AlphaUserPoints

Manual for developers

AlphaUserPoints v.1.8.7 and later for Joomla 2.5.x
AlphaUserPoints v.1.9.6 and later for Joomla 3.x.x

English version

This project is developed by Bernard Gilly.
Official website http://www.alphaplug.com
Documentation Bernard Gilly

Introduction

AlphaUserPoints - API integration in a Third party component (advanced)

This documentation is priorly dedicated to developers and anyone with a sufficient knowledge of PHP language and Joomla! components development.

Author: Bernard Gilly – This project started current summer 2008.

Index

Plugin/Rule creation

A plugin creation (new rule creation for a 3rd party component) is done in 2 steps.

Step1 - API insertion in a component

Just insert the following API in the component code where needed. The best is to make it follow a user action that could give the users some points or take some.
For example : in a comment system component or in a forum, just add the API after the comment or topic INSERT query in the database.

Basic API

```
$api_AUP = JPATH_SITE.DS.'components'.DS.'com_alphauserpoints'.DS.'helper.php';
if ( file_exists($api_AUP))
{
    require_once ($api_AUP);
    AlphaUserPointsHelper::newpoints( 'function_name' );
}
```

function_name is the name of the rule that will be used to attribute points to the current user (if logged in). For each AlphaUserPoints integrated rule (system rules), names syntax is as follows:

example: *sysplgaup_newregistered* for new users points attribution

It is convenient to keep the same name syntax for third party components plugin as follows:

plgaup_functionname

Example: *plgaup_newcomment* or *plgaup_newtopic* for a comment system or forum API integration. To name a rule that would give points when adding a new topic in Kunena: *plgaup_kunena_topic_create*.

Keep in mind that this method only gives points to the current user logged in. This is the Basic API.

Attribute points to another user

To give points to anothe user than the one connected, only the user id is required. To get his AlphaUserPoints (AUPID) Identity, we just need to use the getAnyUserReferreID(). This method is the one used to give points to an article author when the article is being read by someone on the site.

```
$api_AUP = JPATH_SITE.DS.'components'.DS.'com_alphauserpoints'.DS.'helper.php';
if ( file_exists($api_AUP))
{
    require_once ($api_AUP);
    $aupid = AlphaUserPointsHelper::getAnyUserReferreID( $userID );
    if ( $aupid ) AlphaUserPointsHelper::newpoints( 'function_name', $aupid );
}
```

Prevent from attributing points more than once for the same action

To avoid that a user would get points many times for something allready done, we can add a reference key. When calling the AlphaUserPointsHelper::newpoints function, a pre check is done on this reference key. This method is used in the rule where a user reading an article would give points to the author.

```
$api_AUP = JPATH_SITE.DS.'components'.DS.'com_alphauserpoints'.DS.'helper.php';
if ( file_exists($api_AUP))
{
        require_once ($api_AUP);
        $keyreference = AlphaUserPointsHelper::buildKeyreference('function_name', ‘item
id’ );
       AlphaUserPointsHelper::newpoints( 'function_name', '', $keyreference);
}
```

Adding information datas

To add information datas to be displayed in the action details, just add a new parameter as follows:

```
$api_AUP = JPATH_SITE.DS.'components'.DS.'com_alphauserpoints'.DS.'helper.php';
if ( file_exists($api_AUP))
{
  require_once ($api_AUP);
 $keyreference = AlphaUserPointsHelper::buildKeyreference('function_name', ‘item id’ );
AlphaUserPointsHelper::newpoints( function_name','',$keyreference,'information_datas');
}
```

Using different amounts of points on the same rule

A component might also need to add or to take points for different amounts. For example, when buying goods with points. Products have different prices, a fixed amount in the rule would'nt make it. The API $randompoints parameter comes instead of the amount of points set in the rule. It can be negative in case of purchases or penalities.

```
$api_AUP = JPATH_SITE.DS.'components'.DS.'com_alphauserpoints'.DS.'helper.php';
if ( file_exists($api_AUP))
{
 require_once ($api_AUP);
$keyreference = AlphaUserPointsHelper::buildKeyreference('function_name', ‘item id’ );
AlphaUserPointsHelper::newpoints( 'function_name', '', $keyreference, '', -1450);
}
```

Get the result from a successfull operation

In a more advanced code, if the component routine needs to know if the operation has been successfull or not, (enough amount of points for a purchase in a user account), we can add a 'feedback' parameter. It has a Boolean type value.

