](#)]

- **Access** public

void function InviteDBADB::open() [[line 36](#)]

- **Access** public

void function InviteDBADB::replace_invite(\$key, \$email, \$expires) [[line 41](#)]

Function Parameters:

- **\$key**
- **\$email**
- **\$expires**

- **Access** public

PAPIAuthnEngine.php

- **Package** phpPoA2

ATTR_SEPARATOR = "," *[line 31]*

Default assertion delimiters for standard PAPI 1.5 protocol.

NAMEVALUE_SEPARATOR = "=" *[line 33]*

NS_PAPI_ATTRIBUTES = 'urn:mace:rediris.es:papi:attributes' *[line 46]*

NS_PAPI_PROTOCOL = 'urn:mace:rediris.es:papi:protocol' *[line 45]*

Supported namespaces for attributes.

PAPI_ATTRIBUTE_PARSER = "PAPI_ATTRIBUTE_PARSER" *[line 110]*

This hook is executed when returning the attributes found for a user with `getAttributes()` method. It receives a string with the attributes and the array that results of processing the string. Functions for this hook must be defined like this:
function attributeParser(\$assertion, &\$attributes);

Please bear in mind that hooks must return TRUE or they'll keep other hooks from executing.

PAPI_CLEAN_URL = "PAPI_CLEAN_URL" *[line 99]*

This hook is executed when a valid response is found from the AS/GPoA and the engine is about

to end the authentication result. It receives a boolean value that determines if the URL should be cleaned by means of a redirection to the initial URL. Functions for this hook must be defined like this:

function cleanURLHook(&\$clean);

Please bear in mind that hooks must return TRUE or they'll keep other hooks from executing.

PAPI_DBA = 'PAPIDBADB' *[line 38]*

Supported database types.

PAPI_MYSQL = 'PAPIMySQLDB' *[line 39]*

PAPI_REDIRECT_URL_FINISH = "PAPI_REDIRECT_URL_FINISH" *[line 76]*

This hook is executed at the end of the method that returns the URL where to redirect a user.

It can be used to alter parameters in the URL. The hook receives an array of parameters which should be directly modified. Functions for this hook must be defined like this:

```
function redirectURLFinishHook(&$params);
```

Please bear in mind that hooks must return TRUE or they'll keep other hooks from executing.

```
PAPI_RESTORE_ORIGINAL_REQUEST = "PAPI_RESTORE_ORIGINAL_REQUEST" [line 87]
```

This hook is executed when a valid response is found from the AS/GPoA and the original request of the user is about to be restored. It receives an array with the main PHP global variables of the original context. Functions for this hook must be defined like this:

```
function restoreOriginalRequestHook(&$env);
```

Please bear in mind that hooks must return TRUE or they'll keep other hooks from executing.

```
PAPI_SESSION = 'PAPISessionDB' [line 40]
```

```
PHP_VERSION_REQUIRED = "5.2.0" [line 65]
```

Minimum PHP version required.

```
PROTO_ATTR_AS_ID = '__asid' [line 53]
```

```
PROTO_ATTR_EXPIRE_TIME = '__expiretime' [line 55]
```

```
PROTO_ATTR_KEY = '__key' [line 54]
```

```
PROTO_ATTR_PREFIX = '_papi_' [line 52]
```

Prefix for operational attributes inside the protocol and special attributes names.

```
REQUEST_LIFETIME = 300 [line 60]
```

Default timeout for stored requests.

```
require_once dirname(__FILE__)."/crypt.php" [line 26]
```

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **License** [GNU General Public License](#)

VALUE_SEPARATOR = "|" *[line 32]*

PAPIConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

PAPIDB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

PAPIDBADB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

PAPIMySQLDB.php

- **Package** phpPoA2

PAPI_DB_TABLE = 'requests' *[line 26]*

- **Author** Jaime Perez < jaime.perez@rediris.es >
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **License** [GNU General Public License](#)

PAPISessionDB.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

papi1.5-authn-attr.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

include ["./PoA.php"](#)*[line 10]*

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

papi1.5-customattr.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

void function handler(\$assertion, &\$attributes) [*line 20*]

Function Parameters:

- **\$assertion**
- **&\$attributes**

include ["./PoA.php"](#) [*line 10*]

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

papi1.5-papiopoa-hook.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

void function hook(&\$params) [line [14](#)]

Function Parameters:

- **&\$params**

include ["./PoA.php"](#) [line [10](#)]

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

Class AttributeHandler

[line [25](#)]

- **Package** phpPoA2

- **Sub-Package** PAPIAuthenticationEngine

void function AttributeHandler::handle(\$assertion, &\$attributes) [*line* [27](#)]

Function Parameters:

- **\$assertion**
- **&\$attributes**

- **Access** public

Class Example

[*line* [20](#)]

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine

void function Example::hook(&\$clean) [*line* [21](#)]

Function Parameters:

- **&\$clean**

- **Access** public

Class PAPIAuthnEngine

[line 118]

Authentication engine for the PAPI 1.5 protocol.

PLEASE NOTE THAT THIS ENGINE WORKS ONLY FOR WEB-BASED APPLICATIONS.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine

PAPIAuthnEngine::\$assertion

mixed = [line 120]

- **Access** protected

PAPIAuthnEngine::\$as_id

mixed = [line 124]

- **Access** protected

PAPIAuthnEngine::\$attributes

mixed = [line 123]

- **Access** protected

PAPIAuthnEngine::\$cfg

mixed = [line 131]

- **Access** protected

PAPIAuthnEngine::\$clean_url

mixed = true [line 132]

- **Access** protected

PAPIAuthnEngine::\$cookie_name

mixed = "PAPILcook_" [line 134]

- **Access** protected

PAPIAuthnEngine::\$db

mixed = [line 129]

- **Access** protected

PAPIAuthnEngine::\$enforcing

mixed = true [line 135]

- **Access** protected

PAPIAuthnEngine::\$expiration_time

mixed = false [*line 122*]

- **Access** protected

PAPIAuthnEngine::\$global_expire_time

mixed = [*line 128*]

- **Access** protected

PAPIAuthnEngine::\$id

mixed = [*line 130*]

- **Access** protected

PAPIAuthnEngine::\$key

mixed = [*line 125*]

- **Access** protected

PAPIAuthnEngine::\$lkey

mixed = [*line 126*]

- **Access** protected

PAPIAuthnEngine::\$opoa

mixed = "http" [*line 136*]

- **Access** protected

PAPIAuthnEngine::\$pkey

mixed = [*line 127*]

- **Access** protected

PAPIAuthnEngine::\$skip_redirection

mixed = false [*line 133*]

- **Access** protected

PAPIAuthnEngine::\$status

mixed = [*line 121*]

- **Access** protected

PAPIAuthnEngine::\$valid_hooks

mixed = array(PAPI_REDIRECT_URL_FINISH,
PAPI_RESTORE_ORIGINAL_REQUEST,
PAPI_CLEAN_URL,
PAPI_ATTRIBUTE_PARSER) [*line 137*]

- **Access** protected

void function PAPIAuthnEngine::attributeQuery(\$name) [*line 418*]

Function Parameters:

- **\$name**

- **Access** protected

void function PAPIAuthnEngine::authenticate() [*line 185*]

- **Access** public

void function PAPIAuthnEngine::configure(\$file, \$section) [*line 142*]

Function Parameters:

- **\$file**
- **\$section**

- **Access** public

true function PAPIAuthnEngine::deleteCookie() [*line 473*]

Delete the current cookie, if any.

