

Example - API in PHP

<https://campus.barracuda.com/doc/79464623/>

The following is an example of how you can use the Security Awareness Training API written in PHP.

```
<?php
/*
 * This script shows how to authenticate and get an access token, then use
the access token to retrieve a campaign, and that campaign's results.
 *
 * This program assumes that you will have results for your first campaign to
show. If not, you should modify to use a campaign ID for a campaign that
does have results/stats to display.
 * You will need to adjust the values for the following constants:
 *
 * API_URL - change 'phishline_example' to the value for your instance.
Usually, this is 'phishline_yourcompanyname'.
 * AUTH_NAME - This will be the user account which has been given API
access.
 * AUTH_PW - The password for AUTH_NAME.
 * AUTH_API_KEY - the API key will be created by your administrator in
from 'System' -> 'API Keys' menu item.
 *
 */
namespace Barracuda\PhishLine\API;
if (file_exists('.example.env.php')) {
    print("Loading authentication constants from .example.env.php...\n");
    // set the 4 required constants from .example.env.php.
    require_once('.example.env.php');
} else {
    print("Loading default authentication constants...\n");
    // Otherwise, change the constants here.
    define('API_URL',
'https://api.phishline.com/phishline_example/rest/');
    define('AUTH_NAME', 'MyTestUser');
    define('AUTH_PW', 'MyTestPassword');
    define('AUTH_SSO_TYPE', '');
    define('AUTH_SSO_CONFIG_ID', '');
    define('AUTH_API_KEY', 'example-abcd-1234-defg-abcdabcd12345678');
}
// Do not change these items.
const AUTH_ENDPOINT = 'authenticate';
const CAMPAIGN_ENDPOINT = 'campaigns';
const CAMPAIGN_RESULTS_ENDPOINT = 'campaignresults';
const CAMPAIGN_RESULTSTATS_ENDPOINT = 'campaignresultstats';
```

```
// Begin the API calls, and print retrieved data to screen.
ApiUseExample::go();

// exit the program.
exit;

/**
 * ApiUseExample - example for authentication and usage of campaign related
stat endpoints for PhishLine API.
 *
 */
class ApiUseExample {

    /**
     * go - run the example to retrieve an access token, all campaigns, a
single campaign, stats and results from a single campaign.
     *
     * @static
     * @access public
     * @return void
     */
    public static function go() {
        print ("Step 1: get an access token.\n");
        $access_token = ApiUseExample::getAccessToken();
        print("Step 2: get all the campaigns.\n");
        $all_campaigns = ApiUseExample::getCampaigns($access_token);
        print_r($all_campaigns);
        print("Step 3: Get a single campaign ID from the previous call.\n");
            $campaign_id = $all_campaigns['data'][0]['id'];
        print("\tExample campaign ID is: ".$campaign_id."\n");
        print("Step 4. get a single campaign by id. For example purposes only.
            \t You could just use the campaign_id from Step 3, and skip right to
step 5.
            \t This is just an example of using the endpoint to retrieve a single
result if you know the campaign ID you want.\n");
            $single_campaign = ApiUseExample::getCampaigns($access_token,
$campaign_id);
        print_r($single_campaign);
        print("Step 5. get computed/summary campaign stats for a single
campaign.\n");
            $single_campaign_summary_stats =
ApiUseExample::getCampaignResultStats($access_token, $campaign_id);
        print_r($single_campaign_summary_stats);
        print("Step 6. get all raw results for a single campaign.\n");
            $single_campaign_raw_stats =
ApiUseExample::getCampaignResults($access_token, $campaign_id);
```

```
    print_r($single_campaign_raw_stats);
    print("\n\nExample complete!\n\n");
} // End function go
/**
 * getAccessToken - get an access token from the authenticate endpoint.
 *
 * @static
 * @access public
 * @return string the access token
 */
public static function getAccessToken() {
    $ch = curl_init();
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($ch, CURLOPT_URL, API_URL.AUTH_ENDPOINT);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'POST');
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, [
        'Content-Type: application/json; charset=utf-8',
    ]);
    $json_array = [
        'api_key' => AUTH_API_KEY,
        'bof_ticket_pw' => AUTH_PW,
        'bof_ticket_user' => AUTH_NAME,
        'sso' => AUTH_SSO_TYPE,
        'bof_sso_config_id' => AUTH_SSO_CONFIG_ID,
    ];
    $body = json_encode($json_array);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $body);
    $response = curl_exec($ch);
    if (!$response) {
        die('Error: "' . curl_error($ch) . '" - Code: ' . curl_errno($ch));
    }

    $vals = json_decode($response, TRUE);
    if (isset($vals['data']['access_token'])) {
        print("Access token created successfully: ".
    $vals['data']['access_token'] . "\n");
        curl_close($ch);
