



XML API Dokumentation für Entwickler

Version 2.02

23.07.2015

Inhalt

Einführung	4
Wer braucht diese Informationen	4
S-PAY Testumgebung.....	4
URL zur API	4
Test-API:.....	4
Live-API:	4
Vorbereitung.....	5
Voraussetzungen	5
API Funktionen in der Übersicht.....	5
Authentifizierung.....	6
Erzeugen des Authentifizierungs- Hash.....	6
Beispiel für die Hashbildung	6
Anfrage / Antwort Identifikation	6
Benutzerdefinierte Parameter.....	6
Beispiele von XML Anfragen und Antworten in PHP 5	7
Abfrage des Kontosaldos	7
Zu übergebende Parameter.....	7
API Rückgabe	7
Anfrage an die API	7
Antwort von der API	7
Abfrage der Kontobuchungen	8
Zu übergebende Parameter.....	8
API Rückgabe	8
Anfrage an die API	8
Antwort von der API	9
Interne Umbuchung zwischen Konten eines Kundenaccounts	10
Zu übergebende Parameter.....	10
API Rückgabe	10
Anfrage an die API	10
Antwort von der API	10
S-PAY interne Überweisung.....	11
Zu übergebende Parameter.....	11
API Rückgabe	11
Anfrage an die API	11
Antwort von der API	11
Einzelbuchung abfragen	12
Zu übergebende Parameter.....	12
API Rückgabe	12
Anfrage an die API	12
Antwort von der API	12

Kontonummer auf Existenz prüfen.....	13
Zu übergebende Parameter.....	13
API Rückgabe	13
Anfrage an die API	13
Antwort von der API	13
API Abfrage per GET-Methode	14
API Abfrage per Post-Methode (mit Hilfe von Curl)	14
Fehlerbehandlung.....	15
Fehlerhafte Abfrage an die API.....	15
Antwort von der API	15
Tabelle der möglichen Fehlermeldungen	16
Anmerkung zur Sicherheit	17
Noch Fragen oder Vorschläge?.....	17
Änderungsprotokoll.....	18

Einführung

Dieses Dokument enthält alle Funktionen und Spezifikationen der RBH Secure-Pay (S-PAY) API. Die S-PAY API ist ein XML Programm Interface, das Ihnen erlaubt den Zahlungsverkehrsprozess vollständig zu automatisieren. Dazu werden von der API folgende Funktionen zur Verfügung gestellt:

- Senden von Einzelüberweisungen
- Umbuchung zwischen eigenen Konten
- Abruf der Kontenhistorie
- Abruf einzelner Transaktionen
- Prüfung von Konten auf Gültigkeit

Diese Möglichkeiten werden von allen Anbietern benötigt, die S-PAY für Ihre Projekte als Zahlungsprovider anbieten wollen und die Ihren Zahlungsverkehr automatisieren wollen. Mit der API ist es möglich, z.B. Einzahlungen, die über die S-PAY Zahlungsseite erfolgt sind, gegen zu prüfen oder Provisionen automatisch an viele Empfänger auszuzahlen.

Wer braucht diese Informationen

Diese Dokumentation ist für Entwickler bestimmt, die S-PAY als Zahlungsmodul in Ihre Software einbinden wollen. Für eine erfolgreiche Implementierung sollten Entwickler über folgende Kenntnisse verfügen:

- Erstellen und Senden von Serveranfragen
- Empfangen und Auswerten von Daten, die von Webservern gesendet werden
- Kenntnisse des XML Standards

S-PAY Testumgebung

S-PAY stellt Ihnen unter <https://paytest.s-pay.me> eine komplette Testumgebung für das Zahlungssystem zur Verfügung. Wir empfehlen Ihnen, Ihre Anwendung dort ausgiebig zu testen, bevor Sie sie in den Livebetrieb übernehmen.

