



# **Publisher Integration Guide**

## **(Mobile Offer API)**

**Version 1.2**

Version	Publishing Date	Content & Changes
0.1	2011-08-05	Initial version of the Document
0.2	2011-08-17	Changed API path from /api/v1/feed to /feed/v1/offers
1.0	2011-08-31	Changed ip parameter from being mandatory to being optional
1.1	2011-10-07	Added offer_types parameter and offer_types response element
1.2	2011-10-25	Added android_id to request parameters

## Table of Contents

<b>Overview .....</b>	<b>4</b>
<b>A. Integration of the SponsorPay Mobile Offer API .....</b>	<b>5</b>
<b>Overview .....</b>	<b>5</b>
<b>Request Format .....</b>	<b>5</b>
Parameter Overview .....	5
Detailed explanations of all parameters .....	6
Example .....	8
Request Caching .....	9
<b>Response .....</b>	<b>9</b>
Basic response body structure .....	10
Detailed explanation of all elements.....	10
Available offer types.....	12
SponsorPay Response Codes.....	13
Example Responses .....	13
Signed Response .....	15
<b>B. Customization .....</b>	<b>16</b>
<b>Offer filters.....</b>	<b>16</b>

## Overview

SponsorPay is the European market leader in virtual currency monetization and will help you to monetize non-paying users and convert them into paying customers. SponsorPay lets your users earn or buy virtual currency or digital goods by completing offers such as participating in surveys, watching videos, making discounted purchases etc. SponsorPay manages billing, collection, fraud and customer support.

**We are happy to answer any questions:** [publishers@sponsorpay.com](mailto:publishers@sponsorpay.com)

## A. Integration of the SponsorPay Mobile Offer API

### Overview

To be able to use the Mobile Offer API, this feature has to be activated by SponsorPay. Please contact [publishers@sponsorpay.com](mailto:publishers@sponsorpay.com) if you want to know more. As soon as this feature is turned on for your application, SponsorPay will provide you with an API key, that you will need for all requests. Please don't share this API key with any third party. It is specific to you and your Application. You will be personally responsible for the use of the SponsorPay Mobile Offer API key.

### Request Format

All requests to the API have to use HTTP GET, no other HTTP methods are supported and will result in a HTTP Error 404. With the request, you provide SponsorPay with the needed information to send you all offers available to your user.

A typical request will look like following. All [parameters](#) that need to be customized by you are highlighted in **green**.

```
http://api.sponsorpay.com/feed/v1/offers.json?appid=[APP_ID]&uid=[USER_ID]&ip=[IP_ADDRESS]&locale=[LOCALE]&device_id=[DEVICE_ID]&ps_time=[TIMESTAMP]&pub0=[CUSTOM]&timestamp=[UNIX_TIMESTAMP]&offer_types=[OFFER_TYPES]&android_id=[ANDROID_ID]&hash_key=[HASH_KEY]
```

### Parameter Overview

Here is an overview of all possible parameters. You will find deeper explanations beneath this table.

	Parameter	Example Value	Description	Mand.
1	<a href="#">format</a>	json	The response format (lower case), selected by the 'extension' after 'offers'.	yes
2	<a href="#">appid</a>	157	The SponsorPay Application ID for your application.	yes
3	<a href="#">uid</a>	player1	The unique User ID, as used internally in your application.	yes
4	<a href="#">ip</a>	212.45.111.17	The IP address of the device of your user.	no
5	<a href="#">locale</a>	de	The locale used for the offer descriptions.	yes
6	<a href="#">device_id</a>	2b6f0cc904d137be2e1730235f5664094b831186	The ID of the device the user is using.	yes
7	<a href="#">ps_time</a>	1312211903	The creation date of the user's account in your game in Unix Timestamp format.	no
8	<a href="#">pub0</a>	campaign2	Custom parameters.	no
9	<a href="#">page</a>	2	The page of the response set that you are requesting.	no

10	timestamp	1312471066	The time the request is being sent by the device.	yes
11	offer_types	112	Filter the results based on type of offer.	no
12	android_id	ed0e0a7f802b7ad2	The Android ID of the device	yes/no
13	hashkey	eff26c67f527e6817b f6935c75f8cc5cc5cff ac2	The hash that signs the whole request.	yes

## Detailed explanations of all parameters

### 1. Response format (**format**)

The parameter defines the format of the response that you will get. Currently, you can choose either json or xml. This parameter is set by the extension after 'offers' in the URL, e.g. .../offers.**json**?... This parameter is mandatory.