Code example:

```
$api_AUP = JPATH_SITE.DS.'components'.DS.'com_alphauserpoints'.DS.'helper.php';
if ( file_exists($api_AUP))
{
 require_once ($api_AUP);
$keyreference=AlphaUserPointsHelper::buildKeyreference('plgaup_purchasewithvirtuemart', $transactionID );

    if (AlphaUserPointsHelper::newpoints( 'plgaup_purchasewithvirtuemart', '',
    $keyreference, 'Product reference: AM-5245', -1290, true))
    {
        [... code continued ...]
     }
}
```

Remove the constraint on the type of user

In a customized code component, you can force and remove the constraint on a rule to the user level by adding the parameter *force = 1*. The existing rule will be available now for *guest*, *registered* and *special*.

Display an additional system message on frontend

you can display a specific message on frontend by adding the parameter *frontmessage=”You custom message”*. API full implementation

AlphaUserPointsHelper::newpoints(string*$pluginfunction, [*string*$AUPIDotheruser = ''], [*string*$keyreference = ''], [*string*$data = ''], [*integer*$randompoints = 0], [*boolean*$feedback = false], [*integer*$force=0],[*string*$frontmessage='']);*

Note: If the operation is a points substraction, the account has to have at least the same amount of points. If not, a notice warns the user that he doe'snt have enough points to complete the action (by default).

You can set up general parameter in the configuration of AlphaUserPoints, in backend administrator to allows your users to have a negative account.

Step 2 - XML file creation

Then we have to create an xml file to make easier the installation process in the AlphaUserPoints component.
5 elements minimum have to be filled in:

Rule name, its description, the name of the component using that rule (com_example), the used function name (eg: plgaup_vm_purchase) and the last parameter: fixed_points to specify if points are fixed or could change (boolean true or false). If set to false, attributed points for this rule would be zero and points have to be set by the api code parameter included in the third party component (as stated in this documentation).

You can add 3 others fields as the category for the rule based on 2 letters, display message for the current user (0 or 1), and e-mail notification (0 or 1).

```
<?xml version="1.0" encoding="utf-8"?>
<alphauserpoints type="plugin">
    <rule>Short rule name</rule>
    <description>Rule description</description>
    <component>com_example</component>
    <plugin_function>plgaup_function_name</plugin_function>
    <fixed_points>true OR false</fixed_points>
    <category>2 letters for category</category>
    <display_message>1</display_message>
    <email_notification>0</email_notification>
</alphauserpoints>
```

This xml file has to be utf-8 encoded (required) and can then be zipped (not required).

Example :

```
<?xml version="1.0" encoding="utf-8"?>
<alphauserpoints type="plugin">
    <rule>Import article with Feedgator</rule>
    <description>Assigns points on import article with Feedgator</description>
    <component>com_feedgator</component>
    <plugin_function>plgaup_importfeedgator</plugin_function>
    <fixed_points>true</fixed_points>
    <category>ar</category>
    <display_message>1</display_message>
    <email_notification>0</email_notification>
</alphauserpoints>
```

This is the first method to create a simple xml file installation rule for an unique rule and this xml file must be added from the button “plugins” in control panel of AlphaUserPoints. You can name this file as you want.

With AlphaUserPoints version 1.9.6 and later for Joomla 3.x, or AlphaUserPoints version 1.8.7 and later for Joomla 2.5.x, all developers of third extensions for Joomla! can add directly at the root of their **frontend** component an unique xml file containing all the rules for a single component. The file structure is almost similar but you have to respect the ordering and tags:

```
<?xml version="1.0" encoding="utf-8"?>
<alphauserpoints>
  <component>com_example</component>
  <rules>
    <rule>
      <name>Add</name>
      <description>Give points when registered user add something.</description>
      <plugin_function>plgaup_example_add</plugin_function>
      <fixed_points>true</fixed_points>
      <category>ot</category>
      <display_message>1</display_message>
      <email_notification>1</email_notification>
    </rule>
    <rule>
      <name>Upload</name>
      <description>Remove points when registered user upload something.</description>
      <plugin_function>plgaup_example_upload</plugin_function>
      <fixed_points>true</fixed_points>
      <category>mu</category>
      <display_message>1</display_message>
      <email_notification>1</email_notification>
    </rule>
    <rule>
      <name>Remove</name>
      <description>Remove points when registered user remove something.</description>
      <plugin_function>plgaup_example_remove</plugin_function>
      <fixed_points>false</fixed_points>
      <category>ot</category>
      <display_message>1</display_message>
      <email_notification>0</email_notification>
    </rule>
  </rules>
</alphauserpoints>
```

Tag “component” is the name of the third component like “com_kunena” or other. As it is the same component, it is worth repeating for each rule in the tag “rule”

Administrator of the website which install a third component with this xml file can auto-detect directly from the button “auto-detect plugins” in control panel of AlphaUserPoints.