URL zur API

Für die API Anfragen nutzen Sie bitte folgende Adressen:

Test-API: <https://paytest.s-pay.me/api>

Live-API: <https://www.s-pay.me/pay/api>

Vorbereitung

Um mit der API zu arbeiten, müssen Sie folgende Vorbereitungen treffen:

1. Eröffnen Sie ein Kundenkonto im Testsystem unter <https://paytest.s-pay.me>
2. Legen Sie sich Konten in den benötigten Währungen an
3. Aktivieren Sie unter „Konten -> Kontenübersicht -> Bearbeiten“ die API für das jeweilige Konto, indem Sie den, nur für dieses Konto geltenden, API Schlüssel eingeben. Dieses Passwort benötigen Sie für jeden Zugriff auf die API.
4. Wenn Sie den API Zugriff des jeweiligen Kontos nur von bestimmten IP Adressen erlauben wollen, geben Sie die IP Adresse(n) unter IP-Whitelist ein.

Voraussetzungen

Um mit der S-PAY API zu arbeiten benötigen sie eine Programmiersprache, die es Ihnen ermöglicht, XML konforme Daten zu erzeugen und HTTPS POST und GET Anfragen zu versenden. Auch muss die Programmiersprache die sha512 Hashfunktion ermöglichen, dies ist z.B. PHP ab der Version 5.1.2.

Prinzipiell kann die API auch mit anderen Programmiersprachen angesprochen werden. Die Beispiele in dieser Dokumentation basieren auf PHP 5.

API Funktionen in der Übersicht

Die S-PAY API bietet Ihnen folgende Funktionen:

Funktion	Type Attribut	
Kontosaldo	getBalance	Abfrage des aktuellen Saldos eines Kontos in der jeweiligen Kontowährung
Kontoauszug	statement	Abfrage von allen Buchungen eines Datumbereichs
Einzelbuchung	getTransaction	Suchen einer Buchung anhand der Transaktionsnummer
Kontoprüfung	checkAccount	Abfrage, ob das übergebene Konto existiert
Umbuchung	repost	Umbuchung zwischen Konten Ihres Kundenkontos
Überweisung	internal	Überweisung, von einem Ihrer Konten, auf ein S-PAY Kundenkonto. Diese Funktion ermöglicht Ihnen auch Massenüberweisungen.

Authentifizierung

Für jeden Zugriff auf die S-PAY API werden die eigene Kontonummer und das API-Passwort des jeweiligen Kontos benötigt. Für jede API Anfrage sind folgende Authentifizierungsinformationen notwendig:

- Eigene Kontonummer bis zu 12 Zeichen
- Datum im UTC Format Y-m-d
- API Passwort
- unique_id eine einmalige durch Sie generierte ID

Erzeugen des Authentifizierungs- Hash

Aus den Daten wird ein Authentifizierungs-Token wie folgt erzeugt:

Kontonummer:Datum:API-Passwort

Beispiel für die Hashbildung

- Kontonummer= E0123456789
- Datum= 2015-12-06 im UTC Format
- API-Passwort=]d_RS&DIIUYM;PJ#j%DM){wSviiK`s42,!_2C^&jmgh55%~jJ/'1iD%7
- Unique_id= 999719-4599

Aus diesen Daten setzen wir eine durch Doppelpunkte getrennte Zeichenfolge zusammen. Bei unseren Beispieldaten sieht die Zeichenfolge wie folgt aus:

```
'E0123456789:2015-12-06:]d_RS&DIIUYM;PJ#j%DM){wSviiK`s42,!_2C^&jmgh55%~jJ/'1iD%7: 999719-4599'
```

Dieser String wird nun mit der sha512 Funktion gehasht, wir erhalten bei unseren Beispieldaten folgenden Hash:

```
2de672b8187a47d23556a3aa83a43378f8dd785c52fad84196f0a4b430af918f08071b901f27046ca4a3eae78b0fa85b1ef2815cab4380cbb4e3cbe4218fb02
```