- **Access** protected

boolean function PAPIAuthnEngine::deleteRequest(\$key) [*line 755*]

Function Parameters:

- key **\$key** The key that identifies the request.

Delete a request from the request database.

- **Access** protected

void function PAPIAuthnEngine::getAttribute(\$name, [\$namespace = NS_PAPI_ATTRIBUTES]) [*line 353*]

Function Parameters:

- **\$name**
- **\$namespace**

- **Access** public

void function PAPIAuthnEngine::getAttributes() [*line 325*]

- **Access** public

string function PAPIAuthnEngine::getNewCookie() [*line 780*]

Generate a new cookie for the current user.

- **Access** protected

string function PAPIAuthnEngine::getRedirectLocation() [*line 614*]

Retrieve the URL where to redirect a user and store his request.

- **Access** protected

string function PAPIAuthnEngine::getSingleLogoutLocation() [*line 574*]

Retrieve the URL where to redirect a user to perform a single logout.

- **Access** protected

string function PAPIAuthnEngine::getSingleLogoutResponseLocation() [*line 598*]

Retrieve the URL where to redirect a user once he has successfully logged out.

- **Access** protected

void function PAPIAuthnEngine::isAuthenticated() [*line 315*]

- **Access** public

boolean function PAPIAuthnEngine::isSafe() [*line 792*]

Determines if it's safe to assume the user as authenticated.

- **Access** protected

hash function PAPIAuthnEngine::loadRequest(\$key) [*line 711*]

Function Parameters:

- *key* **\$key** The key that identifies the request.

Load a request from the request database.

- **Access** protected

void function PAPIAuthnEngine::logout([\$slo = false]) [*line 383*]

Function Parameters:

- **\$slo**

- **Access** public

void function PAPIAuthnEngine::redirect([\$location = ""]) [*line 554*]

Function Parameters:

- *location* **\$location** If set, the location where to redirect the user. If not, defaults are used.

Redirect user browser to the appropriate URL for authentication.

WARNING: This method ends execution.

- **Access** protected

string|boolean function PAPIAuthnEngine::saveRequest(\$key, \$hli) [*line 683*]

Function Parameters:

- *key* **\$key** The key identifier for this request.
- *hli* **\$hli** The home locator identifier that should be used for this request.

Save a request to the request database. The request includes: \$_REQUEST, \$_GET, \$_POST, \$_SERVER['QUERY_STRING'], \$_SERVER['REQUEST_METHOD'] and php://input.

- **Access** protected

boolean function PAPIAuthnEngine::testCookie([\$name = ""]) [*line 430*]

Function Parameters:

- *cookie* **\$name** The cookie.

Check if a cookie is valid.

- **Access** protected

boolean function PAPIAuthnEngine::testResponse(\$data, \$pubkey) [*line 495*]

Function Parameters:

- *data* **\$data** The data received.
- *key* **\$pubkey** The public key of the AS/GPoA.

Check the response from the AS/GPoA.

- **Access** protected

Class PAPIConfigurator

[line [31](#)]

Configurator class for the PAPI Authentication Engine.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine

PAPIConfigurator::\$mandatory_options

```
mixed = array('Location',
               'CookieDomain',
               'LKey',
               'PubKeyFile',
               'CookieTimeout',
               'DBType',
               'RedirectURL',
               'RedirectType') [line 33]
```

- **Access** protected

string function PAPIConfigurator::getCookieDomain() [[line 54](#)]

- **Access** public

integer function PAPIConfigurator::getCookieTimeout() [[line 78](#)]

- **Access** public

string function PAPIConfigurator::getDBFile() [[line 94](#)]

- **Access** public

string function PAPIConfigurator::getDBHost() [[line 102](#)]

- **Access** public

string function PAPIConfigurator::getDBName() [[line 126](#)]

- **Access** public

string function PAPIConfigurator::getDBPassword() [[line 118](#)]

- **Access** public

string function PAPIConfigurator::getDBPrefix() [[line 134](#)]

- **Access** public

integer function PAPIConfigurator::getDBType() [*line [86](#)*]

- **Access** public

string function PAPIConfigurator::getDBUser() [*line [110](#)*]

- **Access** public

string function PAPIConfigurator::getFriendlyName() [*line [185](#)*]

- **Access** public

string function PAPIConfigurator::getHomeLocatorID() [*line [169](#)*]

- **Access** public

string function PAPIConfigurator::getID() [*line [177](#)*]

- **Access** public

string function PAPIConfigurator::getLKey() [*line [62](#)*]

- **Access** public

string function PAPIConfigurator::getLocation() [[line 46](#)]

- **Access** public

string function PAPIConfigurator::getLogoutURL() [[line 193](#)]

- **Access** public

string function PAPIConfigurator::getPubKeyFile() [[line 70](#)]

- **Access** public

integer function PAPIConfigurator::getRedirectType() [[line 150](#)]

- **Access** public

string function PAPIConfigurator::getRedirectURL() [[line 142](#)]

- **Access** public

seconds function PAPIConfigurator::getRequestLifetime() [[line 201](#)]

- **Access** public

Class PAPIDB

[[line 31](#)]

Generic interface for a PAPI request database.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine

the function PAPIDB::purge(\$gap) [[line 41](#)]

Function Parameters:

- **gap \$gap** The maximum gap to consider a request outdated, between the time it was stored and the current time.

Purge the database of outdated requests.

- **Access** public

void function PAPIDB::replaceContents(\$key, \$get, \$post, \$request, \$query, \$method, \$input, \$hli) [[line 33](#)]

Function Parameters:

- **\$key**
- **\$get**
- **\$post**
- **\$request**
- **\$query**
- **\$method**
- **\$input**

- **\$hli**

- **Access** public

Class PAPIDBADB [line [31](#)]

PAPI binding to a Berkeley DB backend.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine

PAPIDBADB::\$context

mixed = [line [33](#)]

- **Access** protected

void function PAPIDBADB::fetch(\$key) [line [50](#)]

Function Parameters:

- **\$key**

- **Access** public

```
void function PAPIDBADB::purge($gap) [line 60]
```

Function Parameters:

- **\$gap**
- **Access** public

```
void function PAPIDBADB::replaceContents($key, $get, $post, $request, $query, $method, $input, $hli) [line 35]
```

Function Parameters:

- **\$key**
 - **\$get**
 - **\$post**
 - **\$request**
 - **\$query**
 - **\$method**
 - **\$input**
 - **\$hli**
-
- **Access public**

Class PAPIMySQLDB

[line 33]

MySQL database frontend for the PAPI Authentication Engine.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine

PAPIMySQLDB::\$context

mixed = [line 35]

- **Access** protected

PAPIMySQLDB::\$prefix

mixed = "papi_" [line 36]

- **Access** protected

Constructor *void* function PAPIMySQLDB::__construct(\$cfg) [line 38]

Function Parameters:

- **\$cfg**

- **Access** public

void function PAPIMySQLDB::fetch(\$key) [line 76]

Function Parameters:

- **\$key**

- **Access** public

void function PAPIMySQLDB::replaceContents(\$key, \$get, \$post, \$request, \$query, \$method, \$input, \$hli) [*line 53*]

Function Parameters:

- **\$key**
- **\$get**
- **\$post**
- **\$request**
- **\$query**
- **\$method**
- **\$input**
- **\$hli**

- **Access** public

Class PAPISessionDB

[*line 33*]

PAPI binding to a PHP Session DB backend.

