**保護中: PCアプリ向け課金APIについて**

**保护中：面向PC应用的支付API**

PC向け課金APIを利用することにより、mixiポイントを使ったアイテム課金をmixiアプリPC版に組み込むことができます。ここでは、mixiアプリにてPC向け課金APIをどのように利用して組み込むかを説明いたします。

通过使用面向PC的支付API，可以将使用MIXI点数的道具支付组装到MIXI应用PC版中。在此说明在MIXI应用中如何使用组装面向PC的支付API

**制限事項**

**限制事项**

ここで説明するPC向け課金APIの利用は、mixiより許可されたSAPが開発したmixiアプリのみで利用することができます。その際、SAPは開発するmixiアプリごとに許可を得なければなりません。

在此说明的面向PC的支付API的使用，只能用于经过MIXI许可的SAP开发的MIXI应用。届时每个SAP开发的MIXI应用必须得到许可

**利用手順**

**使用步骤**

PC向け課金APIは、OpenSocial Virtual Currency APIをベースに、mixi Platform独自の決済確認フローを追加したAPIです。決済内容を保護するために、mixiアプリから規定のJavaScript APIを呼び出すだけでなく、いくつかバックエンド側(SAP側の外部サーバ)上で行わなければならない処理があります。

面向PC的支付API是基于OpenSocial Virtual Currency API，追加了mixi Platform独自的结算确认流程的API。为保护结算内容，不仅仅从mixi应用呼叫规定的JavaScript API，还有必须在若干个后台（SAP方面的外部服务器）上进行的处理

OpenSocial Virtual Currency APIについての詳しいリファレンスは、以下を参照ください。

关于OpenSocial Virtual Currency API详细的参考，请参照以下内容

【OpenSocial Virtual Currency API】  
<https://docs.google.com/Doc?id=dfjcf7w4_10ddst9xd9&hl=en>  
【上記の日本語訳】

【以上的日本语翻译】  
<http://docs.google.com/View?id=dhsg598c_994hdb9pwcf>

**事前準備**

**事前准备**

PC向け課金APIでは、通信内容の妥当性検証を行うために、署名の生成や検証を行うことになります。使われる署名は、HMAC-SHA1にて生成された文字列となります。この際に使用されるキーは、mixiアプリの設定画面(edit\_appli.pl)にて表示されるConsumer Secretです。予めこのConsumer Secret文字列をその画面より入手しておきます。

面向PC的支付API中，为了进行通信内容的正确性检查，进行署名的生成、验证。使用的署名是用HMAC-SHA1生成的字符串。这时使用的键是在mixi应用的设定画面（edit\_appli.pl）显示的Consumer Secret。预先在那个画面得到这个Consumer Secret字符串

