

# phpPoA2



# Contents

<a href="#">AutoPoA.php</a>	0
<a href="#">definitions.php</a>	0
<a href="#">Define AS_T</a>	0
<a href="#">Define AUTHN_FAILED</a>	0
<a href="#">Define AUTHN_SUCCESS</a>	0
<a href="#">Define AUTHZ_FAILED</a>	0
<a href="#">Define AUTHZ_SUCCESS</a>	0
<a href="#">Define CONFIG_ERR</a>	0
<a href="#">Define GPOA_T</a>	0
<a href="#">Define INVITE_ERR</a>	0
<a href="#">Define NOAUTH_ERR</a>	0
<a href="#">Define SYSTEM_ERR</a>	0
<a href="#">Define USER_ERR</a>	0
<a href="#">AuthenticationEngine.php</a>	0
<a href="#">PAPICrypt.php</a>	0
<a href="#">GenericConfigurator.php</a>	0
<a href="#">GenericEngine.php</a>	0
<a href="#">Hook.php</a>	0
<a href="#">LitePoA.php</a>	0
<a href="#">PoA.php</a>	0
<a href="#">PoAConfigurator.php</a>	0
<a href="#">PoAEventHandler.php</a>	0
<a href="#">PoAException.php</a>	0
<a href="#">PoALog.php</a>	0
<a href="#">PoAUtills.php</a>	0
<a href="#">openid-auto.php</a>	0
<a href="#">openid-ext.php</a>	0
<a href="#">openid-restricted.php</a>	0
<a href="#">openid.php</a>	0
<a href="#">papi1.5-slo-extra.php</a>	0
<a href="#">papi1.5-slo.php</a>	0
<a href="#">authz_convert.php</a>	0
<a href="#">Define ROOT</a>	0
<a href="#">invite_convert.php</a>	0
<a href="#">Class AuthenticationEngine</a>	0
<a href="#">Var \$engine_type</a>	0
<a href="#">Method authenticate</a>	0
<a href="#">Method getAttribute</a>	0
<a href="#">Method getAttributes</a>	0
<a href="#">Method isAuthenticated</a>	0
<a href="#">Method logout</a>	0
<a href="#">Class AuthorizationConfigurator</a>	0
<a href="#">Method getDefaultBehaviour</a>	0
<a href="#">Class AuthorizationEngine</a>	0

<a href="#">Var \$engine_type</a>	0
<a href="#">Method authorize</a>	0
<a href="#">Method getAuthorizedList</a>	0
<a href="#">Method isAuthorized</a>	0
<a href="#">Method revoke</a>	0
<a href="#">Class AutoPoA</a>	0
<a href="#">Method authenticate</a>	0
<a href="#">Method isAuthorized</a>	0
<a href="#">Class GenericConfigurator</a>	0
<a href="#">Var \$cfg</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method validate</a>	0
<a href="#">Class GenericEngine</a>	0
<a href="#">Var \$cfg</a>	0
<a href="#">Var \$engine_type</a>	0
<a href="#">Var \$handler</a>	0
<a href="#">Var \$hooks</a>	0
<a href="#">Var \$valid_hooks</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method addHook</a>	0
<a href="#">Method clean</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method registerHandler</a>	0
<a href="#">Method removeHook</a>	0
<a href="#">Method runHooks</a>	0
<a href="#">Method setHandler</a>	0
<a href="#">Class Hook</a>	0
<a href="#">Var \$ callable_name</a>	0
<a href="#">Var \$ hook</a>	0
<a href="#">Var \$ name</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method getName</a>	0
<a href="#">Method run</a>	0
<a href="#">Class LightOpenID</a>	0
<a href="#">Var \$aliases</a>	0
<a href="#">Var \$ax</a>	0
<a href="#">Var \$ax_to_sreg</a>	0
<a href="#">Var \$cainfo</a>	0
<a href="#">Var \$capath</a>	0
<a href="#">Var \$data</a>	0
<a href="#">Var \$identifier_select</a>	0
<a href="#">Var \$optional</a>	0
<a href="#">Var \$required</a>	0
<a href="#">Var \$returnUrl</a>	0
<a href="#">Var \$server</a>	0
<a href="#">Var \$setup_url</a>	0
<a href="#">Var \$sreg</a>	0

<a href="#">Var \$trustRoot</a>	0
<a href="#">Var \$verify_peer</a>	0
<a href="#">Var \$version</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method authUrl</a>	0
<a href="#">Method authUrl v1</a>	0
<a href="#">Method authUrl v2</a>	0
<a href="#">Method axParams</a>	0
<a href="#">Method build_url</a>	0
<a href="#">Method discover</a>	0
<a href="#">Method getAttributes</a>	0
<a href="#">Method getAxAttributes</a>	0
<a href="#">Method getSregAttributes</a>	0
<a href="#">Method hostExists</a>	0
<a href="#">Method htmlTag</a>	0
<a href="#">Method request</a>	0
<a href="#">Method request_curl</a>	0
<a href="#">Method request_streams</a>	0
<a href="#">Method sregParams</a>	0
<a href="#">Method validate</a>	0
<a href="#">Method get</a>	0
<a href="#">Method set</a>	0
<a href="#">Class LitePoA</a>	0
<a href="#">Var \$handler</a>	0
<a href="#">Method checkAccess</a>	0
<a href="#">Method setIDBuilder</a>	0
<a href="#">Class PAPICrypt</a>	0
<a href="#">Var \$privkey</a>	0
<a href="#">Var \$privkeylen</a>	0
<a href="#">Var \$pubkey</a>	0
<a href="#">Var \$pubkeylen</a>	0
<a href="#">Var \$symmetric</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method decrypt</a>	0
<a href="#">Method decrypt3DES</a>	0
<a href="#">Method decryptAES</a>	0
<a href="#">Method encrypt</a>	0
<a href="#">Method encrypt3DES</a>	0
<a href="#">Method encryptAES</a>	0
<a href="#">Method endSymmetric</a>	0
<a href="#">Method initialize3DES</a>	0
<a href="#">Method initializeAES</a>	0
<a href="#">Method setPrivateKey</a>	0
<a href="#">Method setPublicKey</a>	0
<a href="#">Method setSymmetricKey</a>	0
<a href="#">Class PoA</a>	0
<a href="#">Var \$attributes</a>	0
<a href="#">Var \$authn_engine</a>	0
<a href="#">Var \$authz_engines</a>	0

<a href="#">Var \$autoload</a>	0
<a href="#">Var \$cfg</a>	0
<a href="#">Var \$db_manager</a>	0
<a href="#">Var \$handler</a>	0
<a href="#">Var \$local_site</a>	0
<a href="#">Var \$log</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method addHook</a>	0
<a href="#">Method authenticate</a>	0
<a href="#">Method authorize</a>	0
<a href="#">Method clean</a>	0
<a href="#">Method getAttribute</a>	0
<a href="#">Method getAttributes</a>	0
<a href="#">Method getAuthorizationEngines</a>	0
<a href="#">Method getAuthorizationLevels</a>	0
<a href="#">Method isAuthenticated</a>	0
<a href="#">Method isAuthorized</a>	0
<a href="#">Method logout</a>	0
<a href="#">Method removeHook</a>	0
<a href="#">Method revoke</a>	0
<a href="#">Class PoAConfigurator</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method getAuthnEngine</a>	0
<a href="#">Method getAuthnEngineConfFile</a>	0
<a href="#">Method getAuthzEngineConfFile</a>	0
<a href="#">Method getAuthzEngines</a>	0
<a href="#">Method getAuthzLevels</a>	0
<a href="#">Method getInviteErrorURL</a>	0
<a href="#">Method getLanguage</a>	0
<a href="#">Method getLogFile</a>	0
<a href="#">Method getLogLevel</a>	0
<a href="#">Method getNoAuthErrorURL</a>	0
<a href="#">Method getSystemErrorURL</a>	0
<a href="#">Method isDebug</a>	0
<a href="#">Class PoAEventHandler</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method abort</a>	0
<a href="#">Method autoloadHandler</a>	0
<a href="#">Method errorHandler</a>	0
<a href="#">Method exceptionHandler</a>	0
<a href="#">Method setDebug</a>	0
<a href="#">Method setLogger</a>	0
<a href="#">Class PoAException</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method __toString</a>	0
<a href="#">Class PoALog</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method getLogLevel</a>	0

<a href="#">Method write</a>	0
<a href="#">Class PoAUtils</a>	0
<a href="#">Method lang_code</a>	0
<a href="#">Method msg</a>	0
<a href="#">AttributeFilterAuthzEngine.php</a>	0
<a href="#">Define ATTRIBUTE_BEFORE_FILTERS</a>	0
<a href="#">AttributeFilterConfigurator.php</a>	0
<a href="#">papi1.5-attr-authz.php</a>	0
<a href="#">Function attrHook</a>	0
<a href="#">Class AttributeFilterAuthzEngine</a>	0
<a href="#">Var \$valid_hooks</a>	0
<a href="#">Method authorize</a>	0
<a href="#">Method getAuthorizedList</a>	0
<a href="#">Method isAuthorized</a>	0
<a href="#">Method revoke</a>	0
<a href="#">Class AttributeFilterConfigurator</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method getAllowedAttributes</a>	0
<a href="#">Method getDeniedAttributes</a>	0
<a href="#">ASN1.php</a>	0
<a href="#">Define TAG_BER</a>	0
<a href="#">Define TAG_BITSTRING</a>	0
<a href="#">Define TAG_BOOLEAN</a>	0
<a href="#">Define TAG_INTEGER</a>	0
<a href="#">Define TAG_SEQUENCE</a>	0
<a href="#">RSAPrivateKey.php</a>	0
<a href="#">RSAPublicKey.php</a>	0
<a href="#">Define PUBLIC_KEY_OID</a>	0
<a href="#">Class ASN1</a>	0
<a href="#">Var \$tag</a>	0
<a href="#">Var \$value</a>	0
<a href="#">Constructor __construct</a>	0
<a href="#">Method bin2int</a>	0
<a href="#">Method decode</a>	0
<a href="#">Method encode</a>	0
<a href="#">Method getInteger</a>	0
<a href="#">Method getTag</a>	0
<a href="#">Method getValue</a>	0
<a href="#">Method getValues</a>	0
<a href="#">Method int2bin</a>	0
<a href="#">Method readByte</a>	0
<a href="#">Method readBytes</a>	0
<a href="#">Method setInteger</a>	0
<a href="#">Method setValue</a>	0
<a href="#">Method setValues</a>	0
<a href="#">Method writeByte</a>	0
<a href="#">Method writeBytes</a>	0
<a href="#">Class RSAPrivateKey</a>	0
<a href="#">Var \$bits</a>	0

<a href="#">Var \$coefficient</a>	0
<a href="#">Var \$der</a>	0
<a href="#">Var \$exponent1</a>	0
<a href="#">Var \$exponent2</a>	0
<a href="#">Var \$modulus</a>	0
<a href="#">Var \$pem</a>	0
<a href="#">Var \$prime1</a>	0
<a href="#">Var \$prime2</a>	0
<a href="#">Var \$private_exponent</a>	0
<a href="#">Var \$public_exponent</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method decode</a>	0
<a href="#">Method encode</a>	0
<a href="#">Method fromDER</a>	0
<a href="#">Method fromPEM</a>	0
<a href="#">Method getBits</a>	0
<a href="#">Method getCoefficient</a>	0
<a href="#">Method getDER</a>	0
<a href="#">Method getExponent1</a>	0
<a href="#">Method getExponent2</a>	0
<a href="#">Method getModulus</a>	0
<a href="#">Method getPEM</a>	0
<a href="#">Method getPrime1</a>	0
<a href="#">Method getPrime2</a>	0
<a href="#">Method getPrivateExponent</a>	0
<a href="#">Method getPublicExponent</a>	0
<a href="#">Method setCoefficient</a>	0
<a href="#">Method setExponent1</a>	0
<a href="#">Method setExponent2</a>	0
<a href="#">Method setModulus</a>	0
<a href="#">Method setPrime1</a>	0
<a href="#">Method setPrime2</a>	0
<a href="#">Method setPrivateExponent</a>	0
<a href="#">Method setPublicExponent</a>	0
<a href="#">Class RSAPublicKey</a>	0
<a href="#">Var \$bits</a>	0
<a href="#">Var \$der</a>	0
<a href="#">Var \$exponent</a>	0
<a href="#">Var \$modulus</a>	0
<a href="#">Var \$pem</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method decode</a>	0
<a href="#">Method encode</a>	0
<a href="#">Method fromDER</a>	0
<a href="#">Method fromPEM</a>	0
<a href="#">Method getBits</a>	0
<a href="#">Method getDER</a>	0
<a href="#">Method getExponent</a>	0
<a href="#">Method getModulus</a>	0

<a href="#">Method getPEM</a>	0
<a href="#">Method setModulus</a>	0
<a href="#">Method setPublicExponent</a>	0
<a href="#">DBADB.php</a>	0
<a href="#">GenericDB.php</a>	0
<a href="#">GenericMySQLDB.php</a>	0
<a href="#">SessionDB.php</a>	0
<a href="#">Class DBADB</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method check</a>	0
<a href="#">Method close</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method fetch_all</a>	0
<a href="#">Method open</a>	0
<a href="#">Method replace</a>	0
<a href="#">Class GenericDB</a>	0
<a href="#">Var \$cfg</a>	0
<a href="#">Var \$db</a>	0
<a href="#">Var \$error</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method check</a>	0
<a href="#">Method close</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method fetch_all</a>	0
<a href="#">Method open</a>	0
<a href="#">Method replace</a>	0
<a href="#">Class GenericMySQLDB</a>	0
<a href="#">Var \$create_sql</a>	0
<a href="#">Var \$delete_sql</a>	0
<a href="#">Var \$insert_sql</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Var \$prefix</a>	0
<a href="#">Var \$search_all_sql</a>	0
<a href="#">Var \$search_sql</a>	0
<a href="#">Var \$update_sql</a>	0
<a href="#">Method check</a>	0
<a href="#">Method close</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method fetch_all</a>	0
<a href="#">Method getError</a>	0
<a href="#">Method open</a>	0
<a href="#">Method replace</a>	0



