



# Schnittstellen-Beschreibung Button- und iFrame-Integration

Version 3.5 vom  
20.08.2019

[www.sepa.net](http://www.sepa.net)

**Autor:**  
**SEPA.net**, ein Dienst der  
GRÜN Digital Invest GmbH

Pascalstraße 6  
52076 Aachen

# Inhaltsverzeichnis

|   |    |
|---|----|
| 1. Einleitung.....  | 3  |
| 2. Grundsätzliche Funktionsweise.....                       | 4  |
| 2.1. Art der Einbindung: Kauf-Button oder iFrame .....      | 4  |
| 2.2. Individuelle Anpassung des Bezahldialogs .....         | 4  |
| 2.3. Was passiert nach dem Bezahlvorgang? .....             | 4  |
| 2.3.1. Prüfung der Berechtigung .....                       | 5  |
| 2.3.2. Auslieferung des Inhalts/Dienst-Freischaltung .....  | 5  |
| 2.4. Demo- oder Produktiv-Modus.....                        | 6  |
| 2.5. Vorbefüllung der Felder im Bezahldialog .....          | 6  |
| 3. Einbindung von SEPA.net in die Website .....             | 6  |
| 3.1. Einbindung per Kauf-Button.....                        | 6  |
| 3.1.1. Referenzieren einer JavaScript-Datei: .....          | 6  |
| 3.1.2. Der Aufbau des Kauf-Buttons.....                     | 7  |
| 3.2. Einbindung per iFrame .....                            | 8  |
| 4. Die Schnittstellen-Parameter .....                       | 9  |
| 4.1. „anker“: .....   | 9  |
| 4.2. „oid“: .....   | 9  |
| 4.3. „data“: .....  | 9  |
| 4.3.1. Auflistung aller Unterparameter aus „data“: .....    | 10 |
| 5. Bei Weiterleitung prüfen, ob der Kunde bezahlt hat ..... | 12 |
| 5.1. Aktivieren des PermissionChecks .....                  | 12 |
| 5.2. Auswertung des PermissionChecks.....                   | 12 |
| 6. Über Zahlungsstörungen informiert werden .....           | 13 |
| 7. Nachträglich Status von Bezahlvorgängen abfragen.....    | 14 |
| 7.1. Aufruf der Status-Schnittstelle .....                  | 14 |
| 7.2. Antwort der Status-Schnittstelle .....                 | 14 |
| 8. Support .....  | 15 |
| 9. Anhang: Reseller-Programm .....                          | 15 |

# 1. Einleitung

Das vorliegende Dokument erklärt die technische Einbindung des Bezahlsystems SEPA.net. Über SEPA.net können Endkunden auf einer Internetseite sichere und schnelle Kauf- bzw. Bezahlprozesse durchführen.

Voraussetzung für die Nutzung der Schnittstelle ist ein gültiger Kunden-Account im SEPA.net-System, der über die Website [www.sepa.net](http://www.sepa.net) kostenlos beantragt werden kann.

SEPA.net eignet sich zum Verkauf digitaler Inhalte und um kostenpflichtige Online-Dienste zu aktivieren. Der Einfachheit halber wird im Folgenden nicht mehr zwischen digitalen Inhalten und Online-Diensten unterschieden und einheitlich der Begriff „Artikel“ verwendet.

SEPA.net eignet sich gleichermaßen für „kleine“, statische Internetseiten als auch für große Portale mit breitgefächerten dynamischen Inhalten. Um den Betreibern kleinerer, statischer Seiten einen ganz einfachen Weg zur Integration von SEPA.net zu bieten, stellt SEPA.net fertige Kauf-Buttons zum Verkauf von Dateien zur Verfügung. Diese können an geeigneter Stelle in der Internetseite des Anbieters platziert werden. Beim ersten Klick auf den Button kann der Button konfiguriert werden (welche Datei soll zu welchem Preis verkauft werden). Über folgenden Link können fertige Kauf-Buttons bezogen werden:

<http://www.sepa.net/loesungen/kauf-button/button-generieren/>

Hersteller von HTML-Editoren, Shop-Systemen oder CMS können die genannten „fertigen“ SEPA.net-Kaufbutton ganz einfach selbst generieren und Ihren Kunden als Reseller zur Verfügung stellen (siehe Anhang).