### 2. Application ID (**appid**)

The parameter represents the SponsorPay Application ID of your application. You can find it in the Publisher Control Panel in the section 'Integration' of your application. This parameter is mandatory.

### 3. User ID (**uid**)

This parameter represents the unique User ID, as used internally in your application. This identifies the user so that virtual currency can later be attributed to his account via the callback request. Please note that the user ID has to remain constant so that users are prevented from completing an offer more than once. This parameter is mandatory.

### 4. IP Address (**ip**)

This parameter represents the IP address of the device of your user. SponsorPay is using it to filter the offers available to and targeted at your user. If the parameter is not given, the IP address of the request will be used. This parameter is optional.

Note: During integration, please also use valid 'external' IP addresses for testing, as responses to requests from reserved IP addresses (e.g. 127.0.0.1, 10.0.0.1, etc.) won't contain any offers.

### 5. Locale (**locale**)

This parameter represents the language, which will be used to describe the offers (e.g. the required action). Both iOS and Android offer ways to retrieve the preferred language of the user. This parameter is mandatory.

Note: Do not put country codes here, rather use the language codes, e.g. don't use US, use EN instead.

### 6. Device ID (**device\_id**)

This parameter represents the unique device ID of the mobile device, the user is currently using. For iOS devices this should be the UDID and respectively for Android devices the IMEI / MEID. This parameter is mandatory.

## 7. Account creation date (*ps\_time*)

This parameter represents the creation date of the user's account in your application in Unix Timestamp format. SponsorPay's targeting and fraud prevention systems use this information to optimize which offers to show. This parameter is optional.

## 8. Custom parameters (*pub0, pub1, ...*)

By using these, you can pass along custom parameters (e.g. *pub0*, *pub1*, ...) with each request to your callback URL by attaching these parameters to the API request URL. These parameters are optional.

## 9. Response page (*page*)

SponsorPay is limiting the amount of offers that are returned for a request. If there is more than one page of offers available for your user, an additional parameter will be returned within the response to your first offer request (see below). You can then use this parameter to define, which page of the response set you are requesting. By default, page is 1. This parameter is optional.

## 10. Timestamp (*timestamp*)

This parameter represents the time the request is being sent by the device, using the Unix Timestamp format. It is a security parameter to make sure, the request is valid and wasn't processed before. This parameter is mandatory.

## 11. Offer type filter (*offer\_types*)

This parameter allows the filtering of offers based on their type, so that only offers matching one or more of the specified categories will be returned. This parameter is optional.

More than one offer type can be given as a parameter, concatenated with a comma. For filtering, they will be combined using logical OR.

### Example 1:

- Request: ...&*offer\_types*=112&...
- The Response will contain all offers marked as 'Free'

### Example 2:

- Request: ...&*offer\_types*=101,112&...
- Response will contain all offers marked as 'Download' or marked as 'Free'.

A list of all available *offer\_type* IDs to use in the request filter can be found below: Available offer types.

Note: If you want to request all offers *except* one or more categories (*blacklisting*), please use the filter configuration in the SponsorPay Publisher Control Panel for that.

## 12. Android ID (*android\_id*)

This parameter represents the Android ID of the mobile device, the user is currently using. This parameter is only mandatory for Android applications and should not be provided otherwise.

### 13. Security Hash Key (hashkey)

This parameter represents the hashed key, signing the whole request. It is a security parameter to make sure, the request is valid and no data was tampered. This parameter is mandatory.

This hash key has to be generated by the device itself and has to follow strict rules:

1. Get all request parameters and their values (except hashkey)
2. Order theses pairs alphabetically by parameter name
3. Concatenate all pairs using '=' between key and value and '&' between the pairs.
4. Concatenate the resulting string with '&' and the API Key handed out to you by SponsorPay.
5. Hash the whole resulting string using SHA1.

The resulting hash-key is then appended to the request as a separate parameter.

### Example

This example is following the rules step-by-step.

#### Step 1: Gather all request parameters

Except the to-be-generated hashkey.