<a href="#">Class SessionDB</a>	0
<a href="#">Var \$session_name</a>	0
<a href="#">Method check</a>	0
<a href="#">Method close</a>	0
<a href="#">Method delete</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method fetch_all</a>	0
<a href="#">Method open</a>	0
<a href="#">Method replace</a>	0
<a href="#">messages-en_EN.php</a>	0
<a href="#">Global Variable \$messages</a>	0
<a href="#">messages-es_ES.php</a>	0
<a href="#">AuthorizedDB.php</a>	0
<a href="#">AuthorizedDBADB.php</a>	0
<a href="#">InviteAuthzEngine.php</a>	0
<a href="#">Define AUTHORIZED_DBA</a>	0
<a href="#">Define AUTHORIZED_MYSQL</a>	0
<a href="#">Define AUTHORIZED_SESSION</a>	0
<a href="#">Define INVITES_BEFORE_AUTHORIZATION</a>	0
<a href="#">Define INVITES_DBA</a>	0
<a href="#">Define INVITES_MYSQL</a>	0
<a href="#">Define INVITES_SESSION</a>	0
<a href="#">InviteConfigurator.php</a>	0
<a href="#">InviteDB.php</a>	0
<a href="#">InviteDBADB.php</a>	0
<a href="#">papi1.5-invite-authz.php</a>	0
<a href="#">Class AuthorizedDB</a>	0
<a href="#">Method replace_authorization</a>	0
<a href="#">Class AuthorizedDBADB</a>	0
<a href="#">Var \$authorized</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method fetch_all</a>	0
<a href="#">Method open</a>	0
<a href="#">Method replace_authorization</a>	0
<a href="#">Class InviteAuthzEngine</a>	0
<a href="#">Var \$authz_db</a>	0
<a href="#">Var \$invites_db</a>	0
<a href="#">Var \$valid_hooks</a>	0
<a href="#">Method authorize</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method emailVerify</a>	0
<a href="#">Method finish</a>	0
<a href="#">Method getAuthorizedList</a>	0
<a href="#">Method getPendingInvites</a>	0
<a href="#">Method invite</a>	0
<a href="#">Method isAuthorized</a>	0
<a href="#">Method removeInvite</a>	0
<a href="#">Method revoke</a>	0

<a href="#">Class InviteConfigurator</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method doEmailVerify</a>	0
<a href="#">Method getAdminEmail</a>	0
<a href="#">Method getAuthorizedDBFile</a>	0
<a href="#">Method getAuthorizedDBHost</a>	0
<a href="#">Method getAuthorizedDBName</a>	0
<a href="#">Method getAuthorizedDBPassword</a>	0
<a href="#">Method getAuthorizedDBType</a>	0
<a href="#">Method getAuthorizedDBUser</a>	0
<a href="#">Method getEmailVerifyAlgRegEx</a>	0
<a href="#">Method getEmailVerifyAttribute</a>	0
<a href="#">Method getEmailVerifyRegEx</a>	0
<a href="#">Method getInvitesDBFile</a>	0
<a href="#">Method getInvitesDBHost</a>	0
<a href="#">Method getInvitesDBName</a>	0
<a href="#">Method getInvitesDBPassword</a>	0
<a href="#">Method getInvitesDBType</a>	0
<a href="#">Method getInvitesDBUser</a>	0
<a href="#">Method getInviteSubject</a>	0
<a href="#">Method getInviteText</a>	0
<a href="#">Method getInviteURL</a>	0
<a href="#">Method getUniqueAttributes</a>	0
<a href="#">Class InviteDB</a>	0
<a href="#">Method replace_invite</a>	0
<a href="#">Class InviteDBADB</a>	0
<a href="#">Var \$invite</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method fetch_all</a>	0
<a href="#">Method open</a>	0
<a href="#">Method replace_invite</a>	0
<a href="#">OpenIDConfigurator.php</a>	0
<a href="#">Class OpenIDConfigurator</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method getLoginURL</a>	0
<a href="#">Method getLogoutURL</a>	0
<a href="#">Method getProviders</a>	0
<a href="#">Method isAutoLogin</a>	0
<a href="#">OpenIDAuthnEngine.php</a>	0
<a href="#">Class OpenIDAuthnEngine</a>	0
<a href="#">Var \$attributes</a>	0
<a href="#">Var \$OpenID</a>	0
<a href="#">Var \$reqAttributes</a>	0
<a href="#">Var \$status</a>	0
<a href="#">Constructor construct</a>	0
<a href="#">Method authenticate</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method getAttribute</a>	0

<a href="#">Method <u>getAttributes</u></a>	0
<a href="#">Method <u>isAuthenticated</u></a>	0
<a href="#">Method <u>logout</u></a>	0
<a href="#">PAPIAuthnEngine.php</a>	0
<a href="#">Define <u>ATTR_SEPARATOR</u></a>	0
<a href="#">Define <u>NAMEVALUE_SEPARATOR</u></a>	0
<a href="#">Define <u>NS_PAPI_ATTRIBUTES</u></a>	0
<a href="#">Define <u>NS_PAPI_PROTOCOL</u></a>	0
<a href="#">Define <u>PAPI_ATTRIBUTE_PARSER</u></a>	0
<a href="#">Define <u>PAPI_CLEAN_URL</u></a>	0
<a href="#">Define <u>PAPI_DBA</u></a>	0
<a href="#">Define <u>PAPI_MYSQL</u></a>	0
<a href="#">Define <u>PAPI_REDIRECT_URL_FINISH</u></a>	0
<a href="#">Define <u>PAPI_RESTORE_ORIGINAL_REQUEST</u></a>	0
<a href="#">Define <u>PAPI_SESSION</u></a>	0
<a href="#">Define <u>PROTO_ATTR_AS_ID</u></a>	0
<a href="#">Define <u>PROTO_ATTR_EXPIRE_TIME</u></a>	0
<a href="#">Define <u>PROTO_ATTR_KEY</u></a>	0
<a href="#">Define <u>PROTO_ATTR_PREFIX</u></a>	0
<a href="#">Define <u>REQUEST_LIFETIME</u></a>	0
<a href="#">Define <u>VALUE_SEPARATOR</u></a>	0
<a href="#">PAPIConfigurator.php</a>	0
<a href="#">PAPIDB.php</a>	0
<a href="#">PAPIDBADB.php</a>	0
<a href="#">PAPIMySQLDB.php</a>	0
<a href="#">Define <u>PAPI_DB_TABLE</u></a>	0
<a href="#">PAPISessionDB.php</a>	0
<a href="#">papi1.5-authn-attr.php</a>	0
<a href="#">papi1.5-customattr.php</a>	0
<a href="#">Function <u>handler</u></a>	0
<a href="#">papi1.5-papiopoa-hook.php</a>	0
<a href="#">Function <u>hook</u></a>	0
<a href="#">Class <u>AttributeHandler</u></a>	0
<a href="#">Method <u>handle</u></a>	0
<a href="#">Class <u>Example</u></a>	0
<a href="#">Method <u>hook</u></a>	0
<a href="#">Class <u>PAPIAuthnEngine</u></a>	0
<a href="#">Var <u>\$assertion</u></a>	0
<a href="#">Var <u>\$as_id</u></a>	0
<a href="#">Var <u>\$attributes</u></a>	0
<a href="#">Var <u>\$cfg</u></a>	0
<a href="#">Var <u>\$clean_url</u></a>	0
<a href="#">Var <u>\$cookie_name</u></a>	0
<a href="#">Var <u>\$crypto</u></a>	0
<a href="#">Var <u>\$db</u></a>	0
<a href="#">Var <u>\$enforcing</u></a>	0
<a href="#">Var <u>\$expiration_time</u></a>	0
<a href="#">Var <u>\$global_expire_time</u></a>	0
<a href="#">Var <u>\$id</u></a>	0

<a href="#">Var \$key</a>	0
<a href="#">Var \$lkey</a>	0
<a href="#">Var \$opoa</a>	0
<a href="#">Var \$pkey</a>	0
<a href="#">Var \$skip_redirection</a>	0
<a href="#">Var \$status</a>	0
<a href="#">Var \$valid_hooks</a>	0
<a href="#">Method attributeQuery</a>	0
<a href="#">Method authenticate</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method deleteCookie</a>	0
<a href="#">Method deleteRequest</a>	0
<a href="#">Method getAttribute</a>	0
<a href="#">Method getAttributes</a>	0
<a href="#">Method getNewCookie</a>	0
<a href="#">Method getRedirectLocation</a>	0
<a href="#">Method getSingleLogoutLocation</a>	0
<a href="#">Method getSingleLogoutResponseLocation</a>	0
<a href="#">Method isAuthenticated</a>	0
<a href="#">Method isSafe</a>	0
<a href="#">Method loadRequest</a>	0
<a href="#">Method logout</a>	0
<a href="#">Method redirect</a>	0
<a href="#">Method saveRequest</a>	0
<a href="#">Method testCookie</a>	0
<a href="#">Method testResponse</a>	0
<a href="#">Class PAPIConfigurator</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method getCookieDomain</a>	0
<a href="#">Method getCookieTimeout</a>	0
<a href="#">Method getDBFile</a>	0
<a href="#">Method getDBHost</a>	0
<a href="#">Method getDBName</a>	0
<a href="#">Method getDBPassword</a>	0
<a href="#">Method getDBPrefix</a>	0
<a href="#">Method getDBType</a>	0
<a href="#">Method getDBUser</a>	0
<a href="#">Method getFriendlyName</a>	0
<a href="#">Method getHomeLocatorID</a>	0
<a href="#">Method getID</a>	0
<a href="#">Method getLKey</a>	0
<a href="#">Method getLocation</a>	0
<a href="#">Method getLogoutURL</a>	0
<a href="#">Method getPubKeyFile</a>	0
<a href="#">Method getRedirectType</a>	0
<a href="#">Method getRedirectURL</a>	0
<a href="#">Method getRequestLifetime</a>	0
<a href="#">Class PAPIDB</a>	0
<a href="#">Method purge</a>	0

<a href="#">Method replaceContents</a>	0
<a href="#">Class PAPIDBADB</a>	0
<a href="#">Var \$context</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method purge</a>	0
<a href="#">Method replaceContents</a>	0
<a href="#">Class PAPIMySQLDB</a>	0
<a href="#">Var \$context</a>	0
<a href="#">Var \$prefix</a>	0
<a href="#">Var \$purge_sql</a>	0
<a href="#">Constructor    construct</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method purge</a>	0
<a href="#">Method replaceContents</a>	0
<a href="#">Class PAPISessionDB</a>	0
<a href="#">Var \$context</a>	0
<a href="#">Method fetch</a>	0
<a href="#">Method purge</a>	0
<a href="#">Method replaceContents</a>	0
<a href="#">QueryFilterAuthzEngine.php</a>	0
<a href="#">Define QUERY_BEFORE_FILTERS</a>	0
<a href="#">QueryFilterConfigurator.php</a>	0
<a href="#">papi1.5-query-authz.php</a>	0
<a href="#">Function queryHook</a>	0
<a href="#">Class QueryFilterAuthzEngine</a>	0
<a href="#">Var \$valid_hooks</a>	0
<a href="#">Method authorize</a>	0
<a href="#">Method getAuthorizedList</a>	0
<a href="#">Method isAuthorized</a>	0
<a href="#">Method revoke</a>	0
<a href="#">Class QueryFilterConfigurator</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method getAllowedPatterns</a>	0
<a href="#">Method getDeniedPatterns</a>	0
<a href="#">SimpleSAMLphpAuthnEngine.php</a>	0
<a href="#">Define NS_SAML2_PROTOCOL</a>	0
<a href="#">Class SimpleSAMLphpAuthnEngine</a>	0
<a href="#">Var \$attributes</a>	0
<a href="#">Var \$simplesaml</a>	0
<a href="#">Var \$status</a>	0
<a href="#">Method authenticate</a>	0
<a href="#">Method configure</a>	0
<a href="#">Method getAttribute</a>	0
<a href="#">Method getAttributes</a>	0
<a href="#">Method isAuthenticated</a>	0
<a href="#">Method logout</a>	0
<a href="#">SourceIPAddrAuthzEngine.php</a>	0
<a href="#">Define SOURCEADDR_BEFORE_FILTERS</a>	0
<a href="#">SourceIPAddrConfigurator.php</a>	0

<a href="#">papi1.5-ip-authz.php</a>	0
<a href="#">Function ipHook</a>	0
<a href="#">Class SourceIPAddrAuthzEngine</a>	0
<a href="#">Var \$valid_hooks</a>	0
<a href="#">Method authorize</a>	0
<a href="#">Method getAuthorizedList</a>	0
<a href="#">Method isAuthorized</a>	0
<a href="#">Method revoke</a>	0
<a href="#">Class SourceIPAddrConfigurator</a>	0
<a href="#">Var \$mandatory_options</a>	0
<a href="#">Method getAllowed</a>	0
<a href="#">Method getDenied</a>	0
<a href="#">IPFilters.conf.php</a>	0
<a href="#">Appendix A - Class Trees</a>	0
<a href="#">phpPoA2</a>	0
<a href="#">INSTALL</a>	0
<a href="#">README</a>	0

# AutoPoA.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](mailto: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](mailto: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](#)

# PAPICrypt.php

- **Package** phpPoA2
- **Version** 2.0
- **Copyright** Copyright 2005-2011 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](#)

# GenericConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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 41]*

include\_once ["AutoPoA.php"](#)*[line 40]*

require\_once ["PoAEventHandler.php"](#)*[line 39]*

require\_once ["PoAUtils.php"](#)*[line 38]*



# PoAConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](#)

# openid-auto.php

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

- **Package** phpPoA2
- **Author** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.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** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.es) >
- **Filesource** [Source Code for this file](#)

# openid-ext.php

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

- **Package** phpPoA2
- **Author** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.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** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.es)>
- **Filesource** [Source Code for this file](#)

# openid-restricted.php

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

- **Package** phpPoA2
- **Author** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.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** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.es)>
- **Filesource** [Source Code for this file](#)



## openid.php

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

- **Package** phpPoA2
- **Author** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.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** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.es)>
- **Filesource** [Source Code for this file](#)

# 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](mailto: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](mailto: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](mailto: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](mailto:jaime.perez@rediris.es)>
- **Filesource** [Source Code for this file](#)

# authz\_convert.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](mailto: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](mailto: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](mailto: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 LightOpenID

[line 47]

**This class provides a simple interface for OpenID (1.1 and 2.0) authentication.**

Supports Yadis discovery. The authentication process is stateless/dumb.