WARNING: please note that PHP only allows one session at a time, so using this backend will break any applications that make use of sessions beneath.

- **Package** phpPoA2
- **Sub-Package** PAPIAuthenticationEngine

PAPISessionDB::\$context

mixed = [line [35](#)]

- **Access** protected

void function PAPISessionDB::fetch(\$key) [line [48](#)]

Function Parameters:

- **\$key**

- **Access** public

void function PAPISessionDB::replaceContents(\$key, \$get, \$post, \$request, \$query, \$method, \$input) [line [37](#)]

Function Parameters:

- **\$key**
- **\$get**
- **\$post**
- **\$request**
- **\$query**
- **\$method**

- **\$input**

- **Access** public

QueryFilterAuthzEngine.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

QUERY_BEFORE_FILTERS = "QUERY_BEFORE_FILTERS" [*line 37*]

This hook is executed right after retrieving the current URI, the params (both GET and POST) and the arrays of allowed and denied patterns that will be checked immediately.

It can be used to alter parameters and the URL, and also to configure the filters on runtime. The hook receives the URI string, an array of parameters, the allowed and the denied patterns. Functions for this hook must be defined like this:

```
function queryBeforeFilterHook(&$uri, &$params, &$allowed, &$denied);
```

Please bear in mind that hooks must return TRUE or they'll keep other hooks from executing.

QueryFilterConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

papi1.5-query-authz.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Sub-Package** QueryFilterAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

include ["./PoA.php"](#) [line 10]

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

void function queryHook(&\$uri, &\$params, &\$allowed, &\$denied) [line 17]

Function Parameters:

- **&\$uri**
- **&\$params**
- **&\$allowed**
- **&\$denied**

Class QueryFilterAuthzEngine

[line 46]

Authorization engine that works by checking the query string of the request.

PLEASE NOTE THAT THIS ENGINE SILENTLY IGNORES BOTH USER AND ATTRIBUTES. PLEASE NOTE THAT THIS ENGINE WORKS ONLY FOR WEB-BASED APPLICATIONS.

- **Package** phpPoA2
- **Sub-Package** QueryFilterAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

QueryFilterAuthzEngine::\$valid_hooks

mixed = array(QUERY_BEFORE_FILTERS) [line 48]

- **Access** protected

void function QueryFilterAuthzEngine::authorize(\$user, \$attrs, \$ref, [\$expires = 0]) [line 97]

Function Parameters:

- **\$user**

- **\$attrs**
- **\$ref**
- **\$expires**

- **Access** public

void function QueryFilterAuthzEngine::getAuthorizedList() [*line* [90](#)]

- **Access** public

void function QueryFilterAuthzEngine::isAuthorized(\$user, \$attrs) [*line* [53](#)]

Function Parameters:

- **\$user**
- **\$attrs**

PLEASE NOTE THAT THIS ENGINE SILENTLY IGNORES BOTH USER AND ATTRIBUTES.

- **Access** public

void function QueryFilterAuthzEngine::revoke(\$mail) [*line* [101](#)]

Function Parameters:

- **\$mail**

- **Access** public

Class QueryFilterConfigurator

[line 31]

Configurator for the query filter authorization engine.

- **Package** phpPoA2
- **Sub-Package** QueryFilterAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es >
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

QueryFilterConfigurator::\$mandatory_options

mixed = array("Allowed", "Denied") [line 33]

- **Access** protected

array function QueryFilterConfigurator::getAllowedPatterns() [*line* [39](#)]

Returns the patterns that match allowed URIs or query parameters.

- **Access** public

array function QueryFilterConfigurator::getDeniedPatterns() [*line* [47](#)]

Returns the patterns that match denied URIs or query parameters.

- **Access** public

SimpleSAMLphpAuthnEngine.php

- **Package** phpPoA2

NS_SAML2_PROTOCOL = 'urn:oasis:names:tc:SAML:2.0:protocol' *[line 30]*

Supported namespaces for attributes.

require_once "simplesamlphp/lib/_autoload.php" *[line 26]*

- **Author** Elena Lozano < elena.lozano@rediris.es >
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **License** [GNU General Public License](#)

Class SimpleSAMLphpAuthnEngine

[line 38]

Authentication engine for the PAPI 1.5 protocol.

PLEASE NOTE THAT THIS ENGINE WORKS ONLY FOR WEB-BASED

APPLICATIONS.

- **Package** phpPoA2
- **Sub-Package** SimpleSAMLphpAuthnEngine

SimpleSAMLphpAuthnEngine::\$attributes

mixed = [line 42]

- **Access** protected

SimpleSAMLphpAuthnEngine::\$simplesaml

mixed = [line 40]

- **Access** protected

SimpleSAMLphpAuthnEngine::\$status

mixed = [line 41]

- **Access** protected

void function SimpleSAMLphpAuthnEngine::authenticate() [line 55]

- **Access** public

void function SimpleSAMLphpAuthnEngine::configure(\$file, \$site) [line 44]

Function Parameters:

- **\$file**
- **\$site**

- **Access** public

void function SimpleSAMLphpAuthnEngine::getAttribute(\$name, [\$namespace = NS_SAML2_PROTOCOL]) [line 75]

Function Parameters:

- **\$name**
- **\$namespace**

- **Access** public

void function SimpleSAMLphpAuthnEngine::getAttributes() [line 71]

- **Access** public

void function SimpleSAMLphpAuthnEngine::isAuthenticated() [line 67]

- **Access** public

void function SimpleSAMLphpAuthnEngine::logout([\$slo = false]) [line 83]

Function Parameters:

- **\$slo**
- **Access** public

SourceIPAddrAuthzEngine.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

SOURCEADDR_BEFORE_FILTERS = "SOURCEADDR_BEFORE_FILTERS" [*line 37*]

This hook is executed right after retrieving source address and the arrays of allowed and denied patterns that will be checked immediately.

It can be used to alter the source address, and also to configure the filters on runtime. The hook receives the source IP address, the allowed and the denied patterns. Functions for this hook must be defined like this:

```
function ipBeforeFilterHook(&$ipaddr, &$allowed, &$denied);
```

Please bear in mind that hooks must return TRUE or they'll keep other hooks from executing.

SourceIPAddrConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

papi1.5-ip-authz.php

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Package** phpPoA2
- **Sub-Package** SourceIPAddrAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

include ["./PoA.php"](#) *[line 10]*

This is a sample file to demonstrate the functionality of the phpPoA2 package.

- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Filesource** [Source Code for this file](#)

void function ipHook(&\$ip, &\$allowed, &\$denied) *[line 14]*

Function Parameters:

- **&\$ip**
- **&\$allowed**
- **&\$denied**

Class SourceIPAddrAuthzEngine

[line 46]

Authorization engine that works by checking the source IP address of the request.

PLEASE NOTE THAT THIS ENGINE SILENTLY IGNORES BOTH USER AND ATTRIBUTES. PLEASE NOTE THAT THIS ENGINE WORKS ONLY FOR WEB-BASED APPLICATIONS.

- **Package** phpPoA2
- **Sub-Package** SourceIPAddrAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es>
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

SourceIPAddrAuthzEngine::\$valid_hooks

mixed = array(SOURCEADDR_BEFORE_FILTERS) [*line* [48](#)]

- **Access** protected

void function SourceIPAddrAuthzEngine::authorize(\$user, \$attrs, \$ref, [\$expires = 0]) [*line* [103](#)]

Function Parameters:

- **\$user**
- **\$attrs**

- **\$ref**
- **\$expires**

- **Access** public

void function SourceIPAddrAuthzEngine::getAuthorizedList() [*line* [96](#)]

- **Access** public

void function SourceIPAddrAuthzEngine::isAuthorized(\$user, \$attrs) [*line* [53](#)]

Function Parameters:

- **\$user**
- **\$attrs**

PLEASE NOTE THAT THIS ENGINE SILENTLY IGNORES BOTH USER AND ATTRIBUTES.

- **Access** public

void function SourceIPAddrAuthzEngine::revoke(\$mail) [*line* [107](#)]

Function Parameters:

- **\$mail**

- **Access** public

Class SourceIPAddrConfigurator

[line [31](#)]

Configurator class for the Source IP Address Authorization Engine.

- **Package** phpPoA2
- **Sub-Package** SourceIPAddrAuthorizationEngine
- **Author** Jaime Perez < jaime.perez@rediris.es >
- **Version** 2.0
- **Copyright** Copyright 2005-2010 RedIRIS, <http://www.rediris.es/>

This file is part of phpPoA2.

phpPoA2 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 3 of the License, or (at your option) any later version.

phpPoA2 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 phpPoA2. If not, see <<http://www.gnu.org/licenses/>>.

SourceIPAddrConfigurator::\$mandatory_options

mixed = array("AllowFrom", "DenyFrom") [line [33](#)]

- **Access** protected

array function SourceIPAddrConfigurator::getAllowed() [[line 39](#)]

Returns an array of IP addresses or networks that are authorized.

- **Access** public

array function SourceIPAddrConfigurator::getDenied() [[line 50](#)]

Returns an array of IP addresses or networks that are not authorized.

- **Access** public

Appendix A - Class Trees

Package phpPoA2

AttributeHandler

- [AttributeHandler](#)

AuthorizedDB

- [AuthorizedDB](#)

Example

- [Example](#)

GenericConfigurator

- [GenericConfigurator](#)
 - [AuthorizationConfigurator](#)
 - [AttributeFilterConfigurator](#)
 - [InviteConfigurator](#)
 - [QueryFilterConfigurator](#)
 - [SourceIPAddrConfigurator](#)
 - [PAPConfigurator](#)
 - [PoAConfigurator](#)

GenericDB

- [GenericDB](#)
 - [DBADB](#)
 - [AuthorizedDBADB](#)
 - [InviteDBADB](#)
 - [PAPIDBADB](#)
 - [GenericMySQLDB](#)
 - [PAPIMySQLDB](#)
 - [SessionDB](#)
 - [PAPISessionDB](#)

GenericEngine

- [GenericEngine](#)
 - [AuthenticationEngine](#)
 - [PAPIAuthnEngine](#)
 - [SimpleSAMLphpAuthnEngine](#)

- [AuthorizationEngine](#)
 - [AttributeFilterAuthzEngine](#)
 - [InviteAuthzEngine](#)
 - [QueryFilterAuthzEngine](#)
 - [SourceIPAddrAuthzEngine](#)

Hook

- [Hook](#)

InviteDB

- [InviteDB](#)

PAPIDB

- [PAPIDB](#)

PoA

- [PoA](#)
 - [AutoPoA](#)
 - [LitePoA](#)

PoAEventHandler

- [PoAEventHandler](#)

PoAException

- Exception
 - [PoAException](#)

PoALog

- [PoALog](#)

PoAUtills

- [PoAUtills](#)

INSTALL

REQUISITES:

- * PHP >= 5.2.0
- * PHP-MCRYPT
- * PHP-DBA (if using DBA backends)
- * PHP-MYSQL (if using MySQL backends)
- * SimpleSAMLphp (if using SAML for authentication)

INSTALLATION:

1. Upgrade/install your PHP to meet the requirements.
2. Copy the package with all files right into your server.
3. Ensure the file PoA.php is in your include path.
4. Put the PoA.conf file in your include path, or set the PHP environment variable "poa_conf_file" to point to your PoA.conf file.
5. Configure the PoA.conf file for your PoAs, and any other configuration files you may need. Refer to the samples for more info
on how to configure each engine.

6. If you are using PHP 5.2.X, then ensure the environment option "zend.ze1_compatibility_mode" is off.

README

phpPoA 2.0

Contents of the distribution:

README	This file
GPL	Text of the GNU General Public License, applicable to all the software
LICENSE	Copyright information and short text of the license, including responsibility disclaimer
doc	Documentation for this distribution
lib	Auxiliary classes that implement most of the functionality that is shipped with this distribution
messages	Support files for message internationalization
samples	Sample usage files, including configuration
PoA.php	The main class of phpPoA2. Include only this file in your applications
PoAConfigurator.php	The main configuration class
PoAException.php	Generic phpPoA exception implementation
PoALog.php	Logging class
definitions.php	Useful constants
utils.php	Utility functions

