

JavaScript Plugin for Hangoutt Reward Platform

JavaScript Plugin for Hangoutt Reward Platform provides developers an interface to integrate Hangoutt Rewards on their website.

Procedure to integrate Hangoutt Reward Platform:

1. Download the Hangoutt jsPlugin zip file and copy the following files into their respective directories:
 - a. js/hangoutt.min.js to your js folder.
 - b. css/hangoutt.css to your css folder.
 - c. images/hangoutt to your images folder.
2. Include the files (js, css) in your webpage as mentioned in Figure 1. Ensure that the path to hangoutt.css and hangoutt.min.js are correct.

```
<link href="css/hangoutt.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="js/hangoutt.min.js"></script>
```

Figure 1: Including files in the webpage

3. Once you include the files in your webpage, initialize the variables and create an object for HangouttRewards Class. Refer Figure 2.

```
<script type="text/javascript">
    HangouttRewards.pathToImages = "";
    HangouttRewards.captchaSiteKey = "Enter your site key";
    HangouttRewards.captchaSecretKey = "Enter your secret key";
    HangouttRewards.site_verify_url =
        "https://www.google.com/recaptcha/api/siteverify";
    HangouttRewards.token_url = "link to token url";
    var hangouttRewards = new HangouttRewards();
</script>
```

Figure 2: Initialize variables and create an object for HangouttRewards

The above script tag should be added in the head tag of your webpage.

Regarding Captcha:

We currently use Google's reCaptcha in the Javascript Plugin. This reCaptcha uses SiteKey and SecretKey to display and validate the captcha. These Keys are domain specific. For your website, you can get a new pair of keys by visiting the following link:

<https://www.google.com/recaptcha/admin>.

Once the captcha is solved by the user, Google verifies the user's response using the secretkey, which should be kept secret on the server. This is discussed in detail in the appendix section.

4. If you want to award hangoutt points to your users, then you can do so, by calling add method of HangouttRewards Class, as mentioned in Figure 3.

```
<script type="text/javascript">
    hangouttRewards.add("100");
</script>
```

Figure 3: Rewarding Hangoutt Points to the user

5. Once you identify the rewarding moment, you can call the reward_user method of HangouttRewards Class. Please refer Figure 4. You can also pass additional parameters to reward_user method such as title, sub_title, labels, email etc.

```
<div id="rewards_section" style="width:500px; height:400px;">
</div>
<script type="text/javascript">
    // The labels array is used in analytics.
    var labels = [];
    labels[0] = "IndexPage";
    labels[1] = "After3rdEvent";

    var params = {};
    params['labels'] = JSON.stringify(labels);
    params['email'] = "user_email@example.com";
    params['title'] = "Enter your Title Message";
    params['sub_title'] = "We are offering this reward coz you have
        completed level 4.";

    hangouttRewards.reward_users("rewards_section", true, false, params,
        function(count){
            if (count > 0) {
                hangouttRewards.reward_users("rewards_section",
                    false, false);
            }
        });
</script>
```

Figure 4: Rewarding the user

6. You can also get the current points of the user by calling get method of HangouttRewards Class, as mentioned in Figure 5.

```
<script type="text/javascript">
    var points = hangouttRewards.get();
</script>
```

Figure 5: Get the current user points

7. If you'd like to get a callback, then you can use the methods, setRewardsCloseCallback and setRewardRedeemedCallback, as shown in Figure 6.

```
//setup callbacks
hangouttRewards.setRewardsCloseCallback(function(){
    //do something
    //this is called when rewards screen is closed.
});
hangouttRewards.setRewardRedeemedCallback(function(){
    //do something
    //this is called when a reward is redeemed.
});
```

Figure 6: Setup callbacks

You can also find the above code snippets in template file provided along with the plugin.

Features:

1. Serendipitous rewarding - A way of identifying rewardable instances for the user in the application lifecycle, and rewarding the user at these instances.

One such moment is, when the user makes a purchase on your website.

