|  |
| --- |
| Metaps, Inc |
| test面向开发者的 |
| 基于Native应用的OfferBoard SDK嵌入手册 | |
|  |
| 3.0.2 |
| 2014/07/23 |

|  |
| --- |
|  |

目录

[1. 概要 3](#_Toc391311457)

[使用积分墙的方法 3](#_Toc391311458)

[用户下载成果的接收方法 3](#_Toc391311459)

[2. 对应版本 4](#_Toc391311460)

[SDK的版本 4](#_Toc391311461)

[Google Play Services SDK 4](#_Toc391311462)

[Android版本 4](#_Toc391311463)

[3. OfferBoard SDK的嵌入流程 5](#_Toc391311464)

[4. OfferBoard SDK的嵌入准备 6](#_Toc391311465)

[注册账号 6](#_Toc391311466)

[注册应用 6](#_Toc391311467)

[下载OfferBoard SDK 6](#_Toc391311468)

[在Android Project中嵌入SDK 6](#_Toc391311469)

[嵌入Google Play Services SDK 7](#_Toc391311470)

[Google Play Services SDK的嵌入准备 7](#_Toc391311471)

[在Library里面追加Google Play Services 7](#_Toc391311472)

[在Android-Manifest.xml中添加设定 8](#_Toc391311473)

[添加用于OfferBoard的Activity 8](#_Toc391311474)

[添加uses-permission 8](#_Toc391311475)

[添加meta-data 8](#_Toc391311476)

[应用作为媒体的设置 10](#_Toc391311477)

[5. 通过OfferBoard SDK显示积分墙 11](#_Toc391311478)

[在应用中导入SDK显示积分墙 11](#_Toc391311479)

[SDK的初始化 11](#_Toc391311480)

[验证初始化处理完成状况 12](#_Toc391311481)

[积分墙(OfferBoard)页面显示入口设定 13](#_Toc391311482)

[6. 接收成果通知 14](#_Toc391311483)

[共通注意事项 14](#_Toc391311484)

[接收安装成果通知 15](#_Toc391311485)

[通过指定服务器接收成果通知 15](#_Toc391311486)

[应用内接收成果 17](#_Toc391311487)

[7. 附录 18](#_Toc391311488)

[通过ProGuard等工具实现代码的模糊处理 18](#_Toc391311489)

[ProGuard的设定 18](#_Toc391311490)

[请将Google Play Services SDK也排除到加密对象之外 18](#_Toc391311491)

[修订记录 19](#_Toc391311492)

# 概要

作为开发者，嵌入OfferBoard积分墙型的激励型广告（reward Ads），自社广告（house Ads）和换量型广告（trade）等类型的奖励型（incentive）广告时，关于表示方法和成果的通知有以下多种方法可以选择。