Index

A

AttributeFilterConfigurator::getAllowedAttributes()	0
<i>Returns an array of attributes that would be allowed if their values match any of the patterns for each attribute.</i>	
AttributeFilterConfigurator::\$mandatory_options	0
AttributeFilterConfigurator::getDeniedAttributes()	0
<i>Returns an array of attributes that would be denied if their values match any of the patterns for each attribute.</i>	
AuthorizedDB.php	0
AUTHORIZED_DBA	0
AuthorizedDBADB.php	0
AttributeFilterConfigurator	0
<i>Configurator for the Attribute Filter Authorization Engine.</i>	
AttributeFilterAuthzEngine::revoke()	0
AttributeFilterAuthzEngine::\$valid_hooks	0
AttributeFilterAuthzEngine	0
<i>Authorization engine that works by checking the attributes of the user. The first match of an attribute against one of the filters will trigger the authorization result, no matter if it's positive or negative.</i>	
AttributeFilterAuthzEngine::authorize()	0
AttributeFilterAuthzEngine::getAuthorizedList()	0
AttributeFilterAuthzEngine::isAuthorized()	0
AUTHORIZED_MYSQL	0
AUTHORIZED_SESSION	0
AuthorizedDBADB::replace_authorization()	0
AuthorizedDBADB::open()	0
ATTR_SEPARATOR	0
<i>Default assertion delimiters for standard PAPI 1.5 protocol.</i>	
AttributeHandler	0
AttributeHandler::handle()	0
AuthorizedDBADB::fetch_all()	0
AuthorizedDBADB::fetch()	0
AuthorizedDB::replace_authorization()	0
AuthorizedDB	0
<i>Generic interface for a PAPI request database.</i>	
AuthorizedDBADB	0
<i>Authorization lists binding to a Berkeley DB backend.</i>	
AuthorizedDBADB::\$authorized	0
AuthorizedDBADB::\$mandatory_options	0
attrHook()	0
AttributeFilterConfigurator.php	0
AuthenticationEngine::\$engine_type	0
AuthenticationEngine	0
<i>Abstract authentication engine.</i>	
AuthenticationEngine::authenticate()	0

<i>Trigger the authentication of the user in the current context. May perform HTTP redirections or any other procedure to gather the authentication status if the user, so don't expect it to return control once called.</i>	
AuthenticationEngine::getAttribute()	0
<i>Get an specific attribute for the current user, by specifying its name and namespace. May trigger some attribute query procedure.</i>	
AuthenticationEngine::isAuthenticated()	0
<i>Check the current authentication status withing this context. Does not trigger any authentication procedure.</i>	
AuthenticationEngine::getAttributes()	0
<i>Get an associative array with the set of common attributes for the current user. May trigger some attribute recollection.</i>	
authz_convert.php	0
AuthenticationEngine.php	0
AUTHN_FAILED	0
<i>Authentication failed.</i>	
AS_T	0
<i>'AS' type redirection in the PAPI 1.5 protocol.</i>	
AUTHN_SUCCESS	0
<i>Authentication succeeded.</i>	
AUTHZ_FAILED	0
<i>Authorization failed.</i>	
AUTHZ_SUCCESS	0
<i>Authorization succeeded.</i>	
AuthenticationEngine::logout()	0
<i>Perform a logout (locally or global).</i>	
AuthorizationConfigurator	0
AutoPoA::authenticate()	0
AutoPoA	0
<i>Class that automatically redirects to error pages if any error is detected.</i>	
AutoPoA::isAuthorized()	0
AttributeFilterAuthzEngine.php	0
ATTRIBUTE_BEFORE_FILTERS	0
<i>This hook is executed right after retrieving the arrays of allowed and denied attributes that will be checked immediately.</i>	
AuthorizationEngine::revoke()	0
AuthorizationEngine::isAuthorized()	0
<i>Check authorization for the specified user.</i>	
AuthorizationEngine	0
AuthorizationConfigurator::getDefaultBehaviour()	0
<i>Determines the default behaviour of the authorization engine, whether it should authorize or not if no pattern (nor allowed nor denied) matches.</i>	
AuthorizationEngine::\$engine_type	0
AuthorizationEngine::authorize()	0
AuthorizationEngine::getAuthorizedList()	0
AutoPoA.php	0

C

constructor PoAException::__construct()	0
constructor PoAEventHandler::__construct()	0
<i>Main constructor.</i>	

constructor PoALog::__construct()	0
<i>Build a new logger.</i>	
constructor GenericDB::__construct()	0
<i>Main constructor.</i>	
constructor PAPIMySQLDB::__construct()	0
constructor PoAConfigurator::__construct()	0
constructor PoA::__construct()	0
<i>Main constructor. Configures the PoA and performs initialization.</i>	
crypt.php	0
constructor GenericConfigurator::__construct()	0
<i>Main constructor.</i>	
constructor GenericEngine::__construct()	0
<i>Main constructor for the engine.</i>	
constructor Hook::__construct()	0
<i>Build a new hook.</i>	
CONFIG_ERR	0
<i>A configuration related error.</i>	

D

DBADB::delete()	0
DBADB::configure()	0
DBADB::fetch()	0
DBADB::fetch_all()	0
DBADB::replace()	0
DBADB::open()	0
DBADB::close()	0
DBADB::check()	0
decrypt_AES()	0
decrypt_3DES()	0
DBADB.php	0
DBADB	0
<i>Berkeley DB backend.</i>	
DBADB::\$mandatory_options	0
definitions.php	0

E

Example::hook()	0
Example	0
encrypt_AES()	0
encrypt_3DES()	0

G

GenericMySQLDB::\$create_sql	0
GenericMySQLDB	0
<i>Generic MySQL database class.</i>	
GenericMySQLDB::\$delete_sql	0
GenericMySQLDB::\$insert_sql	0

GenericMySQLDB::\$mandatory_options	0
GenericDB::replace()	0
<i>Replaces the specified key with a new value. If the key does not exist previously, it will be created.</i>	
GenericDB::open()	0
<i>Opens the database.</i>	
GenericDB::configure()	0
<i>Configure the database handler.</i>	
GenericDB::close()	0
<i>Closes the database.</i>	
GenericDB::delete()	0
<i>Removes the specified key from the database.</i>	
GenericDB::fetch()	0
<i>Gets the value of the specified key.</i>	
GenericDB::fetch_all()	0
<i>Gets all the contents stored in the database.</i>	
GenericMySQLDB::\$prefix	0
GenericMySQLDB::\$search_all_sql	0
GenericMySQLDB::getError()	0
GenericMySQLDB::fetch_all()	0
GenericMySQLDB::open()	0
GenericMySQLDB::replace()	0
global \$messages	0
<i>An array with all the internationalized messages used by phpPoA2.</i>	
GenericMySQLDB::fetch()	0
GenericMySQLDB::delete()	0
GenericMySQLDB::\$update_sql	0
GenericMySQLDB::\$search_sql	0
GenericMySQLDB::check()	0
GenericMySQLDB::close()	0
GenericMySQLDB::configure()	0
GenericDB::check()	0
<i>Checks if the specified key exists in the database.</i>	
GenericDB::\$mandatory_options	0
GenericEngine::\$cfg	0
GenericEngine	0
<i>A generic engine class.</i>	
GenericEngine::\$engine_type	0
GenericEngine::\$handler	0
GenericEngine::\$hooks	0
GenericConfigurator::validate()	0
<i>Check all mandatory attributes are set.</i>	
GenericConfigurator::configure()	0
<i>Read the configuration from the specified file and section.</i>	
GenericEngine.php	0
GenericConfigurator.php	0
GenericConfigurator	0
<i>Generic configuration class.</i>	
GenericConfigurator::\$cfg	0
GenericConfigurator::\$mandatory_options	0
GenericEngine::\$valid_hooks	0
GenericEngine::addHook()	0
<i>Adds a function to the specified hook, which will be executed at some point of the code.</i>	

