
Call2Pay API Event v2.0

Copyright © 2008 micropayment™ gmbh

Sämtliche Inhalte dieses Dokuments unterliegen den internationalen Copyrightbestimmungen.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Allgemein	1
2. Schnittstelle	2
2.1. Allgemein	2
2.1.1. Abstrakt	2
2.1.2. Standardparameter und -rückgaben	2
2.1.3. Fehlercodes	3
2.1.4. Serviceprotokolle	3
2.2. Funktionen	5
2.2.1. Verfügbare Länder ermitteln mit country	5
2.2.2. Rufnummer reservieren mit init	6
2.2.3. Anrufüberwachung mit status	8
2.2.4. Detaillierte Statusabfrage mit info	10
2.2.5. Anrufsimitation mit testcall	12
2.3. Beispiel	13
A. Kurzreferenz	18
B. Fehlercodes	21

1. Einleitung

1.1. Allgemein

Die Call2Pay API Event bietet dem Partner die Möglichkeit, die Micropayment Zahlart "Call2Pay" für seine Kunden anzubieten und gleichzeitig maximale Kontrolle über die Kundeninteraktionen zu erhalten. Hierfür ist der Partner selbst verantwortlich, dem Kunden die benötigten Eingabeformulare und Informationen anzuzeigen und ihn durch den Bezahlvorgang zu führen. Da seit dem 01.04.08 in Deutschland nur noch ein Maximalbetrag von 10 EUR per Dropcharge gebucht werden kann, gibt es den erweiterten Modus "Multicall" der es erleichtert größere Beträge abzurechnen.

Im wesentlichen erfolgt die Bezahlung in 3 Schritten:

1. Auswahl des Landes, aus dem der Kunde anrufen möchte und ggf. Wahl des gewünschten Tarifs
2. Anzeige der reservierten Call2Pay Rufnummer mit den gesetzlichen Preisinformationen, und Wartefunktion, bis erforderliche Anrufdauer vollständig ist.
3. Bestätigung über den Erfolg der Bezahlung an den Kunden.

Da seit dem 01.04.08 in Deutschland nur noch ein Maximalbetrag von 10 EUR per Dropcharge gebucht werden kann, gibt es den erweiterten Modus "Multicall" der es erleichtert, größere Beträge abzurechnen. Wird ein Betrag größer 10 EUR für Deutschland angefordert ergeben sich nun 2 Möglichkeiten.

1. Multicall nicht erwünscht:

Der Betrag wird für einen Call reserviert. Abrechnung erfolgt ausschließlich über Zeittakt. Dies ist auch

der Modus für die Call2Pay API Event 1.0

2. Multicall erwünscht:

Der Gesamtbetrag wird in kleinere Calls (ebend der Maxbetrag von derzeit 10 EUR) aufgesplittet, und wenn nötig einem Call mit dem Restbetrag < 10 EUR.

Bsp.: angefordert 29,99 EUR ergibt 3 Calls (2* 10 EUR + 1* 9,99 EUR)

Alle Calls des Multicalls haben den gleichen Wert für "handle".

Alle Calls haben den gleichen Wert für "number", d.h. der Endkunde kann/soll immer die Wahlwiederholung benutzen. In der Statistik erscheint dann auch nur immer ein Satz über den Gesamtbetrag.

Für einen typischen Ablauf findet sich am Ende des Dokuments ein Beispiel.

2. Schnittstelle

2.1. Allgemein

2.1.1. Abstrakt

Die Beschreibunsyntax der Parameter und Rückgaben in den folgenden Abschnitten definiert sich wie folgt:

name :type

Pflichtparameter mit Parametertyp

(name :type)

optionaler Parameter, wird er nicht angegeben wird abhängig vom Typ Leere Zeichenkette (""), numerische Null (0) oder logisches Falsch (0) angenommen

(name :type) =defaultwert

optionaler Parameter, wird dieser nicht angegeben wird für diesen *defaultwert* angenommen. Wichtig! Wird der Parameter mit leerer Zeichenkette angegeben, wird diese, und nicht der Defaultwert verwendet.

name :type wenn: bedingung

Pflichtparameter, wenn die bezeichnete Bedingung zutrifft.

(name :type) wenn: bedingung

Parameter ist auch dann optional, wenn die Bedingung zutrifft, hat aber anderenfalls keine Bedeutung

(name :type) =defaultwert wenn: abhängigkeit

wie voriger, mit definierter Voreinstellung

name[n] :type

indizierter Parameter name[0], name[1], ... name[n-1], koexistiert typischerweise mit einem parameter mit dem wert von *n*

2.1.2. Standardparameter und -rückgaben

Jeder Aufruf erfolgt mittels definierter URL der Serviceschnittstelle und den erforderlichen Parametern zur Identifizierung des Aufrufers. Die Standardparameter sind für alle Funktionen gültig und werden im folgenden Text nicht mehr beschrieben.

Parameter:

accesskey :string

enthält den persönlichen AccessKey des Partners. Dieser wird automatisch bei der Partnerregistrierung im Micropayment-System erzeugt, und kann im Controlcenter ermittelt werden.

(*testmode* :boolean) =0

schaltet optional auf eine interne Testumgebung um, unter der der Partner ohne die Kosten des Bezahlsystems seine Implementierung testen kann

Rückgabe:

error :integer

enthält 0 bei Erfolg bzw. liefert die Fehlernummer des gescheiterten Aufrufs

errormessage :string wenn: *error* <> 0

enthält die nähere Beschreibung des Fehlers im Klartext

2.1.3. Fehlercodes

Um Fehlerbehandlung zu vereinfachen, sind die Fehlercodes der Rückgabe *error* in vier verschiedene Klassen mit eigenem Nummernkreis unterteilt.

permanenter Serverfehler 1xxx

Bei diese Fehlerklasse liegt meist ein permanentes Problem beim Webdienst vor. Bei Fehlern dieser Klasse sollte der Support informiert werden.

temporärer Serverfehler 2xxx

Fehler dieses Typs sind auch vom Webdienst bedingt, haben aber nur eine vorübergehenden Ursache, wie Wartungsarbeiten oder erschöpfte Ressourcen. Der Kunde kann hierbei auf spätere Nutzung des Dienstes getröstet werden.