Soll SEPA.net für eine größere oder wechselnde Anzahl von Artikeln eingesetzt werden, empfiehlt sich die Einbindung mit selbst (dynamisch) generierten Kauf-Buttons oder iFrames. Diese Art der Einbindung wird in der vorliegenden Dokumentation näher beschrieben.

## 2. Grundsätzliche Funktionsweise

### 2.1. Art der Einbindung: Kauf-Button oder iFrame

Das Kundeninterface von SEPA.net kann auf zwei verschiedene Weisen in die Seite des Anbieters integriert werden:

- Kauf-Button: Das Kundeninterface von SEPA.net öffnet sich bei Klick auf einen „SEPA.net-Kaufbutton“. Hierzu muss auf der Seite lediglich ein SEPA.net-Kaufbutton eingebunden werden. Bei Klick auf diesen Button wird der Seite automatisch eine neue Division (Layer) hinzugefügt, die über dem Content der Seite platziert wird. Durch die geringe Größe des SEPA.net-Kundeninterfaces (ca. 290 x 272 Pixel) bleibt die Ursprungsseite größtenteils sichtbar. Dennoch werden alle für Endkunden notwendigen Informationen zu SEPA.net innerhalb dieses Layers dargestellt.
- iFrame: Das Kundeninterface von SEPA.net ist innerhalb der Seite des Anbieters von Anfang an geöffnet. Hierzu wird das SEPA.net-Kundeninterface auf der Seite des Anbieters innerhalb eines iFrames mit den Maßen 370 x 260 Pixel fest eingebunden.

### 2.2. Individuelle Anpassung des Bezahldialogs

Die Texte im SEPA.net-Fenster sind standardmäßig so formuliert, dass sie für eine Vielzahl von Produkten bzw. Diensten passen. Auf Anfrage sind diese Texte individuell auf die zu verkaufende Leistung anpassbar. (siehe Parameter „template“ unter Punkt 4.3).

### 2.3. Was passiert nach dem Bezahlvorgang?

Hat ein Kunde über SEPA.net einen Bezahlvorgang durchgeführt, wird ihm abschließend von SEPA.net per E-Mail ein Link zugeschickt, über den er den bezahlten Inhalt abrufen kann (z.B. Download eines Artikels, Aktivierung eines Dienstes, etc.).

### 2.3.1. Prüfung der Berechtigung

Der Link verweist auf einen speziellen Download-Bereich von SEPA.net, der die Berechtigung des Kunden zum Abruf des bezahlten Inhalts prüft. Das System prüft dabei folgende Kriterien:

- Hat der Kunde tatsächlich über SEPA.net bezahlt?
- Liegt der Abruf innerhalb des zulässigen Zeitfensters?  
(kann vom Anbieter festgelegt werden)
- Hat der Kunde die maximale Downloadzahl noch nicht überschritten?  
(kann vom Anbieter festgelegt werden)

### 2.3.2. Auslieferung des Inhalts/Dienst-Freischaltung

Bei positiver Prüfung wird eine Auslieferungs-URL aufgerufen, die der Anbieter festgelegt und über den Kauf-Button an SEPA.net kommuniziert hat. Der zurückgelieferte Inhalt wird an den Browser des Client geschickt.