Anfrage / Antwort Identifikation

Die S-PAY XML API braucht einen Parameter (unique_id) für die Identifikation von Anfrage und Antwort. Damit können Sie in Ihrem Programm prüfen, ob die S-PAY API Antwort zu Ihrer vorhergegangenen Abfrage passt. Der unique_id Parameter muss für das abzufragende Konto einmalig und eindeutig sein. Er muss im Datenblock der API Abfrage eingefügt werden und wird bei der Antwort vom S-PAY Server im Datenblock unverändert zurückgesendet. S-PAY nutzt diese ID um Doppelanfragen an die API zu unterbinden. Nur wenn für das betreffende Konto keine Anfrage mit der gleichen unique_id vorliegt, wird die aktuelle Anfrage ausgeführt. Falls die übergebene ID bei dem jeweiligen Konto schon einmal genutzt wurde, liefert die API den Fehler „Doppelanfrage“ zurück. Die unique_id ist ein Pflichtfeld und muss auch bei der Hashbildung genutzt werden.

Benutzerdefinierte Parameter

Die S-PAY XML API erlaubt beliebige zusätzliche Parameter im Datablock zu übergeben. Diese werden bei der Antwort an Ihren Server unverändert zurückgegeben.

Beispiele von XML Anfragen und Antworten in PHP 5

Für die Beispiele nutzen wir die folgenden Variablen:

```
$api_key = ']d_RS&DIIUYM;PJ#j%DM){wSviiK`s42,!_2C^&jmgh55%~jJ/'1iD%7'; // API Passwort
$account_no = 'E0123456789'; // eigene Kontonummer (11 Stellen, bei Exchangekonten 12 Stellen!)
$unique_id = '999719-4599';
$api_path = 'https://paytest.s-pay.me/api/'; // Pfad zur Test-API
```

Abfrage des Kontosaldos

Liefert den aktuellen Kontostand des abgefragten Kontos.

Zu übergebende Parameter

- unique_id durch Sie generierte einmaliger ID String hier: 999719-4599

API Rückgabe

- Kontostand, Nachkommastellen durch Punkt(.) getrennt. EUR/USD 2, Gold 3 und BTC 8 Nachkommastellen.
- Währung : EUR, USD, BTC, XAU in Gramm

Anfrage an die API

```
$request_xml = '
<request type="getBalance">
  <auth>
    <hash>'.hash('sha512', $account_no.'.'.gmdate('Y-m-d')).':'. $api_key.'.'. $unique_id).</hash>
    <account>'. $account_no.</account>
  </auth>
  <data>
    <unique_id>'. $unique_id.</unique_id>
  </data>
</request>
';
```

Antwort von der API

```
$response_xml = '
<response type="getBalance">
  <data>
    <unique_id>999719-4599</unique_id>
  </data>
  <result>
    <result>1</result>
    <balance>9543.91</balance>
    <currency>EUR</currency>
  </result>
</response>';
```

Abfrage der Kontobuchungen

Liefert die letzten 1000 Buchungen

Zu übergebende Parameter

- `datefrom` Startdatum (optional), Format YYYY-MM-DD
- `dateto` Enddatum (optional) , Format YYYY-MM-DD
- `unique_id` eine einmalige durch Sie generierte ID

API Rückgabe

- `account` Kontonummer des Empfängers / Senders
- `subject1` Verwendungszweck 1
- `subject2` Verwendungszweck 2
- `refid` S-Pay Referenznummer
- `reference_id` Referenz Nummer aus einer SCI Zahlung (falls gesetzt)
- `amount` Betrag
- `type` 1 = interne Umbuchung
2 = RBH-interne Überweisung
101 = Gebühren für interne Umbuchung
102 = Gebühren für interne Überweisung
- `rate` Umrechnungskurs zwischen Kontowährung und Ausgangswährung
- `currency` Ausgangswährung
- `currencyamount` Betrag in Ausgangswährung
- `valuta` Zeitpunkt Valuta

Anfrage an die API

```
$request_xml = '
```

```
<request type="statement">
  <auth>
    <hash>'.hash('sha512', $account_no.'.'.gmdate('Y-m-d').':'. $api_key.'.'. $unique_id).'
```