Clientfehler 3xxx

Die Ursache der Fehler aus dieser Klasse liegt typischerweise bei der aufrufende Applikation. Es ist sinnvoll diese Fehler mit dem Text aus *errormessage* mitzuloggen, und die Anwendung entsprechend zu modifizieren. Zur Behebung kann die Dokumentation oder der Support zu Rate gezogen werden.

Userfehler 4xxx

Zu dieser Klasse gehören alle Fehler, die typischerweise aus falschen Eingaben des Kunden resultieren, z.B. falsches Passwort etc. Dem User ist hier eine aussagekräftige Fehlerbeschreibung anzuzeigen, damit er die Falscheingabe korregieren kann. Die Ursache kann aber auch hier bei der Applikation liegen, die z.B. eingegebene Werte fehlerhaft formatiert.

In Anhang sind alle Fehlercodes, Ursachen und Tipps zur Behebung zusammengefasst.

2.1.4. Serviceprotokolle

Die Schnittstelle ist unter der Adresse <http://webservices.micropayment.de/public/c2p/v2/> erreichbar und wird im weiteren als {service-url} angesprochen. Momentan ist sie mit Hilfe zweier alternativen Technologien implementiert.

2.1.4.1. SOAP Webservice

An die Service-URL werden Funktion und Parameter als Soap-Envelop gesendet (HTTP Methode POST). Alle Parameter sind hierbei in einem strukturierten Typ enthalten:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:c2p="http://webservices.micropayment.de/public/call2pay/version2.0"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <c2p:{function}>
      <param xsi:type="c2p:C2P{function}RequestTyp">
        <{name} xsi:type="{type}">{value}</{name}>
        <{name} xsi:type="{type}">{value}</{name}>
        ...
      </param>
    </c2p:{function}>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Das Ergebnisdokument enthält im Erfolgsfall die Rückgabestruktur, ...

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:c2p="http://webservices.micropayment.de/public/call2pay/version2.0"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <c2p:{function}Response>
      <return xsi:type="c2p:C2P{function}ResponseType">
        <{name} xsi:type="{type}">{value}</{name}>
        <{name} xsi:type="{type}">{value}</{name}>
        ...
      </return>
    </c2p:{function}Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

... oder einen Soap-Fault mit den Standardrückgaben error und errormessage:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>{error value}</faultcode>
      <faultstring>{errormessage value}</faultstring>
      <faultactor></faultactor>
      <detail></detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.1.4.2. Simple HTTP

Die Webservice-URL wird um die Parameterliste als HTTP Querystring (GET-Methode) erweitert. Für den Funktionsnamen ist der Parameter *action* vorgesehen:

```
http://{service-url}/?action={function}&{name}={value}&{name}={value}&...
```

Die Rückgabewerte werden zeilenweise als Name-Wert-Paare im Ergebnisdokument geliefert:

```
error=0
{name}={value}
{name}={value}
...
```

Im Fehlerfall werden nur die beiden Standardrückgaben `error` und `errormessage` geliefert:

```
error={error value}
errormessage={errormessage value}
```

Die jeweiligen Parameter- und Rückgabewerte sind URL-codiert. Sonderzeichen sind nach ISO-8859-1 Standard zu codieren.

2.2. Funktionen

2.2.1. Verfügbare Länder ermitteln mit `country`

Die Funktion ermittelt für einen bestimmten Zahlbetrag die Liste der verfügbaren Länder. Einerseits kann mit Hilfe dieser Funktion dem Kunden eine Liste der verfügbaren Länder angeboten, andererseits sein Herkunftsland vorausgewählt werden.

Parameter:

project :string

Bezeichnung des bezogenen Projekts, für den die Bezahlung initiiert werden soll

(*amount* :integer)

Gibt den Betrag in 100stel der angeforderten Währung (z.B in Cent) an, den der Kunde bezahlen soll. Ist der Parameter leer, wird auf die Vorgabe aus der Projekteinstellung zurückgegriffen. Der Betrag wird für jedes Land in dessen Währung umgerechnet und anschließend geprüft, ob für diesen eine Rufnummer verfügbar wäre.

(*currency* :string(3)) ="EUR" wenn: *amount* angegeben

Gibt an, in welcher Währung der Parameter *amount* angegeben ist.

(*ip* :string)

Anhand der IP des Kunden kann mit dieser Funktion gleichzeitig das bevorzugte Land ermittelt werden (Rückgabe `ipcountry` und `ipprovider`).

Rückgabe:

`countrycount` :integer

Enthält die Anzahl der gefundenen Länder und zurückgelieferter `country[n]` Werte.

`country[n]` :string(2) wenn: `countrycount` > 0

Liefert jeweils einen Ländercode (z.B: "DE", "AT") in denen der umgerechnete Preis bezahlt werden kann.

`ipcountry` :string(2) wenn: `ip` angegeben

Enthält das Land, in dem die übergebene IP gehostet ist. Dies muss zwar nicht zwangsläufig die Herkunft des Kunden sein, ist es aber dennoch mit hoher Wahrscheinlichkeit. Der Wert muss nicht zwangsläufig in der Länderliste enthalten sein.

`ipprovider` :string wenn: `ip` angegeben

Enthält, sofern ermittelbar, die Bezeichnung größerer ISP. Derzeit mögliche Werte:

"UNKNOWN" = ISP nicht bekannt, hat aber seinen Sitz in `ipcountry`

"AOL" = IP ist im AOL-Netz, der Kunde kann aber aus jedem anderen Land stammen. `ipcountry` enthält

meistens "US"

Alle Parameter und Rückgaben sind in der Kurzreferenz zusammengefasst.

2.2.2. Rufnummer reservieren mit `init`

Mit dieser Funktion wird ein neuer Bezahlvorgang gestartet. Als Ergebnis wird eine Rufnummer für den Kunden reserviert und zurückgeliefert. Der Kunde sollte anschließend aufgefordert werden diese Nummer anzurufen.