- appid = 157
- uid = player1
- ip = 212.45.111.17
- locale = de
- device\_id = 2b6f0cc904d137be2e1730235f5664094b831186
- ps\_time = 1312211903
- pub0 = campaign2
- page = 2
- timestamp = 1312553361

#### Step 2: Order all request alphabetically

- appid = 157
- device\_id = 2b6f0cc904d137be2e1730235f5664094b831186
- ip = 212.45.111.17
- locale = de
- page = 2
- ps\_time = 1312211903
- pub0 = campaign2
- timestamp = 1312553361
- uid = player1

#### Step 3: Concatenate all request parameters

Resulting string:

appid=157&device\_id=2b6f0cc904d137be2e1730235f5664094b831186&ip=212.45.111.17&locale=de&page=2&ps\_time=1312211903&pub0=campaign2&timestamp=1312553361&uid=player1



## **Step 4: Concatenate the resulting string with your API Key**

Your API Key, e.g.:

e95a21621a1865bcb3bee89c4d4f84

Resulting string:

appid=157&device\_id=2b6f0cc904d137be2e1730235f5664094b831186&ip=212.45.111.17&locale=de&page=2&ps\_time=1312211903&pub0=campaign2&timestamp=1312553361&uid=player1&e95a21621a1865bcb3bee89c4d4f84

## **Step 5: Hash the resulting string using SHA1**

Resulting string:

7a2b1604c03d46eec1ecd4a686787b75dd693c4d

Note: The SP Mobile Offer API expects the hashkey to be lower case.

The complete request URL to SponsorPay's Mobile Offer API for these parameters would then look like the following:

```
http://api.sponsorpay.com/feed/v1/offers.json?appid=157&device_id=2b6f0cc904d137be2e1730235f5664094b831186&ip=212.45.111.17&locale=de&page=2&ps_time=1312211903&pub0=campaign2&uid=player1&timestamp=1312471066&hashkey=7a2b1604c03d46eec1ecd4a686787b75dd693c4d
```

## **Request Caching**

There is currently no limit to the amount of requests done to the Mobile Offer API, as it is subject to fair use limits. Thus, it is recommended to cache the results requested for a unique user either in the application or on your server for at least 5-10 minutes. By doing this, you'll avoid unnecessary calls to our API backend and reduce the amount of data transferred to the user.

To guarantee that there are no expired offers shown to the user, please request new data if the cached data is older than 15 minutes.

## **Response**

The SponsorPay Mobile Offer API will respond with one of the following HTTP response codes:

### **Successful request**

- 200 OK
- 200 No Content (correct and successful request, but no offers are available for this user)

### **Error during request**

- 400 Bad Request (e.g. invalid or missing mandatory request parameters)
- 401 Unauthorized (e.g. wrong hashkey for this request and appid)
- 404 Not found (e.g. feed.xml instead of offers.xml, or using a non-existent version in the URL)
- 500 Internal Server Error (Error on the SponsorPay server)
- 502 Bad Gateway (Error on the SponsorPay server)

Upon a successful request, the response body will contain the offers in the format you chose upon requesting.

## Basic response body structure

For a successful response, depending on the format that you chose upon requesting, SponsorPay will either respond with JSON or XML. Both format follow basically the same structure, detailed in the following, with examples for both format.

- response (only present in XML format)
  - code
  - message
  - count
  - pages
  - information
    - app\_name
    - appid
    - virtual\_currency
    - country
    - language
    - support\_url
  - offers [ ]
    - link
    - title
    - teaser
    - required\_action
    - thumbnail
      - lowres
      - hires
    - offer\_types [ ]
      - offer\_type\_id
      - readable
    - payout
    - time\_to\_payout
      - amount
      - readable

## Detailed explanation of all elements

### *response*

The complete response will be wrapped inside this element.

### *code*

The code represents the general outcome of your request. See the table below for more details about all possible response codes. This code will be especially helpful during integration and for debugging.

### *message*

This element contains a more human readable explanation of the general outcome of your request, helping you to quickly identify possible problems.

Note: Don't use this element message to evaluate the response of your request, as we may change the texts over time. Only use the above mentioned element code for this.

## ***count***

This element contains the total number of offers available, depending on the parameters of your request.

## ***pages***

If there are more offers available than the amount that SponsorPay is responding with to a single request, this parameter will indicate how many more pages of results are available. You can then issue additional requests using the optional parameter page to request all the remaining offers.

## ***information***

This element will contain basic information about application and offers that are being returned, see below.

## ***app\_name***

The title of the application as entered in the SponsorPay Publisher Control Panel (<http://publishers.sponsorpay.com>).