Antwort von der API

```
$response_xml = '
```

```
<response type="statement">
  <data>
    <datefrom>2015-06-01</datefrom>
    <dateto>2015-06-13</dateto>
    <unique_id>999719-4599</unique_id>
  </data>
  <result>
    <bookings>
      <booking>
        <account>E9876543210</account>
        <subject1>Verwendungszweck 1</subject1>
        <subject2>Verwendungszweck 2</subject2>
        <refid>1234</refid>
        <amount>123.45</amount>
        <type>2</type>
        <valuta>2015-06-03 10:19:21</valuta>
      </booking>
      <booking>
        <account>E1234567890</account>
        <subject1>Verwendungszweck 1</subject1>
        <refid>1239</refid>
        <amount>1234.56</amount>
        <type>2</type>
        <valuta>2015-06-09 10:18:58</valuta>
      </booking>
      // (max. 1000 Buchungen pro Abfrage)
    </bookings>
  </result>
</response>;
```

Interne Umbuchung zwischen Konten eines Kundenaccounts

Zu übergebende Parameter

- account Kontonummer des Empfängers
- amount Betrag
- currency Währung (EUR, USD, BTC, XAU)
- subject1 Verwendungszweck 1
- subject2 Verwendungszweck 2
- unique_id durch Sie generierte einmaliger ID String hier: 999719-4599

API Rückgabe

- transaction ID der Transaktion
- balance Kontostand, Nachkommastellen durch Punkt(.) getrennt. EUR/USD 2, Gold 3 und BTC 8

Anfrage an die API

\$request_xml = '

```
<request type="repost">
  <auth>
    <hash>'.hash('sha512', $account_no.':'.gmdate('Y-m-d').':'. $api_key.':'. $unique_id).'
```

Antwort von der API

\$response_xml = '

```
<response type="repost">
  <data>
    <account>E1234567890</account>
    <amount>123.45</amount>
    <currency>EUR</currency>
    <subject1>Verwendungszweck 1</subject1>
    <subject2>Verwendungszweck 2</subject2>
    <unique_id>999719-4599</unique_id>
  </data>
  <result>
    <result>1</result>
    <transaction>1234</transaction>
    <balance>9543.91</balance>
  </result>
</response>;
```

S-PAY interne Überweisung

Zu übergebende Parameter

- account Kontonummer des Empfängers
- amount Betrag
- currency Währung (EUR, USD, BTC, XAU)
- subject1 Verwendungszweck 1
- subject2 Verwendungszweck 2
- unique_id durch Sie generierte einmaliger ID String hier: 999719-4589

API Rückgabe

- transaction ID der Transaktion
- balance Kontostand, Nachkommastellen durch Punkt(.) getrennt. EUR/USD 2, Gold 3 und BTC 8

Anfrage an die API

\$request_xml = '

```
<request type="internal">
  <auth>
    <hash>'.hash('sha512', $account_no.':'.gmdate('Y-m-d').':'. $api_key.':'. $unique_id).'
```

Antwort von der API

\$response_xml = '

```
<response type="internal">
  <data>
    <account>E1234567890</account>
    <amount>123.45</amount>
    <currency>EUR</currency>
    <subject1>Verwendungszweck 1</subject1>
    <subject2>Verwendungszweck 2</subject2>
    <unique_id>999719-4589</unique_id>
  </data>
  <result>
    <result>1</result>
    <transaction>1234</transaction>
    <balance>9543.91</balance>
  </result>
</response>;
```

Einzelbuchung abfragen

Zu übergebende Parameter

- transaction Transaktions ID (Buchungsnummer)
- unique_id durch Sie generierte einmaliger ID String hier: 999719-4589