GenericDB	0
<i>Provides the basic operations and defines the interface of any DB operating class.</i>	
GenericMySQLDB.php	0
GenericDB::\$cfg	0
GenericDB::\$db	0
GenericDB::\$error	0
GenericDB.php	0
GenericEngine::setHandler()	0
<i>Set the event handler to the one specified.</i>	
GenericEngine::configure()	0
<i>Configure the engine.</i>	
GenericEngine::clean()	0
<i>Unregister error and exception handlers. Use it only for methods not declared in the interface that previously called registerHandler() method.</i>	
GenericEngine::registerHandler()	0
<i>Register error and exception handlers for logging. Use it only for methods not declared in the interface that could trigger errors.</i>	
GenericEngine::removeHook()	0
<i>Removes a function from the specified hook.</i>	
GenericEngine::runHooks()	0
<i>Run all hooks attached to an specific action.</i>	
GPOA_T	0
<i>'GPoA' type redirection in the PAPI 1.5 protocol.</i>	

H

Hook::run()	0
<i>Run the hook with the specified params.</i>	
handler()	0
hook()	0
Hook::getName()	0
<i>Get the name of the hook.</i>	
Hook::\$ name	0
Hook	0
<i>Hook class that allows to run a function or a method specified by their name.</i>	
Hook::\$ callable_name	0
Hook::\$ hook	0
Hook.php	0

I

InviteConfigurator::getInvitesDBHost()	0
<i>Returns</i>	
InviteConfigurator::getInvitesDBFile()	0
<i>Returns</i>	
InviteConfigurator::getInvitesDBName()	0
<i>Returns</i>	
InviteConfigurator::getInvitesDBPassword()	0
<i>Returns</i>	
InviteConfigurator::getInvitesDBUser()	0
<i>Returns</i>	

InviteConfigurator::getInvitesDBType()	0
Returns	
InviteConfigurator::getEmailVerifyRegEx()	0
InviteConfigurator::getEmailVerifyAttribute()	0
InviteConfigurator::getAuthorizedDBPassword()	0
Returns	
InviteConfigurator::getAuthorizedDBName()	0
Returns	
InviteConfigurator::getAuthorizedDBType()	0
Returns	
InviteConfigurator::getAuthorizedDBUser()	0
Returns	
InviteConfigurator::getEmailVerifyAlgRegEx()	0
InviteConfigurator::getInviteSubject()	0
Returns	
InviteConfigurator::getInviteText()	0
Returns	
InviteDBADB::fetch_all()	0
InviteDBADB::fetch()	0
InviteDBADB::open()	0
InviteDBADB::replace_invite()	0
INSTALL	0
ipHook()	0
InviteDBADB::\$mandatory_options	0
InviteDBADB::\$invite	0
InviteConfigurator::getUniqueAttributes()	0
Returns	
InviteConfigurator::getInviteURL()	0
Returns	
InviteDB	0
Generic interface for a database of invites.	
InviteDB::replace_invite()	0
InviteDBADB	0
Invites binding to a Berkeley DB backend.	
InviteConfigurator::getAuthorizedDBHost()	0
Returns	
InviteConfigurator::getAuthorizedDBFile()	0
Returns	
InviteDBADB.php	0
InviteDB.php	0
InviteAuthzEngine	0
Invitation based authorization engine.	
InviteAuthzEngine::\$authz_db	0
InviteAuthzEngine::\$valid_hooks	0
InviteAuthzEngine::\$invites_db	0
InviteConfigurator.php	0
INVITES_SESSION	0
InviteAuthzEngine.php	0
invite_convert.php	0
INVITES_BEFORE_AUTHORIZATION	0
This hook is executed right before it is checked whether the user was invited or not to continue with authorization.	
INVITES_DBA	0

Supported database types.	0
INVITES_MYSQL	0
InviteAuthzEngine::authorize()	0
<i>Authorize the specified user.</i>	
InviteAuthzEngine::configure()	0
<i>Configure the authorization engine.</i>	
InviteConfigurator	0
<i>Configurator for the Invitation based authorization engine.</i>	
InviteAuthzEngine::revoke()	0
<i>Revoke authorization for the specified user.</i>	
InviteConfigurator::\$mandatory_options	0
InviteConfigurator::doEmailVerify()	0
InviteConfigurator::getAdminEmail()	0
<i>Returns</i>	
InviteAuthzEngine::removeInvite()	0
<i>Remove an invitation from the database.</i>	
InviteAuthzEngine::isAuthorized()	0
<i>Check if the specified user is authorized with the given attributes.</i>	
InviteAuthzEngine::finish()	0
<i>Close databases and return the same value received as a parameter.</i>	
InviteAuthzEngine::emailVerify()	0
<i>Perform e-mail verification for the current user according to the configuration for this site.</i>	
InviteAuthzEngine::getAuthorizedList()	0
<i>Get a list of all authorized users.</i>	
InviteAuthzEngine::getPendingInvites()	0
<i>Get a list of all pending invitations.</i>	
InviteAuthzEngine::invite()	0
<i>Send an invitation to an e-mail address (that is, send an e-mail to that address with instructions on how to get authorized and an URL to follow).</i>	
INVITE_ERR	0
<i>An invitation related error.</i>	

L

LitePoA::setIDBuilder()	0
<i>Set an attribute handler to build identifiers for the users according to their attributes. Method must receive a hash of attributes ("name" => "value").</i>	
LitePoA::checkAccess()	0
<i>A shortcut for the whole authentication and authorization process.</i>	
LitePoA::\$handler	0
LitePoA	0
<i>Class that embeds all functionality (authentication, attribute collection and authorization) in one method, just like older versions of phpPoA.</i>	
LitePoA.php	0

M

messages-es_ES.php	0
------------------------------------	---

[messages-en_EN.php](#) 0

N

[NS_SAML2_PROTOCOL](#) 0

Supported namespaces for attributes.

[NS_PAPI_PROTOCOL](#) 0

Supported namespaces for attributes.

[NS_PAPI_ATTRIBUTES](#) 0

[NAMEVALUE_SEPARATOR](#) 0

[NOAUTH_ERR](#) 0

Authentication failed error.

O

[openssl_encrypt\(\)](#) 0

[openssl_decrypt\(\)](#) 0

P

[PAPIAuthnEngine::deleteRequest\(\)](#) 0

Delete a request from the request database.

[PAPIAuthnEngine::getAttribute\(\)](#) 0

[PAPIAuthnEngine::getAttributes\(\)](#) 0

[PAPIAuthnEngine::getNewCookie\(\)](#) 0

Generate a new cookie for the current user.

[PAPIAuthnEngine::deleteCookie\(\)](#) 0

Delete the current cookie, if any.

[PAPIAuthnEngine::configure\(\)](#) 0

[PAPIAuthnEngine::\\$valid_hooks](#) 0

[PAPIAuthnEngine::attributeQuery\(\)](#) 0

[PAPIAuthnEngine::authenticate\(\)](#) 0

[PAPIAuthnEngine::getRedirectLocation\(\)](#) 0

Retrieve the URL where to redirect a user and store his request.