Der Aufruf der Auslieferungs-URL kann auf zwei verschiedene Arten erfolgen. Welche der beiden Varianten zum Einsatz kommen soll, kann der Anbieter durch einen Parameter innerhalb des Kauf-Buttons festlegen:

- Proxy: Bei dieser Variante ruft nicht der Client sondern ein spezieller Proxy-Server von SEPA.net die Auslieferungs-URL ab und reicht den zurückgelieferten Inhalt eins zu eins an den Client weiter. Bei dieser Variante erhält der Endkunde also keinen eigenen Zugriff auf die Auslieferungs-URL, sondern ruft den Inhalt über den Proxy-Server von SEPA.net ab, der mit zusätzlichen Sicherungsmechanismen ausgestattet ist. Auf diese Weise ist sichergestellt, dass die Auslieferungs-URL (und damit der gekaufte Inhalt) nicht beliebig oft abgerufen werden kann, z.B. weil der Endkunde die Auslieferungs-URL an Freunde weitergibt oder in ein Forum stellt.
- Weiterleitung: Bei dieser Variante wird der Browser des Endkunden nach Prüfung der Berechtigung (siehe 2.3.1) direkt auf die Auslieferungs-URL (bzw. Freischalt-URL) des Anbieters weitergeleitet. Die Auslieferungs-URL ist dabei in der Adresszeile des Browsers zu sehen. Der Anbieter muss also selber dafür sorgen tragen, dass die Auslieferungs-URL nicht beliebig oft oder gar von Dritten abgerufen werden kann. SEPA.net unterstützt den Anbieter dabei, indem optional von SEPA.net ein zusätzlicher Parameter an die Auslieferungs-URL gehängt wird, mit dem der Anbieter die Berechtigung selbständig überprüfen kann („PermissionCheck“, siehe Punkt 5).

## 2.4. Demo- oder Produktiv-Modus

Um die Funktionsweise von SEPA.net während der Integration problemlos testen zu können, kann SEPA.net im Demo-Modus betrieben werden. Im Demo-Modus werden keine echten Zahlungsvorgänge veranlasst. Der Demo-Modus kann über einen Parameter aktiviert werden (siehe 4.3). Im Demo-Modus blinkt im BezahlDialog unten links ein entsprechender Hinweis und die Felder müssen nicht ausgefüllt werden. Standardmäßig startet SEPA.net im Produktiv-Modus.

## 2.5. Vorbefüllung der Felder im BezahlDialog

Die Abfragefelder im SEPA.net-BezahlDialog können vom Anbieter vorbelegt werden, wenn die Daten bereits anderweitig erhoben wurden. Die entsprechenden Daten werden als „Data-Parameter“ übergeben (siehe z.B. Parameter „customer\_email“ unter 4.3)

# 3. Einbindung von SEPA.net in die Website

## 3.1. Einbindung per Kauf-Button

Damit SEPA.net innerhalb einer Internetseite über die Kauf-Button-Variante gestartet werden kann, sind folgende zwei Änderungen an der Seite vorzunehmen:

- Referenzieren einer JavaScript-Datei (stellt notw. SEPA.net-Funktionen bereit)
- Kauf-Buttons innerhalb der Seite (öffnen den SEPA.net-BezahlDialog)

### 3.1.1. Referenzieren einer JavaScript-Datei:

Soll auf einer Internetseite SEPA.net genutzt per Kauf-Button werden, muss an geeigneter Stelle (etwa im Head-Bereich) eine JavaScript-Datei eingebunden werden. Hierzu bitte folgenden Code-Schnipsel verwenden (ohne Zeilenumbruch; dieser entsteht hier nur aus Platzgründen):

```
<script type="text/javascript"
src="https://payment.sepa.net/capp/javascripts/minipay"></script>
```

Im obigen Code-Schnipsel wird die JavaScript-Datei über das Protokoll „https“ eingebunden. Bitte achten Sie für eine größtmögliche Browserkompatibilität darauf, dass hier immer das Protokoll der Seite verwendet wird, auf der die JavaScript-Datei eingebunden wird (also unter Umständen „http“).