* 报酬型广告（Reward Ads）

积分墙上的广告应用一旦被用户安装，开发者将会得到广告的收益，用户将会获得应用内的虚拟货币。

* 自社广告（House Ads）

在积分墙上免费刊载自家公司的应用。与报酬型广告一样，用户通过下载应用来获得应用内的货币。

* 换量型广告（Trade）

开发者之间可以相互在自己的积分墙上刊载对方的应用，自由的交换用户的安装数量从而实现互换用户的目的。用户可以通过下载应用来获得应用内的虚拟货币。

## 使用积分墙的方法

Native应用调用广告积分墙的方法如下：

**通过SDK表示广告的积分墙**

**Native应用**

## 用户下载成果的接收方法

用户一经完成积分墙上广告的下载，即作为广告成果发送至目标接收地点。开发者可以任意指定以下两种接收办法：1)直接发送到应用内，2）发送至开发者指定的服务器。

**Native应用**

**将该成果发送至应用**

或

**将该成果发送至您指定的服务器**

**成果を通知**

# 对应版本

在嵌入OfferBoard SDK version 3.0.0时，请注意如下版本信息是否符合。

## SDK的版本

本手册仅针对Metaps于Offerboard官网提供的以下最新SDK版本，请确认SDK的文件名如下所示：

『**metaps\_sdk\_android\_ver3\_0\_0.zip**』

若您当前所持SDK版本并非本最新版本，请前往官网获取最新版本的SDK。

## Google Play Services SDK

作为嵌入OfferBoard SDK的前提，您应用中必须已嵌入Google Play Services SDK 4.0.30以上版本。

## Android版本

支持Platform Version 1.6 / API Level 4以上的Android版本。

# OfferBoard SDK的嵌入流程

Native应用

**STEP 1-1 下载OfferBoard SDK（P6）**

**STEP.1 OfferBoard SDK的嵌入准备（P6）**

**STEP 1-2 在Android Project中嵌入SDK（P6）**

**STEP 1-3 嵌入Google Play Services SDK（P7）**

**STEP 1-4 应用作为媒体在OfferBoard网站上的设置（P10）**

**STEP 3-1 接收广告offer成果信息（P15）**

**STEP 3-2 应用内接收成果（P17）**

**STEP 3-2 通过指定服务器接收成果（P15）**

**STEP 2-1 SDK的初始化（P11）**

**STEP.2 通过OfferBoard SDK显示积分墙（P11）**

**STEP 2-2 设定积分墙的显示（P13）**

**STEP.3 接收应用的安装成果信息（P14）**

# OfferBoard SDK的嵌入准备

## 注册账号

在没有OfferBoard账号的情况下，请**<http://offerboard.metaps.com/>**进行账号注册。

账号注册完成后，请登陆。

## 注册应用

请在Offerboard官网上注册账号并登录后，通过“新登录应用程序”页面新建应用，为应用获取唯一的APP ID。

其中，“应用名”，“应用的简要说明”，“应用的详细说明”为必填项目。

**※应用注册后，Metaps将针对该新应用进行必要的审查。**

关于审查进度及后续支持，可通过以下格式内容邮件垂询：

Subject: Approval for App Registration

To： [support@metaps.com](mailto:support@metaps.com)

※在应用通过审查前，OfferBoard SDK积分墙将不会正常显示。

## 下载OfferBoard SDK

应用注册完成后，请通过首页进入“文档下载”页面，下载最新版的OfferBoard SDK 。下载后的文件名如下：

**metaps\_sdk\_android\_ver3\_x\_x.zip** (**x部分为版本数字**)

## 在Android Project中嵌入SDK

解压zip文件后，将其中的**metaps\_sdk\_android\_3\_x\_x.jar**文件添加至project文件夹，然后在您的工程里的Java build path中设置添加这个jar包。

(※若开发工具为Eclipse，请按照如下步骤设置)

1. 在PackageExplorer中target project下新建libs文件夹，并将**metaps\_sdk\_android\_3\_x\_x.jar**文件拷贝进去。
2. 在target project上右键选择属性（Properties）。
3. 左侧部分中选择「Java Build Path」选项。
4. 在中间部分的标签中选择「Libraries」标签。
5. 在右侧部分中点击「Add JARs…」后，点击“OK”对弹出的对话框，选择 **metaps\_sdk\_android\_ver3\_x\_x.jar**后确认,从而导入SDK的jar包。
6. 点击OK按钮结束。

## 嵌入Google Play Services SDK

### Google Play Services SDK的嵌入准备

请参考<http://developer.android.com/google/play-services/setup.html>（仅提供英文版）后嵌入Google Play Services SDK 4.0.30以上版本。

### 在Library里面追加Google Play Services

在使用Eclipse的情况下，请将library project的拷贝import至workspace中。

1. 拷贝library project「<android>/extras/google/google\_play\_services/libproject/google-play-services-lib/」
2. 将拷贝的Library project导入(import)到Eclipse中

（<android>是Android SDK的安装路径，请根据您所使用的环境替换该字段。）

* Eclipse下，请妥善备份 Library以防止文件被改写/覆盖等情况发生。

## 在Android-Manifest.xml中添加设定

### 添加用于OfferBoard的Activity

请在应用所使用的Activity中添加以下Activity。

* 竖版显示应用请添加如下代码（积分墙将以竖版进行显示 ）

|  |
| --- |
| <activity android:name="**net.metaps.sdk.WallActivity**" android:screenOrientation="portrait"  android:theme="@android:style/Theme.NoTitleBar"></activity> |

* 横板显示应用请添加如下代码（积分墙将以横板进行显示 ）

|  |
| --- |
| <activity android:name="**net.metaps.sdk.WallActivity**" android:screenOrientation="landscape" android:theme="@android:style/Theme.NoTitleBar"></activity> |