Parameter:

`project` :string

Bezeichnung des bezogenen Projekts, für den die Bezahlung initiiert werden soll

(`projectcampaign` :string)

legt fest welcher Projekt-Kampagne dieser Vorgang zugeordnet werden soll

(`account` :string)

Accountnummer bzw. -bezeichnung des partizipierenden Webmasters. Standardmäßig wird der Account des Projektinhabers angenommen.

(`webmastercampaign` :string)

legt fest welcher Webmaster-Kampagne dieser Vorgang zugeordnet werden soll

`sessionid` :string

Ein eindeutiges Handle, das die Kundensitzung identifiziert. Diese wird für den offenen Vorgang gespeichert. Sollte der Kunde den Bezahlvorgang ein zweites mal starten, erhält er die selbe Rufnummer.

`ip` :string

Die IP des Kunden, der den Vorgang startet. Sie wird einerseits für die Fehlersuche gespeichert und andererseits verwendet, um ein Flooding des Dienstes zu vermeiden.

`country` :string(2)

Das Land des Anrufers um dem Kunden eine nationale Service-Rufnummer anzubieten.

(`language` : string(2))

Legt die Sprache der telefonischen Ansagen fest. Standardmäßig wird die landestypische Sprache für `country` herangezogen.

(`amount` :integer)

Gibt den Betrag in 100stel der angeforderten Währung (z.B in Cent) an, den der Kunde bezahlen soll. Ist der Parameter leer, wird auf die Vorgabe aus der Projekteinstellung zurückgegriffen.

Achtung! Der dem Kunden anzuzeigende Betrag und dessen Währung hängt von der gewählten Landeseinstellung in `country` ab. Er wird bei Bedarf aus den übergebenen Werten errechnet und von der Funktion zurückgeliefert.

(`currency` :string(3)) ="EUR" wenn: `amount` angegeben

Gibt an, in welcher Währung der Parameter `amount` angegeben ist.

(`title` :string)

Benennt die zu kaufende Sache (z.B: Kundennummer, Username, Artikelbezeichnung, o.ä.). Diese wird in den Statistikdetails des Partners mit angezeigt. Wird der Parameter leer gelassen, wird der Standard-Artikel in der Projekteinstellung verwendet.

(`freeparam` :string)

Bindet einen beliebigen Text an den Vorgang und kann mit anderen Funktionen wieder abgefragt werden.

Das kann ein externes Handle des Partners sein oder aber auch mehrere kommaseparierte Werte.

multicall :integer

0 = Multicall nicht erwünscht (default)

1 = Multicall erwünscht

Rückgabe:

status :string(20)

Der Reservierungsstatus des Vorgangs. Im Fehlerfall werden die Standardrückgaben error und errormessage gesetzt. Mögliche Werte sind:

"INIT" = Ein neuer Vorgang wurde gestartet.

"REINIT" = Ein unterbrochener Vorgang wurde reinitialisiert oder ein Call eines Multicalls wurde abgeschlossen.

"CALL" = Der Kunde ruft grade an

handle :string(50)

Die eindeutige ID des Vorgangs. Diese wird für andere Funktionen benötigt, um den aktuellen Status abzufragen.

expire :datetimestring

Der Zeitpunkt, wann die Reservierung abläuft und die Rufnummer verfällt. Die Zeitspanne ist sehr kurz gehalten (30 sek.), um den Rufnummernvorrat zu schonen. Durch zyklisches Aufrufen der Funktion status wird die Lebensdauer der Reservierung aufgefrischt.

number :string

Liefert die vom Kunden anzurufende Servicenummer. Sie ist bereits optisch formatiert.

(numberinfo :string)

Enthält, falls gesetzlich erforderlich, zusätzliche Tarifangaben und Hinweise zur angezeigten Rufnummer. Der Text ist dem Kunden in unmittelbarem Zusammenhang mit der Rufnummer anzuzeigen und mit dieser z.B: mit Hilfe einer Fußnote (Sternchen *) visuell zu verknüpfen.

origin :string

Gibt an, ob die zurückgegebene Rufnummer nur aus einem bestimmten Netz (Mobilfunk- oder Festnetz) erreichbar ist. Dies kann gelegentlich vorkommen, typischerweise enthält diese Rückgabe den Wert "BOTH". Bei einer Einschränkung sollte der Kunde aber entsprechend darauf hingewiesen werden (z.B: "Rufnummer ist nur aus dem Festnetz erreichbar").

"BOTH"

"LANDLINE"

"MOBILE"

amount :integer

Betrag, den der Kunde für den gesamten Anruf bezahlt. Er ist, unabhängig von numberinfo, zusammen mit currency dem Kunden anzuzeigen.

currency :string

Währung des Betrags in Rückgabe amount.

mode: string

Diese Rückgabe legt den Modus für die Rufnummer fest.

"DIRECT" = Die Rufnummer ist eindeutig für den Kunden reserviert, alleine durch den Anruf kann er zugeordnet werden.

"DTMF" = Gelegentlich, meist bei ausländischen Rufnummern, kann keine durchwahlfähige Rufnummer reserviert werden. In diesem Fall identifiziert sich der Kunde mit einer TAN durch Eingabe auf der Telefon-tastatur während des Anrufs.

`tan :string wenn: mode = "DTMF"`

Im Modus "DTMF" die vom Kunden via Telefon eingegebene TAN.

`duration : integer`

Gesamtdauer des Anrufs in Sekunden.

`durationpart :integer`

Bisherige Anrufdauer in Sekunden. Diese ist beim Status "INIT" Null, bei "REINIT" oder "CALL" größer Null und kleiner als "duration" (Siehe `durationpart` in Funktion `status`)

`split :integer`

Gleiches Format wie Parameter "`amount`".

Wenn "`split`" > 0 handelt es sich um einen Multicall und gibt den Betrag des aktuellen Calls an.

Abhängig von "`split`" wird dann auch "`numberinfo`" entsprechend formatiert. Für obiges Beispiel würde beim ersten Aufruf von `init` folgendes zurückgegeben werden:

`"split" => 1000`

`"amount" => 2999`

`"numberinfo" => 10,00 EUR/Anruf aus dt. Festnetz, ggf. abweichend aus Mobilnetz.`

beim letzten Aufruf entsprechend:

`"split" => 999`

`"amount" => 2999`

`"numberinfo" => 9,99 EUR/Anruf aus dt. Festnetz, ggf. abweichend aus Mobilnetz.`

`paid :integer`

Gleiches Format wie Parameter "`amount`".