## ***appid***

The application id used by SponsorPay to identify your specific application. You can find your ID in the SponsorPay Publisher Control Panel in the section 'Integration' of your application. This element should match the parameter appid of your request.

## ***virtual\_currency***

The name of your currency as entered in the SponsorPay Publisher Control Panel.

## ***country***

This element represents the country of the user, which SponsorPay derived from the IP given in the request (via parameter ip). The offers returned are all targeted to users from this country.

## ***language***

This element represents the language used to describe the offers (e.g. the required action) and should match the locale parameter of your request.

## ***support\_url***

This element represents the customer support URL specific to your app.

## ***offers [ ]***

This element contains an array of offers returned. Each offer contains more detailed information, as seen below.

## ***link***

This element represents the URL that the user should be sent to upon selecting this offer.

Note: Please open the link in the system browser instead of a web view, as some offers don't work within the context of a web view.

## ***title, teaser and required\_action***

These elements are describing the offer as well as giving a description of the steps needed to complete the offer. If there is no offer specific description available, the element `teaser` will contain the same content as the element `required_action`

## ***thumbnail***

This element contains URL's to the image used for the offer. Both a hi-res (`hires`) and a low-res (`lowres`) version of the image are included.

## ***offer\_types [ ]***

This element contains a list of one or more `offer_types` attributed to this offer, both as `offer_type_id` as well in a human readable format (using the language requested via the `locale` parameter). A list of all `offer_types` can be found below.

## ***payout***

This element contains the amount of virtual currency paid out to the user upon completion of the offer. It is calculated using the money payout and the exchange rate setup for your application in the SponsorPay Publisher Control Panel.

## ***time\_to\_payout***

This element contains the amount of seconds a user normally has to wait until the payout is done after completing an offer, both in the numerical representation as well as in a human readable format (using the language requested via the `locale` parameter). If the time to payout is unknown, the element will contain "N/A" as a string.

## **Available offer types**

This list is valid for both the `offer_types` request parameter as well as the response element `offer_types`.

<b>offer_type_id</b>	<b>readable</b>	<b>Description</b>
100	Mobile	Mobile subscription offers
101	Download	Download offers
102	Trial	Trial offers
103	Sale	Shopping offers
104	Registration	Information request offers
105	Registration	Registration offers
106	Games	Gaming offers
107	Games	Gambling offers
108	Registration	Data generation offers
109	Games	Games offers
110	Surveys	Survey offers
111	Registration	Dating offers
112	Free	Free offers
113	Video	Video offers

## SponsorPay Response Codes

In addition to the already mentioned HTTP response codes, we are adding a more specific response code the response body. Following a table containing all possible values for code, their corresponding message and an explanation.

HTTP Code	SponsorPay Response Code	Explanation
200	OK	Successful request, containing at least one offer in the response body.
200	NO_CONTENT	Successful request, but no offers are currently available for this user.
400	ERROR_INVALID_PAGE	A non-existing page was requested with the parameter page.
400	ERROR_INVALID_APPID	An invalid or missing application id (appid) was given as a parameter in the request.
400	ERROR_INVALID_UID	An invalid or missing user id (uid) was given as a parameter in the request.
401	ERROR_INVALID_HASHKEY	An invalid or missing hashkey for this appid was given as a parameter in the request.
400	ERROR_INVALID_DEVICE_ID	An invalid or missing unique device identifier (device_id) was given as a parameter in the request.
400	ERROR_INVALID_IP	An invalid IP address (ip) was given as a parameter in the request.
400	ERROR_INVALID_TIMESTAMP	An invalid, expired or missing timestamp was given as a parameter in the request.
400	ERROR_INVALID_LOCALE	An invalid or missing language (locale) was given as a parameter in the request.
400	ERROR_INVALID_ANDROID_ID	An invalid or missing Android ID (android_id) was given as a parameter in the request.
400	ERROR_INVALID_CATEGORY	An invalid or non-existing category (offer/mobile_types) was given as a parameter in the request.
500	ERROR_INTERNAL_SERVER_ERROR	An unknown error happened on the SponsorPay server.