### 添加uses-permission

接下来，请在</application>标签的后面或<application>的前面添加android应用的许可（uses-permission）。若该许可已存在则无需重复添加。

|  |
| --- |
| <uses-permission android:name="**android.permission.INTERNET**"> </uses-permission>  <uses-permission android:name="**android.permission.ACCESS\_NETWORK\_STATE**"> </uses-permission> |

### 添加meta-data

请在<application>tag里面加入Google Play Services SDK的meta-data设定。若Google Play Services SDK已使用的情况下，则不需要此操作 。

|  |
| --- |
| <meta-data android:name="**com.google.android.gms.version**" android:value=”**@integer/google\_play\_services\_version**” /> |

#### 完整的AndroidManifest.xml设定示例

OfferBoard的嵌入准备，以下内容的设定将被完成：

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="net.metaps.sdk" android:versionCode="1" android:versionName="1.0.0" >  <uses-sdk android:minSdkVersion="4" />  <application android:name=".MainActivity" android:icon="@drawable/icon" android:label="@string/app\_name" >  ...  <!-- for metaps reward ads -->  <activity android:name="**net.metaps.sdk.WallActivity**" android:screenOrientation="portrait" android:theme="@android:style/Theme.NoTitleBar"></activity>  <meta-data android:name=”**com.google.android.gms.version**” android:value=”**@integer/google\_play\_services\_version**” />  </application>  <uses-permission android:name="**android.permission.INTERNET**"></uses-permission>  <uses-permission android:name="**android.permission.ACCESS\_NETWORK\_STATE**"></uses-permission>  </manifest> |

## 应用作为媒体的设置

在完成应用注册并通过审核之后，请登录Offerboard官网后在应用一览页面中目标应用的右侧点击“悬赏广告设定”下“编辑”按钮。

设定内容如下：

|  |  |
| --- | --- |
| **项目** | **内容** |
| **悬赏画面标题** | 目前的SDK版本没有使用这个字段，请在此填写应用名称即可。 |
| **用户的报酬单位（应用内货币）** | 媒体应用内的虚拟货币单位名称。比如“金币”。 |
| **用户报酬量** | 用户完成积分墙内广告应用下载任务后，媒体应用给予用户奖励的点数转换比例，即每100points相对应虚拟货币的个数。  假设100points=10个宝石，若积分墙内广告商对每个下载用户提供50个points的话，那么实际用户下载并启动该广告应用后，将会获得媒体应用内5个宝石作为奖励。 |
| **成果确认**  **（二选一即可）** | **向OfferBoard服务器发送确认通知**：若您选择通过应用内确认成果时使用的选项。该选项下将不需要填写下面的「成果确认URL」  **向目标应用服务器发送确认通知**：即向开发者自有服务器发送成果的确认通知的选项（若开发者需要户通过自己公司的服务器接收通知时，请选择此项） |
| **成果确认URL** | 即接收成果的服务器地址（URL）。成果确认信息将在用户安装后，通过该地址发送成果通知。 |

# 通过OfferBoard SDK显示积分墙

## 在应用中导入SDK显示积分墙

奖励型广告（报酬型广告/自社广告/换量型广告），即媒体应用的用户通过积分墙下载安装并启动广告应用后获取原应用内的虚拟货币作为奖励的一种广告类型。要使用OfferBoard广告仅需简单地嵌入SDK即可实现 。

※若需使用自社广告（house Ads）或换量型广告（trade）功能时，请通过另外的设置进行实现，SDK嵌入步骤相同 。

※若您以前也是OfferBoard的用户，并嵌入了之前版本的SDK，换为本次的版本后，在选择了发送成果信息到您指定的服务器的情况下，有一部分的发送信息会有所变动，具体请参照第14页「通过指定服务器接收成果」的红色字体部分。

## SDK的初始化

通过SDK展示积分墙之前，请务必调用Factory类（Class）的静态方法（Static Method）initialize()进行初始化。

initialize()须在其他进程（Process）执行之前进行调用。建议在安卓应用的Activity中的onResume方法中进行调用。

**（1）示例代码（将成果发送至指定服务器）**

|  |
| --- |
| import net.metaps.sdk.Factory;  import net.metaps.sdk.Const;  …  **Factory.initialize(this, null, "应用程序用户名", ”运行模式”);** |