Enthält den aufsummierten Betrag aller bisherigen (abgeschlossenen) Calls eines Multicalls.

`callcnt :integer`

Anzahl der erfolgreich abgeschlossenen Calls eines Multicalls

Alle Parameter und Rückgaben sind in der Kurzreferenz zusammengefasst.

2.2.3. Anrufüberwachung mit `status`

Die Funktion ermittelt den aktuellen Status einer Reservierung. Sie sollte zyklisch aufgerufen werden, einerseits um die Reservierung zu erhalten, andererseits um festzustellen, ob der Kunde gerade anruft oder bereits angerufen hat. Zusätzlich kann dem Kunden der Fortschritt bei zeitabhängigen Anrufen angezeigt werden, z.B. durch eine AJAX-Lösung.

Parameter:

`handle :string(40)`

Die ID des Vorgangs aus der Funktion `init`.

Rückgabe:

`status` :string(20)

Der Reservierungsstatus des Vorgangs. Im Fehlerfall werden die Standardrückgaben `error` und `errormessage` gesetzt. Mögliche Werte sind:

"INIT" = Reservierung wartet auf Anruf.

"REINIT" = Reservierung wartet auf Anruf nach unterbrochener Verbindung.

"CALL" = Der Kunde ruft grade an

"RECALL" = Der Anruf wurde vorzeitig abgebrochen, der Kunde muss erneut anrufen um die Bezahlung abzuschließen. Bei diesem Wert muss die Funktion `init` mit den gleichen Werten aufgerufen werden, da u.a. eine neue Servicrufnummer reserviert werden kann.

"COMPLETE" = Anruf ist vollständig und erfolgreich. Dieser Status wird nur unmittelbar nach Abschluss geliefert (ca. 10 min. lang). Danach liefert die Funktion einen Fehler für das übergebene `handle`. Die Eigenschaften des Vorgangs können aber jederzeit mit `info` ermittelt werden.

`expire` :datetimestring

Der Zeitpunkt, wann die Reservierung abläuft und die Rufnummer verfällt. Durch wiederholtes Aufrufen der Funktion wird dieser Wert immer aktualisiert. (siehe Erklärung `expire` in Funktion `init`)

(`caller` :string) wenn: `status` <> "INIT"

Liefert die Rufnummer des Anrufers während und nach einem Anruf, falls sie ermittelbar ist. In jedem Fall ist sie aber verkürzt angegeben (z.B. "01719988XXX")

(`origin` :string)

Enthält unabhängig von `caller` das Herkunftnetz, während oder nach dem Anruf. Bei Auslandsnummern ist dies meist nicht ermittelbar, und die Rückgabe leer.

"LANDLINE"

"MOBILE"

`duration` :integer

Gesamtdauer des Anrufs in Sekunden. Dieser Wert kann während eines Anrufs von dem aus der Funktion `init` abweichen, wenn der Kunde z.B. aus dem Mobilnetz anruft.

`durationpart` :integer

Noch benötigte Anrufdauer in Sekunden. Dieser ist beim Status "CALL" oder "RECALL" typischerweise kleiner als `duration`. Somit kann aus `durationpart` und `duration` ein prozentualer Wert ermittelt werden, um dem Kunden z.B. einen Fortschrittsbalken anzuzeigen. Beim Status "COMPLETE" ist er 0.

(`freeparam` :string)

enthält den Wert, der bei der Funktion `init` übergeben wurde

`split` :integer

Gleiches Format wie Parameter "`amount`".

Wenn "`split`" > 0 handelt es sich um einen Multicall und gibt den Betrag des aktuellen Calls an.

Abhängig von "`split`" wird dann auch "`numberinfo`" entsprechend formatiert. Für obiges Beispiel würde beim ersten Aufruf von `init` folgendes zurückgegeben werden:

"`split`" => 1000

"`amount`" => 2999

"*numberinfo*" => 10,00 EUR/Anruf aus dt. Festnetz, ggf. abweichend aus Mobilnetz.

beim letzten Aufruf entsprechend:

"*split*" => 999

"*amount*" => 2999

"*numberinfo*" => 9,99 EUR/Anruf aus dt. Festnetz, ggf. abweichend aus Mobilnetz.

paid :integer

Gleiches Format wie Parameter "*amount*".

Enthält den aufsummierten Betrag aller bisherigen (abgeschlossenen) Calls eines Multicalls.

callcnt :integer

Anzahl der erfolgreich abgeschlossenen Calls eines Multicalls

Alle Parameter und Rückgaben sind in der Kurzreferenz zusammengefasst.

2.2.4. Detaillierte Statusabfrage mit *info*

Im Unterschied zur Funktion *status* liefert die Funktion alle Eigenschaften für einen Zahlungsvorgang. Außerdem lassen sich mit ihr auch abgelaufene Reservierungen, erfolgreiche und gescheiterte Zahlungsvorgänge abfragen. Die Funktion sollte nicht zyklisch aufgerufen werden (z.B. beim Warten auf den Anruf), da sie deutlich höhere Antwortzeiten benötigt und mehr Last erzeugt. Mit ihr kann beispielsweise ein Zahlungsvorgang überprüft werden, bevor der Kunde den bezahlten Dienst in Anspruch nimmt..

Parameter:

handle :string(40)

Die ID des Vorgangs aus der Funktion *init*.

Rückgabe:

status :string(20)

Der Status des Vorgangs. Mögliche Werte sind:

"INIT" = Reservierung wartet auf Anruf.

"REINIT" = Reservierung wartet auf Anruf nach unterbrochener Verbindung.

"EXPIRED" = Reservierung ist ohne Anruf abgelaufen

"CALL" = Der Kunde ruft grade an

"RECALL" = Der Anruf wurde vorzeitig abgebrochen, der Kunde muss erneut anrufen um die Bezahlung abzuschließen.