### 3.1.2. Der Aufbau des Kauf-Buttons

Überall da, wo innerhalb der Internetseite ein SEPA.net-Bezahlvorgang gestartet werden soll, muss ein SEPA.net-Kauf-Button eingebaut werden. Über die Kauf-Buttons werden verschlüsselt alle Daten an SEPA.net kommuniziert, die für den Verkaufsprozess notwendig sind, z.B. der Name oder Preis des Artikels. Der Kauf-Button besteht aus einer verlinkten Grafik. Im Link ist ein JavaScript-Aufruf mit fest definiertem Aufbau integriert. Ein Kauf-Button könnte z.B. wie folgt aussehen (ohne Zeilenumbrüche; diese entstehen hier nur aus Platzgründen):

#### Beispiel

```
<a href="https://www.sepa.net/sicher-bezahlen/"
onclick="minipay_open(this, '1000', 'S0D24wAFeNdLdLzRpPlVPjpacyg0gaXVLaXo00wX
32dgoeLrgErt+H42GqVkhGGipWuqKtn30FBjX6eapu8cI7xXzSQwKUgej6vi0tPTQJVZTbwlSHM
I9xmdwq4yFzfhqHPDqAzhKc0sYX2ZX/PWXG25YprueSMZvlag9zqodAE5+VUuNqSPGWQDZq7C9+
hi3+IU8JTrVtPJF51d8KtIep0PmhUcR37I3CyK0iQW4s3aK86vKUmfKMhwojxWNMwMRtPANnxFL
nagbzJjBJAapl2/c0hpCxBpoB15dJ7M3jVr3Zhk5p50ePPEz1TcnDWYVc3rVXz7xdQwbI+NwzjT
8/cmV3Iz8tuztaFa8FrTtKA+kWJ70wW+dRvu3BKpApF9aCLDQ/PaVTJTDkMqfvsXTE5v1s0yB8P
VLmy3ZVSjlaaT6J3eRoesrWqav/2UIP4UgPAityPnPfyrAGQ/CWBrkyYd1PCb5uLqQM4GpK6v08
s=');return false;"></a>
```

Jeder Kauf-Button gilt nur für einen Artikel, da in ihm alle notwendigen Daten zu diesem Artikel codiert sind. Er ist zeitlich unbegrenzt gültig.<sup>1</sup> Es können mit einem Kauf-Button beliebig viele Bezahlvorgänge beliebig vieler Kunden durchgeführt werden.

Ein Kauf-Button ist folgendermaßen aufgebaut:

```
<a href="https://www.sepa.net/sicher-bezahlen/" "
onclick="minipay_open( Anker, 'Kundennummer', 'Data' );return false;">
</a>
```

SEPA.net wird geöffnet durch einen JavaScript-Funktionsaufruf, der beim Event „onclick“ ausgeführt wird. Das im Link enthaltene „href“ wird hingegen nur aufgerufen, wenn der User JavaScript deaktiviert hat.

Wie im obigen Kasten ersichtlich, hat der Funktionsaufruf „minipay\_open“ drei Parameter: 'Anker', 'Kundennummer' und 'Data'. Diese werden im Abschnitt „Schnittstellen-Parameter“ näher erläutert.

## 3.2. Einbindung per iFrame

Soll SEPA.net mit der Einbindungs-Variante „iFrame“ in die Seite des Anbieters integriert werden, muss an geeigneter Stelle ein „iFrame“ mit den Maßen 370x220 Pixel platziert werden. Bitte Scrolling im iFrame deaktivieren.