Usage: Sign-on with OpenID is a two step process: Step one is authentication with the provider:

```
1 $openid = new LightOpenID;
2 $openid->identity = 'ID supplied by user';
3 header('Location: ' . $openid->authUrl());
```

The provider then sends various parameters via GET, one of them is openid\_mode. Step two is verification:

```
1 if ($this->data['openid_mode']) {
2     $openid = new LightOpenID;
3     echo $openid->validate() ? 'Logged in.' : 'Failed';
4 }
```

Optionally, you can set \$returnUrl and \$realm (or \$trustRoot, which is an alias). The default values for those are:  
\$openid->realm = (!empty(\$\_SERVER['HTTPS']) ? 'https' : 'http') . '://' . \$\_SERVER['HTTP\_HOST']; \$openid->returnUrl = \$openid->realm . \$\_SERVER['REQUEST\_URI']; If you don't know their meaning, refer to any openid tutorial, or specification. Or just guess.

AX and SREG extensions are supported. To use them, specify \$openid->required and/or \$openid->optional before calling \$openid->authUrl(). These are arrays, with values being AX schema paths (the 'path' part of the URL). For example: \$openid->required = array('namePerson/friendly', 'contact/email'); \$openid->optional = array('namePerson/first'); If the server supports only SREG or OpenID 1.1, these are automatically mapped to SREG names, so that user doesn't have to know anything about the server.

To get the values, use \$openid->getAttributes().

The library requires PHP >= 5.1.2 with curl or http/https stream wrappers enabled.

- **Package** phpPoA2
- **Author** Mewp
- **Copyright** Copyright (c) 2010, Mewp
- **License** [MIT](#)

## LightOpenID::\$aliases

*mixed* = [line 57]

- **Access** protected

### LightOpenID::\$sax

*mixed* = false [*line 58*]

- **Access** protected

### LightOpenID::\$sax\_to\_sreg

```
mixed = array(  
    'namePerson/friendly' => 'nickname',  
    'contact/email'       => 'email',  
    'namePerson'          => 'fullname',  
    'birthDate'           => 'dob',  
    'person/gender'       => 'gender',  
    'contact/postalCode/home' => 'postcode',  
    'contact/country/home' => 'country',  
    'pref/language'       => 'language',  
    'pref/timezone'       => 'timezone',  
) [line 59]
```

- **Static**
- **Access** protected

### LightOpenID::\$sainfo

*mixed* = [*line 54*]

- **Access** public

### LightOpenID::\$scapath

*mixed* = [*line 53*]

- **Access** public

#### **LightOpenID::\$data**

*mixed* = [line 55]

- **Access** public

#### **LightOpenID::\$identifier\_select**

*mixed* = false  
[line 57]

- **Access** protected

#### **LightOpenID::\$optional**

*mixed* = array() [line 51]

- **Access** public

#### **LightOpenID::\$required**

*mixed* = array() [line 50]

- **Access** public

#### **LightOpenID::\$returnUrl**

*mixed* = [line 49]

- **Access** public

#### **LightOpenID::\$server**

*mixed = [line 57]*

- **Access** protected

#### **LightOpenID::\$setup\_url**

*mixed = null [line 58]*

- **Access** protected

#### **LightOpenID::\$sreg**

*mixed = false [line 58]*

- **Access** protected

#### **LightOpenID::\$trustRoot**

*mixed = [line 57]*

- **Access** protected

#### **LightOpenID::\$verify\_peer**

*mixed = [line 52]*

- **Access** public

## LightOpenID::\$version

*mixed = [line 57]*

- **Access** protected

Constructor *void* function LightOpenID::\_\_construct() *[line 71]*

*String* function LightOpenID::authUrl([\$immediate = false], \$select\_identifier) *[line 600]*

### **Function Parameters:**

- *String* **\$select\_identifier** Whether to request OP to select identity for an user in OpenID 2. Does not affect OpenID 1.
- **\$immediate**

**Returns authentication url. Usually, you want to redirect your user to it.**

- **Throws** `ErrorException`

*void* function LightOpenID::authUrl\_v1(\$immediate) *[line 541]*

### **Function Parameters:**

- **\$immediate**

- **Access** protected



*void function* LightOpenID::authUrl\_v2(\$immediate) [*line 562*]

***Function Parameters:***

- **\$immediate**
- **Access** protected

*void function* LightOpenID::axParams() [*line 502*]

- **Access** protected

*void function* LightOpenID::build\_url(\$url, \$parts) [*line 303*]

***Function Parameters:***

- **\$url**
- **\$parts**
- **Access** protected

*String function* LightOpenID::discover(\$url) [*line 341*]

***Function Parameters:***

- *\$url* **\$url** Identity URL.

**Performs Yadis and HTML discovery. Normally not used.**

- **Throws** `ErrorException`

*void function LightOpenID::getAttributes() [line 753]*

**Gets AX/SREG attributes provided by OP. should be used only after successful validation.**

Note that it does not guarantee that any of the required/optional parameters will be present, or that there will be no other attributes besides those specified. In other words. OP may provide whatever information it wants to. \* SREG names will be mapped to AX names. \* @return Array Array of attributes with keys being the AX schema names, e.g. 'contact/email'

- See <http://www.axschema.org/types/>

*void function LightOpenID::getAxAttributes() [line 678]*

- **Access** protected

*void function LightOpenID::getSregAttributes() [line 725]*

- **Access** protected

*true, function LightOpenID::hostExists(\$url) [line 132]*

**Function Parameters:**

- **\$url** **\$url** url to check

**Checks if the server specified in the url exists.**

*void function LightOpenID::htmlTag(\$content, \$tag, \$attrName, \$attrValue, \$valueName) [line 326]*

***Function Parameters:***

- **\$content**
- **\$tag**
- **\$attrName**
- **\$attrValue**
- **\$valueName**

**Helper function used to scan for <meta>/<link> tags and extract information from them**

- **Access protected**

*void function LightOpenID::request(\$url, [\$method = 'GET'], [\$params = array()]) [line 293]*

***Function Parameters:***

- **\$url**
- **\$method**
- **\$params**

- **Access protected**

*void function LightOpenID::request\_curl(\$url, [\$method = 'GET'], [\$params = array()]) [line 147]*

***Function Parameters:***

- **\$url**

- **\$method**
- **\$params**

- **Access** protected

*void function LightOpenID::request\_streams(\$url, [\$method = 'GET'], [\$params = array()]) [line 203]*

**Function Parameters:**

- **\$url**
- **\$method**
- **\$params**

- **Access** protected

*void function LightOpenID::sregParams() [line 475]*

- **Access** protected

*Bool function LightOpenID::validate() [line 616]*

**Performs OpenID verification with the OP.**

- **Throws** `ErrorException`

*void function LightOpenID::\_\_get(\$name) [line 110]*

**Function Parameters:**

- **\$name**

*void function LightOpenID::\_\_set(\$name, \$value) [line 88]*

**Function Parameters:**

- **\$name**
- **\$value**

## 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 PAPICrypt

[line [29](#)]

**Cryptographic routines for PAPI protocol v1.**

- **Package** phpPoA2

**PAPICrypt::\$privkey**

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

- **Access** protected

#### **PAPICrypt::\$privkeylen**

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

- **Access** protected

#### **PAPICrypt::\$pubkey**

*mixed* = [[line 32](#)]

- **Access** protected

#### **PAPICrypt::\$pubkeylen**

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

- **Access** protected

#### **PAPICrypt::\$symmetric**

*mixed* = [[line 38](#)]

- **Access** protected

Constructor *void* function PAPICrypt::\_\_construct([\$symmetric = ""], [\$pubkey = ""],  
[\$privkey = ""], [\$passphrase = ""]) [[line 43](#)]

#### **Function Parameters:**

- **\$symmetric**
- **\$pubkey**
- **\$privkey**
- **\$passphrase**

- **Access public**

*void function* PAPICrypt::decrypt(\$input) [*line* [248](#)]

***Function Parameters:***

- **\$input**

- **Access public**

*void function* PAPICrypt::decrypt3DES(\$input) [*line* [206](#)]

***Function Parameters:***

- **\$input**

- **Access public**

*void function* PAPICrypt::decryptAES(\$input) [*line* [172](#)]

***Function Parameters:***



- **\$input**

- **Access** public

*void* function PAPICrypt::encrypt(\$input) [*line* [223](#)]

***Function Parameters:***

- **\$input**

- **Access** public

*void* function PAPICrypt::encrypt3DES(\$input) [*line* [189](#)]

***Function Parameters:***

- **\$input**

- **Access** public

*void* function PAPICrypt::encryptAES(\$input) [*line* [155](#)]

***Function Parameters:***

- **\$input**

- **Access** public

*void function* PAPICrypt::endSymmetric(\$module) [[line 147](#)]

***Function Parameters:***

- **\$module**

- **Access** protected

*void function* PAPICrypt::initialize3DES() [[line 128](#)]

- **Access** protected

*void function* PAPICrypt::initializeAES() [[line 109](#)]

- **Access** protected

*void function* PAPICrypt::setPrivateKey(\$privkey, [\$passphrase = ""]) [[line 85](#)]

***Function Parameters:***

- **\$privkey**
- **\$passphrase**

- **Access** public

*void function* PAPICrypt::setPublicKey(\$pubkey) [[line 61](#)]

***Function Parameters:***

- **\$pubkey**

- **Access** public

void function PAPICrypt::setSymmetricKey(\$symmetric) [[line 54](#)]

**Function Parameters:**

- **\$symmetric**

- **Access** public

## Class PoA

[[line 47](#)]

Standard class that implements all the functionality of the phpPoA.

- **Package** phpPoA2

### PoA::\$attributes

*mixed* = [[line 52](#)]

- **Access** protected

#### **PoA::\$authn\_engine**

*mixed* = [[line 51](#)]

- **Access** protected

#### **PoA::\$authz\_engines**

*mixed* = [[line 53](#)]

- **Access** protected

#### **PoA::\$autoload**

*mixed* = [[line 55](#)]

- **Access** protected

#### **PoA::\$cfg**

*mixed* = [[line 49](#)]

- **Access** protected

#### **PoA::\$db\_manager**

*mixed* = [[line 54](#)]

- **Access** protected

#### **PoA::\$handler**

*mixed* = [[line 56](#)]

- **Access** protected

#### **PoA::\$local\_site**

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

- **Access** protected

#### **PoA::\$log**

*mixed* = [[line 50](#)]

- **Access** protected

Constructor *void* function PoA::\_\_construct(\$site, [\$file = null]) [[line 63](#)]

#### **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* [128](#)]

**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* [179](#)]

**Perform a federated login for the user.**

- **Access** public

*AUTHZ\_SUCCESS* function PoA::authorize(\$user, \$attrs, [\$reference = null], [\$expires = 0], [\$engine = null]) [*line* [376](#)]

**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 114](#)]

- **Access** protected

*the* function PoA::getAttribute(\$name, \$namespace) [[line 266](#)]

#### **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 243](#)]

### **Retrieve the attributes provided by the user when logged in.**

- **Access** public

The function PoA::getAuthorizationEngines([\$engine = null]) [[line 457](#)]

**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 491](#)]

**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 214](#)]

**Query the current status of the user in the federation.**



- **Access** public

*AUTHZ\_SUCCESS* function PoA::isAuthorized(\$user, \$attrs, [\$engine = null]) [*line* [319](#)]

**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* [288](#)]

**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 153](#)]

**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 416](#)]

**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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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

# ASN1.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](#)

TAG\_BER = 0x00 [*line 31*]

## **Basic tags.**

TAG\_BITSTRING = 0x03 [*line 34*]

TAG\_BOOLEAN = 0x01 [*line 32*]

TAG\_INTEGER = 0x02 [*line 33*]

TAG\_SEQUENCE = 0x30 [*line 40*]

## **Constructed tags.**

# RSAPrivateKey.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto:jaime.perez@rediris.es)>
- **Version** 2.0
- **Copyright** Copyright 2005-2011 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](#)

# RSAPublicKey.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto:jaime.perez@rediris.es)>
- **Version** 2.0
- **Copyright** Copyright 2005-2011 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](#)

PUBLIC\_KEY\_OID = "1.2.840.113549.1.1.1" [[line 30](#)]

**The OID for public keys.**

## Class ASN1

[[line 50](#)]

**This class implements ASN.1 encoding.**

Please, be aware that this class is INCOMPLETE, as it is intended for BER/DER encoding primarily.