This xml file has to be utf-8 encoded (required) but not be zipped! Just put this file at the root of **frontend** component folder or plugin or module and include this file in your installer extension. This file **must be named** exactly as follows: **alphauserpoints_rule.xml**

Code list for categories :

ar -> articles
cd -> coupons
co -> community
fo -> forums/comments
li -> links
mu -> music
ot -> other
ph -> photo
po -> polls
pu -> purchase
re -> recommend/send to a friend
sh -> shopping
su -> private
sy -> system rules
us -> users
vi -> video

NOTE :

Optionally, you can avoid this step manually by filling all the fields in the creation of a new rule directly in the rule manager (button new in toolbar).

Plugin/Rule installation

- First manually method with simple xml file (one rule):

Install the new rule (plugin) using the Plugins button in the AlphaUserPoints component administration panel. The plugin installation can also be completed from the 'New' menu item in AlphaUserPoints 'Rules manager' (manually method) or be completed from the ‘Plugins’ menu item in toolbar in AlphaUserPoints 'Rules manager' to insert the new rule with an unique xml file for a rule (see above).

- Second method with auto-detect xml file at the root of frontend component:

Click on the button “Auto-detect plugins” in the control panel of AlphaUserPoints after installation of third component, plugin or module. Check regularly or periodically by clicking on this button.

Using AlphaUserPoints informations in a third component

You can use easily the profil informations of a user directly in a third component.

=> Before using a function, you must include the API at least once in your page like this:

```
$api_AUP = JPATH_SITE.DS.'components'.DS.'com_alphauserpoints'.DS.'helper.php';
if ( file_exists($api_AUP))
{
    require_once ($api_AUP);
}
```

Profil informations

To get the entire profil information, just use the function ***getUserInfo() ;***

Just use the referreid of AlphaUserPoints user or the joomla ID of the user (Id of Joomla users table).

Use the first method with the referreid to get user Information profile like this :
AlphaUserPointsHelper::getUserInfo ($referrerid) ;

If you have not the referreid, you can use the ID of user in second parameter like this :

```
$user = & JFactory::getUser();
$userid = $user->id ;
$profil = AlphaUserPointsHelper::getUserInfo ( '',  $user->id ) ;
```

Then you can get the user informations.

```
$profil->name
$profil->username
$profil->registerDate
$profil->lastvisitDate
$profil->email
$profil->referreid
$profil->points
$profil->max_points
$profil->last_update
$profil->referraluser
$profil->referrees
$profil->blocked
$profil->birthdate
$profil->avatar
$profil->levelrank
$profil->leveldate
$profil->gender
$profil->aboutme
$profil->website
$profil->phonehome
```

$profil->phonemobile
$profil->address
$profil->zipcode
$profil->city
$profil->country
$profil->education
$profil->graduationyear
$profil->job
$profil->facebook
$profil->twitter
$profil->icq
$profil->aim
$profil->yim
$profil->msn
$profil->skype
$profil->gtalk
$profil->xfire
$profil->profileviews

Get AUP avatar of user

Display the image of avatar from AlphaUserPoints.

$avatar = AlphaUserPointsHelper:: getAupAvatar($userid, [$linktoprofil=0],[$width=48], [$height=48], [$class=''], [$otherprofileurl=''])
echo $avatar ;

if *$linktoprofil* set to 1, display avatar with the link to the AUP profil of this user.

Get link to AUP user profil

Get the url to show the profil of user.