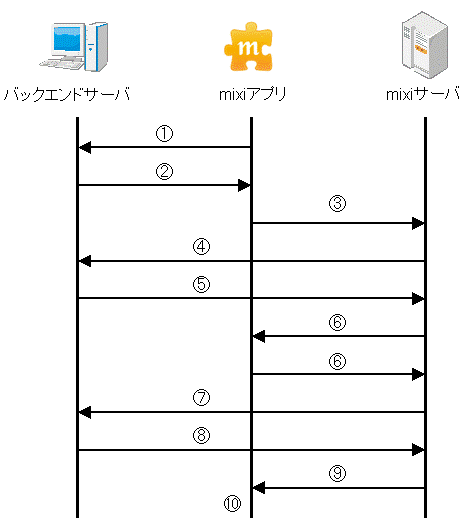
**基本的な処理フロー**

**基本的处理流程**

mixiアプリ内でPC向け課金APIを利用する際の代表的な処理フローを以下に示します。

mixi应用中使用面向PC的支付API时的代表性的处理流程如下

后台 mixi应用 mixi服务器



1. mixiアプリからバックエンドサーバに決済情報の作成を依頼する
2. 从mixi应用向后台服务器请求结算信息的制作
3. バックエンドサーバにて決済情報を作成し、mixiアプリに返却する
4. 在后台服务器制作结算信息，返回给mixi应用
5. 決済情報を元に、決済のためのAPIを呼び出す
6. 基于结算信息，呼叫结算API
7. mixiサーバからバックエンドサーバに、ポイントコードが送信される
8. mixi服务器向后台服务器发送Point code
9. バックエンドサーバはポイントコードを保持し、mixiサーバに結果を返却する
10. 后台服务器保持Point code，向mixi服务器返回结果
11. 決済確認ポップアップ画面が表示され、ユーザが確認を行う
12. 显示结算确认弹出画面，用户进行确认
13. mixiサーバからバックエンドサーバに、決済処理ステータスが送信される
14. 从mixi服务器向后台服务器发送结算处理状态
15. バックエンドサーバは決済情報を確認後、mixiサーバに結果を返却する
16. 后台服务器确认结算信息后，将结果返回给mixi服务器
17. 決済完了ポップアップ画面が表示され、ユーザがそれを消去する
18. 显示结算完了弹出画面，用户将其消去
19. 指定したコールバック関数が呼び出され、結果が渡される
20. 调用指定的回叫（callback）函数，传递结果

**課金APIの利用宣言**

**支付API的使用声明**

mixiアプリPC版から課金APIを利用するためには、Gadget XMLファイルにて課金APIを利用するための宣言を記述する必要があります。課金APIは「opensocial-payment」Featureとして提供されるため、以下のようにRequire要素を使って利用宣言を行ってください。

为了从mixi应用PC版使用支付API，需要在Gadget XML中记述使用支付API的声明。支付API是作为「opensocial-payment」Feature提供的，请按照如下内容使用Require要素进行使用声明

<Module>

<ModulePrefs ...>

<Require feature="opensocial-payment" />

...

</ModulePrefs>

</Module>

**決済情報の作成**

**结算信息的创建**

ユーザが何らかのアイテムの購入をmixiアプリ内で希望した際に、最初にmixiアプリからバックエンドサーバに対して、決済情報の作成を依頼します。この時点で、購入対象のアイテムIDが決定します。

在mixi应用中用户希望购入某些道具时，首先从mixi应用对后台服务器请求结算信息的创建。此时，决定购入对象的道具ID

作成の依頼は、gadgets.io.makeRequest()関数を利用すると良いでしょう。その際に、署名付きリクエストを利用すると、バックエンド側サーバにて通信内容の妥当性検証を行うことが可能です。

创建的请求使用gadgets.io.makeRequest()函数就可以了。那时，使用附加署名的请求的话，在后台一侧的服务器中可以进行通信内容的正确性检查

以下に、サンプルコードを記載いたします。

如下记载演示代码

var itemId = 123; // アイテムID 道具ID

var url = "http://sap.server.host/create\_order"; // バックエンドサーバのURL 后台服务器的URL

var p = {};

p[gadgets.io.RequestParameters.AUTHORIZATION] = gadgets.io.AuthorizationType.SIGNED;

p[gadgets.io.RequestParameters.POST\_DATA] = gadgets.io.encodeValues({item\_id: itemId});

p[gadgets.io.RequestParameters.METHOD] = gadgets.io.MethodType.POST;

gadgets.io.makeRequest(url, function(response) {

// do something

}, p);

**決済情報の作成**

**结算信息的创建**

決済情報の作成要求を受け取ったバックエンドサーバでは、アイテムIDを元に必要な情報を準備し、それらを元に署名を作成します。これらを結果としてmixiアプリに返却します。

在收到结算信息创建请求的后台服务器中，准备基于道具ID的必要的信息，基于这些信息创建署名。将这些作为结果返回给mixi应用

決済情報として必要な項目を以下に示します。

作为结算信息需要的项目如下

|  |  |
| --- | --- |
| **項目名** | **説明** |
| callback\_url | mixiサーバから呼び出されるバックエンドサーバのURL  从mixi服务器呼叫的后台服务器URL |
| inventory\_code | バックエンドサーバ側でこの決済を特定するための任意の文字列  后台服务器中，为了特别指定这个结算的任意的字符串 |
| is\_test | 決済処理のテストの場合は”true”, 実際の決済の場合は”false”  结算处理的测试的情况为“true”,实际结算的情况为“false” |
| item\_id | アイテムID  道具ID |
| item\_price | アイテムのポイント数  道具的点数 |

