
Securepin API v2.0

Copyright © 2009 micropayment™ gmbh

Sämtliche Inhalte dieses Dokuments unterliegen den internationalen Copyrightbestimmungen.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Allgemein	1
2. Schnittstelle	1
2.1. Allgemein	1
2.1.1. Abstrakt	1
2.1.2. Standardparameter und -rückgaben	2
2.1.3. Fehlercodes	2
2.1.4. Serviceprotokolle	3
2.2. Funktionen	4
2.2.1. Verfügbare Länder ermitteln mit country	4
2.2.2. Rufnummer reservieren mit init	5
2.2.3. Anrufüberwachung mit auth	7
2.2.4. Anrufsimitation mit testcall	7
A. Kurzreferenz	8

1. Einleitung

1.1. Allgemein

Die Securepin API bietet dem Partner die Möglichkeit, einen Kunden zu verifizieren. Dazu ruft der Kunde eine eindeutige Nummer des Micropayment-Systems auf. Mit diesem Call wird der Anschluß des Kunden ermittelt. Auf diese Art kann sich ein Kunde legitimieren und erhöht somit die Manipulationshürde entscheidend.

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

2. Schnittstelle

2.1. Allgemein

2.1.1. Abstrakt

Die Beschreibungssyntax 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 Bezahlssystems seine Implementierung testen kann

Rückgabe:

status :integer

enthält 0 bei Systemfehler; 1 Anfrage ist OK; 2 Anfrage existiert bereits und alte Daten werden zurückgegeben.

callstate :integer

enthält 0 wenn session unbekannt; 1 Reservierung existiert; 2 Anruf ist erfolgt

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 Webservice bedingt, haben aber nur eine vorübergehende Ursache, wie Wartungsarbeiten oder erschöpfte Ressourcen. Der Kunde kann hierbei auf spätere Nutzung des Dienstes vertröstet werden.

Clientfehler 3xxx

Die Ursache der Fehler aus dieser Klasse liegt typischerweise bei der aufrufenden Applikation. Es ist sinnvoll diese Fehler mit dem Text aus der 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 korrigieren 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/securepin/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-Envelope 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/securepin/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/securepin/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 wichtige Informationen zur Verfügbarkeit des Dienstes: eine Liste der verfügbaren Länder und der möglichen Telefongassen. 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

(*country*) :string

Ländercode des abzufragenden Landes. Wenn nicht gesetzt, wird das Land angenommen, das über ip ermittelt werden kann (Ist eine Auflösung der ip nicht möglich, wird DE angenommen).

(*ip* :string)

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

Rückgabe:

prefixcountry :string

Ländercode des abzufragenden Landes (entspricht *country* aus dem Request, d.h. Parameter *country* bzw. *ip*).

prefixcount :integer

Enthält die Anzahl der gefundenen Prefixe (normalerweise Nummerngruppen). Diese können bei "init" verwendet werden.

prefix[n] :string wenn: *prefixcount* > 0

Liefert jeweils einen gültigen Prefix für das abzufragende Land (*prefixcountry*).

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 Service verwendet 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 Securepinvorgang 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

userparam :string

Ein eindeutiges Handle, das die Kundensitzung identifiziert. Diese wird für den offenen Vorgang gespeichert. Sollte der Kunde den Vorgang 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.

timeout :integer

Maximale Dauer der Reservierung. Wenn 0 oder nicht gesetzt wird das Systemtimeout verwendet (ca. 2 Minuten)

(*prefix* :string)

(Nummern)Gasse bzw. Nummerpräfix oder Identifier der Nummer. z.B. 0800 oder 01802. Gültige Werte werden mit "country" ermittelt. Wenn der prefix nicht gesetzt wird, wird eine beliebige Nummer genommen.

(*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.

Rückgabe:

auth :string(50)

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

basenumber :string

Liefert die vom Kunden anzurufende Servicenummer. Bei Durchwahlnummer (siehe type) zusätzlich suffix als zu wählende Nummer anzeigen! Somit muß bei einer basenumber=12121212 sowie mode=0 und suffix=555 dem Kunden als zu wählende Nummer 12121212555 angezeigt werden.

(*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"

booking :datetime

Zeitpunkt der max Reservierung Formatbsp: 2006-04-30 23:01:00. Wenn die Session nicht durch Aufrufe mit Funktion "auth" refresht wird, wird die Session nach dieser Zeit freigegeben!

type: integer

Diese Rückgabe legt den Modus für die Rufnummer fest.

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

1 = Gelegentlich, meist bei ausländischen Rufnummern, kann keine durchwahlfähige Rufnummer reserviert werden. In diesem Fall identifiziert sich der Kunde mit einer TAN (Parameter suffix) durch Eingabe auf der Telefontastatur während des Anrufs.

price : double

Tarif Minutenpreis. Nur zur Info. Wenn nötig bitte numberinfo anzeigen..

currency :string

Währung von price.

dc: integer

Art des Tarifes.