**initialize()方法的各参数请参照下一页的表格**

**（2）示例代码（将成果发送至应用）**

|  |
| --- |
| import net.metaps.sdk.Factory;  import net.metaps.sdk.Const;  import net.metaps.sdk.Receiver;  …  Receiver receiver = new MyReceiver(); // 开发商请自行实现此类  **Factory.initialize(this, receiver, "应用程序用户名", ”运行模式”);** |

#### 关于Receiver

* 示例代码里面出现的Receiver类的目的是将用户下载积分墙上的广告应用所得到的point转换为开发者应用内的虚拟货币最后交付给用户，我们假设您通过实现了net.metaps.sdk.Receiver接口的MyReceiver方法来实现。详细的示例代码请参照第16页的「接收成果」这一章的「应用内接收成果」部分。

**initialize()方法的各个变量说明**

|  |  |
| --- | --- |
| **变量** | **内容** |
| this  （Activity） | 初始化时候运行的Activity的Instance |
| receiver （Receiver） | 实现了Receiver接口的Instance |
| 应用程序用户名（String） | Metaps为新注册应用发行的唯一ID 。请在登录Offerboard主页后，在应用一览的“详细内容确认・编辑”下点击目标应用对应的“编辑”按钮进入该应用详细页面即可确认“应用程序用户名”信息。 |
| 运行模式 | SDK的运行模式分为“测试环境模式”和“真实环境模式”。可根据需要调用初始化模式 。在测试完毕后请选择“真实环境模式” 「net.metaps.sdk.Const.SDK\_MODE\_PRODUCTION」。 |

**运行模式**

* 测试环境模式

填写「net.metaps.sdk.Const.SDK\_MODE\_TEST」。

此模式下，多次安装积分墙上同一广告应用后，每次均可获得奖励货币 。

* 真实环境模式

填写「net.metaps.sdk.Const.SDK\_MODE\_PRODUCTION」。

此模式下，多次安装积分墙上同一广告应用，仅第一次产生安装成果。

### 验证初始化处理完成状况

可通过以下方法（Method）验证初始化处理完成情况：

**验证初始化处理的示例代码**

|  |
| --- |
| import net.metaps.sdk.Factory;  …  **if ( !Factory.isInitializationFinished() ) {**  **// 初始化处理没有正常完成。或者在该环境下不能正常使用。**  } |

|  |  |
| --- | --- |
| **返回值** | **说明** |
| true | 初始化处理正常完成，可以启动积分墙。 |
| false | 初始化处理没有正常结束，或应用正处于不能正常使用的环境下。 |

### 积分墙(OfferBoard)页面显示入口设定

开发者可以根据应用特点在不同的位置设定积分墙显示入口。请将如下示例代码将 launchOfferWall()放置在目标展现位置。

**示例代码**

|  |
| --- |
| import.net.metaps.sdk.Factory;  …  **Factory.launchOfferWall(this,"end\_user\_ID","scenario");** |

指定end\_user\_ID和scenario参数后进行调用即可展示积分墙。

**launchOfferWall()方法的变量**

|  |  |
| --- | --- |
| **变量** | **说明** |
| **end\_user\_ID** | end\_user\_ID即媒体应用内用于识别用户所设置的字符串。该字符串应为该应用内**唯一字符串（unique）**。限半角英文数字，64位以内。  若没有进行用户的管理的话，请指定为[空字符]或者[null]。 |
| **scenario** | **scenario**字符串是标识在显示积分墙时应用所处的场景。限半角字符，64个字符以内。  仅在接收成果时除**end\_user\_ID**之外需要其他信息时使用。不需要时请指定为[空字符]或者[null]。 |

# 接收成果通知

用于接收媒体应用用户成功安装并启动积分墙上广告的成果有两种方式：1，成果发送到开发者指定的服务器上，再由开发者进行处理；2，直接发送成果至应用，由应用程序处理后直接将虚拟货币交付给用户。

## 共通注意事项

OfferBoard SDK­通过在便携设备内部查找是否安装了积分墙上的应用来向OfferBoard的服务器发送用户的安装成果信息。

为了确认积分墙上的应用被用户安装与否请将如下方法在应用内调用。例如您可以在用户下载安装完积分墙的应用后重返您的媒体应用时，调用该方法。

**Factory.runInstallReport();**

**测试环境模式**

为了便于测试，在该模式下，积分墙上的同一应用多次在设备上重复安装仍可以产生成果，并向用户发送虚拟货币。此外，如果有在积分墙上的下载运行后需要花些时间才能产生成果的应用，也会在下载运行后立即产生成果。（对于成果直接发送到应用内的成果通知接收方式，用户获得的报酬将一律为“10point”）