"FAILED" = Anruf wurde vorzeitig abgebrochen, Reservierung ist aber bereits abgelaufen

"COMPLETE" = Anruf ist vollständig und erfolgreich

Wurde kein Vorgang unter *handle* gefunden, sind *error* und *errormessage* gesetzt.

expire :datetimestring

Der Zeitpunkt, wann die Reservierung abläuft oder abgelaufen ist. Im Gegensatz zur Funktion *status* wird eine bestehende Reservierung durch diese Funktion nicht aufgefrischt.

`project` :string
die bei `init` übergebene Bezeichnung des bezogenen Projekts

`projectcampaign` :string
liefert die Kampagne der die Transaktion zugeordnet wurde

`account` :string
die übergebene Accountnummer bzw. -bezeichnung des partizipierenden Webmasters bzw. die des Projektbetreibers

`webmastercampaign` :string
Kampagne des Webmasters

`country` :string(2)
das initialisierte Land des Anrufers

`number` :string
Liefert die für den Vorgang reservierte Servicenummer. Ist dieser abgeschlossen, darf die Nummer nicht mehr angerufen werden, da sie ggf. bereits für einen anderen Kunden reserviert sein könnte.

`amount` :integer
der Gesamtbetrag für den Anruf

`currency` :string
die Währung des Wertes in `amount`.

`mode` :string
Der Modus des Bezahlvorgangs "DIRECT" oder "DTMF" aus Funktion `init`

`tan` :string wenn: `mode = "DTMF"`
Für Modus "DTMF" die TAN.

(`caller` :string) wenn: `status >= "CALL"`
Liefert die Rufnummer des Anrufers während und nach einem Anruf, falls sie ermittelbar ist. In jedem Fall ist sie aber verkürzt angegeben (z.B "01719988XXX")

(`origin` :string)
Enthält unabhängig von `caller` das Herkunftnetz, während oder nach dem Anruf. Bei Auslandsnummern ist dies meist nicht ermittelbar, und die Rückgabe leer.

"LANDLINE"

"MOBILE"

`duration` : integer
Gesamtdauer des Anrufs in Sekunden.

`durationpart` :integer
die bereits verstrichene Anrufdauer

`title` :string
die Artikelbezeichnung für diesen Vorgang

(`freeparam` :string)
enthält den Wert, der bei der Funktion `init` übergeben wurde

`split` :integer
Gleiches Format wie Parameter "`amount`".

Wenn "*split*" > 0 handelt es sich um einen Multicall und gibt den Betrag des aktuellen Calls an.

Abhängig von "*split*" wird dann auch "*numberinfo*" entsprechend formatiert. Für obiges Beispiel würde beim ersten Aufruf von *init* folgendes zurückgegeben werden:

"*split*" => 1000

"*amount*" => 2999

"*numberinfo*" => 10,00 EUR/Anruf aus dt. Festnetz, ggf. abweichend aus Mobilnetz.

beim letzten Aufruf entsprechend:

"*split*" => 999

"*amount*" => 2999

"*numberinfo*" => 9,99 EUR/Anruf aus dt. Festnetz, ggf. abweichend aus Mobilnetz.

paid :integer

Gleiches Format wie Parameter "*amount*".

Enthält den aufsummierten Betrag aller bisherigen (abgeschlossenen) Calls eines Multicalls.

callcnt :integer

Anzahl der erfolgreich abgeschlossenen Calls eines Multicalls

Alle Parameter und Rückgaben sind in der Kurzreferenz zusammengefasst.

2.2.5. Anrufsimulation mit *testcall*

Die Funktion *testcall* kann bei der Anbindung des Webservices und für Testzwecke hilfreich sein. Sie ermöglicht es einen Kundenanruf zu simulieren, und ist deshalb auch nur im Testmodus verfügbar. Wurde ein Bezahlvorgang mit *init* im Testmodus erzeugt, kann durch separaten Aufruf dieser Funktion, der Status auf "CALL", "RECALL" oder "COMPLETE" gesetzt werden, um das Verhalten der eigenen Applikation zu testen..

testmode :boolean

Dieser allgemeine Parameter muss für die Funktion auf "1" gesetzt sein.

number: string

Die Servicrufnummer aus der Funktion *init*, die der Kunde anrufen würde.

(*origin*: string) ="LANDLINE"

Legt fest, aus welchem Netz der Anruf stammen soll. Möglich sind "LANDLINE" oder "MOBILE"

(*caller* :string)

Kann die Anrufernummer enthalten, sie wird dann von der Funktion *status* direkt zurückgegeben.

tan :string wenn: *mode* = "DTMF"

Im Modus "DTMF" die TAN.

durationpart: integer

Gibt an, wie lange der simulierte Anruf dauern soll. Obwohl der Aufruf sofort zurückkehrt, wird die angegebene Dauer gewartet, bis der Anruf beendet wird. Wiederholte Aufrufe der Funktion *status* liefert also den Status "CALL" und aufsteigende Werte für *durationpart*. Wurde dieser Parameter kleiner als *duration* gewählt taucht er am Ende zusammen mit Status "RECALL" auf, ansonsten sind *duration* und *durationpart* gleich und der Status auf "COMPLETE".

Rückgabe:

handle :string(50)

Die ID des Vorgangs, wie aus der Funktion `init`.

Alle Parameter und Rückgaben sind in der Kurzreferenz zusammengefasst.

2.3. Beispiel

Hier wird ein typischer Ablauf mit den erforderlichen Webserviceinteraktionen skizziert. Das Beispiel demonstriert ausserdem den Fall, in dem der Kunde zu früh die Verbindung unterbricht und erneut zum Anruf aufgefordert wird.