まず、これらの項目を「&」で連結して、1つの文字列を作成します。その際に、項目名はアルファベット順とします。また、全ての項目値はURIエスケープを行っておきます。

首先，将这些项目用「&」连接，创建一个字符串。届时，项目名按照字母顺序排列。还有，所有的项目值预先进行URI escape

callback\_url=http%3A%2F%2Fsap.server.host%2Fcreate\_order&inventory\_code=123&is\_test=true&item\_id=123&item\_price=500

この文字列に対して、さらに以下の加工を行います。その結果は、上記のパラメータに対する署名となります。

对这个字符串再次进行以下的加工，结果是对以上参数的署名

1. 上記文字列全体をURIエスケープする – (1)
2. 将以上字符串全体进行URI escape– (1)
3. 予め入手したConsumer Secretの末尾に”&”を付与する – (2)
4. 在预先得到的Consumer Secret的末尾添加”&” – (2)
5. (2)をキーとしてHMAC-SHA1アルゴリズムを使用し、(1)のダイジェスト値を得る – (3)
6. 将(2)作为键，使用HMAC-SHA1算法，获得(1)的摘要值
7. (3)をBASE64エンコードする
8. 将(3)进行BASE64 encoding

上記のパラメータ値および生成した署名を、mixiアプリに結果として返却します。

以上参数值以及生成的署名，作为结果返回给mixi应用

※ URIエスケープは、こちら(<http://oauth.net/core/1.0a#encoding_parameters>)に則って行ってください。

※ URI escape，请按照(<http://oauth.net/core/1.0a#encoding_parameters>)进行

**決済APIの呼び出し**

**呼叫结算API**

バックエンドサーバから得られた決済情報に基づいて、決済のためのAPIを呼び出します。このAPIとして、opensocial.requestPayment()関数を利用します。

基于从后台服务器得到的结算信息，呼叫结算API。作为这个API，使用opensocial.requestPayment()函数

var paymentHandlerUrl = "http://sap.server.host/create\_order";

var params = {};

params[opensocial.Payment.Field.AMOUNT] = 250;

params[mixi.Payment.Field.ITEM\_NAME] = "エクスカリバー";

params[mixi.Payment.Field.SIGNATURE] = "TmihyproUc02HOh17W0uz++WdYM=";

params[mixi.Payment.Field.ITEM\_ID] = 123

params[mixi.Payment.Field.IS\_TEST] = 'true';

params[mixi.Payment.Field.INVENTORY\_CODE] = 123;

params[opensocial.Payment.Field.PAYMENT\_TYPE] = opensocial.Payment.PaymentType.PAYMENT;

var payment = opensocial.newPayment(params);

opensocial.requestPayment(payment, paymentHandlerUrl, function(response) {

// do something

});

まず、決済オブジェクトをopensocial.newPayment()関数で生成します。その際に必要となる項目は、以下となります。

首先，用opensocial.newPayment()函数生成结算对象，需要的项目如下

|  |  |
| --- | --- |
| **定数** | **説明** |
| opensocial.Payment.Field.AMOUNT | アイテムのポイント数  道具点数 |
| mixi.Payment.Field.ITEM\_NAME | アイテム名  道具名 |
| mixi.Payment.Field.SIGNATURE | 決済情報の署名文字列  结算信息的署名字符串 |
| mixi.Payment.Field.ITEM\_ID | アイテムID  道具ID |
| mixi.Payment.Field.IS\_TEST | 決済処理のテストの場合は”true”, 実際の決済の場合は”false”  结算处理测试的情况为“true”，实际的结算时为“false” |
| mixi.Payment.Field.INVENTORY\_CODE | SAP側でこの決済を特定するための任意の文字列  在SAP一侧，为了特别指定这个结算的任意字符串 |
| opensocial.Payment.Field.PAYMENT\_TYPE | 決済種別  结算类别  (opensocial.Payment.PaymentType.PAYMENT) |

