AlphaUserPoints

Manual for developers

from AlphaUserPoints v.1.5.6 and later

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 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 convinient 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_newtopic_kunena*.

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);
    AlphaUserPointsHelper::newpoints( 'function_name', '', 'reference_key');
}
```

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);
    AlphaUserPointsHelper::newpoints( 'function_name', '', '', '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 diferent 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);
    AlphaUserPointsHelper::newpoints( 'function_name', '', '', '', -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);
    if (AlphaUserPointsHelper::newpoints( 'plgaup_purchasewithvirtuemart', '',
    'payment ID:      20080831-YHMU', '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 a 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='']);
```

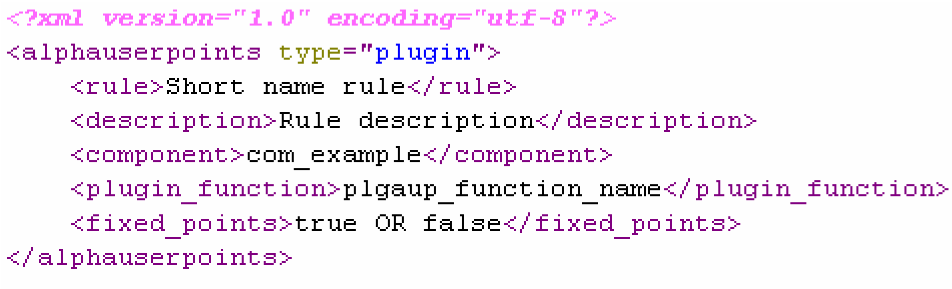
Note: If a the operation is a points substraction, the account has to have at least the same amount of points. If not, a notice warns the user that he doe'snt have enough points to complete the action.

The user account points can’t be negative.

Step 2 - XML file creation

Then we have to create an xml file to make easier theinstallation process in the AlphaUserPoints component.
5 elements 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 compnent (as stated in this documentation).



```xml
<?xml version="1.0" encoding="utf-8"?>
<alphauserpoints type="plugin">
    <rule>Short name rule</rule>
    <description>Rule description</description>
    <component>com_example</component>
    <plugin_function>plgaup_function_name</plugin_function>
    <fixed_points>true OR false</fixed_points>
</alphauserpoints>
```

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

Plugin installation

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'.

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  = getUserMedals ( '', $userid ) ;

for each ( $medalslistuser as $medallistuser ) {

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

}
```

Get the RefferreID of a any user

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

Get version of AUP (added in 1.5.6)

```
 $num_aup_version = AlphaUserPointsHelper::getAupVersion();
```

Version 1.0.1 of document released on 14 april 2010