**真实环境模式**

在真实环境模式下，积分墙上同一应用多次安装启动仅在第一次产生成果。

## 接收安装成果通知

### 通过指定服务器接收成果通知

通过在登录OfferBoard官网后台页面设定指定服务器URL后，即可实现用户成功安装启动广告后从OfferBoard的服务器向指定开发者服务器发送成果信息。

设定方式如下：

1. 登陆OfferBoard官网后，点击上方[主页]菜单
2. 点击目标应用右侧“悬赏广告设定”项下“编辑”按钮，进入“悬赏广告设定”页面
3. 在“成果确认程序”中，选择“向客户本公司服务器通知”
4. 在“成果确认URL”中，填入接收成果通知的目标服务器。

**※“成果确认URL”中指定了SSL方式的情况下，需在目标接收通知的服务器端设定正式的SSL证书。（不支持self-signed certificate）**

**从Offerboard服务器向开发者的服务器发送的成果内容:**

|  |  |  |
| --- | --- | --- |
| **变量名** | **内容** | **数据类型** |
| **cuid** | 通过媒体应用中SDK发送而来的应用内用户的ID字符串 | VARCHAR(64) |
| **ucur** | 媒体应用内，用户完成任务后可获得的虚拟货币。在Offerboard主页的“悬赏广告设定”页面中，基于“用户报酬量”设定的换算标准给用户提供的虚拟货币的数值。[[1]](#footnote-1)该数值即为最终发给用户的虚拟货币数量 | INTEGER |
| **point** | 媒体应用从OfferBoard获得的本次成果的point数量。  ※在自社广告（house型广告）下，该值将会被置为0。 | INTEGER |
| **grid** | 通过Offerboard获得成果的ID。每次获得都会产生唯一（unique）的字符串。  **※在自社广告/换量型广告情况下，该字段可能产生重复值。请确认是否存在【house】【trade】变量值来判断。** | VARCHAR(32) |
| **scn** | 初始化显示积分墙方法时设定的**scenario**字符串 | VARCHAR(64) |
| **aid** | Offerboard上广告应用的ID | VARCHAR(64) |
| **house** | 用于判断是否为自社广告（house型广告）的变量。如非自社型广告（house型广告），该变量将不会被发送。  1 : 指自社广告（house型广告） | VARCHAR(8) |
| **trade** | 用于判断是否为换量型广告（trade型广告）的变量。若非换量型广告（trade型广告），该变量将不会被发送。  1 : 指换量型广告（trade型广告） | VARCHAR(8) |

**※上述变量由GET方法发送。**

**※ 请注意在使用「自社广告」「换量型广告」时候，【grid】的数值有可能不唯一，可能有重复的数值。**

**※「udid」参数已从通知内容中删除。**

* 报酬型广告（reward Ads）的发送成果的URL例子

https://example.com/metaps?cuid=endUserId&ucur=30&point=50&grid=1234567&ucur=30&scn=scenario&aid=APPID0001

* 自社广告（house Ads）的发送成果的URL例子

https://example.com/metaps?cuid=endUserId&ucur=30&point=0&grid=101234567&scn=scenario&aid=APPID0001**&house=1**

* 换量型广告（trade）的发送成果的URL例子

https://example.com/metaps?cuid=endUserId&ucur=30&point=50&grid=201234567&scn=scenario&aid=APPID0001**&trade=1**

**开发者指定服务器在接收到成果后，请向给Offerboard服务器返回Response：**

|  |  |
| --- | --- |
| **BODY** | **内容** |
| **1** | 用户得到虚拟货币后正常结束 |
| **0** | 用户未得到虚拟货币，但也不再希望OfferBoard继续尝试发送成果信息 |
| 上述之外 | 发生异常，希望Offerboard继续发送成果信息（OfferBoard将会记录返回的24byte以内的字符串） |