実際の決済処理は、opensocial.requestPayment()関数によって呼び出します。この関数は、以下の引数を受け付けます。

实际的结算处理由opensocial.requestPayment()函数呼叫。这个函数接受以下参数

1. 決済オブジェクト
2. 结算对象
3. mixiサーバから呼び出されるバックエンドサーバのURL
4. 从mixi服务器呼叫的后台服务器的URL
5. コールバック関数
6. 回叫（callback）函数

requestPayment()関数を呼び出すと、mixiサーバに決済開始要求がmixiアプリから送信されます。

呼叫requestPayment()函数的话，结算开始请求由mixi应用发往mixi服务器

**残ポイントが足りなかった場合**

**剩余点数不足的情况**

決済情報に含まれるポイント数について、対象ユーザが必要なポイント数を持っていなかった場合は、ユーザに残高不足を表すポップアップ画面が表示されます。ユーザがポイントチャージを行い、ユーザがポップアップ画面から続行を選択することで、再度決済処理が行われます。

关于包含在结算信息中的点数，对象用户没有需要的点数的时候，向用户显示余额不足弹出画面。用户进行点数充值，用户从弹出画面选择继续，再次进行结算处理

**ポイントコードの送信**

**Point code 的发送**

mixiアプリからrequestPayment()関数が呼び出された後、mixiサーバはpaymentHandlerUrl引数にて指定されたバックエンドサーバのURLにポイントコードを送信します。その際に送られるパラメータは以下となります。

从mixi应用调用requestPayment()函数后，mixi服务器向用paymentHandlerUrl参数指定的后台服务器的URL发送Point code。届时发送的参数如下

|  |  |
| --- | --- |
| **パラメータ** | **説明** |
| opensocial\_app\_id | mixiアプリのID  mixi应用的ID |
| opensocial\_owner\_id | mixiアプリの利用者のユーザID  mixi应用的使用者的用户ID |
| inventory\_code | バックエンドサーバにて採番された任意の文字列  后台中采集编号的任意字符串 |
| point\_code | この決済を特定するためのポイントコード  为了特别指定这个结算的Point code |
| item\_id | アイテムのID  道具ID |
| item\_price | アイテムのポイント数  道具点数 |
| item\_name | アイテムの名前  道具名 |
| signature | バックエンドサーバにて生成された署名文字列  后台服务器中生成的署名字符串 |
| is\_test | 決済処理のテストの場合は”true”, 実際の決済の場合は”false”  结算处理测试的情况为“true”，实际的结算时为“false” |

このリクエストは、2-legged OAuthによる署名が施されています。署名は、Authenticationヘッダに記載されています。不正な購入を防ぐために、必ずこの署名の検証を行ってください。検証方法は、mixiアプリモバイルでの仕様と同じです。ただし、検証時のベース文字列として、上記にあげたパラメータに関しても含めるようにしてください。

这个请求施加了依据2-legged OAuth的署名。署名在Authentication头部记载。为了防止不正常的购入，请务必对这个署名进行验证。验证方法和mixi应用手机中的规格相同。但是作为验证时的基础（BASE）字符串，以上列举的参数也请包含

**結果の返却**

**结果的返回**

署名の検証後、ポイントコードなど決済内容をバックエンドサーバ側で保持します。その後、成功のレスポンスを返却します。

署名验证后，在后台服务器一侧保持Point code等结算内容。之后返回成功的响应

Status: 200

Content-Type: text/plain

OK

このレスポンスは、10秒以内に行う必要があります。10秒以内にレスポンスがなかった場合、決済処理は行われません。

这个响应需要在10秒以内进行。10秒以内没有响应的情况，不进行结算处理

**ユーザによる決済確認**

**用户进行的结算确认**

mixiサーバが結果を受信後、ユーザに決済確認のためのポップアップ画面が表示されます。そこでユーザは、決済するかどうかの判断を行います。ユーザが決済を要求することで、mixiサーバ内で決済処理が実行されます。

mixi服务器收到结果后，向用户显示结算确认弹出画面。于是用户判断是否结算。用户通过要求结算，在mixi服务器中执行结算处理

**決済処理ステータスの送信**

**结算处理状态的发送**

mixiサーバ内で決済処理が行われた後に、再度バックエンドサーバに決済確定のためのリクエストが送信されます。リクエストは、 requestPayment()関数の呼び出し時に渡されたpaymentHandlerUrlで指定したURLとなります。リクエストに含まれるパラメータは、以下となります。