1. Der Kunde entscheidet sich für einen Kauf und startet die Call2Pay Applikation des Partners. Diese ermittelt die verfügbaren Länder und bietet sie dem Kunden zur Auswahl an. Für das wahrscheinlichste Land initiiert sie einen neuen Bezahlvorgang.

```
http://{service-url}/?action=country&accesskey=0123abc
&project=demo&amount=100&currency=EUR&ip=127.0.0.1
```

```
error=0
countrycount=3
country[0]=DE
country[1]=CH
country[2]=AT
ipcountry=DE
ipprovider=UNKNOWN
```

```
http://{service-url}/?action=init&accesskey=0123abc
&project=demo&sessionid=aabbccddeeff&ip=127.0.0.1
&country=DE&amount=100&currency=EUR&title=10 Coins&multicall=1
```

```
error=0
status=INIT
handle=1000abcdef
expire=2007-01-15 12:00:00
number=09005 000 111 22
numberinfo=2,00 EUR/min aus dt. Festnetz, ggf. abweichend aus Mobilnetz.
origin=BOTH
amount=100
currency=EUR
mode=DIRECT
tan=
duration=30
durationpart=0
split=0
paid=0
callcnt=0
```

Diese Informationen werden dem Kunden angezeigt.

Rufen Sie die folgende Nummer an und halten Sie sie 30 Sekunden, bis automatisch aufgelegt wird!
Anschließend werden Ihrem Konto 10 Coins gutgeschrieben.

09005 000 111 22 *

* 2,00 EUR/min aus dt. Festnetz, ggf. abweichend aus Mobilnetz.
Die Gesamtkosten des Anrufs betragen 1,00 EUR.

Wählen Sie Ihr Land: Deutschland | Schweiz | Österreich

2. Auf der Webseite läuft ein zyklischer Aufruf (z.B. alle 5 sek. per Javascript) an den Applikationserver. Dieser fragt den Status des Vorgangs ab und erhält somit die Reservierung.

```
http://{service-url}/?action=status&accesskey=0123abc&handle=1000abcdef  
error=0  
status=INIT  
expire=2007-01-15 12:00:05  
caller=  
duration=30  
durationpart=0  
split=0  
paid=0  
callcnt=0
```

Sobald der Kunde den Anruf tätigt, liefert die Funktion z.B: folgendes

```
http://{service-url}/?action=status&accesskey=0123abc&handle=1000abcdef  
error=0  
status=CALL  
expire=2007-01-15 12:00:10  
caller=03012345xxx  
duration=30  
durationpart=5  
split=0  
paid=0  
callcnt=0
```

Der zyklische Aufruf aktualisiert die Webseite und zeigt den Fortschritt dem Kunden an:

```
Der Bezahlvorgang läuft ... halten Sie die Verbindung  
noch 25 Sekunden  
bis automatisch getrennt wird!
```

Jetzt legt der Kunde 10 sek. zu früh auf. status liefert:

```
http://{service-url}/?action=status&accesskey=0123abc&handle=1000abcdef  
error=0  
status=RECALL  
expire=2007-01-15 12:00:25  
caller=03012345xxx  
duration=30  
durationpart=20  
split=0  
paid=0  
callcnt=0
```

3. Die Applikation fordert nun neue Daten vom Webservice an, ...

```
http://{service-url}/?action=init&accesskey=0123abc  
&project=demo&sessionid=aabbccddeeff&ip=127.0.0.1  
&country=DE&amount=100&currency=EUR&title=10 Coins&multicall=1  
error=0  
status=REINIT  
handle=1000abcdef  
expire=2007-01-15 12:00:25
```

```
number=09005 000 111 88
numberinfo=2,00 EUR/min aus dt. Festnetz, ggf. abweichend aus Mobilnetz.
origin=BOTH
amount=100
currency=EUR
mode=DIRECT
tan=
duration=30
durationpart=20
split=0
paid=0
callcnt=0
```

... und fordert den Kunden auf, erneut anzurufen:

Sie haben leider zu früh aufgelegt!

Rufen Sie erneut an und halten Sie die Verbindung 10 Sekunden,
bis automatisch aufgelegt wird!

```
09005 000 111 88 *
```

* 2,00 EUR/min aus dt. Festnetz, ggf. abweichend aus Mobilnetz.
Die Gesamtkosten beider Anrufe betragen insgesamt 1,00 EUR.

Die Status-Funktion liefert jetzt:

```
http://{service-url}/?action=status&accesskey=0123abc&handle=1000abcdef
```

```
error=0
status=REINIT
expire=2007-01-15 12:00:25
caller=03012345xxx
duration=30
durationpart=20
split=0
paid=0
callcnt=0
```

... und nach jeweils 5 sek seit Kundenanruf:

```
http://{service-url}/?action=status&accesskey=0123abc&handle=1000abcdef
```

```
error=0
status=CALL
expire=2007-01-15 12:00:30
caller=03012345xxx
duration=30
durationpart=25
split=0
paid=0
callcnt=0
```

```
http://{service-url}/?action=status&accesskey=0123abc&handle=1000abcdef
```

```
error=0
status=COMPLETE
expire=2007-01-15 12:00:35
caller=03012345xxx
duration=30
durationpart=30
split=0
```

```
paid=0
callcnt=0
```

4. Die Bezahlvorgang ist nun abgeschlossen, und der Kunde kann den bezahlten Dienst in Anspruch nehmen. Für Wartung und Support kann ab hier jederzeit die Funktion `info` benutzt werden, um die Transaktion nachzuvollziehen:

```
http://{service-url}/?action=info&accesskey=0123abc&handle=1000abcdef

error=0
expire=2007-01-15 12:00:35
project=demo
projectcampaign=
account=10010
webmastercampaign=
country=DE
number=09005 000 111 88
amount=100
currency=EUR
mode=DIRECT
tan=
caller=03012345xxx
duration=30
durationpart=20
title=10 Coins
freeparam=
split=0
paid=0
callcnt=0
```

5. Das Beispiel aus Punkt 1 wird alternativ mit 13, 50 EUR abgerechnet und es soll wenn nötig von der `Multicall`-Methode gebrauch gemacht werden. Das sieht dann wie folgt aus.

```
http://{service-url}/?action=init&accesskey=0123abc
&project=demo&sessionid=aabbccddeeff&ip=127.0.0.1
&country=DE&amount=1350&currency=EUR&title=10 Coins&multicall=1

error=0
status=INIT
handle=1000abcdef
expire=2007-01-15 12:00:00
number=09005 000 111 22
numberinfo=10,00 EUR/Anruf aus dt. Festnetz, ggf. abweichend aus Mobilnetz.
origin=BOTH
amount=1350
currency=EUR
mode=DIRECT
tan=
duration=45
durationpart=0
split=1000
paid=0
callcnt=0
```

Diese Informationen werden dem Kunden angezeigt.