- **Package** phpPoA2
- **Sub-Package** crypto

### ASN1::\$tag

*mixed* = [line [52](#)]

- **Access** public

### ASN1::\$value

*mixed* = [line [53](#)]

- **Access** public

Constructor *void* function ASN1::\_\_construct([\$tag = 0x00], [\$value = "]) [line [58](#)]

#### **Function Parameters:**

- **\$tag**
- **\$value**

**Build a new ASN1 from its tag and value.**

- **Access** public

*void* function ASN1::bin2int(\$bin) [line [260](#)]

#### **Function Parameters:**

- **\$bin**

**Decode an integer from its binary representation.**

- **Access** protected

*void function ASN1::decode(&\$buffer) [line [97](#)]*

**Function Parameters:**

- **&\$buffer**

**Decode an object in ASN.1 format.**

- **Access** public

*void function ASN1::encode() [line [66](#)]*

**Encode this object into ASN.1.**

- **Access** public

*void function ASN1::getInteger() [line [181](#)]*

**Get the integer value of this ASN.1.**

- **Access** public



*void function* ASN1::getTag() [*line* [144](#)]

**Get the tag of this ASN.1.**

- **Access** public

*void function* ASN1::getValue() [*line* [151](#)]

**Get the binary value of this ASN.1.**

- **Access** public

*void function* ASN1::getValues() [*line* [195](#)]

**Get the sequence of values stored in this ASN.1.**

- **Access** public

*void function* ASN1::int2bin(\$int) [*line* [275](#)]

**Function Parameters:**

- **\$int**

**Encode an integer in binary representation.**

- **Access** protected

*void function ASN1::readByte(&\$buffer) [line [237](#)]*

***Function Parameters:***

- **&\$buffer**

**Read first byte from the buffer and move the internal pointer.**

- **Access** protected

*void function ASN1::readBytes(&\$buffer, \$length) [line [228](#)]*

***Function Parameters:***

- **&\$buffer**
- **\$length**

**Read n bytes from a buffer and mov the internal pointer.**

- **Access** protected

*void function ASN1::setInteger(\$int) [line [188](#)]*

***Function Parameters:***

- **\$int**

**Set the value of this ASN.1 from an integer.**

- **Access public**

*void function* ASN1::setValue(\$value) [*line* [163](#)]

***Function Parameters:***

- **\$value**

**Set the binary value of this ASN.1.**

- **Access public**

*void function* ASN1::setValues(\$values) [*line* [211](#)]

***Function Parameters:***

- **\$values**

**Set the value of this ASN.1 to be a sequence of values.**

- **Access public**

*void function* ASN1::writeByte(&\$buffer, \$byte) [*line* [253](#)]

***Function Parameters:***

- **&\$buffer**
- **\$byte**

**Write a byte to a buffer.**

- **Access** protected

*void* function ASN1::writeBytes(&\$buffer, \$bytes) [*line* [244](#)]

**Function Parameters:**

- **&\$buffer**
- **\$bytes**

**Write bytes to a buffer.**

- **Access** protected

## Class RSAPrivateKey

[*line* [33](#)]

**Class to manage private keys.**

- **Package** phpPoA2
- **Sub-Package** crypto

**RSAPrivateKey::\$bits**

*mixed* = [*line* [38](#)]

- **Access** protected

#### **RSAPrivateKey::\$coefficient**

*mixed* = [line [45](#)]

- **Access** protected

#### **RSAPrivateKey::\$der**

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

- **Access** protected

#### **RSAPrivateKey::\$exponent1**

*mixed* = [line [43](#)]

- **Access** protected

#### **RSAPrivateKey::\$exponent2**

*mixed* = [line [44](#)]

- **Access** protected

#### **RSAPrivateKey::\$modulus**

*mixed* = [line [37](#)]

- **Access** protected

#### **RSAPrivateKey::\$pem**

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

- **Access** protected

#### **RSAPrivateKey::\$prime1**

*mixed* = [[line 41](#)]

- **Access** protected

#### **RSAPrivateKey::\$prime2**

*mixed* = [[line 42](#)]

- **Access** protected

#### **RSAPrivateKey::\$private\_exponent**

*mixed* = [[line 40](#)]

- **Access** protected

#### **RSAPrivateKey::\$public\_exponent**

*mixed* = [line [39](#)]

- **Access** protected

Constructor *void* function RSAPrivateKey::\_\_construct([\$pem = "]) [line [50](#)]

**Function Parameters:**

- **\$pem**

**Build a new private key from its PEM representation.**

- **Access** public

*void* function RSAPrivateKey::decode() [line [92](#)]

**Extract private key details from its DER representation.**

- **Access** protected

*void* function RSAPrivateKey::encode() [line [120](#)]

**Build a new private key represented by it DER and PEM formats from its modulus and public exponent.**

- **Access** public

*void* function RSAPrivateKey::fromDER(\$der) [line [59](#)]

**Function Parameters:**

- **\$der**

**Build the private key from its DER representation.**

- **Access** public

*void function* RSAPrivateKey::fromPEM(\$pem) [*line* [74](#)]

**Function Parameters:**

- **\$pem**

**Build the private key from its PEM representation.**

- **Access** protected

*void function* RSAPrivateKey::getBits() [*line* [162](#)]

**Get the length of the key in bits.**

- **Access** public

*void function* RSAPrivateKey::getCoefficient() [*line* [197](#)]

**Get the coefficient of the key.**



- **Access public**

*void function RSAPrivateKey::getDER() [line [134](#)]*

**Get the DER representation of the key.**

- **Access public**

*void function RSAPrivateKey::getExponent1() [line [183](#)]*

**Get the exponent 1 of the key.**

- **Access public**

*void function RSAPrivateKey::getExponent2() [line [190](#)]*

**Get the exponent 2 of the key.**

- **Access public**

*void function RSAPrivateKey::getModulus() [line [141](#)]*

**Get the modulus of the key.**

- **Access public**

*void function RSAPrivateKey::getPEM() [line [127](#)]*

**Get the PEM representation of the key.**

- **Access public**

*void* function RSAPrivateKey::getPrime1() [[line 169](#)]

**Get the prime 1 of the key.**

- **Access public**

*void* function RSAPrivateKey::getPrime2() [[line 176](#)]

**Get the prime 2 of the key.**

- **Access public**

*void* function RSAPrivateKey::getPrivateExponent() [[line 155](#)]

**Get the private exponent of the key.**

- **Access public**

*void* function RSAPrivateKey::getPublicExponent() [[line 148](#)]

**Get the public exponent of the key.**

- **Access public**

`void function RSAPrivateKey::setCoefficient($coefficient) [line 253]`

***Function Parameters:***

- **\$coefficient**

**Set the coefficient of the key.**

- **Access public**

`void function RSAPrivateKey::setExponent1($exponent) [line 239]`

***Function Parameters:***

- **\$exponent**

**Set the exponent 1 of the key.**

- **Access public**

`void function RSAPrivateKey::setExponent2($exponent) [line 246]`

***Function Parameters:***

- **\$exponent**

**Set the exponent 2 of the key.**

- **Access public**

`void function RSAPrivateKey::setModulus($modulus) [line 204]`

***Function Parameters:***

- **\$modulus**

**Set the modulus of the key.**

- **Access** public

`void function RSAPrivateKey::setPrime1($prime) [line 225]`

***Function Parameters:***

- **\$prime**

**Set the prime 1 of the key.**

- **Access** public

`void function RSAPrivateKey::setPrime2($prime) [line 232]`

***Function Parameters:***

- **\$prime**

**Set the prime 2 of the key.**

- **Access** public

*void* function RSAPrivateKey::setPrivateExponent(\$exponent) [*line* [218](#)]

**Function Parameters:**

- **\$exponent**

**Set the private exponent of the key.**

- **Access** public

*void* function RSAPrivateKey::setPublicExponent(\$exponent) [*line* [211](#)]

**Function Parameters:**

- **\$exponent**

**Set the public exponent of the key.**

- **Access** public

## Class RSAPublicKey

[*line* [38](#)]

**Class to manage public keys.**

- **Package** phpPoA2
- **Sub-Package** crypto

#### RSAPublicKey::\$bits

*mixed* = [[line 44](#)]

- **Access** protected

#### RSAPublicKey::\$der

*mixed* = [[line 41](#)]

- **Access** protected

#### RSAPublicKey::\$exponent

*mixed* = [[line 43](#)]

- **Access** protected

#### RSAPublicKey::\$modulus

*mixed* = [[line 42](#)]

- **Access** protected

#### RSAPublicKey::\$pem

*mixed* = [line [40](#)]

- **Access** protected

Constructor *void* function RSAPublicKey::\_\_construct([\$pem = ""]) [line [49](#)]

**Function Parameters:**

- **\$pem**

**Build a new public key from its PEM representation.**

- **Access** public

*void* function RSAPublicKey::decode() [line [91](#)]

**Extract public key details from its DER representation.**

- **Access** protected

*void* function RSAPublicKey::encode() [line [121](#)]

**Build a new public key represented by it DER and PEM formats from its modulus and public exponent.**

- **Access** public

*void* function RSAPublicKey::fromDER(\$der) [line [58](#)]

**Function Parameters:**

- **\$der**

**Build the public key from its DER representation.**

- **Access** public

*void function RSAPublicKey::fromPEM(\$pem) [line [73](#)]*

**Function Parameters:**

- **\$pem**

**Build the public key from its PEM representation.**

- **Access** protected

*void function RSAPublicKey::getBits() [line [156](#)]*

**Get the length of the key in bits.**

- **Access** public

*void function RSAPublicKey::getDER() [line [135](#)]*

**Get the DER representation of the key.**



- **Access public**

*void function* RSAPublicKey::getExponent() [*line* [149](#)]

**Get the exponent of the key.**

- **Access public**

*void function* RSAPublicKey::getModulus() [*line* [142](#)]

**Get the modulus of the key.**

- **Access public**

*void function* RSAPublicKey::getPEM() [*line* [128](#)]

**Get the PEM representation of the key.**

- **Access public**

*void function* RSAPublicKey::setModulus(\$modulus) [*line* [163](#)]

***Function Parameters:***

- **\$modulus**

**Set the modulus of the key.**

- **Access public**

`void function RSAPublicKey::setPublicExponent($exponent) [line 170]`

***Function Parameters:***

- **\$exponent**

**Set the public exponent of the key.**

- **Access** public

# DBADB.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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 [178](#)]

#### **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 386](#)]

**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 460](#)]

**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 141](#)]

**Get a list of all authorized users.**

- **Access** public

*array* function InviteAuthzEngine::getPendingInvites() [[line 158](#)]

**Get a list of all pending invitations.**

- **Access** public

*boolean* function InviteAuthzEngine::invite(\$mail, [\$expires = 0]) [[line 325](#)]

**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* [365](#)]

**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* [291](#)]

**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](mailto: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

# OpenIDConfigurator.php

- **Package** phpPoA2
- **Author** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.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 OpenIDConfigurator

[line [31](#)]

**Configurator class for the OpenID Authentication Engine.**

- **Package** phpPoA2
- **Sub-Package** OpenIDAuthenticationEngine

**OpenIDConfigurator::\$mandatory\_options**

*mixed* = array('Providers') [[line 33](#)]

- **Access** protected

*string* function OpenIDConfigurator::getLoginURL() [[line 39](#)]

- **Access** public

*string* function OpenIDConfigurator::getLogoutURL() [[line 47](#)]

- **Access** public

*array* function OpenIDConfigurator::getProviders() [[line 62](#)]

- **Access** public

*boolean* function OpenIDConfigurator::isAutoLogin() [[line 54](#)]

- **Access** public

# OpenIDAuthnEngine.php

**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/>>.

- **Package** phpPoA2
- **Author** Miguel Macías < [miguel.macias@upv.es](mailto:miguel.macias@upv.es) >
- **Version** 2.0
- **Filesource** [Source Code for this file](#)
- **License** [GNU General Public License](#)

## Class OpenIDAuthnEngine

*[line 31]*

**Authentication engine for the OpenID protocol.**

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

- **Package** phpPoA2
- **Sub-Package** OpenIDAuthnEngine



### OpenIDAuthnEngine::\$attributes

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

- **Access** protected

### OpenIDAuthnEngine::\$lOpenID

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

- **Access** protected

### OpenIDAuthnEngine::\$reqAttributes

```
mixed = array (  
    'namePerson/friendly' => 'nickname',  
    'contact/email'      => 'email'  
) [line 38]
```

- **Access** protected

### OpenIDAuthnEngine::\$status

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

- **Access** protected

Constructor *void* function OpenIDAuthnEngine::\_\_construct(\$file, \$site) [line [43](#)]

#### **Function Parameters:**

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

- **Access** public

*void* function OpenIDAuthnEngine::authenticate() [*line* [60](#)]

- **Access** public

*void* function OpenIDAuthnEngine::configure(\$file, \$site) [*line* [56](#)]

***Function Parameters:***

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

- **Access** public

*void* function OpenIDAuthnEngine::getAttribute(\$name, [\$namespace = null]) [*line* [124](#)]

***Function Parameters:***

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

- **Access** public

*void* function OpenIDAuthnEngine::getAttributes() [*line* [120](#)]

- **Access** public

*void* function OpenIDAuthnEngine::isAuthenticated() [*line* [116](#)]

- **Access** public

*void* function OpenIDAuthnEngine::logout([\$slo = false]) [*line* [132](#)]

***Function Parameters:***

- **\$slo**

- **Access** public

# PAPIAuthnEngine.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](#)

ATTR\_SEPARATOR = "," *[line 29]*

**Default assertion delimiters for standard PAPI 1.5 protocol.**

NAMEVALUE\_SEPARATOR = "=" *[line 31]*

NS\_PAPI\_ATTRIBUTES = 'urn:mace:rediris.es:papi:attributes' *[line 44]*

NS\_PAPI\_PROTOCOL = 'urn:mace:rediris.es:papi:protocol' *[line 43]*

**Supported namespaces for attributes.**

PAPI\_ATTRIBUTE\_PARSER = "PAPI\_ATTRIBUTE\_PARSER" *[line 103]*

**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 proccessing 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 92]*

**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 36]*

**Supported database types.**

PAPI\_MYSQL = 'PAPIMySQLDB' *[line 37]*

PAPI\_REDIRECT\_URL\_FINISH = "PAPI\_REDIRECT\_URL\_FINISH" *[line 69]*

**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 80]*

**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 38]*

PROTO\_ATTR\_AS\_ID = '\_\_asid' *[line 51]*

PROTO\_ATTR\_EXPIRE\_TIME = '\_\_expiretime' *[line 53]*

PROTO\_ATTR\_KEY = '\_\_key' *[line 52]*

PROTO\_ATTR\_PREFIX = '\_papi\_' *[line 50]*

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

REQUEST\_LIFETIME = 300 *[line 58]*

**Default timeout for stored requests.**

```
VALUE_SEPARATOR = "|" [line 30]
```

# PAPIConfigurator.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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 111](#)]

**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 113](#)]

- **Access** protected

## PAPIAuthnEngine::\$as\_id

*mixed* = [[line 117](#)]