API Rückgabe

- account Kontonummer des Empfängers / Senders
- subject1 Verwendungszweck 1
- subject2 Verwendungszweck 2
- refid S-Pay Referenznummer
- reference_id Referenz Nummer aus einer SCI Zahlung
- amount Betrag
- type 1 = interne Umbuchung
2 = interne Überweisung
101 = Gebühren für interne Umbuchung
102 = Gebühren für interne Überweisung
- rate Umrechnungskurs zwischen Kontowährung und Ausgangswährung
- currency Ausgangswährung
- currencyamount Betrag in Ausgangswährung
- valuta Zeitpunkt Valuta

Anfrage an die API

```
$request_xml = '  
  <request type="getTransaction">  
    <auth>  
      <hash>.hash('sha512', $account_no.'.'.gmdate('Y-m-d').'.').$api_key.</hash>  
      <account>'.$account_no.</account>  
    </auth>  
    <data>  
      <transaction>1478</transaction>  
      <unique_id>'.$unique_id.</unique_id>  
    </data>  
  </request>';
```

Antwort von der API

```
$response_xml = '  
<response type="getTransaction">  
  <data>  
    <transaction>1478</transaction>  
  </data>  
  <result>  
    <bookings>  
      <booking>  
        <account>XA0004471041</account>  
        <subject1>Verwendungszweck 1</subject1>  
        <subject2>Verwendungszweck 2</subject2>  
        <refid>1478</refid>  
        <amount>-123.45</amount>  
        <type>2</type>  
        <valuta>2015-06-11 11:43:29</valuta>  
        <unique_id>999719-4589</unique_id>  
      </booking>  
    </bookings>  
  </result>  
</response>
```

```
</result>  
</response>;
```

Kontonummer auf Existenz prüfen

Zu übergebende Parameter

- account Kontonummer
- unique_id durch Sie generierte einmaliger ID String hier: 999719-4599

API Rückgabe

- status 1 = Kontonummer ok
 0 = Kontonummer nicht ok

Anfrage an die API

```
$request_xml = '  
  <request type="checkAccount">  
    <auth>  
      <hash>'.hash('sha512', $account_no.'.'.gmdate('Y-m-d')).'.'. $api_key.'.'. $unique_id).'  
      <account>'. $account_no.'  
    </auth>  
    <data>  
      <account>XA0004471041</account>  
      <unique_id>$.unique_id.</unique_id>  
    </data>  
  </request>;
```

Antwort von der API

```
$response_xml = '  
  <response type="checkAccount">  
    <data>  
      <account>XA0004471041</account>  
      <unique_id>999719-4589</unique_id>  
    </data>  
    <result>  
      <status>1</status>  
    </result>  
  </response>;
```

API Abfrage per GET-Methode

```
$response_xml = file_get_contents($api_path.'?data='.urlencode($request_xml));  
echo $response_xml;
```

API Abfrage per Post-Methode (mit Hilfe von Curl)

```
$ch = curl_init($api_path);  
curl_setopt($ch, CURLOPT_POST, 1);  
curl_setopt($ch, CURLOPT_POSTFIELDS, 'data='.$request_xml);  
curl_setopt($ch, CURLOPT_HEADER, 0);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
$response_xml = curl_exec($ch);  
echo $response_xml;
```

Fehlerbehandlung

Beispiel für eine mögliche Fehlermeldung beim Abfragen der Buchungen. Der Fehler wird erzeugt, da das Datum (31. Februar) nicht gültig ist. Im Falle eines Fehlers gibt die API immer eine Fehlernummer und eine kurze Beschreibung des Fehlers zurück.