0 = Zeittakt.

1 = Dropcharge (d.h. mögliche Gebühren sind unabhängig von der Anrufdauer des Kunden).

suffix :string

entweder zusätzlich zur Hauptnummer wählen (*type*=0) oder als einzugebende Pin (*type*=1)

Alle Parameter und Rückgaben sind in der Kurzreferenz zusammengefasst.

2.2.3. Anrufüberwachung mit `auth`

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 grade anruft oder bereits angerufen hat. Zusätzlich kann dem Kunden der Fortschritt bei zeitabhängigen Anrufen angezeigt werden, z.B durch eine AJAX-Lösung.

Parameter:

auth :string(40)

Die ID des Vorgangs aus der Funktion `init`.

timeout :integer

wenn 0 bleibt bisherige Reservierungsdauer erhalten. wenn > 0 wird die max. Reservierung auf (aktuelle zeit) + *timeout* gesetzt. default 0

Rückgabe:

booking :datetimestring

Der Zeitpunkt, wann die Reservierung abläuft und die Rufnummer verfällt. Durch wiederholtes Aufrufen der Funktion (*timeout* >0) wird dieser Wert immer aktualisiert.

(*caller* :string)

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")

(*freeparam* :string)

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

userparam :string

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

Alle Parameter und Rückgaben sind in der Kurzreferenz zusammengefasst.

2.2.4. 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 ein Anruf simuliert werden, um das Verhalten der eigenen Applikation zu testen..

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)

Falls erforderlich der Wert einer simulierten Tan-Eingabe (Parameter suffix aus "init" wenn type=1)

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. Bei Securepin ist die Anruferdauer irrelevant, da der Service sofort erfolgreich ist, wenn eine Verbindung zustande kommt.

Rückgabe:

auth :string(50)

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

Alle Parameter und Rückgaben sind in der Kurzreferenz zusammengefasst.

Kurzreferenz

Funktion `country`

Tabelle A.1. Funktion `country` Parameter

Parameter	Typ	Standard	Beschreibung
<i>accesskey</i>	string		Zugangsschlüssel
<i>testmode</i>	boolean	0	Testmodus aktivieren
<i>project</i>	string		Projektname
<i>country</i>	string		abzufragendes Land
<i>ip</i>	string	leer	Land für IP ermitteln

Tabelle A.2. Funktion `country` Rückgabe

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

Funktion `init`

Tabelle A.3. Funktion `init` Parameter

Parameter	Typ	Standard	Beschreibung
<i>accesskey</i>	string		Zugangsschlüssel
<i>testmode</i>	boolean	0	Testmodus aktivieren
<i>project</i>	string		Projektname
<i>userparam</i>	string		Usersession
<i>ip</i>	string		User IP
<i>country</i>	string(2)		Land des Users
<i>prefix</i>	string		Prefix/Gasse der Service- nummer
<i>freeparam</i>	string	leer	freier Parameter

Tabelle A.4. Funktion `init` Rückgabe

Parameter	Typ	Standard	Beschreibung
<code>error</code>	integer	0	Fehlercode
<code>errormessage</code>	string	bei <code>error <> 0</code>	Fehlermeldung
<code>callstate</code>	integer		Reservierungsstatus 0, 1, 2
<code>auth</code>	string(50)		Reservierungs ID
<code>basenumber</code>	string		Rufnummer
<code>suffix</code>	string		suffix bzw tan zur base- number
<code>booking</code>	datetime		Ablauf der Reservierung
<code>numberinfo</code>	string		gesetzl. Tarifangaben
<code>origin</code>	string		Erreichbarkeit d. Nummer "MOBILE", "LANDLINE", "BOTH"
<code>price</code>	integer		Tarif der Nummer
<code>currency</code>	string(3)		Währung
<code>type</code>	integer		Nummertyp 0,1
<code>dc</code>	integer		Nummerart 0,1 (takt oder dropcharge)

Funktion `status`

Tabelle A.5. Funktion `auth` Parameter

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

Tabelle A.6. Funktion `auth` Rückgabe

Parameter	Typ	Standard	Beschreibung
error	integer	0	Fehlercode
errormessage	string	bei error <> 0	Fehlermeldung
callstate	integer		Reservierungsstatus 0, 1, 2
expire	datetimestring		Ablauf der Reservierung
caller	string	wenn status >= "REINIT" und verfügbar	Telefonnummer d. Users
<i>userparam</i>	string		Usersession
freeparam	string	leer	freier Parameter
booking	datetime		Ablauf der Reservierung

Funktion testcall

Tabelle A.7. 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.8. Funktion testcall Rückgabe

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