- **Access** protected

## PAPIAuthnEngine::\$attributes

*mixed* = [[line 116](#)]

- **Access** protected



### **PAPIAuthnEngine::\$cfg**

*mixed* = [line [124](#)]

- **Access** protected

### **PAPIAuthnEngine::\$clean\_url**

*mixed* = true [line [126](#)]

- **Access** protected

### **PAPIAuthnEngine::\$cookie\_name**

*mixed* = "PAPILcook\_" [line [128](#)]

- **Access** protected

### **PAPIAuthnEngine::\$crypto**

*mixed* = [line [125](#)]

- **Access** protected

### **PAPIAuthnEngine::\$db**

*mixed* = [line [122](#)]

- **Access** protected

### **PAPIAuthnEngine::\$enforcing**

*mixed* = true [[line 129](#)]

- **Access** protected

### **PAPIAuthnEngine::\$expiration\_time**

*mixed* = false [[line 115](#)]

- **Access** protected

### **PAPIAuthnEngine::\$global\_expire\_time**

*mixed* = [[line 121](#)]

- **Access** protected

### **PAPIAuthnEngine::\$id**

*mixed* = [[line 123](#)]

- **Access** protected

### **PAPIAuthnEngine::\$key**

*mixed* = [[line 118](#)]

- **Access** protected

#### **PAPIAuthnEngine::\$lkey**

*mixed* = [line [119](#)]

- **Access** protected

#### **PAPIAuthnEngine::\$opoa**

*mixed* = "http" [line [130](#)]

- **Access** protected

#### **PAPIAuthnEngine::\$pkey**

*mixed* = [line [120](#)]

- **Access** protected

#### **PAPIAuthnEngine::\$skip\_redirection**

*mixed* = false [line [127](#)]

- **Access** protected

#### **PAPIAuthnEngine::\$status**

*mixed* = [line [114](#)]

- **Access** protected

#### **PAPIAuthnEngine::\$valid\_hooks**

```
mixed = array(PAPI_REDIRECT_URL_FINISH,  
              PAPI_RESTORE_ORIGINAL_REQUEST,  
              PAPI_CLEAN_URL,  
              PAPI_ATTRIBUTE_PARSER) [line 131]
```

- **Access** protected

*void* function PAPIAuthnEngine::attributeQuery(\$name) [*line 401*]

#### **Function Parameters:**

- **\$name**

- **Access** protected

*void* function PAPIAuthnEngine::authenticate() [*line 168*]

- **Access** public

*void* function PAPIAuthnEngine::configure(\$file, \$section) [*line 136*]

#### **Function Parameters:**

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

- **Access** public

*true* function PAPIAuthnEngine::deleteCookie() [[line 456](#)]

**Delete the current cookie, if any.**

- **Access** protected

*boolean* function PAPIAuthnEngine::deleteRequest(\$key) [[line 738](#)]

**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 336](#)]

**Function Parameters:**

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

- **Access** public

*void* function PAPIAuthnEngine::getAttributes() [[line 308](#)]

- **Access** public

*string* function PAPIAuthnEngine::getNewCookie() [[line 763](#)]

**Generate a new cookie for the current user.**

- **Access** protected

*string* function PAPIAuthnEngine::getRedirectLocation() [[line 597](#)]

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

- **Access** protected

*string* function PAPIAuthnEngine::getSingleLogoutLocation() [[line 557](#)]

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

- **Access** protected

*string* function PAPIAuthnEngine::getSingleLogoutResponseLocation() [[line 581](#)]

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

- **Access** protected

*void* function PAPIAuthnEngine::isAuthenticated() [[line 298](#)]

- **Access** public

*boolean* function PAPIAuthnEngine::isSafe() [[line 775](#)]

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

- **Access** protected

*hash* function PAPIAuthnEngine::loadRequest(\$key) [[line 694](#)]

**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 366](#)]

**Function Parameters:**

- **\$slo**

- **Access** public

`void` function PAPIAuthnEngine::redirect([\$location = ""]) [[line 537](#)]

**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 666](#)]

**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 413](#)]

**Function Parameters:**

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

**Check if a cookie is valid.**



- **Access** protected

*boolean* function PAPIAuthnEngine::testResponse(\$data, \$pubkey) [*line* [478](#)]

**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

### PAPIMySQLDB::\$purge\_sql

*mixed = [line 37]*



- **Access** protected

Constructor *void* function PAPIMySQLDB::\_\_construct(\$cfg) [*line 39*]

***Function Parameters:***

- **\$cfg**

- **Access** public

*void* function PAPIMySQLDB::fetch(\$key) [*line 78*]

***Function Parameters:***

- **\$key**

- **Access** public

*void* function PAPIMySQLDB::purge(\$gap) [*line 95*]

***Function Parameters:***

- **\$gap**

- **Access** public

*void* function PAPIMySQLDB::replaceContents(\$key, \$get, \$post, \$request, \$query, \$method, \$input, \$hli) [*line 55*]

***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::purge(\$gap) [[line 54](#)]

***Function Parameters:***

- **\$gap**

- **Access** public

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

***Function Parameters:***

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

- **\$input**
- **\$hli**

- **Access** public

# QueryFilterAuthzEngine.php

- **Package** phpPoA2
- **Author** Jaime Perez < [jaime.perez@rediris.es](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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](mailto: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

This is a sample configuration file.

- **Package** phpPoA2
- **Sub-Package** SourceIPAddressAuthorizationEngine

## Appendix A - Class Trees

Package phpPoA2

### ASN1

- [ASN1](#)

### AttributeHandler

- [AttributeHandler](#)

### AuthorizedDB

- [AuthorizedDB](#)

### Example

- [Example](#)

### GenericConfigurator

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

## GenericDB

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

## GenericEngine

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



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

## Hook

- [Hook](#)

## InviteDB

- [InviteDB](#)

## LightOpenID

- [LightOpenID](#)

## PAPICrypt

- [PAPICrypt](#)

## PAPIDB

- [PAPIDB](#)

## PoA

- [PoA](#)

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

## PoAEventHandler

- [PoAEventHandler](#)

## PoAException

- Exception
  - [PoAException](#)

## PoALog

- [PoALog](#)

## PoAUtils

- [PoAUtils](#)

## RSAPrivateKey

- [RSAPrivateKey](#)

## RSAPublicKey

- [RSAPublicKey](#)

# INSTALL

## REQUISITES:

- \* PHP >= 4.X
- \* 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.php file in your include path, or set the PHP environment variable "poa\_conf\_file" to point to your PoA.conf.php file.
5. Configure the PoA.conf.php 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. Please make sure all the configuration files are outside the web server root directory.
7. 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
doc/api	API documentation for developers
doc/tutorials	User tutorials to demonstrate the use of the library
lib	Auxiliary classes that implement most of the functionality that is shipped with this distribution
lib/authn	Authentication engines
lib/authz	Authorization engines
lib/crypto	Cryptographic classes
lib/db	Database classes
lib/third-party	Third party libraries
messages	Support files for message internationalization
samples	Sample usage files, including configuration
tools	Useful tools that ship with the library

PoA.php	The main class of phpPoA2. Include only this file in your applications
AutoPoA.php	PoA class that automatically redirects on error
LitePoA.php	PoA class with backwards compatibility
PoAConfigurator.php	The main configuration class
PoAEventHandler.php	The main event handler used by phppoA2
PoAException.php	Generic phpPoA exception implementation
PoALog.php	Logging class
PoAUtils.php	Utility class
definitions.php	Useful constants

# Index

## A

<a href="#">ASN1::int2bin()</a>	0
<i>Encode an integer in binary representation.</i>	
<a href="#">ASN1::getValues()</a>	0
<i>Get the sequence of values stored in this ASN.1.</i>	
<a href="#">ASN1::getValue()</a>	0
<i>Get the binary value of this ASN.1.</i>	
<a href="#">ASN1::readByte()</a>	0
<i>Read first byte from the buffer and move the internal pointer.</i>	
<a href="#">ASN1::readBytes()</a>	0
<i>Read n bytes from a buffer and mov the internal pointer.</i>	
<a href="#">ASN1::setValues()</a>	0
<i>Set the value of this ASN.1 to be a sequence of values.</i>	
<a href="#">ASN1::setValue()</a>	0
<i>Set the binary value of this ASN.1.</i>	
<a href="#">ASN1::setInteger()</a>	0
<i>Set the value of this ASN.1 from an integer.</i>	
<a href="#">ASN1::getTag()</a>	0
<i>Get the tag of this ASN.1.</i>	
<a href="#">ASN1::getInteger()</a>	0
<i>Get the integer value of this ASN.1.</i>	
<a href="#">ASN1</a>	0
<i>This class implements ASN.1 encoding.</i>	
<a href="#">ASN1.php</a>	0
<a href="#">AttributeFilterConfigurator::getDeniedAttributes()</a>	0
<i>Returns an array of attributes that would be denied if their values match any of the patterns for each attribute.</i>	
<a href="#">ASN1::\$tag</a>	0
<a href="#">ASN1::\$value</a>	0
<a href="#">ASN1::encode()</a>	0
<i>Encode this object into ASN.1.</i>	
<a href="#">ASN1::decode()</a>	0
<i>Decode an object in ASN.1 format.</i>	
<a href="#">ASN1::bin2int()</a>	0
<i>Decode an integer from its binary representation.</i>	
<a href="#">ASN1::writeByte()</a>	0
<i>Write a byte to a buffer.</i>	
<a href="#">ASN1::writeBytes()</a>	0
<i>Write bytes to a buffer.</i>	
<a href="#">AuthorizedDBADB::fetch_all()</a>	0
<a href="#">AuthorizedDBADB::fetch()</a>	0
<a href="#">AuthorizedDBADB::\$mandatory_options</a>	0
<a href="#">AuthorizedDBADB::open()</a>	0
<a href="#">AuthorizedDBADB::replace_authorization()</a>	0
<a href="#">AttributeHandler::handle()</a>	0

<a href="#">AttributeHandler</a>	0
<a href="#">ATTR_SEPARATOR</a>	0
<i>Default assertion delimiters for standard PAPI 1.5 protocol.</i>	
<a href="#">AuthorizedDBADB::\$authorized</a>	0
<a href="#">AuthorizedDBADB</a>	0
<i>Authorization lists binding to a Berkeley DB backend.</i>	
<a href="#">AUTHORIZED_DBA</a>	0
<a href="#">AuthorizedDBADB.php</a>	0
<a href="#">AuthorizedDB.php</a>	0
<a href="#">AUTHORIZED_MYSQL</a>	0
<a href="#">AUTHORIZED_SESSION</a>	0
<a href="#">AuthorizedDB::replace_authorization()</a>	0
<a href="#">AuthorizedDB</a>	0
<i>Generic interface for a PAPI request database.</i>	
<a href="#">AttributeFilterConfigurator::getAllowedAttributes()</a>	0
<i>Returns an array of attributes that would be allowed if their values match any of the patterns for each attribute.</i>	
<a href="#">AttributeFilterConfigurator::\$mandatory_options</a>	0
<a href="#">AuthenticationEngine::getAttributes()</a>	0
<i>Get an associative array with the set of common attributes for the current user. May trigger some attribute recollection.</i>	
<a href="#">AuthenticationEngine::getAttribute()</a>	0
<i>Get an specific attribute for the current user, by specifying its name and namespace. May trigger some attribute query procedure.</i>	
<a href="#">AuthenticationEngine::authenticate()</a>	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>	
<a href="#">AuthenticationEngine::isAuthenticated()</a>	0
<i>Check the current authentication status withing this context. Does not trigger any authentication procedure.</i>	
<a href="#">AuthenticationEngine::logout()</a>	0
<i>Perform a logout (locally or global).</i>	
<a href="#">AuthorizationEngine</a>	0
<a href="#">AuthorizationConfigurator::getDefaultBehaviour()</a>	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>	
<a href="#">AuthorizationConfigurator</a>	0
<a href="#">AuthenticationEngine::\$engine_type</a>	0
<a href="#">AuthenticationEngine</a>	0
<i>Abstract authentication engine.</i>	
<a href="#">AUTHN_SUCCESS</a>	0
<i>Authentication succeeded.</i>	
<a href="#">AUTHN_FAILED</a>	0
<i>Authentication failed.</i>	
<a href="#">AS_T</a>	0
<i>'AS' type redirection in the PAPI 1.5 protocol.</i>	
<a href="#">AUTHZ_FAILED</a>	0
<i>Authorization failed.</i>	
<a href="#">AUTHZ_SUCCESS</a>	0
<i>Authorization succeeded.</i>	
<a href="#">authz_convert.php</a>	0
<a href="#">AuthenticationEngine.php</a>	0

<a href="#">AuthorizationEngine::\$engine_type</a>	0
<a href="#">AuthorizationEngine::authorize()</a>	0
<a href="#">AttributeFilterAuthzEngine::\$valid_hooks</a>	0
<a href="#">AttributeFilterAuthzEngine</a>	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>	
<a href="#">attrHook()</a>	0
<a href="#">AttributeFilterAuthzEngine::authorize()</a>	0
<a href="#">AttributeFilterAuthzEngine::getAuthorizedList()</a>	0
<a href="#">AttributeFilterConfigurator</a>	0
<i>Configurator for the Attribute Filter Authorization Engine.</i>	
<a href="#">AttributeFilterAuthzEngine::revoke()</a>	0
<a href="#">AttributeFilterAuthzEngine::isAuthorized()</a>	0
<a href="#">AttributeFilterConfigurator.php</a>	0
<a href="#">ATTRIBUTE BEFORE FILTERS</a>	0
<i>This hook is executed right after retrieving the arrays of allowed and denied attributes that will be checked immediately.</i>	
<a href="#">AuthorizationEngine::revoke()</a>	0
<a href="#">AuthorizationEngine::isAuthorized()</a>	0
<i>Check authorization for the specified user.</i>	
<a href="#">AuthorizationEngine::getAuthorizedList()</a>	0
<a href="#">AutoPoA</a>	0
<i>Class that automatically redirects to error pages if any error is detected.</i>	
<a href="#">AutoPoA::authenticate()</a>	0
<a href="#">AttributeFilterAuthzEngine.php</a>	0
<a href="#">AutoPoA::isAuthorized()</a>	0
<a href="#">AutoPoA.php</a>	0