Fehlerhafte Abfrage an die API

```
$request_xml = '
```

```
<request type="statement">
  <auth>
    <hash>'.hash('sha512', $account_no.'.'.gmdate('Y-m-d').'. $api_key.'.'. $unique_id).'
```

Antwort von der API

```
$response_xml = '
```

```
<response type="error">
  <data>
    <datefrom>2015-02-31</datefrom>
    <unique_id>999719-4589</unique_id>
  </data>
  <error>
    <code>53</code>
    <text>Invalid date value in field (datefrom)</text>
  </error>
</response>;
```

Tabelle der möglichen Fehlermeldungen

Fehlercode	Fehlermeldung
1	Received no POST or GET data
2	There was no XML data
3	No request type
4	Unknown request type
10	Authentication failed: no account and/or hash
11	Authentication failed: wrong account and/or hash
12	API is not active for this account
13	IP XXX.XXX.XXX.XXX is not whitelisted for this account
50	Empty mandatory fields (Name of Field with Error)
51	Additional field is not set (Name of Field with Error)
52	Invalid value for field (Name of Field with Error)
53	Invalid date value in field (Name of Field with Error)
60	Unknown currency
100	Not enough money in account
101	Different customers
102	Unknown beneficiary account
9997	unique_id was already used for this account
9998	API is temporarily disabled
9999	Unknown Error

Anmerkung zur Sicherheit

Um sicher zu stellen, dass der S-PAY XML API Kontozugriff unter sicheren und optimalen Bedingungen stattfindet, empfiehlt S-PAY Ihnen folgende Sicherheitsmaßnahmen:

- Geben Sie nie Ihre Zugangsdaten zum S-PAY Kundenkonto an Dritte weiter!
- Achten Sie darauf, dass Ihr API Passwort nur Ihnen bekannt ist! Es ist der ultimative Schlüssel zu Ihrem Konto und den Zugriff auf Ihr dort liegendes Geld!
- Wenn Sie externe Programmierer beschäftigen, lassen Sie Ihre Anwendung mit dem S-PAY Testsystem entwickeln und achten Sie darauf, dass beim Umzug von der Testumgebung zum Livesystem nur Sie Zugriff auf die Authentifizierung Daten zu Ihrem Live-Account haben.
- Loggen Sie sich nicht in der Öffentlichkeit in Ihren S-PAY Account ein. Achten Sie darauf, sich nur von privaten Geräten einzuloggen, die sicher sind! (Alle Sicherheitsupdates, aktuelle Schutzprogramme usw.)
- Speichern Sie Ihre Zugangsdaten nicht in einfachen unverschlüsselten Textdateien. Wenn Sie Passwörter sicher auf Ihrem Rechner speichern wollen, nutzen Sie dazu Programme wie Keypass.
- Nutzen Sie für den Zugriff auf die API nur Rechner, die eine fixe IP Adresse haben! Durch die Zugriffbeschränkung auf diese IP verhindern Sie, dass von anderen Rechnern aus versucht werden kann, Ihren API Zugang zu hacken.

Noch Fragen oder Vorschläge?

Falls Sie Fragen haben oder Vorschläge die API zu verbessern, wenden Sie sich einfach über das Ticketsystem an unseren Service. Für noch schnellere Kommunikation können Sie uns auch unter Skype (Skype Name: rbh-service) kontaktieren. Wir sind stets bemüht unsere Produkte sicherer zu machen und zu verbessern. Helfen Sie uns dabei!

Änderungsprotokoll

Datum	Art	Beschreibung
26.05.2015	API System	IP Whitelisting je Konto zugefügt
08.06.2015	API Rückgabe erweitert	Bei Buchungsabfragen wird die reference_id zurückgegeben, wenn Sie bei einer SCI Zahlung gesetzt wurde.
11.06.2015	API Anfrage zugefügt	Abfrage auf Existenz eines Kontos im S-PAY System
11.06.2015	API Anfrage zugefügt	Abfrage einer Buchung anhand der Transaktionsnummer
12.06.2015	Anfrage Parameter zugefügt	Parameter unique_id zugefügt um Doppelanfragen zu vermeiden
12.06.2015	API Fehlercode	Fehlercode 13, 9997, 9998 zugefügt
12.06.2015	API System	Datum zur Hashbildung auf UTC umgestellt
20.07.2015	API System	Hashbildung um unique_id erweitert und unique_id ist nun Pflichtfeld
23.07.2015	API Rückgabe erweitert	repost und internal geben zusätzlich den aktuellen Kontosaldo zurück