[PAPIAuthnEngine::getSingleLogoutLocation\(\)](#) 0

Retrieve the URL where to redirect a user to perform a single logout.

[PAPIAuthnEngine::redirect\(\)](#) 0

Redirect user browser to the appropriate URL for authentication.

[PAPIAuthnEngine::saveRequest\(\)](#) 0

Save a request to the request database. The request includes: \$_REQUEST, \$_GET, \$_POST, \$_SERVER['QUERY_STRING'], \$_SERVER['REQUEST_METHOD'] and php://input.

[PAPIAuthnEngine::testCookie\(\)](#) 0

Check if a cookie is valid.

[PAPIAuthnEngine::logout\(\)](#) 0

[PAPIAuthnEngine::loadRequest\(\)](#) 0

Load a request from the request database.

[PAPIAuthnEngine::getSingleLogoutResponseLocation\(\)](#) 0

Retrieve the URL where to redirect a user once he has successfully logged out.

[PAPIAuthnEngine::isAuthenticated\(\)](#) 0

PAPIAuthnEngine::isSafe()	0
<i>Determines if it's safe to assume the user as authenticated.</i>	
PAPIAuthnEngine::\$status	0
PAPIAuthnEngine::\$skip_redirection	0
PAPIAuthnEngine::\$as_id	0
PAPIAuthnEngine::\$attributes	0
PAPIAuthnEngine::\$cfg	0
PAPIAuthnEngine::\$clean_url	0
PAPIAuthnEngine::\$assertion	0
PAPIAuthnEngine	0
<i>Authentication engine for the PAPI 1.5 protocol.</i>	
papi1.5-authn-attr.php	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
papi1.5-customattr.php	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
papi1.5-papiopoa-hook.php	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
PAPIAuthnEngine::\$cookie_name	0
PAPIAuthnEngine::\$db	0
PAPIAuthnEngine::\$lkey	0
PAPIAuthnEngine::\$opoa	0
PAPIAuthnEngine::\$pkey	0
PAPIAuthnEngine::\$key	0
PAPIAuthnEngine::\$id	0
PAPIAuthnEngine::\$enforcing	0
PAPIAuthnEngine::\$expiration_time	0
PAPIAuthnEngine::\$global_expire_time	0
PAPIAuthnEngine::testResponse()	0
<i>Check the response from the AS/GPoA.</i>	
PAPICConfigurator	0
<i>Configurator class for the PAPI Authentication Engine.</i>	
PAPIDBADB::fetch()	0
PAPIDBADB::purge()	0
PAPIDBADB::replaceContents()	0
PAPIMySQLDB	0
<i>MySQL database frontend for the PAPI Authentication Engine.</i>	
PAPIDBADB::\$context	0
PAPIDBADB	0
<i>PAPI binding to a Berkeley DB backend.</i>	
PAPIDB	0
<i>Generic interface for a PAPI request database.</i>	
PAPIDB::purge()	0
<i>Purge the database of outdated requests.</i>	
PAPIDB::replaceContents()	0
PAPIMySQLDB::\$context	0
PAPIMySQLDB::\$prefix	0
PAPISessionDB::replaceContents()	0
papi1.5-query-authz.php	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
papi1.5-ip-authz.php	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
PAPISessionDB::fetch()	0
PAPISessionDB::\$context	0

PAPIMySQLDB::fetch()	0
PAPIMySQLDB::replaceContents()	0
PAPISessionDB	0
<i>PAPI binding to a PHP Session DB backend.</i>	
PAPIConfigurator::getRequestLifetime()	0
PAPIConfigurator::getRedirectURL()	0
PAPIConfigurator::getDBName()	0
PAPIConfigurator::getDBPassword()	0
PAPIConfigurator::getDBPrefix()	0
PAPIConfigurator::getDBHost()	0
PAPIConfigurator::getDBFile()	0
PAPIConfigurator::\$mandatory_options	0
PAPIConfigurator::getCookieDomain()	0
PAPIConfigurator::getCookieTimeout()	0
PAPIConfigurator::getDBType()	0
PAPIConfigurator::getDBUser()	0
PAPIConfigurator::getLogoutURL()	0
PAPIConfigurator::getPubKeyFile()	0
PAPIConfigurator::getRedirectType()	0
PAPIConfigurator::getLocation()	0
PAPIConfigurator::getLKey()	0
PAPIConfigurator::getFriendlyName()	0
PAPIConfigurator::getHomeLocatorID()	0
PAPIConfigurator::getID()	0
PAPISessionDB.php	0
PAPI_DB_TABLE	0
PoA::getAuthorizationEngines()	0
<i>Returns the authorization engines configured for the current PoA, or the one specified.</i>	
PoA::getAuthorizationLevels()	0
<i>Get the authorization levels that match for the user specified, according to the configuration.</i>	
PoA::isAuthenticated()	0
<i>Query the current status of the user in the federation.</i>	
PoA::isAuthorized()	0
<i>Perform authorization for the a given subject.</i>	
PoA::getAttributes()	0
<i>Retrieve the attributes provided by the user when logged in.</i>	
PoA::getAttribute()	0
<i>Get the value (or values) of an attribute, if present.</i>	
PoA::authenticate()	0
<i>Perform a federated login for the user.</i>	
PoA::authorize()	0
<i>Authorize a given subject with the data retrieved from federated login.</i>	
PoA::clean()	0
PoA::logout()	0
<i>Remove the user's session and trigger a logout for the specified authentication protocol.</i>	
PoA::removeHook()	0
<i>Remove a hook from the specified entry point of the available authentication or authorization engines.</i>	
PoAConfigurator::getAuthzEngineConfFile()	0
PoAConfigurator::getAuthzEngines()	0
PoAConfigurator::getAuthzLevels()	0

PoAConfigurator::getAuthnEngineConfFile()	0
PoAConfigurator::getAuthnEngine()	0
PoA::revoke()	0
<i>Revoke authorization for a given subject identified by an e-mail.</i>	
PoAConfigurator	0
<i>Main configurator for the phpPoA.</i>	
PoAConfigurator::\$mandatory_options	0
PoA::addHook()	0
<i>Attach a hook object to the appropriate entry point of the available authentication or authorization engines.</i>	
PoA::\$log	0
PoAException.php	0
PoALog.php	0
PoAUtils.php	0
papi1.5-slo-extra.php	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
PoAEventHandler.php	0
PoAConfigurator.php	0
papi_openssl_decrypt()	0
papi_openssl_encrypt()	0
PoA.php	0
papi1.5-slo.php	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
PoA	0
<i>Standard class that implements all the functionality of the phpPoA.</i>	
PoA::\$db_manager	0
PoA::\$handler	0
PoA::\$local_site	0
PoA::\$cfg	0
PoA::\$autoload	0
PoA::\$attributes	0
PoA::\$authn_engine	0
PoA::\$authz_engines	0
PoAConfigurator::getInviteErrorURL()	0
PoAConfigurator::getLanguage()	0
PAPI DBA	0
<i>Supported database types.</i>	
PAPI MYSQL	0
PAPI REDIRECT_URL_FINISH	0
<i>This hook is executed at the end of the method that returns the URL where to redirect a user.</i>	
PAPI RESTORE_ORIGINAL_REQUEST	0
<i>This hook is executed when a valid response is found from the AS/GPoA and the original request of the user is about to be restored. It receives an array with the main PHP global variables of the original context. Functions for this hook must be defined like this:</i>	
PAPI CLEAN_URL	0
<i>This hook is executed when a valid response is found from the AS/GPoA and the engine is about</i>	
PAPI ATTRIBUTE_PARSER	0
<i>This hook is executed when returning the attributes found for a user with getAttributes() method. It receives a string with the attributes and the array that results of processing the string. Functions for this hook must be defined like this:</i>	
papi1.5-attr-authz.php	0