Mixi服务器中进行结算处理后，再次向后台服务器发送结算确认的请求。请求是呼叫requestPayment()函数时传递的用paymentHandlerUrl指定的URL。请求中包含的参数如下

|  |  |
| --- | --- |
| パラメータ名 | 説明 |
| opensocial\_app\_id | mixiアプリのID  mixi应用ID |
| opensocial\_owner\_id | mixiアプリの利用者のユーザID  mixi应用的使用者的用户ID |
| point\_code | この決済を特定するためのポイントコード  为了特别指定这个结算的Point code |
| status | ステータス 10=購入成功  状态 10=购入成功 |
| updated | 購入確定日時(UTC)  购入确定日期时间（UTC） |

このリクエストは、2-legged OAuthによる署名が施されています。署名は、Authenticationヘッダに記載されています。不正な購入を防ぐために、必ずこの署名の検証を行ってください。検証方法は、mixiアプリモバイルでの仕様と同じです。ただし、検証時のベース文字列として、上記にあげたパラメータに関しても含めるようにしてください。

这个请求施加了依据2-legged OAuth的署名。署名在Authentication头部记载。为了防止不正常的购入，请务必对这个署名进行验证。验证方法和mixi应用手机中的规格相同。但是作为验证时的基础（BASE）字符串，以上列举的参数也请包含

**結果の返却**

**结果的返回**

署名の検証後、リクエストに含まれる各パラメータに基づいて、バックエンドサーバ側にて決済内容の確認とアイテム購入処理などを行います。その後、その後、成功のレスポンスを返却します。

署名验证后，基于包含在请求中的各个参数，在后台服务器一侧确认结算内容、进行道具购入处理等。之后返回成功的响应

Status: 200

Content-Type: text/plain

OK

このレスポンスは、10秒以内に行う必要があります。10秒以内にレスポンスがなかった場合、決済確定の処理は行われません。

这个响应需要在10秒以内进行。10秒以内没有响应的情况，不进行结算确定的处理

**ユーザへの決済完了表示**

**向用户显示结算完了**

バックエンドサーバから結果を受け取ったmixiサーバは、ユーザに決済完了を示すポップアップ画面を表示します。ユーザは、その画面を閉じます。

从后台服务器得到结果的mixi服务器，向用户显示结算完了的弹出画面。用户关闭那个画面

**コールバック関数の呼び出し**

**回叫（callback）函数的呼叫**

最後に、requestPayment()関数にて引数に指定されたコールバック関数が呼び出されます。この際、引数にはopensocial.ResponseItemオブジェクトが渡されます。

最后在requestPayment()函数中呼叫在参数中指定的回叫（callback）函数。此时，参数传递opensocial.ResponseItem对象

決済に成功したかどうかをmixiアプリにて確認するために、ResponseItemオブジェクトのhadError()関数を利用します。 hadError()関数の結果がtrueだった場合は、何らかの理由で決済が完了しなかったことを示しています。その場合は、 getErrorCode()関数によりエラーコードを取得し、適切なエラー処理を行ってください。

为了mixi应用中确认结算是否成功，使用ResponseItem对象的hadError()函数。hadError()的结果是true的情况，表明因某种理由结算没有完成。那时用getErrorCode()函数取得错误代码，请进行适当的错误处理

opensocial.requestPayment(payment, paymentHandlerUrl, function(response) {

if (response.hadError()) {

var code = response.getErrorCode();

if (code == opensocial.Payment.ResponseCode.USER\_CANCELLED) {

// ユーザが決済をキャンセルしたときの処理 用户取消结算时的处理

} else {

// その他のエラーに対する処理 其他错误的处理

}

} else {

var data = response.getData(); // opensocial.Paymentオブジェクト opensocial.Payment对象

// 決済完了時の処理 结算完成时的处理

}

});

決済処理完了後のopensocial.Paymentオブジェクトを、ResponseItemオブジェクトのgetData()関数によって取得することが可能です。各値は、PaymentオブジェクトのgetField()関数に取得したい項目の定数を渡すことで得ることができます。

结算处理完成后的opensocial.Payment对象，可以用ResponseItem对象的getData()函数取得。各个值可以通过向Payment对象的getField()函数传递想要取得的项目的常量的方式取得