Beispiel:

```
<iframe src="https://payment.sepa.net/capp/payments/start?
oid=1000&layout=extended&data=S5kxhQACbMR%2FbgqxYEE0f0ZfvHZWs5d5kwL9r%2FLXj
U7jv8ninIIX0v76h%2B7MzokNATIitzCdkKkk0l0TE7WPKCyybE8EPMPiPM8bnQvi7IuNqa5To4n
6ilbYdXlppahxYCJQre0FKX4yGxEsP9ippbqE6asTh7ZCkxZx%2BXL70qEA5%2Fs f61zs5XVdv
E0Wzz1ePWQK6k8%2FIDHeMl0vDReb5cplso4LD4%2FfkJTPNt%2BmpwoYA4XzbpW6g5rtFPeWtj
fw9HMxUPx1wKfIE3JdlSwZ61IDx20Xbn5kXcMRTyIUg%2FAs0Rdo%2F8BGL9w0mM%2BeJcsZB%2
BMxD5YqCyqxXBAXmIwx7ygsSQz%2Be7%2Bz2m8uc25x0x5kzYyLuSq0Mda4zSwIx0SDCt3mT7v
ia%2BGAR%2BVXgSPqar07vEd05mk0ivWXFVxnftn0de8DFuNNUg"
style="width:370px;height:220px" scrolling="no">
</iframe>
```

Wie im obigen Beispiel ersichtlich, referenziert der iFrame auf die URL <https://payment.sepa.net/capp/payments/start/>. Dabei werden folgende GET-Parameter übergeben, von denen „oid“ und „data“ im Abschnitt „Schnittstellen-Parameter“ näher erläutert werden:

- oid (Ihre SEPA.net-Kundennummer)
- layout (bei iFrame-Einbindung immer folgender String: „extended“)
- data (enthält verschlüsselt alle für den Bezahlprozess wichtigen Daten)

1. Bitte beachten Sie, dass daher der Kauf-Button entfernt werden muss, sobald der zu verkaufende digitale Inhalt oder Dienst nicht mehr verfügbar ist.



## 4. Die Schnittstellen-Parameter

### 4.1. „anker“:

Dieser Parameter bestimmt bei der Einbindungsvariante „Kauf-Button“, über welchem HTML-Element der Internetseite das SEPA.net-Fenster geöffnet werden soll. In den meisten Fällen soll SEPA.net direkt über dem Link geöffnet werden, auf den der Kunde gerade klickt. In diesem Fall ist als Parameter „Anker“ einfach „this“ zu verwenden (ohne Anführungsstriche). Alternativ kann hier aber auch ein anderes Element der HTML-Seite notiert werden. Beispiel:  
„document.getElementById('test')“.

### 4.2. „oid“:

Die Kundennummer wird Ihnen nach der Registrierung als SEPA.net-Verkäufer per E-Mail zugeschickt. Sie gilt für alle Bezahlvorgänge der mit der Kundennummer verknüpften natürlichen oder juristischen Person.

### 4.3. „data“:

Der Data-Parameter ist eine verschlüsselte Zeichenkette, in der alle Daten des jeweiligen Artikels enthalten sind. Die Verschlüsselung des Data-Parameters erfolgt mit dem Verschlüsselungs-Algorithmus „xtea“ (Extended Tiny Encryption Algorithm), der sehr sicher ist und dank seiner großen Verbreitung in vielen Systemumgebungen leicht einzubinden ist. Als Verschlüsselungs-Key ist der „Security-String“ zu verwenden, der bei der Registrierung zusammen mit der Kundennummer von SEPA.net vergeben wird.

Unverschlüsselt könnte der Parameter „Data“ z.B. folgendermaßen aussehen:

```
sec=ab56b4d92b40713acc5af89985d4b786&shopDomain=www.ihredomain.de&offer  
Type=sale&productName=Testartikel&productId=TestId&price=0.99&tax=19.00  
&url=http%3A%2F%2Fwww.minipay.de%2Finfo%2Ffiles%2F  
minipay_agb_verkaeufer.pdf&useProxy=0&fileType=pdf&testmode=1&username=test  
&password=test
```

Am Beispiel wird ersichtlich, dass der Parameter „Data“ in unverschlüsselter Form aus einer Reihe von „Unter-Parametern“ besteht, die gemäß der GET-Parameterübergabe in URLs notiert sind:

- NameDesUnterparameters = WertDesUnterparameters
- die Trennung der einzelnen Unterparameter erfolgt durch ein „&“.