## C

<a href="#">constructor RSAPrivateKey::__construct()</a>	0
<i>Build a new private key from its PEM representation.</i>	
<a href="#">constructor ASN1::__construct()</a>	0
<i>Build a new ASN1 from its tag and value.</i>	
<a href="#">constructor PoALog::__construct()</a>	0
<i>Build a new logger.</i>	
<a href="#">constructor RSAPublicKey::__construct()</a>	0
<i>Build a new public key from its PEM representation.</i>	
<a href="#">constructor GenericDB::__construct()</a>	0
<i>Main constructor.</i>	
<a href="#">constructor PAPIMySQLDB::__construct()</a>	0
<a href="#">constructor OpenIDAuthnEngine::__construct()</a>	0
<a href="#">constructor PoAException::__construct()</a>	0
<a href="#">constructor PoAEventHandler::__construct()</a>	0
<i>Main constructor.</i>	
<a href="#">constructor Hook::__construct()</a>	0
<i>Build a new hook.</i>	
<a href="#">constructor GenericEngine::__construct()</a>	0
<i>Main constructor for the engine.</i>	
<a href="#">constructor GenericConfigurator::__construct()</a>	0
<i>Main constructor.</i>	

<a href="#">constructor LightOpenID::__construct()</a>	0
<a href="#">constructor PAPICrypt::__construct()</a>	0
<a href="#">constructor PoAConfigurator::__construct()</a>	0
<a href="#">constructor PoA::__construct()</a>	0
<i>Main constructor. Configures the PoA and performs initialization.</i>	
<a href="#">CONFIG_ERR</a>	0
<i>A configuration related error.</i>	

## D

<a href="#">DBADB::fetch()</a>	0
<a href="#">DBADB::delete()</a>	0
<a href="#">DBADB::fetch_all()</a>	0
<a href="#">DBADB::open()</a>	0
<a href="#">DBADB::replace()</a>	0
<a href="#">DBADB::configure()</a>	0
<a href="#">DBADB::close()</a>	0
<a href="#">DBADB.php</a>	0
<a href="#">DBADB</a>	0
<i>Berkeley DB backend.</i>	
<a href="#">DBADB::\$mandatory_options</a>	0
<a href="#">DBADB::check()</a>	0
<a href="#">definitions.php</a>	0

## E

<a href="#">Example::hook()</a>	0
<a href="#">Example</a>	0

## G

<a href="#">GenericMySQLDB::\$create_sql</a>	0
<a href="#">GenericMySQLDB</a>	0
<i>Generic MySQL database class.</i>	
<a href="#">GenericMySQLDB::\$delete_sql</a>	0
<a href="#">GenericMySQLDB::\$insert_sql</a>	0
<a href="#">GenericMySQLDB::\$mandatory_options</a>	0
<a href="#">GenericDB::replace()</a>	0
<i>Replaces the specified key with a new value. If the key does not exist previously, it will be created.</i>	
<a href="#">GenericDB::open()</a>	0
<i>Opens the database.</i>	
<a href="#">GenericDB::configure()</a>	0
<i>Configure the database handler.</i>	
<a href="#">GenericDB::close()</a>	0
<i>Closes the database.</i>	
<a href="#">GenericDB::delete()</a>	0
<i>Removes the specified key from the database.</i>	
<a href="#">GenericDB::fetch()</a>	0
<i>Gets the value of the specified key.</i>	



<a href="#">GenericDB::fetch_all()</a>	0
<i>Gets all the contents stored in the database.</i>	
<a href="#">GenericMySQLDB::\$prefix</a>	0
<a href="#">GenericMySQLDB::\$search_all_sql</a>	0
<a href="#">GenericMySQLDB::getError()</a>	0
<a href="#">GenericMySQLDB::fetch_all()</a>	0
<a href="#">GenericMySQLDB::open()</a>	0
<a href="#">GenericMySQLDB::replace()</a>	0
<a href="#">global \$messages</a>	0
<i>An array with all the internationalized messages used by phpPoA2.</i>	
<a href="#">GenericMySQLDB::fetch()</a>	0
<a href="#">GenericMySQLDB::delete()</a>	0
<a href="#">GenericMySQLDB::\$update_sql</a>	0
<a href="#">GenericMySQLDB::\$search_sql</a>	0
<a href="#">GenericMySQLDB::check()</a>	0
<a href="#">GenericMySQLDB::close()</a>	0
<a href="#">GenericMySQLDB::configure()</a>	0
<a href="#">GenericDB::check()</a>	0
<i>Checks if the specified key exists in the database.</i>	
<a href="#">GenericDB::\$mandatory_options</a>	0
<a href="#">GenericEngine::\$cfg</a>	0
<a href="#">GenericEngine</a>	0
<i>A generic engine class.</i>	
<a href="#">GenericEngine::\$engine_type</a>	0
<a href="#">GenericEngine::\$handler</a>	0
<a href="#">GenericEngine::\$hooks</a>	0
<a href="#">GenericConfigurator::validate()</a>	0
<i>Check all mandatory attributes are set.</i>	
<a href="#">GenericConfigurator::configure()</a>	0
<i>Read the configuration from the specified file and section.</i>	
<a href="#">GenericEngine.php</a>	0
<a href="#">GenericConfigurator.php</a>	0
<a href="#">GenericConfigurator</a>	0
<i>Generic configuration class.</i>	
<a href="#">GenericConfigurator::\$cfg</a>	0
<a href="#">GenericConfigurator::\$mandatory_options</a>	0
<a href="#">GenericEngine::\$valid_hooks</a>	0
<a href="#">GenericEngine::addHook()</a>	0
<i>Adds a function to the specified hook, which will be executed at some point of the code.</i>	
<a href="#">GenericDB</a>	0
<i>Provides the basic operations and defines the interface of any DB operating class.</i>	
<a href="#">GenericMySQLDB.php</a>	0
<a href="#">GenericDB::\$cfg</a>	0
<a href="#">GenericDB::\$db</a>	0
<a href="#">GenericDB::\$error</a>	0
<a href="#">GenericDB.php</a>	0
<a href="#">GenericEngine::setHandler()</a>	0
<i>Set the event handler to the one specified.</i>	
<a href="#">GenericEngine::configure()</a>	0
<i>Configure the engine.</i>	
<a href="#">GenericEngine::clean()</a>	0
<i>Unregister error and exception handlers. Use it only for methods not declared in the interface that previously called registerHandler() method.</i>	

<a href="#">GenericEngine::registerHandler()</a>	0
<i>Register error and exception handlers for logging. Use it only for methods not declared in the interface that could trigger errors.</i>	
<a href="#">GenericEngine::removeHook()</a>	0
<i>Removes a function from the specified hook.</i>	
<a href="#">GenericEngine::runHooks()</a>	0
<i>Run all hooks attached to an specific action.</i>	
<a href="#">GPOA_T</a>	0
<i>'GPoA' type redirection in the PAPI 1.5 protocol.</i>	

## H

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

## I

<a href="#">InviteConfigurator::getInvitesDBName()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getInvitesDBHost()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getInvitesDBPassword()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getInvitesDBType()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getInviteSubject()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getInvitesDBUser()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getInvitesDBFile()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getEmailVerifyRegEx()</a>	0
<a href="#">InviteConfigurator::getAuthorizedDBType()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getAuthorizedDBPassword()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getAuthorizedDBUser()</a>	0
<i>Returns</i>	
<a href="#">InviteConfigurator::getEmailVerifyAlgRegEx()</a>	0
<a href="#">InviteConfigurator::getEmailVerifyAttribute()</a>	0
<a href="#">InviteConfigurator::getInviteText()</a>	0

Returns	
<a href="#">InviteConfigurator::getInviteURL()</a>	0
Returns	
<a href="#">InviteDBADB::open()</a>	0
<a href="#">InviteDBADB::fetch_all()</a>	0
<a href="#">InviteDBADB::replace_invite()</a>	0
<a href="#">ipHook()</a>	0
<a href="#">INSTALL</a>	0
<a href="#">IPFilters.conf.php</a>	0
This is a sample configuration file.	
<a href="#">InviteDBADB::fetch()</a>	0
<a href="#">InviteDBADB::\$mandatory_options</a>	0
<a href="#">InviteDB</a>	0
Generic interface for a database of invites.	
<a href="#">InviteConfigurator::getUniqueAttributes()</a>	0
Returns	
<a href="#">InviteDB::replace_invite()</a>	0
<a href="#">InviteDBADB</a>	0
Invites binding to a Berkeley DB backend.	
<a href="#">InviteDBADB::\$invite</a>	0
<a href="#">InviteConfigurator::getAuthorizedDBName()</a>	0
Returns	
<a href="#">InviteConfigurator::getAuthorizedDBHost()</a>	0
Returns	
<a href="#">InviteDBADB.php</a>	0
<a href="#">InviteDB.php</a>	0
<a href="#">InviteAuthzEngine</a>	0
Invitation based authorization engine.	
<a href="#">InviteAuthzEngine::\$authz_db</a>	0
<a href="#">InviteAuthzEngine::\$valid_hooks</a>	0
<a href="#">InviteAuthzEngine::\$invites_db</a>	0
<a href="#">InviteConfigurator.php</a>	0
<a href="#">INVITES_SESSION</a>	0
<a href="#">InviteAuthzEngine.php</a>	0
<a href="#">invite_convert.php</a>	0
<a href="#">INVITES_BEFORE_AUTHORIZATION</a>	0
This hook is executed right before it is checked whether the user was invited or not to continue with authorization.	
<a href="#">INVITES_DBA</a>	0
Supported database types.	
<a href="#">INVITES_MYSQL</a>	0
<a href="#">InviteAuthzEngine::authorize()</a>	0
Authorize the specified user.	
<a href="#">InviteAuthzEngine::configure()</a>	0
Configure the authorization engine.	
<a href="#">InviteConfigurator</a>	0
Configurator for the Invitation based authorization engine.	
<a href="#">InviteAuthzEngine::revoke()</a>	0
Revoke authorization for the specified user.	
<a href="#">InviteConfigurator::\$mandatory_options</a>	0
<a href="#">InviteConfigurator::doEmailVerify()</a>	0
<a href="#">InviteConfigurator::getAuthorizedDBFile()</a>	0
Returns	

<a href="#">InviteConfigurator::getAdminEmail()</a>	0
<i>Returns</i>	
<a href="#">InviteAuthzEngine::removeInvite()</a>	0
<i>Remove an invitation from the database.</i>	
<a href="#">InviteAuthzEngine::isAuthorized()</a>	0
<i>Check if the specified user is authorized with the given attributes.</i>	
<a href="#">InviteAuthzEngine::finish()</a>	0
<i>Close databases and return the same value received as a parameter.</i>	
<a href="#">InviteAuthzEngine::emailVerify()</a>	0
<i>Perform e-mail verification for the current user according to the configuration for this site.</i>	
<a href="#">InviteAuthzEngine::getAuthorizedList()</a>	0
<i>Get a list of all authorized users.</i>	
<a href="#">InviteAuthzEngine::getPendingInvites()</a>	0
<i>Get a list of all pending invitations.</i>	
<a href="#">InviteAuthzEngine::invite()</a>	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>	
<a href="#">INVITE_ERR</a>	0
<i>An invitation related error.</i>	

## L

<a href="#">LightOpenID::getSregAttributes()</a>	0
<a href="#">LightOpenID::hostExists()</a>	0
<i>Checks if the server specified in the url exists.</i>	
<a href="#">LightOpenID::htmlTag()</a>	0
<i>Helper function used to scan for &lt;meta&gt;/&lt;link&gt; tags and extract information</i>	
<a href="#">LightOpenID::request()</a>	0
<a href="#">LightOpenID::getAxAttributes()</a>	0
<a href="#">LightOpenID::getAttributes()</a>	0
<i>Gets AX/SREG attributes provided by OP. should be used only after successful validation.</i>	
<a href="#">LightOpenID::axParams()</a>	0
<a href="#">LightOpenID::build_url()</a>	0
<a href="#">LightOpenID::discover()</a>	0
<i>Performs Yadis and HTML discovery. Normally not used.</i>	
<a href="#">LightOpenID::request_curl()</a>	0
<a href="#">LightOpenID::request_streams()</a>	0
<a href="#">LitePoA::\$handler</a>	0
<a href="#">LitePoA::checkAccess()</a>	0
<i>A shortcut for the whole authentication and authorization process.</i>	
<a href="#">LitePoA::setIDBuilder()</a>	0
<i>Set an attribute handler to build identifiers for the users according to their attributes. Method must receive a hash of attributes ("name" =&gt; "value").</i>	
<a href="#">LitePoA</a>	0
<i>Class that embeds all functionality (authentication, attribute collection and authorization) in one method, just like older versions of phpPoA.</i>	
<a href="#">LightOpenID::set()</a>	0
<a href="#">LightOpenID::sregParams()</a>	0
<a href="#">LightOpenID::validate()</a>	0

*Performs OpenID verification with the OP.*

<a href="#">LightOpenID::get()</a>	0
<a href="#">LightOpenID::authUrl_v2()</a>	0
<a href="#">LightOpenID::authUrl_v1()</a>	0
<a href="#">LightOpenID::\$capath</a>	0
<a href="#">LightOpenID::\$data</a>	0
<a href="#">LightOpenID::\$identifier_select</a>	0
<a href="#">LightOpenID::\$cainfo</a>	0
<a href="#">LightOpenID::\$ax_to_sreg</a>	0
<a href="#">LightOpenID</a>	0

*This class provides a simple interface for OpenID (1.1 and 2.0) authentication.*

<a href="#">LightOpenID::\$aliases</a>	0
<a href="#">LightOpenID::\$ax</a>	0
<a href="#">LightOpenID::\$optional</a>	0
<a href="#">LightOpenID::\$required</a>	0
<a href="#">LightOpenID::\$verify_peer</a>	0
<a href="#">LightOpenID::\$version</a>	0
<a href="#">LightOpenID::authUrl()</a>	0