To reward your users serendipitously, set the third parameter (boolean serendipity) of reward_user method to true. Please refer Figure 7.

You can call the method reward_user multiple times, if you've multiple rewarding moments in a given session.

```
<div id="rewards_section" style="width:500px; height:400px;">
</div>
<script type="text/javascript">
    // This labels array is used in analytics.
    var labels = [];
    labels[0] = "FinalPage";
    labels[1] = "AfterSomeEvent";

    var params = {};
    params['labels'] = JSON.stringify(labels);
    params['email'] = "user_email@example.com";
    params['title'] = "Enter your Title Message";
    params['sub_title'] = "We are offering this reward coz you have done
        some purchase on our website.";

    // In this mode, you need to specify the score i.e., params['score']
    parameter
    (default is set to 100).
    params['score'] = '200';

    hangouttRewards.reward_users("rewards_section", true, true, params,
        function(count){
            if (count > 0) {
                hangouttRewards.reward_users("rewards_section",
                    false, true);
            }
        });
</script>
```

Figure 7: Rewarding the user

Here are the method signatures of HangoutRewards Class:

a. **add**("points")

Parameters:

- I. Hangout Points you want to award to your user.

b. **get**()

Return Value:

Returns user's hangout points as string.

c. **reward_user**("document id of div", count_only, serendipity, overrideParameters, callback)

Parameters:

- I. Document id of the div where the layout is to be displayed. For example, "rewards_section" in Figure 4.
- II. Boolean value to get the count of rewards or not. If set to true, the callback function passed as fifth parameter, is called with number of available rewards.
- III. To reward your user serendipitously, set this parameter to true.
- IV. Pass additional parameters such as title, sub_title, score, labels etc. Any parameters you pass will override the existing values.
- V. If second parameter is set to true, this callback will be called.

Return Value:

On error, returns json string with error description.

d. **setRewardsCloseCallback** (function)

Parameters:

- I. Callback function, which will be called once the rewards screen, is closed.

e. **setRewardRedeemedCallback** (function)

Parameters:

- I. Callback function, which will be called once a reward, is redeemed.

APPENDIX

Upon registering a domain with Google reCaptcha, you are provided with a SiteKey and a SecretKey.

The SiteKey is necessary to render the captcha, and it is not required to keep it a secret. As such, it can be included directly in any webpage where you want captcha's to appear. The SecretKey, however, is used to verify that the captcha was solved successfully, and it is desirable to keep it a secret.

In our example above (Figure 2), we have kept both these keys public. This is because we do not have control on your server side scripts, and thus a working example is only possible by keeping the SecretKey public.

If you wish to keep the SecretKey private, you will need to place it on a server side script, which in turn makes the call to Google reCaptcha for verification. We show an example below. First, create a server side script which will in turn contact Google reCaptcha to verify whether the captcha was solved successfully. We show an example of how this can be done using PHP.

php/site_verify_helper.php

```
<?php
    $reponse = "";
    if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
        isset($_POST['response'])) {
        $google_site_verify_url =
            "https://www.google.com/recaptcha/api/siteverify";
        $secretKey = "your secret key";
        $grecaptcha_response = $_POST['response'];

        $curl = curl_init();
        curl_setopt_array($curl, array(
            CURLOPT_RETURNTRANSFER => 1,
            CURLOPT_URL => $google_site_verify_url,
            CURLOPT_POST => 1,
            CURLOPT_POSTFIELDS => array(
                'secret' => $secretKey,
                'response' => $grecaptcha_response
            )
        ));
        $response = curl_exec($curl);
        if (!$response){
        } else {
            echo $response;
        }
        curl_close($curl);
    }
?>
```

Now, change your initialization script in Step 3 above as follows:

```
<script type="text/javascript">
  HangouttRewards.pathToImages = "";
  HangouttRewards.captchaSiteKey = "Enter your site key";
  HangouttRewards.site_verify_url = "php/site_verify_helper.php";
  HangouttRewards.token_url = "link to token url";
  var hangouttRewards = new HangouttRewards();
</script>
```