- Der Wert jedes einzelnen Unterparameters ist URL-Encodiert.

Eine Liste sämtlicher Unterparameter finden Sie in der folgenden Tabelle. Abschließend wird der **gesamte** String mit dem Verschlüsselungs-Algorithmus „xtea“ verschlüsselt; als Verschlüsselungs-Key ist der „Security-String“ zu verwenden.

#### 4.3.1. Auflistung aller Unterparameter aus „data“:

| Parameter-Name | Pflicht | Format              | Erklärung  |
|----------------|---------|---------------------|--|
| sec            | ja      | String(32)          | Security-String (von SEPA.net vergeben)  |
| shopDomain     | ja      | String (50)         | Domain, über die der Artikel verkauft wird. Taucht später z.B. in der Rechnung an Endkunden auf.   |
| offerType      | ja      | 'sale'    'service' | Bei Verkauf digitaler Inhalte: „sale“; Bei Bezahlung von Diensten: „service“   |
| productName    | ja      | String (100)        | Wie heißt das Produkt gegenüber Endkunden? Taucht im BezahlDialog und auf Rechnung an Endkunden auf.   |
| productId      | nein    | String (50)         | Optionaler Identifier des Produkts (erscheint in monatlicher Anbieter-Abrechnung)  |
| price          | ja      | Decimal (5,2)       | Preis des Produkts (brutto) in Euro. Muss innerhalb der erlaubten Preisspanne liegen   |
| tax            | ja      | Decimal (5,2)       | In „price“ enthaltener MwSt. -Satz (%)   |
| url            | ja      | String (1000)       | Welche Auslieferungs-URL soll nach dem Bezahlvorgang durch SEPA.net bzw. den Endkunden aufgerufen werden? Hier liegt bei offerType=„sale“ der gekaufte Artikel. Gültige URL oder IP. |
| useProxy       | nein    | 0    1              | Soll SEPA.net zum Schutz des verkauften Inhalts die o.g. URL via Proxy abrufen und so den Content an den Endkunde ausliefern (1=Ja)?   |
| username       | nein    | String (50)         | Falls useProxy=1 und die URL passwortgeschützt ist (HTACCESS)  |
| Password       | nein    | String (50)         | Zu „username“ gehörendes Passwort  |
| fileType       | nein    | String (4)          | Dateityp des Contents (z.B. „pdf“)   |
| fileName       | nein    | String (100)        | Opt. Dateiname bei Download via Proxy  |
| testmode       | nein    | 0    1              | Soll SEPA.net zum Testen im Demo-Modus gestartet werden (1=Ja)? Falls Ja werden keine echte Zahlung ausgelöst und die  |

|                     |      |              |  |
|---------------------|------|--------------|--|
|                     |      |              | Felder im Dialog müssen nicht zwingend ausgefüllt werden.  |
| maxDownloadNumbers  | nein | Integer      | Wie oft darf Kunde nach Bezahlvorgang Download durchführen. Minimum=1, Maximum=10, Default=10  |
| maxDownloadDuration | nein | Integer      | Wie lange darf Kunde nach Bezahlvorgang Download durchführen (in Minuten). Minimum=1, Maximum=43200, Default=43200   |
| template            | nein | String (20)  | Name des gewünschten Templates zur Feinjustierung des Bezahldialogs  |
| processId           | nein | String (100) | In Ihrem System generierte ID, mit der Sie später per XML-Schnittstelle den Status der Zahlung abfragen können.  |
| eventUrl            | nein | String (900) | URL, die bei Änderungen des Status einer Zahlung aufgerufen werden soll.   |
| customer_email      | nein | String (50)  | E-Mail des Kunden (zur Vorbefüllung)   |
| customer_phone      | nein | String (50)  | Handy-Nr. des Kunden (s.o.)  |
| customer_name       | nein | String (50)  | Vor- und Nachname des Kunden (s.o.)  |
| customer_account    | nein | String (50)  | Konto-Nr. des Kunden (s.o.)  |
| customer_bankcode   | nein | String (50)  | BLZ des Kunden (s.o.)  |
| reseller            | nein | Integer (4)  | Zugehörige Reseller-ID   |
| billing             | nein | 0    1       | Soll SEPA.net per E-Mail im Namen des Anbieters eine Rechnung an den Endkunden verschicken (Default=1)   |
| url_afterDialog     | nein | String (255) | Optionale URL zum Triggern, ob der Bezahldialog vom Endkunden vollständig durchlaufen wurde. Wird vom Endkunden unbemerkt clientseitig auf der Abschlussseite des Bezahldialogs aufgerufen (also bevor Endkunde den per Email zugesandten Link aufruft). |