*Returns authentication url. Usually, you want to redirect your user to it.*

<a href="#">LightOpenID::\$trustRoot</a>	0
<a href="#">LightOpenID::\$sreg</a>	0
<a href="#">LightOpenID::\$returnUrl</a>	0
<a href="#">LightOpenID::\$server</a>	0
<a href="#">LightOpenID::\$setup_url</a>	0
<a href="#">LitePoA.php</a>	0

## M

<a href="#">messages-es_ES.php</a>	0
<a href="#">messages-en_EN.php</a>	0

## N

<a href="#">NS_SAML2_PROTOCOL</a>	0
<i>Supported namespaces for attributes.</i>	
<a href="#">NS_PAPI_PROTOCOL</a>	0
<i>Supported namespaces for attributes.</i>	
<a href="#">NS_PAPI_ATTRIBUTES</a>	0
<a href="#">NAMEVALUE_SEPARATOR</a>	0
<a href="#">NOAUTH_ERR</a>	0
<i>Authentication failed error.</i>	

## O

<a href="#">OpenIDAuthnEngine::\$status</a>	0
<a href="#">OpenIDAuthnEngine::\$reqAttributes</a>	0
<a href="#">OpenIDAuthnEngine::\$lOpenID</a>	0
<a href="#">OpenIDAuthnEngine::\$attributes</a>	0
<a href="#">OpenIDAuthnEngine::authenticate()</a>	0
<a href="#">OpenIDAuthnEngine::configure()</a>	0

<a href="#">OpenIDAuthnEngine::logout()</a>	0
<a href="#">OpenIDAuthnEngine::isAuthenticated()</a>	0
<a href="#">OpenIDAuthnEngine::getAttributes()</a>	0
<a href="#">OpenIDAuthnEngine::getAttribute()</a>	0
<a href="#">OpenIDAuthnEngine</a>	0
<i>Authentication engine for the OpenID protocol.</i>	
<a href="#">OpenIDAuthnEngine.php</a>	0
<i>This file is part of phpPoA2.</i>	
<a href="#">OpenIDConfigurator.php</a>	0
<a href="#">openid.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">openid-restricted.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">openid-ext.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">OpenIDConfigurator</a>	0
<i>Configurator class for the OpenID Authentication Engine.</i>	
<a href="#">OpenIDConfigurator::\$mandatory_options</a>	0
<a href="#">OpenIDConfigurator::isAutoLogin()</a>	0
<a href="#">OpenIDConfigurator::getProviders()</a>	0
<a href="#">OpenIDConfigurator::getLogoutURL()</a>	0
<a href="#">OpenIDConfigurator::getLoginURL()</a>	0
<a href="#">openid-auto.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	

## P

<a href="#">PAPIAuthnEngine::attributeQuery()</a>	0
<a href="#">PAPIAuthnEngine::authenticate()</a>	0
<a href="#">PAPIAuthnEngine::configure()</a>	0
<a href="#">PAPIAuthnEngine::deleteCookie()</a>	0
<i>Delete the current cookie, if any.</i>	
<a href="#">PAPIAuthnEngine::\$valid_hooks</a>	0
<a href="#">PAPIAuthnEngine::\$status</a>	0
<a href="#">PAPIAuthnEngine::\$lkey</a>	0
<a href="#">PAPIAuthnEngine::\$opoa</a>	0
<a href="#">PAPIAuthnEngine::\$pkey</a>	0
<a href="#">PAPIAuthnEngine::\$skip_redirection</a>	0
<a href="#">PAPIAuthnEngine::deleteRequest()</a>	0
<i>Delete a request from the request database.</i>	
<a href="#">PAPIAuthnEngine::getAttribute()</a>	0
<a href="#">PAPIAuthnEngine::isSafe()</a>	0
<i>Determines if it's safe to assume the user as authenticated.</i>	
<a href="#">PAPIAuthnEngine::loadRequest()</a>	0
<i>Load a request from the request database.</i>	
<a href="#">PAPIAuthnEngine::logout()</a>	0
<a href="#">PAPIAuthnEngine::redirect()</a>	0
<i>Redirect user browser to the appropriate URL for authentication.</i>	
<a href="#">PAPIAuthnEngine::isAuthenticated()</a>	0
<a href="#">PAPIAuthnEngine::getSingleLogoutResponseLocation()</a>	0
<i>Retrieve the URL where to redirect a user once he has successfully logged out.</i>	
<a href="#">PAPIAuthnEngine::getAttributes()</a>	0

<a href="#">PAPIAuthnEngine::getNewCookie()</a>	0
<i>Generate a new cookie for the current user.</i>	
<a href="#">PAPIAuthnEngine::getRedirectLocation()</a>	0
<i>Retrieve the URL where to redirect a user and store his request.</i>	
<a href="#">PAPIAuthnEngine::getSingleLogoutLocation()</a>	0
<i>Retrieve the URL where to redirect a user to perform a single logout.</i>	
<a href="#">PAPIAuthnEngine::\$key</a>	0
<a href="#">PAPIAuthnEngine::\$id</a>	0
<a href="#">papi1.5-authn-attr.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">papi1.5-customattr.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">papi1.5-papiopoa-hook.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">PAPIAuthnEngine</a>	0
<i>Authentication engine for the PAPI 1.5 protocol.</i>	
<a href="#">PAPISessionDB.php</a>	0
<a href="#">PAPI_DB_TABLE</a>	0
<a href="#">PAPIConfigurator.php</a>	0
<a href="#">PAPIDB.php</a>	0
<a href="#">PAPIDBADB.php</a>	0
<a href="#">PAPIMySQLDB.php</a>	0
<a href="#">PAPIAuthnEngine::\$assertion</a>	0
<a href="#">PAPIAuthnEngine::\$as_id</a>	0
<a href="#">PAPIAuthnEngine::\$db</a>	0
<a href="#">PAPIAuthnEngine::\$enforcing</a>	0
<a href="#">PAPIAuthnEngine::\$expiration_time</a>	0
<a href="#">PAPIAuthnEngine::\$global_expire_time</a>	0
<a href="#">PAPIAuthnEngine::\$crypto</a>	0
<a href="#">PAPIAuthnEngine::\$cookie_name</a>	0
<a href="#">PAPIAuthnEngine::\$attributes</a>	0
<a href="#">PAPIAuthnEngine::\$cfg</a>	0
<a href="#">PAPIAuthnEngine::\$clean_url</a>	0
<a href="#">PAPIAuthnEngine::saveRequest()</a>	0
<i>Save a request to the request database. The request includes: \$_REQUEST, \$_GET, \$_POST, \$_SERVER['QUERY_STRING'], \$_SERVER['REQUEST_METHOD'] and php://input.</i>	
<a href="#">PAPIAuthnEngine::testCookie()</a>	0
<i>Check if a cookie is valid.</i>	
<a href="#">PAPIDBADB::replaceContents()</a>	0
<a href="#">PAPIMySQLDB</a>	0
<i>MySQL database frontend for the PAPI Authentication Engine.</i>	
<a href="#">PAPIMySQLDB::\$context</a>	0
<a href="#">PAPIMySQLDB::\$prefix</a>	0
<a href="#">PAPIDBADB::purge()</a>	0
<a href="#">PAPIDBADB::fetch()</a>	0
<a href="#">PAPIDB::purge()</a>	0
<i>Purge the database of outdated requests.</i>	
<a href="#">PAPIDB::replaceContents()</a>	0
<a href="#">PAPIDBADB</a>	0
<i>PAPI binding to a Berkeley DB backend.</i>	
<a href="#">PAPIDBADB::\$context</a>	0
<a href="#">PAPIMySQLDB::\$purge_sql</a>	0

<a href="#">PAPIMySQLDB::fetch()</a>	0
<a href="#">PAPISessionDB::purge()</a>	0
<a href="#">PAPISessionDB::replaceContents()</a>	0
<a href="#">papi1.5-query-authz.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">papi1.5-ip-authz.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">PAPISessionDB::fetch()</a>	0
<a href="#">PAPISessionDB::\$context</a>	0
<a href="#">PAPIMySQLDB::purge()</a>	0
<a href="#">PAPIMySQLDB::replaceContents()</a>	0
<a href="#">PAPISessionDB</a>	0
<i>PAPI binding to a PHP Session DB backend.</i>	
<a href="#">PAPIDB</a>	0
<i>Generic interface for a PAPI request database.</i>	
<a href="#">PAPIConfigurator::getRequestLifetime()</a>	0
<a href="#">PAPIConfigurator::getDBHost()</a>	0
<a href="#">PAPIConfigurator::getDBName()</a>	0
<a href="#">PAPIConfigurator::getDBPassword()</a>	0
<a href="#">PAPIConfigurator::getDBPrefix()</a>	0
<a href="#">PAPIConfigurator::getDBFile()</a>	0
<a href="#">PAPIConfigurator::getCookieTimeout()</a>	0
<a href="#">PAPIAuthnEngine::testResponse()</a>	0
<i>Check the response from the AS/GPoA.</i>	
<a href="#">PAPIConfigurator</a>	0
<i>Configurator class for the PAPI Authentication Engine.</i>	
<a href="#">PAPIConfigurator::\$mandatory_options</a>	0
<a href="#">PAPIConfigurator::getCookieDomain()</a>	0
<a href="#">PAPIConfigurator::getDBType()</a>	0
<a href="#">PAPIConfigurator::getDBUser()</a>	0
<a href="#">PAPIConfigurator::getLogoutURL()</a>	0
<a href="#">PAPIConfigurator::getPubKeyFile()</a>	0
<a href="#">PAPIConfigurator::getRedirectType()</a>	0
<a href="#">PAPIConfigurator::getRedirectURL()</a>	0
<a href="#">PAPIConfigurator::getLocation()</a>	0
<a href="#">PAPIConfigurator::getLKey()</a>	0
<a href="#">PAPIConfigurator::getFriendlyName()</a>	0
<a href="#">PAPIConfigurator::getHomeLocatorID()</a>	0
<a href="#">PAPIConfigurator::getID()</a>	0
<a href="#">PROTO_ATTR_PREFIX</a>	0
<i>Prefix for operational attributes inside the protocol and special attributes names.</i>	
<a href="#">PROTO_ATTR_KEY</a>	0
<a href="#">PoA::\$authz_engines</a>	0
<a href="#">PoA::\$autoload</a>	0
<a href="#">PoA::\$cfg</a>	0
<a href="#">PoA::\$db_manager</a>	0
<a href="#">PoA::\$authn_engine</a>	0
<a href="#">PoA::\$attributes</a>	0
<a href="#">PAPICrypt::setPrivateKey()</a>	0
<a href="#">PAPICrypt::setPublicKey()</a>	0
<a href="#">PAPICrypt::setSymmetricKey()</a>	0
<a href="#">PoA</a>	0



<i>Standard class that implements all the functionality of the phpPoA.</i>	
<a href="#">PoA::\$handler</a>	0
<a href="#">PoA::\$local_site</a>	0
<a href="#">PoA::getAttributes()</a>	0
<i>Retrieve the attributes provided by the user when logged in.</i>	
<a href="#">PoA::getAuthorizationEngines()</a>	0
<i>Returns the authorization engines configured for the current PoA, or the one specified.</i>	
<a href="#">PoA::getAuthorizationLevels()</a>	0
<i>Get the authorization levels that match for the user specified, according to the configuration.</i>	
<a href="#">PoA::isAuthenticated()</a>	0
<i>Query the current status of the user in the federation.</i>	
<a href="#">PoA::getAttribute()</a>	0
<i>Get the value (or values) of an attribute, if present.</i>	
<a href="#">PoA::clean()</a>	0
<a href="#">PoA::\$log</a>	0
<a href="#">PoA::addHook()</a>	0
<i>Attach a hook object to the appropriate entry point of the available authentication or authorization engines.</i>	
<a href="#">PoA::authenticate()</a>	0
<i>Perform a federated login for the user.</i>	
<a href="#">PoA::authorize()</a>	0
<i>Authorize a given subject with the data retrieved from federated login.</i>	
<a href="#">PAPICrypt::initializeAES()</a>	0
<a href="#">PAPICrypt::initialize3DES()</a>	0
<a href="#">papi1.5-slo-extra.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">papi1.5-slo.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">PAPICrypt</a>	0
<i>Cryptographic routines for PAPI protocol v1.</i>	
<a href="#">PAPICrypt::\$privkey</a>	0
<a href="#">PoAUtils.php</a>	0
<a href="#">PoALog.php</a>	0
<a href="#">PoA.php</a>	0
<a href="#">PoAConfigurator.php</a>	0
<a href="#">PoAEventHandler.php</a>	0
<a href="#">PoAException.php</a>	0
<a href="#">PAPICrypt::\$privkeylen</a>	0
<a href="#">PAPICrypt::\$pubkey</a>	0
<a href="#">PAPICrypt::encrypt()</a>	0
<a href="#">PAPICrypt::encrypt3DES()</a>	0
<a href="#">PAPICrypt::encryptAES()</a>	0
<a href="#">PAPICrypt::endSymmetric()</a>	0
<a href="#">PAPICrypt::decryptAES()</a>	0
<a href="#">PAPICrypt::decrypt3DES()</a>	0
<a href="#">PAPICrypt::\$pubkeylen</a>	0
<a href="#">PAPICrypt::\$symmetric</a>	0
<a href="#">PAPICrypt::decrypt()</a>	0
<a href="#">PoA::isAuthorized()</a>	0
<i>Perform authorization for the a given subject.</i>	
<a href="#">PoA::logout()</a>	0
<i>Remove the user's session and trigger a logout for the specified authentication</i>	