Rufen Sie die folgende Nummer an und halten Sie die Verbindung, bis automatisch getrennt wird!
Anschließend werden Ihrem Konto 10 Coins gutgeschrieben.

Bei Anrufen aus dem deutschen Festnetz werden Sie gebeten, mit 2 Anrufen zu bezahlen. Erst wenn Sie den Gesamtbetrag bezahlt haben,

wird dieser auf Ihrer Telefonrechnung belastet.

```
09005 000 111 22 *
```

* 10,00 EUR aus dem deutschen Festnetz, ggf. abweichende Preise im Mobilnetz.
Die Gesamtkosten betragen 13,50 EUR.

nach erfolgtem Anruf per Festnetz ergibt die Abfrage mit status folgendes:

```
http://{service-url}/?action=status&accesskey=0123abc&handle=1000abcdef
```

```
error=0
status=REINIT
expire=2007-01-15 12:00:35
caller=03012345xxx
duration=45
durationpart=0
origin=LANDLINE
freeparam=
split=350
paid=1000
callcnt=1
```

daraufhin muß erneut init aufgerufen werden. Alles so lange bis alle Multicalls erfolgreich abgeschlossen wurden, siehe status.

```
http://{service-url}/?action=init&accesskey=0123abc
&project=demo&sessionid=aabbccddeeff&ip=127.0.0.1
&country=DE&amount=1350&currency=EUR&title=10 Coins&multicall=1
```

```
error=0
status=REINIT
handle=1000abcdef
expire=2007-01-15 12:10:00
number=09005 000 111 22
numberinfo=10,00 EUR/Anruf aus dt. Festnetz, ggf. abweichend aus Mobilnetz.
origin=BOTH
amount=1350
currency=EUR
mode=DIRECT
tan=
duration=45
durationpart=0
split=350
paid=1000
callcnt=1
```

```
http://{service-url}/?action=status&accesskey=0123abc&handle=1000abcdef
```

```
error=0
status=COMPLETE
expire=2007-01-15 12:00:35
caller=03012345xxx
duration=45
durationpart=45
origin=LANDLINE
freeparam=
split=0
paid=1350
callcnt=2
```

Kurzreferenz

Funktion `country`

Tabelle A.1. Funktion `country` Parameter

Parameter	Typ	Standard	Beschreibung
<code>accesskey</code>	string		Zugangsschlüssel
<code>testmode</code>	boolean	0	Testmodus aktivieren
<code>project</code>	string		Projektname
<code>amount</code>	integer	Betrag aus Projekteinstellung	Zahlbetrag
<code>currency</code>	string(3)	"EUR", wenn <code>amount</code>	Währung
<code>ip</code>	string	leer	Land für IP ermitteln

Tabelle A.2. Funktion `country` Rückgabe

Parameter	Typ	Standard	Beschreibung
<code>error</code>	integer	0	Fehlercode
<code>errormessage</code>	string	bei <code>error</code> <> 0	Fehlermeldung
<code>countrycount</code>	integer		Anzahl <code>country[n]</code>
<code>country[n]</code>	string(2)	bei <code>countrycount</code> > 0	Unterstützte Länder
<code>ipcountry</code>	string(2)	bei <code>ip</code> <> ""	Sitz des ISP mit IP
<code>ipprovider</code>	string	bei <code>ip</code> <> ""	ISP "UNKNOWN", "AOL"

Funktion `init`

Tabelle A.3. Funktion `init` Parameter

Parameter	Typ	Standard	Beschreibung
<code>accesskey</code>	string		Zugangsschlüssel
<code>testmode</code>	boolean	0	Testmodus aktivieren
<code>project</code>	string		Projektname
<code>projectcampaign</code>	string	keine	Kampagne des Projektbetreibers
<code>account</code>	string	Account des Projektbetr.	Webmasteraccount
<code>webmastercampaign</code>	string	keine	Kampagne des Webmasters
<code>sessionid</code>	string		Usersession
<code>ip</code>	string		User IP
<code>country</code>	string(2)		Land des Users
<code>language</code>	string(2)	Landessprache	Ansagesprache bei Anruf
<code>amount</code>	integer	Betrag aus Projekteinstellung	Zahlbetrag
<code>currency</code>	string(3)	"EUR", wenn <code>amount</code>	Währung
<code>title</code>	string	Artikel aus Projekteinstellung	Kaufgegenstand

Parameter	Typ	Standard	Beschreibung
<i>freeparam</i>	string	leer	freier Parameter
<i>multicall</i>	integer	0	Multicall erlauben

Tabelle A.4. Funktion *init* Rückgabe

Parameter	Typ	Standard	Beschreibung
error	integer	0	Fehlercode
errormessage	string	bei error <> 0	Fehlermeldung
status	string		Reservierungsstatus "INIT", "REINIT", "CALL"
handle	string(50)		Reservierungs ID
expire	datetimestring		Ablauf der Reservierung
number	string		Rufnummer
numberinfo	string		gesetzl. Tarifangaben
origin	string		Erreichbarkeit d. Nummer "MOBILE", "LANDLINE", "BOTH"
amount	integer		Zahlbetrag
currency	string(3)		Währung
mode	string		Modus d. Useridentifikation "DIRECT", "DTMF"
tan	string	bei mode = "DTMF"	TAN-Eingabe beim Anruf
duration	integer		gesamte Anrufdauer in Sek.
durationpart	integer		verstrichene Anrufsekunden
split	integer		Betrag des aktuellen (Multi)Calls
paid	integer		Betrag abgeschlossene (Multi)Calls
callcnt	integer		Anzahl abgeschlossene (Multi)Calls

Funktion *status*

Tabelle A.5. Funktion *status* Parameter

Parameter	Typ	Standard	Beschreibung
<i>accesskey</i>	string		Zugangsschlüssel
<i>testmode</i>	boolean	0	Testmodus aktivieren
<i>handle</i>	string(50)		Reservierungs ID