Für die Kodierung des data-Parameters in XTEA kann in unserem Downloadbereich eine Bibliothek für PHP oder ASP heruntergeladen werden. Dort werden unter anderem auch Beispiele für die Einbindung als i-Frame bereitgestellt.

## 5. Bei Weiterleitung prüfen, ob der Kunde bezahlt hat

Soll der Endkunde den erworbenen Inhalt direkt von den Seiten des Anbieters beziehen (also per „Redirect“ und ohne Einsatz des SEPA.net-Proxy-Servers), muss der Anbieter selbst dafür Sorge tragen, dass nur berechtigte Personen, die bezahlt haben, den Inhalt abrufen können. SEPA.net unterstützt den Anbieter hierbei über einen optionalen Parameter, der von SEPA.net in die Auslieferungs-URL eingefügt werden kann (PermissionCheck). Durch Auswertung dieses Parameters kann der Anbieter selbständig überprüfen, ob der User die notwendige Berechtigung hat.

### 5.1. Aktivieren des PermissionChecks

SEPA.net fügt den Parameter PermissionCheck optional in die vom Anbieter festgelegte Auslieferungs-URL ein. Wird der Endkunde nach dem Bezahlvorgang auf die Auslieferungs-URL geleitet, kann der Anbieter den Parameter auslesen und überprüfen.

Der Anbieter kann selbst festlegen, an welcher Stelle innerhalb der Auslieferungs-URL der Parameter PermissionCheck nachträglich von SEPA.net eingefügt werden soll. Hierzu muss der Anbieter in der Auslieferungs-URL an gewünschter Stelle den String \*PermissionCheck\* als Platzhalter vorsehen; dieser wird von SEPA.net durch den Wert des PermissionCheck ersetzt.

Beispiele von URLs mit dem Platzhalter:

```
www.anbieter.de/pfad/file.php?id=123&checkwert=*PermissionCheck*  
www.anbieter.de/controller/action/param1/*PermissionCheck*/param2
```

Vor dem Weiterleiten des Endkunden beim Abruf des Inhalts wird der String \*PermissionCheck\* von SEPA.net durch den eigentlichen Wert ersetzt.

### 5.2. Auswertung des PermissionChecks

Der Anbieter kann anhand des Wertes PermissionCheck selbständig überprüfen, ob der User die notwendige Berechtigung zum Abruf des angefragten Inhalts hat. Hierzu werden im PermissionCheck durch SEPA.net folgende Werte verschlüsselt an den Anbieter übergeben (Unterparameter):

- ip: aktuelle IP des Users (bei Klick auf Auslieferungslink in Email)
- url: Auslieferungs-URL, mit noch nicht ersetzttem \*PermissionCheck\*
- productId: ID des Produkts, sofern als Data-Unterparameter übergeben
- time: Unix-Timestamp der Erzeugung des PermissionCheck
- sec: md5-Hashwert des Security-Strings des Anbieters

Beispiel eines entschlüsselten PermissionChecks:

```
ip=123.456.789.1&url=www.anbieter.de%2Fpfad%2Ffile.php
%3Fid%3D123%26checkwert%3D%2APermissionCheck%2A&productId=123456&time=12640
67264&sec=ab56b4d92b40713acc5af89985d4b786
```

Wie das Beispiel zeigt, sind die einzelnen „Unter-Parameter“ des PermissionChecks gemäß der GET-Parameterübergabe in URLs notiert (wie beim Aufbau des Parameter „Data“, vgl. Abschnitt 4.3).