**注意**

* Response的status请返回200。若返回其他的status都会被判断为发生异常后结束。
* 请将Response设定为平文（plain text）。如果Response包含有其他值时，会被判断为发生异常后结束。
* 发生异常结束的情况下，在5分钟后到6个小时之内Offerboard将尝试重新发送成果，上限10次。  
  （在测试环境模式下将不会重复发送成果）

### 应用内接收成果

OfferBoard的奖励型广告可直接将用户成功安装并启动OfferBoard广告的成果通知发送到媒体应用中。通过应用的定时处理，应用内接收成果进行处理需要添加一个实现（implements）**net.metaps.sdk.Receiver**接口的类，并在实例化后，在初始化SDK（请参照第10页的示例代码）的时候将该对象的引用传入。

Receiver接口的retrieve、finalizeOnError方法将在成果正常时返回true，失败时返回false。在返回false时，这些方法将持续被回调执行直到成功为止。

**Receive的实现例子（通过MyReceiver类将Receive实现并支付虚拟货币的例子）：**

|  |
| --- |
| **import net.metaps.sdk.Receiver;**  **…**  class MyReceiver implements Receiver{  private String userId;  private String userName;  private int point = 0;  …  public boolean retrieve(int currencyAmount, Offer offer){  …  **this.point += currencyAmount;**  **return true;**  }  public boolean finalizeOnError(Offer offer){  …  return true;  }  } |

**（注意）Receiver不仅可用于提供虚拟货币，也可用作用户类的状态变化和物品的交付等处理。**

**直接调用实例化后的MyReceiver类的示例代码**

|  |
| --- |
| import net.metaps.sdk.Factory;  …  **Factory.initialize(this, new MyReceiver(), "应用程序用户名", "运行模式");** |

此外，在应用中任何时候都可以接收成果并进行确认。在任何地方都可以调用Factory.runInstallReport方法。

**在Activity直接确认成果的示例代码**

|  |
| --- |
| import net.metaps.sdk.Factory;  …  **Factory.runInstallReport();** |

※ 请注意在使用runInstallReport()方法时必须要在Activity里面先调用Factory.initialize方法。

# 附录

## 通过ProGuard等工具实现代码的模糊处理

在使用ProGuard等工具将代码加密的情况下，请确保以下内容不再加密对象内。

### ProGuard的设定

使用ProGuard时，请修改proguard.cfg文件将metapsSDK排除。

请在Proguard.cfg文件内追加如下代码。

|  |
| --- |
| -keep class net.metaps.\*\* {  \*;  } |

#### Android API Level 17以上版本的情况下

请确认下面的代码是否已经添加。

|  |
| --- |
| -keepattributes \*Annotation\* |

### 请将Google Play Services SDK也排除到加密对象之外

请参考<http://developer.android.com/google/play-services/setup.html#Proguard>（仅提供英文版），并确保将Google Play Services SDK排除到加密处理对象外。

|  |
| --- |
| -keep class com.google.android.gms.\*\* {  \*;  }  -keep class \* extends java.util.ListResourceBundle {  protected Object[][] getContents();  }  -keep public class com.google.android.gms.common.internal.safeparcel.SafeParcelable {  public static final \*\*\* NULL;  }  -keepnames @com.google.android.gms.common.annotation.KeepName class \*  -keepclassmembernames class \* {  @com.google.android.gms.common.annotation.KeepName \*;  }  -keepnames class \* implements android.os.Parcelable {  public static final \*\* CREATOR;  } |

## 修订记录

|  |  |
| --- | --- |
| **2014-07-23** | 根据SDK 3.0.2修订了该说明书。 |
| **2014-07-11** | 根据SDK 3.0.1修订了该说明书。 |
| **2014-06-20** | 追加了Google Play Services SDK的版本号。  更改了一部分ProGuard的设定内容。 |
| **2014-06-02** | 根据SDK 3.0.0修订了该说明书。  追加Google Play Services SDK的内容。  接口修改。  AndroidManifest.xml中的内容修订。 |
| **2014-03-12** | 由于Metaps的网站整体URL变更，导致[用户登录]页面的URL变更 |
| **2014-01-17** | 与Ver2.2.2合并更新 |
| **2014-01-10** | 初版 |

1. 主页应用一览的“悬赏广告设定”→“编辑”页面内通过“用户报酬量”中设定的数值为基准计算出的提供给用户的虚拟货币报酬。 [↑](#footnote-ref-1)