Tabelle A.6. Funktion *status* Rückgabe

Parameter	Typ	Standard	Beschreibung
error	integer	0	Fehlercode
errormessage	string	bei error <> 0	Fehlermeldung

Parameter	Typ	Standard	Beschreibung
status	string		Reservierungsstatus "INIT", "REINIT", "CALL", "RECALL", "COMPLETE"
expire	datetimestring		Ablauf der Reservierung
caller	string	wenn status >= "REINIT" und verfügbar	Telefonnummer d. Users
origin	string	leer	Netz des Anrufers "MOBI- LE", "LANDLINE"
duration	integer		gesamte Anrufdauer in Sek.
durationpart	integer		verstrichene Anrufsekun- den
freeparam	string	leer	freier Parameter
split	integer		Betrag des aktuellen (Multi)Calls
paid	integer		Betrag abgeschlossene (Multi)Calls
callcnt	integer		Anzahl abgeschlossene (Multi)Calls

Funktion info

Tabelle A.7. Funktion info Parameter

Parameter	Typ	Standard	Beschreibung
<i>accesskey</i>	string		Zugangsschlüssel
<i>testmode</i>	boolean	0	Testmodus aktivieren
<i>handle</i>	string(50)		Reservierungs ID

Tabelle A.8. Funktion info Rückgabe

Parameter	Typ	Standard	Beschreibung
error	integer	0	Fehlercode
errormessage	string	bei error <> 0	Fehlermeldung
status	string		Reservierungsstatus "INIT", "REINIT", "EXPI- RED", "CALL", "RECALL", "FAILED", "COMPLETE"
expire	datetimestring		Ablauf der Reservierung
project	string		Projektname
projectcampaign	string	leer	Kampagne des Projektbe- treibers
account	string		Webmasteraccount
webmastercampaign	string	leer	Kampagne des Webma- sters
country	string(2)		Land der Rufnummer
number	string		Rufnummer
amount	integer		Zahlbetrag

Parameter	Typ	Standard	Beschreibung
currency	string(3)		Währung
mode	string		Modus d. Useridentifikation "DIRECT", "DTMF"
tan	string	bei mode = "DTMF"	TAN-Eingabe beim Anruf
caller	string	wenn status >= "REINIT" und verfügbar	Telefonnummer d. Users
origin	string	leer	Netz des Anrufers "MOBILE", "LANDLINE"
duration	integer		gesamte Anrufdauer in Sek.
durationpart	integer		verstrichene Anrufsekunden
title	string		Kaufgegenstand
freeparam	string	leer	freier Parameter
split	integer		Betrag des aktuellen (Multi)Calls
paid	integer		Betrag abgeschlossene (Multi)Calls
callcnt	integer		Anzahl abgeschlossene (Multi)Calls

Funktion `testcall`

Tabelle A.9. Funktion `testcall` Parameter

Parameter	Typ	Standard	Beschreibung
<i>accesskey</i>	string		Zugangsschlüssel
<i>testmode</i>	boolean		Muss "1" sein
<i>number</i>	string		Rufnummer
<i>origin</i>	string	"LANDLINE"	Anrufnetz "LANDLINE", "MOBILE"
<i>caller</i>	string	leer	Telefonnummer d. Users
<i>tan</i>	string	bei mode = "DTMF"	TAN-Eingabe beim Anruf
<i>durationpart</i>	integer		anzurufende Sekunden

Tabelle A.10. Funktion `testcall` Rückgabe

Parameter	Typ	Standard	Beschreibung
error	integer	0	Fehlercode
errormessage	string	bei error <> 0	Fehlermeldung
handle	string(50)		Reservierungs ID

Fehlercodes

Tabelle B.1. durch Server verursachte Fehler (permanent)

error	Funktionen	Ursache	Reaktion
1000	alle	unbekannter Fehler	Information an micropayment Support
1001	alle	Fehler während Datenbankoperation	Information an micropayment Support
1002	alle	Initialisierung des Services ist fehlgeschlagen	Information an micropayment Support

Tabelle B.2. durch Server verursachte Fehler (temporär)

error	Funktionen	Ursache	Reaktion
2001	alle	Dienst vorübergehend nicht verfügbar (meist bei kurzzeitige Wartungsarbeiten)	Anzeige für User: "Bitte versuchen Sie es später noch einmal"
2002	init	Reservierung ist vorübergehend nicht möglich (meist ist der Rufnummernpool erschöpft)	Anzeige für User: "Bitte versuchen Sie es später noch einmal"

Tabelle B.3. vom Clienten verursachte Fehler

error	Funktionen	Ursache	Reaktion
3001	alle	Authorisierung ist fehlgeschlagen (accesskey oder IP des Clienten ist falsch)	Prüfen der Projekteinstellungen
3002	alle	Funktion nicht unterstützt oder verfügbar	Prüfen der Dokumentation, evtl. ist Funktion nur im Testmodus erlaubt
3003	alle	Parameter im falschen Format oder mit falschen Werten (meist bei fehlenden Pflichtparametern)	Prüfen der Dokumentation, Angabe aller erforderlichen Parameter, Angabe korrekter Werte
3004	alle	Fehler in Projektkonfiguration	Prüfen der Projekteinstellungen, errormessage liefert genauere Details
3005	init	Angefordertes Land ist nicht verfügbar	Anzeige für User: "Land wird nicht unterstützt"
3006	init, country	Angeforderter Betrag ist nicht erlaubt	Wahl eines anderen Zahlungsbetrags
3007	init, country	Angeforderte Währung kann nicht verarbeitet werden	Angabe der Währung nach ISO 4217
3008	status, info	handle ist ungültig	Angabe eines gültigen Handles, status liefert nur begrenzte Zeit Daten, verwenden der Funktion info

Tabelle B.4. durch User verursachte Fehler

error	Funktionen	Ursache	Reaktion
4001	testcall	Rufnummer bzw. TAN nicht reserviert oder Anruf läuft bereits	Angabe korrekter Werte, einmaliger Aufruf