<i>protocol.</i>	0
<a href="#">PoAUtils::lang_code()</a>	0
<i>Return the language code identifying applicable to a messages file.</i>	
<a href="#">PoAUtils::msg()</a>	0
<i>Return the internationalized message identified by the code 's'.</i>	
<a href="#">papi1.5-attr-authz.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">PUBLIC_KEY_OID</a>	0
<i>The OID for public keys.</i>	
<a href="#">PoAUtils</a>	0
<a href="#">PoALog::write()</a>	0
<i>Write a message to the log file if the current log level allows it.</i>	
<a href="#">PoAException</a>	0
<i>Exception handler class for the phpPoA.</i>	
<a href="#">PoAException:: toString()</a>	0
<a href="#">PoALog</a>	0
<i>A simple class to write log messages to a file.</i>	
<a href="#">PoALog::getLogLevel()</a>	0
<i>Get the current log level.</i>	
<a href="#">papi1.5-invite-authz.php</a>	0
<i>This is a sample file to demonstrate the functionality of the phpPoA2 package.</i>	
<a href="#">PAPIAuthnEngine.php</a>	0
<a href="#">PAPI RESTORE ORIGINAL REQUEST</a>	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>	
<a href="#">PAPI_SESSION</a>	0
<a href="#">PROTO_ATTR_AS_ID</a>	0
<a href="#">PROTO_ATTR_EXPIRE_TIME</a>	0
<a href="#">PAPI_REDIRECT_URL_FINISH</a>	0
<i>This hook is executed at the end of the method that returns the URL where to redirect a user.</i>	
<a href="#">PAPI_MYSQL</a>	0
<a href="#">PAPI_ATTRIBUTE_PARSER</a>	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>	
<a href="#">PAPI_CLEAN_URL</a>	0
<i>This hook is executed when a valid response is found from the AS/GPoA and the engine is about</i>	
<a href="#">PAPI_DBA</a>	0
<i>Supported database types.</i>	
<a href="#">PoAEventHandler::setLogger()</a>	0
<i>Set the logger.</i>	
<a href="#">PoAEventHandler::setDebug()</a>	0
<i>Set debugging mode.</i>	
<a href="#">PoAConfigurator::getAuthzEngineConfFile()</a>	0
<a href="#">PoAConfigurator::getAuthzEngines()</a>	0
<a href="#">PoAConfigurator::getAuthzLevels()</a>	0
<a href="#">PoAConfigurator::getInviteErrorURL()</a>	0
<a href="#">PoAConfigurator::getAuthnEngineConfFile()</a>	0
<a href="#">PoAConfigurator::getAuthnEngine()</a>	0
<a href="#">PoA::removeHook()</a>	0

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

<a href="#">PoA::revoke()</a>	0
Revoke authorization for a given subject identified by an e-mail.	
<a href="#">PoAConfigurator</a>	0
Main configurator for the phpPoA.	
<a href="#">PoAConfigurator::\$mandatory_options</a>	0
<a href="#">PoAConfigurator::getLanguage()</a>	0
<a href="#">PoAConfigurator::getLogFile()</a>	0
<a href="#">PoAEventHandler::abort()</a>	0
Finish execution and print an error message with the current backtrace.	
<a href="#">PoAEventHandler::autoloadHandler()</a>	0
Override __autoload standard function to allow loading classes dinamically.	
<a href="#">PoAEventHandler::errorHandler()</a>	0
A generic error handler that logs the error message.	
<a href="#">PoAEventHandler::exceptionHandler()</a>	0
A generic exception handler that logs the exception message.	
<a href="#">PoAEventHandler</a>	0
Class to handle events inside the library.	
<a href="#">PoAConfigurator::isDebug()</a>	0
<a href="#">PoAConfigurator::getLogLevel()</a>	0
<a href="#">PoAConfigurator::getNoAuthErrorURL()</a>	0
<a href="#">PoAConfigurator::getSystemErrorURL()</a>	0
<a href="#">PAPICrypt.php</a>	0

## Q

<a href="#">QueryFilterAuthzEngine::revoke()</a>	0
<a href="#">QueryFilterAuthzEngine::isAuthorized()</a>	0
PLEASE NOTE THAT THIS ENGINE SILENTLY IGNORES BOTH USER AND ATTRIBUTES.	
<a href="#">QueryFilterConfigurator</a>	0
Configurator for the query filter authorization engine.	
<a href="#">QueryFilterConfigurator::\$mandatory_options</a>	0
<a href="#">QueryFilterConfigurator::getDeniedPatterns()</a>	0
Returns the patterns that match denied URIs or query parameters.	
<a href="#">QueryFilterConfigurator::getAllowedPatterns()</a>	0
Returns the patterns that match allowed URIs or query parameters.	
<a href="#">QueryFilterAuthzEngine::getAuthorizedList()</a>	0
<a href="#">QueryFilterAuthzEngine::authorize()</a>	0
<a href="#">QueryFilterConfigurator.php</a>	0
<a href="#">QUERY_BEFORE_FILTERS</a>	0
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.	
<a href="#">queryHook()</a>	0
<a href="#">QueryFilterAuthzEngine</a>	0
Authorization engine that works by checking the query string of the request.	
<a href="#">QueryFilterAuthzEngine::\$valid_hooks</a>	0
<a href="#">QueryFilterAuthzEngine.php</a>	0

## R

<a href="#">RSAPublicKey</a>	0
<i>Class to manage public keys.</i>	
<a href="#">RSAPrivateKey::setPublicExponent()</a>	0
<i>Set the public exponent of the key.</i>	
<a href="#">RSAPublicKey::\$bits</a>	0
<a href="#">RSAPublicKey::\$der</a>	0
<a href="#">RSAPublicKey::\$modulus</a>	0
<a href="#">RSAPublicKey::\$exponent</a>	0
<a href="#">RSAPrivateKey::setPrivateExponent()</a>	0
<i>Set the private exponent of the key.</i>	
<a href="#">RSAPrivateKey::setPrime2()</a>	0
<i>Set the prime 2 of the key.</i>	
<a href="#">RSAPrivateKey::setExponent1()</a>	0
<i>Set the exponent 1 of the key.</i>	
<a href="#">RSAPrivateKey::setCoefficient()</a>	0
<i>Set the coefficient of the key.</i>	
<a href="#">RSAPrivateKey::setExponent2()</a>	0
<i>Set the exponent 2 of the key.</i>	
<a href="#">RSAPrivateKey::setModulus()</a>	0
<i>Set the modulus of the key.</i>	
<a href="#">RSAPrivateKey::setPrime1()</a>	0
<i>Set the prime 1 of the key.</i>	
<a href="#">RSAPublicKey::\$pem</a>	0
<a href="#">RSAPublicKey::decode()</a>	0
<i>Extract public key details from its DER representation.</i>	
<a href="#">RSAPublicKey::setModulus()</a>	0
<i>Set the modulus of the key.</i>	
<a href="#">RSAPublicKey::getPEM()</a>	0
<i>Get the PEM representation of the key.</i>	
<a href="#">RSAPublicKey::setPublicExponent()</a>	0
<i>Set the public exponent of the key.</i>	
<a href="#">REQUEST_LIFETIME</a>	0
<i>Default timeout for stored requests.</i>	
<a href="#">README</a>	0
<a href="#">RSAPublicKey::getModulus()</a>	0
<i>Get the modulus of the key.</i>	
<a href="#">RSAPublicKey::getExponent()</a>	0
<i>Get the exponent of the key.</i>	
<a href="#">RSAPublicKey::fromDER()</a>	0
<i>Build the public key from its DER representation.</i>	
<a href="#">RSAPublicKey::encode()</a>	0
<i>Build a new public key represented by it DER and PEM formats from its modulus and public exponent.</i>	
<a href="#">RSAPublicKey::fromPEM()</a>	0
<i>Build the public key from its PEM representation.</i>	
<a href="#">RSAPublicKey::getBits()</a>	0
<i>Get the length of the key in bits.</i>	
<a href="#">RSAPublicKey::getDER()</a>	0
<i>Get the DER representation of the key.</i>	
<a href="#">RSAPrivateKey::getPublicExponent()</a>	0
<i>Get the public exponent of the key.</i>	

<a href="#">RSAPrivateKey::getPrivateExponent()</a>	0
<i>Get the private exponent of the key.</i>	
<a href="#">RSAPrivateKey::\$modulus</a>	0
<a href="#">RSAPrivateKey::\$exponent2</a>	0
<a href="#">RSAPrivateKey::\$pem</a>	0
<a href="#">RSAPrivateKey::\$prime1</a>	0
<a href="#">RSAPrivateKey::\$private_exponent</a>	0
<a href="#">RSAPrivateKey::\$prime2</a>	0
<a href="#">RSAPrivateKey::\$exponent1</a>	0
<a href="#">RSAPrivateKey::\$der</a>	0
<a href="#">RSAPublicKey.php</a>	0
<a href="#">RSAPrivateKey.php</a>	0
<a href="#">RSAPrivateKey</a>	0
<i>Class to manage private keys.</i>	
<a href="#">RSAPrivateKey::\$bits</a>	0
<a href="#">RSAPrivateKey::\$coefficient</a>	0
<a href="#">RSAPrivateKey::\$public_exponent</a>	0
<a href="#">RSAPrivateKey::decode()</a>	0
<i>Extract private key details from its DER representation.</i>	
<a href="#">RSAPrivateKey::getModulus()</a>	0
<i>Get the modulus of the key.</i>	
<a href="#">RSAPrivateKey::getExponent2()</a>	0
<i>Get the exponent 2 of the key.</i>	
<a href="#">RSAPrivateKey::getPEM()</a>	0
<i>Get the PEM representation of the key.</i>	
<a href="#">RSAPrivateKey::getPrime1()</a>	0
<i>Get the prime 1 of the key.</i>	
<a href="#">RSAPrivateKey::getPrime2()</a>	0
<i>Get the prime 2 of the key.</i>	
<a href="#">RSAPrivateKey::getExponent1()</a>	0
<i>Get the exponent 1 of the key.</i>	
<a href="#">RSAPrivateKey::getDER()</a>	0
<i>Get the DER representation of the key.</i>	
<a href="#">RSAPrivateKey::fromDER()</a>	0
<i>Build the private key from its DER representation.</i>	
<a href="#">RSAPrivateKey::encode()</a>	0
<i>Build a new private key represented by it DER and PEM formats from its modulus and public exponent.</i>	
<a href="#">RSAPrivateKey::fromPEM()</a>	0
<i>Build the private key from its PEM representation.</i>	
<a href="#">RSAPrivateKey::getBits()</a>	0
<i>Get the length of the key in bits.</i>	
<a href="#">RSAPrivateKey::getCoefficient()</a>	0
<i>Get the coefficient of the key.</i>	
<a href="#">ROOT</a>	0
<i>Authorized DB converter tool from phpPoA1 to phpPoA2.</i>	

## S

<a href="#">SOURCEADDR_BEFORE_FILTERS</a>	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>	

<a href="#">SourceIPAddrConfigurator.php</a>	0
<a href="#">SourceIPAddrAuthzEngine</a>	0
<i>Authorization engine that works by checking the source IP address of the request.</i>	
<a href="#">SourceIPAddrAuthzEngine.php</a>	0
<a href="#">SimpleSAMLphpAuthnEngine::logout()</a>	0
<a href="#">SimpleSAMLphpAuthnEngine::getAttributes()</a>	0
<a href="#">SimpleSAMLphpAuthnEngine::isAuthenticated()</a>	0
<a href="#">SourceIPAddrAuthzEngine::\$valid_hooks</a>	0
<a href="#">SourceIPAddrAuthzEngine::authorize()</a>	0
<a href="#">SourceIPAddrConfigurator::\$mandatory_options</a>	0
<a href="#">SourceIPAddrConfigurator::getAllowed()</a>	0
<i>Returns an array of IP addresses or networks that are authorized.</i>	
<a href="#">SourceIPAddrConfigurator::getDenied()</a>	0
<i>Returns an array of IP addresses or networks that are not authorized.</i>	
<a href="#">SourceIPAddrConfigurator</a>	0
<i>Configurator class for the Source IP Address Authorization Engine.</i>	
<a href="#">SourceIPAddrAuthzEngine::revoke()</a>	0
<a href="#">SourceIPAddrAuthzEngine::getAuthorizedList()</a>	0
<a href="#">SourceIPAddrAuthzEngine::isAuthorized()</a>	0
<i>PLEASE NOTE THAT THIS ENGINE SILENTLY IGNORES BOTH USER AND ATTRIBUTES.</i>	
<a href="#">SimpleSAMLphpAuthnEngine::getAttribute()</a>	0
<a href="#">SimpleSAMLphpAuthnEngine::configure()</a>	0
<a href="#">SessionDB::close()</a>	0
<a href="#">SessionDB::delete()</a>	0
<a href="#">SessionDB::fetch()</a>	0
<a href="#">SessionDB::check()</a>	0
<a href="#">SessionDB::\$session_name</a>	0
<a href="#">SessionDB.php</a>	0
<a href="#">SessionDB</a>	0
<i>Session database backend.</i>	
<a href="#">SessionDB::fetch_all()</a>	0
<a href="#">SessionDB::open()</a>	0
<a href="#">SimpleSAMLphpAuthnEngine::\$simplesaml</a>	0
<a href="#">SimpleSAMLphpAuthnEngine::\$status</a>	0
<a href="#">SimpleSAMLphpAuthnEngine::authenticate()</a>	0
<a href="#">SimpleSAMLphpAuthnEngine::\$attributes</a>	0
<a href="#">SimpleSAMLphpAuthnEngine</a>	0
<i>Authentication engine for the PAPI 1.5 protocol.</i>	
<a href="#">SessionDB::replace()</a>	0
<a href="#">SimpleSAMLphpAuthnEngine.php</a>	0
<a href="#">SYSTEM_ERR</a>	0
<i>A system related error.</i>	

## T

<a href="#">TAG_SEQUENCE</a>	0
<i>Constructed tags.</i>	
<a href="#">TAG_INTEGER</a>	0
<a href="#">TAG_BOOLEAN</a>	0
<a href="#">TAG_BITSTRING</a>	0
<a href="#">TAG_BER</a>	0
<i>Basic tags.</i>	

## U

<a href="#">USER_ERR</a> . . . . .	0
<i>An error triggered by the user.</i>	

## V

<a href="#">VALUE_SEPARATOR</a> . . . . .	0
---	---