## Example Responses

### JSON

```
{
  "code": "OK",
  "message": "OK",
  "count": "1",
  "pages": "1",
  "information": {
    "app_name": "SP Test App",
    "appid": "157",
    "virtual_currency": "Coins",
```

```

    "country": "US",
    "language": "EN",
    "support_url": "http://iframe.sponsorpay.com/mobile/DE/157/my_offers"
  },
  "offers": [
    {
      "title": "Tap Fish",
      "teaser": " Download and START ",
      "required_action": "Download and START",
      "link": "http://iframe.sponsorpay.com/mbrowser?appid=157&lpid=11387&uid=player1",
      "offer_types": [
        {
          "offer_type_id": "101",
          "readable": "Download"
        },
        {
          "offer_type_id": "112",
          "readable": "Free"
        }
      ]
    },
    {
      "thumbnail": {
        "lowres": "http://cdn.sponsorpay.com/assets/1808/icon175x175-2_square_60.png",
        " hires": "http://cdn.sponsorpay.com/assets/1808/icon175x175-2_square_175.png"
      },
      "payout": "90",
      "time_to_payout": {
        "amount": "1800",
        "readable": "30 minutes"
      }
    }
  ]
}

```

## XML

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <code>OK</code>
  <message>OK</message>
  <count>1</count>
  <pages>1</pages>
  <information>
    <app_name>SP Test App</app_name>
    <appid>157</appid>
    <virtual_currency>Coins</currency>
    <country>US</country>
    <language>EN</language>
    <support_url>http://iframe.sponsorpay.com/mobile/DE/157/my_offers</support_url>
  </information>
  <offers>
    <offer>
      <title>Tap Fish</title>
      <teaser>Download and START </teaser>
      <required_action>Download and START</required_action>
      <link>http://iframe.sponsorpay.com/mbrowser?appid=157&lpid=11387&uid=player1</link>
      <offer_types>
        <offer_type id="101">Download</offer_type>
        <offer_type id="112">Free</offer_type>
      </offer_types>
      <thumbnail>
        <lowres>http://cdn.sponsorpay.com/assets/1808/icon175x175-2_square_60.png</lowres>
        < hires>http://cdn.sponsorpay.com/assets/1808/icon175x175-2_square_175.png</ hires>
      </thumbnail>
      <payout>90</payout>
      <time_to_payout>
        <amount>1800</amount>
        <readable>30 minutes</readable>
      </time_to_payout>
    </offer>
  </offers>
</response>

```

## Signed Response

To ensure the integrity and validity of the response, we are adding a special security parameter to the HTTP response header. It is called X-Sponsorpay-Response-Signature and created by using rules very similar to those for the request signature. Although the use of this parameter is optional, it is strongly recommended to evaluate it on your side to prevent anyone from tampering any responses sent to you.

### Rules

1. Concatenate the full response body with your API key
2. Hash the whole resulting string using SHA1

### Example header

```
X-Sponsorpay-Response-Signature: 96cf9acc1c1d1800ba8aa1e095e3a032ec49bca3
```

## B. Customization

### Offer filters

By default, the SponsorPay Mobile Offer API contains all of our high-quality offers. However, we also give you the possibility to control precisely which offers are returned. Please note that limiting offers in that way will decrease the revenues that can be generated.

The category filter gives you a matrix view of offer categories and countries. This allows you to block offer categories on a per-country basis, giving you full control over the offers exposed to your users from around the world (note that **checking the boxes means blocking** the associated categories, not enabling them). If none of the boxes are checked, all categories are active in all countries.

On the bottom of the page you can select a default country. If SponsorPay introduces new countries, the filter settings from the default country will be used. In addition, you can block specific offers by selecting the [Offer Filter] tab.














[Category Filter](#)
[Offer Filter](#)
[Settings](#)

## Category Filter

Through the category filter, you can prevent offers of certain categories to be displayed in your SponsorPay iFrame. Please note that changes may require up to 60 minutes to take effect.

**Example:** If you do not want to show gambling offers to your users, simply block them by selecting the appropriate checkbox below.

If you want to block certain offers of a category, but allow others, please leave the category enabled and use offer filtering for more granular control. For further questions, feel free to contact your account manager or [publishers@sponsorpay.com](mailto:publishers@sponsorpay.com).

Which categories should be blocked?	Download	Trial	Sale	Information request	Registration	Gaming	Gambling	Skill games	Survey	Dating	Mobile subscription
											
All	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Afghanistan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Argentina	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Australia	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Austria	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
⋮											
United Arab Emirates	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
United Kingdom	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
United States	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Default Country

If SponsorPay introduces new countries, the filter settings from the default country will be used.