$linktoAUPprofil = AlphaUserPointsHelper::getAupLinkToProfil($userid);

Get link to AUP users list

Get the url to show the list of users with points etc…

$linktoAUPusersList = AlphaUserPointsHelper:: getAupUsersURL();

Get avatar Path of a user

Get the path avatar path of a specific user

$avatarPath = AlphaUserPointsHelper:: getAvatarPath($userid);

Get avatar Live Path of a user

Get the live url avatar path of a specific user

```
$avatarLivePath = AlphaUserPointsHelper:: getAvatarLivePath( $userid );
```

Get the medals list of a user

```
$medalslistuser  = getUserMedals ( $referrerid ) ;
```
or
```
$medalslistuser  = getUserMedals ( '', $userid ) ;
```

return $medallistuser->id
return $medallistuser->rank (name of medal)
return $medallistuser->description (reason for awarded)
return $medallistuser->icon (name file icon)
return $medallistuser->image (name file image)

Complete example to display large image of medals:

```
$user = & JFactory::getUser();
$userid = $user->id ;

if(!defined("_AUP_MEDALS_PATH")) {
    define('_AUP_ MEDALS _PATH', JURI::root() .
    'components'.DS.'com_alphauserpoints'.DS.'assets'.DS.'images'.DS.'awards'.
   DS.'large'.DS);
}
if(!defined("_AUP_MEDALS_LIVE_PATH")) {
    define('_AUP_ MEDALS _LIVE_PATH', JURI::base(true) .
   '/components/com_alphauserpoints/assets/images/awards /large/');
}


$medalslistuser  = AlphaUserPointsHelper::getUserMedals ( '', $userid ) ;

for each ( $medalslistuser as $medallistuser ) {

        echo '<img src="' . _AUP_ MEDALS _LIVE_PATH'.$medallistuser->image . '"
              alt="" />' ;

}
```

Get the ReferreID of any user

$referreid = AlphaUserPointsHelper::getAnyUserReferreID($userID);

Get the current total points of any user

Use the first method with the referreid to get the total of points

$totalPoints = AlphaUserPointsHelper::getCurrentTotalPoints($referrerid);
If you have not the referreid, you can use the ID of user (joomla) in second parameter like this :

$user = & JFactory::getUser();
$userid = $user->id ;

$totalPoints = AlphaUserPointsHelper::getCurrentTotalPoints(‘‘, $userid);

Get the list of activities

A new function has been added in version 1.7 :

$listActivities = AlphaUserPointsHelper:: getListActivity($type='all', $userid='all', $numrows=0);

$params $type = all | positive | negative
$params $userid = all or unique userID
$params $limit = int (0 by default)
example-1 -> ----------
example-1 -> $listActivities = AlphaUserPointsHelper::getListActivity('all', 'all');
example-1 SAME AS -> $listActivities = AlphaUserPointsHelper::getListActivity();
example-1 -> show all activities with pagination, positive and negative points of activity for all users
---
example-2 -> ----------
example-2 -> $user = JFactory::getUser();
example-2 -> $userID = $user->id;
example-2 -> $listActivities = AlphaUserPointsHelper::getListActivity('positive',$userID,20);
example-2 -> show only positive points of activity for the current user logged in and show only 20 rows of recent activities

return an array or $rows

list of available fields :

insert_date
referreid
points (of this activity)
datareference
rule_name
plugin_function
category

Get the next user rank

$nextrankinfo = AlphaUserPointsHelper::getNextUserRank($referrerid='', $userid='0', currentrank);

return $nextrankinfo->id
return $nextrankinfo->rank (name of rank)
return $nextrankinfo->description (description of rank)
return $nextrankinfo->levelpoints (level points to reach this rank)
return $nextrankinfo->typerank (return 0)
return $nextrankinfo->icon (name file icon)
return $nextrankinfo->image (name file image)

Get version of AUP

$num_aup_version = AlphaUserPointsHelper::getAupVersion();

Return the current version of AlphaUserPoints like -> 1.7.4

AlphaUserPoints is open for third component

Create your own plugin for AlphaUserPoint !

Plugins provide functions which are associated with trigger events.

Available events in AlphaUserPoints:

- onBeforeInsertUserActivityAlphaUserPoints
- onUpdateAlphaUserPoints
- onAfterUpdateAlphaUserPoints
- onChangeLevelAlphaUserPoints
- onUnlockMedalAlphaUserPoints
- onGetNewRankAlphaUserPoints

- onResetGeneralPointsAlphaUserPoints
- onBeforeDeleteUserActivityAlphaUserPoints
- onAfterDeleteUserActivityAlphaUserPoints
- onBeforeDeleteAllUserActivitiesAlphaUserPoints
- onAfterDeleteAllUserActivitiesAlphaUserPoints
- onBeforeMakeRaffleAlphaUserPoints
- onAfterMakeRaffleAlphaUserPoints

You can see the example file /plugins/alphauserpoints/example_plugin_aup.php in your Joomla directory with parameters for each functions.