        return $vals['data']['access_token'];
    } else {
        echo 'HTTP Status Code: ' . curl_getinfo($ch, CURLINFO_HTTP_CODE) .
PHP_EOL;
        echo 'Response Body: ' . $response . PHP_EOL;
        print("ERROR: Something went wrong in \"__FUNCTION__\". Response was:
```

```
\n");
    curl_close($ch);
    die;
}

} // End function getAccessToken

/**
 * getCampaigns - get all campaigns in your instance, or a single campaign
if campaign_id is passed.
 *
 * @param string $access_token (required) The access token acquired from
the authenticate endpoint.
 * @param int $campaign_id (optional) The campaign ID. If no campaign ID
is passed, all results are retrieved.
 * @static
 * @access public
 * @return array results transformed to an associative array
 */
public static function getCampaigns($access_token, $campaign_id=NULL) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    // add the campaign_id to the endpoint if present. If not, we retrieve
all campaigns.
    if ($campaign_id != NULL) {
        curl_setopt($ch, CURLOPT_URL,
API_URL.CAMPAIGN_ENDPOINT.'/'.$intval($campaign_id));
    } else {
        curl_setopt($ch, CURLOPT_URL, API_URL.CAMPAIGN_ENDPOINT);
    }
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, [
        'Content-Type: application/json; charset=utf-8',
        'Authorization: Bearer '.$access_token,
    ]);
    $response = curl_exec($ch);
    if (!$response) {
        die('Error: "' . curl_error($ch) . '" - Code: ' . curl_errno($ch));
    }

    $vals = json_decode($response, TRUE);
    if (isset($vals['data'][0]['id'])) {
        print("Campaigns found. Returning all campaigns.\n");
        curl_close($ch);
        return $vals;
    }
}
```

```
    } else {
        echo 'HTTP Status Code: ' . curl_getinfo($ch, CURLINFO_HTTP_CODE) .
PHP_EOL;
        echo 'Response Body: ' . $response . PHP_EOL;
        print("ERROR: Something went wrong in ".__FUNCTION__."\n");
        curl_close($ch);
        die;
    }
} // End function getCampaigns
/**
 * getCampaignResultStats
 *
 * @param string $access_token (required) The access token acquired from
the authenticate endpoint.
 * @param int $campaign_id (required) The campaign ID.
 * @static
 * @access public
 * @return array results transformed to an associative array
 */
public static function getCampaignResultStats($access_token, $campaign_id)
{
    $ch = curl_init();
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
        curl_setopt($ch, CURLOPT_URL,
API_URL.CAMPAIGN_RESULTSTATS_ENDPOINT.'/'.$intval($campaign_id));
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_HTTPHEADER, [
            'Content-Type: application/json; charset=utf-8',
            'Authorization: Bearer '.$access_token,
        ]);
        $response = curl_exec($ch);
        if (!$response) {
            die('Error: "' . curl_error($ch) . '" - Code: ' . curl_errno($ch));
        }
        $vals = json_decode($response, TRUE);
        if (isset($vals['data'][0])) {
            print("Campaign Result Stats found. This is a summary of results.\n");
            curl_close($ch);
            return $vals;
        } else {
            echo 'HTTP Status Code: ' . curl_getinfo($ch, CURLINFO_HTTP_CODE) .
PHP_EOL;
            echo 'Response Body: ' . $response . PHP_EOL;
            print("ERROR: Something went wrong in ".__FUNCTION__."\n");
            curl_close($ch);
        }
    }
}
```

```
        die;
    }
} // End function getCampaignResultStats
/**
 * getCampaignResults - get raw results of your campaign.
 *
 * @param string $access_token (required) The access token acquired from
the authenticate endpoint.
 * @param int $campaign_id (required) The campaign ID.
 * @static
 * @access public
 * @return array results transformed to an associative array
 */
public static function getCampaignResults($access_token, $campaign_id) {
    $ch = curl_init();
        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($ch, CURLOPT_URL,
API_URL.CAMPAIGN_RESULTS_ENDPOINT.'/'.$intval($campaign_id));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_HTTPHEADER, [
        'Content-Type: application/json; charset=utf-8',
        'Authorization: Bearer '.$access_token,
    ]);
    $response = curl_exec($ch);
    if (!$response) {
        die('Error: "' . curl_error($ch) . '" - Code: ' . curl_errno($ch));
    }

    $vals = json_decode($response, TRUE);
    if (isset($vals['data'][0])) {
        print("Campaign Results found. This is all raw results.\n");
        curl_close($ch);
        return $vals;
    } else {
        echo 'HTTP Status Code: ' . curl_getinfo($ch, CURLINFO_HTTP_CODE) .
PHP_EOL;
        echo 'Response Body: ' . $response . PHP_EOL;
        print("ERROR: Something went wrong in ".__FUNCTION__."\n");
        curl_close($ch);
        die;
    }

} // End function getCampaignResults
} // End class ApiUseExample
```


© Barracuda Networks Inc., 2022 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.