papi1.5-invite-authz.php	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
PAPIAuthnEngine.php	0
PAPI_SESSION	0
PHP_VERSION_REQUIRED	0
<i>Minimum PHP version required.</i>	
PAPIDB.php	0
PAPIDBADB.php	0
PAPIMySQLDB.php	0
PAPIConfigurator.php	0
PROTO_ATTR_PREFIX	0
<i>Prefix for operational attributes inside the protocol and special attributes names.</i>	
PROTO_ATTR_AS_ID	0
PROTO_ATTR_EXPIRE_TIME	0
PROTO_ATTR_KEY	0
PoAUtils::msg()	0
<i>Return the internationalized message identified by the code 's'.</i>	
PoAUtils::lang_code()	0
<i>Return the language code identifying applicable to a messages file.</i>	
PoAEventHandler	0
<i>Class to handle events inside the library.</i>	
PoAEventHandler::abort()	0
<i>Finish execution and print an error message with the current backtrace.</i>	
PoAEventHandler::autoloadHandler()	0
<i>Override __autoload standard function to allow loading classes dinamically.</i>	
PoAConfigurator::isDebug()	0
PoAConfigurator::getSystemErrorURL()	0
PoAConfigurator::getLogFile()	0
PoAConfigurator::getLogLevel()	0
PoAConfigurator::getNoAuthErrorURL()	0
PoAEventHandler::errorHandler()	0
<i>A generic error handler that logs the error message.</i>	
PoAEventHandler::exceptionHandler()	0
<i>A generic exception handler that logs the exception message.</i>	
PoALog::getLogLevel()	0
<i>Get the current log level.</i>	
PoALog::write()	0
<i>Write a message to the log file if the current log level allows it.</i>	
PoAUtils	0
PoALog	0
<i>A simple class to write log messages to a file.</i>	
PoAException::toString()	0
PoAEventHandler::setDebug()	0
<i>Set debugging mode.</i>	
PoAEventHandler::setLogger()	0
<i>Set the logger.</i>	
PoAException	0
<i>Exception handler class for the phpPoA.</i>	
PADDINGSIZE	0

Q

QueryFilterAuthzEngine::revoke()	0
QueryFilterAuthzEngine::isAuthorized()	0
<i>PLEASE NOTE THAT THIS ENGINE SILENTLY IGNORES BOTH USER AND ATTRIBUTES.</i>	
QueryFilterConfigurator	0
<i>Configurator for the query filter authorization engine.</i>	
QueryFilterConfigurator::\$mandatory_options	0
QueryFilterConfigurator::getDeniedPatterns()	0
<i>Returns the patterns that match denied URIs or query parameters.</i>	
QueryFilterConfigurator::getAllowedPatterns()	0
<i>Returns the patterns that match allowed URIs or query parameters.</i>	
QueryFilterAuthzEngine::getAuthorizedList()	0
QueryFilterAuthzEngine::authorize()	0
QueryFilterConfigurator.php	0
QUERY_BEFORE_FILTERS	0
<i>This hook is executed right after retrieving the current URI, the params (both GET and POST) and the arrays of allowed and denied patterns that will be checked immediately.</i>	
queryHook()	0
QueryFilterAuthzEngine	0
<i>Authorization engine that works by checking the query string of the request.</i>	
QueryFilterAuthzEngine::\$valid_hooks	0
QueryFilterAuthzEngine.php	0

R

README	0
REQUEST_LIFETIME	0
<i>Default timeout for stored requests.</i>	
ROOT	0
<i>Authorized DB converter tool from phpPoA1 to phpPoA2.</i>	

S

SOURCEADDR_BEFORE_FILTERS	0
<i>This hook is executed right after retrieving source address and the arrays of allowed and denied patterns that will be checked immediately.</i>	
SourceIPAddrConfigurator.php	0
SourceIPAddrAuthzEngine	0
<i>Authorization engine that works by checking the source IP address of the request.</i>	
SourceIPAddrAuthzEngine.php	0
SimpleSAMLphpAuthnEngine::logout()	0
SimpleSAMLphpAuthnEngine::getAttributes()	0
SimpleSAMLphpAuthnEngine::isAuthenticated()	0
SourceIPAddrAuthzEngine::\$valid_hooks	0
SourceIPAddrAuthzEngine::authorize()	0
SourceIPAddrConfigurator::\$mandatory_options	0
SourceIPAddrConfigurator::getAllowed()	0
<i>Returns an array of IP addresses or networks that are authorized.</i>	
SourceIPAddrConfigurator::getDenied()	0

<i>Returns an array of IP addresses or networks that are not authorized.</i>	
SourceIPAddrConfigurator	0
<i>Configurator class for the Source IP Address Authorization Engine.</i>	
SourceIPAddrAuthzEngine::revoke()	0
SourceIPAddrAuthzEngine::getAuthorizedList()	0
SourceIPAddrAuthzEngine::isAuthorized()	0
<i>PLEASE NOTE THAT THIS ENGINE SILENTLY IGNORES BOTH USER AND ATTRIBUTES.</i>	
SimpleSAMLphpAuthnEngine::getAttribute()	0
SimpleSAMLphpAuthnEngine::configure()	0
SessionDB::close()	0
SessionDB::delete()	0
SessionDB::fetch()	0
SessionDB::check()	0
SessionDB::\$session_name	0
SessionDB.php	0
SessionDB	0
<i>Session database backend.</i>	
SessionDB::fetch_all()	0
SessionDB::open()	0
SimpleSAMLphpAuthnEngine::\$simplesaml	0
SimpleSAMLphpAuthnEngine::\$status	0
SimpleSAMLphpAuthnEngine::authenticate()	0
SimpleSAMLphpAuthnEngine::\$attributes	0
SimpleSAMLphpAuthnEngine	0
<i>Authentication engine for the PAPI 1.5 protocol.</i>	
SessionDB::replace()	0
SimpleSAMLphpAuthnEngine.php	0
SYSTEM_ERR	0
<i>A system related error.</i>	

U

USER_ERR	0
<i>An error triggered by the user.</i>	

V

VALUE_SEPARATOR	0
---------------------------------	---