- NameDesUnterparameters = WertDesUnterparameters
- Die Trennung der einzelnen Unterparameter erfolgt durch ein „&“.
- Der Wert jedes einzelnen Unterparameters ist URL-Encodiert.

Der PermissionCheck wird verschlüsselt übergeben. Die Verschlüsselung erfolgt auch hier mit dem Verschlüsselungs-Algorithmus „xtea“ (Extended Tiny Encryption Algorithm). Als Verschlüsselungs-Key ist der „Security-String“ zu verwenden, der bei der Registrierung von SEPA.net zusammen mit der Kundennummer vergeben wird.

## 6. Über Zahlungstörungen informiert werden

Über eine Event-Schnittstelle kommuniziert SEPA.net dem Anbieter optional, sobald Probleme bei einer Zahlung auftreten (z.B. weil der Endkunde einer Lastschrift widerspricht) bzw. sobald die Probleme wieder gelöst werden (weil der Endkunde den offenen Betrag im Mahnverfahren ausgleicht). Dies gibt dem Anbieter z.B. die Möglichkeit, den Endkunden in seinem System so lange zu sperren, bis der offene Betrag ausgeglichen wurde.

SEPA.net kommuniziert diese Änderungen im Status einer Zahlung über einen HTTP-Request an das Anbieter-System. Die URL, die SEPA.net dabei aufruft, kann der Anbieter selbst bestimmen (siehe Parameter „eventUrl“ unter 4.3.1). Dabei sieht der Anbieter in der URL einen Platzhalter vor (\*status\*), der von SEPA.net beim Aufruf zur Übermittlung der Statusänderung ersetzt wird.

Beispiel einer Anbieter-URL für SEPA.net-Events:

```
http://www.anbieter.de/minipay_event?process_id=12345&status_new=*status*
```

Sobald eine Status-Änderung bei dem betreffenden Zahlungsvorgang auftritt, ruft SEPA.net obige URL auf. Dabei wird der Platzhalter \*status\* durch folgende möglichen Werte ersetzt:

- „problem“ wenn es bei der Zahlung Probleme gibt (z.B. Widerspruch)
- „ok“ wenn die Probleme gelöst wurden (z.B. durch Überweisung)

Es wird empfohlen, in der URL einen Schutz gegen Missbrauch vorzusehen, etwa durch ein als GET-Parameter übergebenes Passwort.

## 7. Nachträglich Status von Bezahlvorgängen abfragen

Über eine Status-Schnittstelle kann der Anbieter jederzeit den Status eines zurückliegenden Bezahlvorgangs überprüfen. Auf diese Weise kann z.B. vom Anbieter in Echtzeit abgefragt werden, ob eine Lastschriftzahlung z.B. mangels Kontodeckung oder wegen Widerspruchs „geplatzt“ ist.

### 7.1. Aufruf der Status-Schnittstelle

Die Status-Schnittstelle wird folgendermaßen aufgerufen:

```
https://payment.sepa.net/capp/gateways/status?oid=1000&sec=ab56b4d92b40713acc5af89985d4b786&process_id=123456
```

Wie im Beispiel ersichtlich, werden beim Aufruf folgende Parameter per GET-Methode übergeben:

- **oid**: SEPA.net-Kundennummer des Anbieters
- **sec**: md5-Hashwert des Security-Strings des Anbieters
- **process\_id**: Die bei Bezahlvorgang des Endkunden über die Schnittstelle übergebene Process-ID des Anbieters. Bitte achten Sie darauf, dass jede Process-ID nur einmal vergeben wird, da sonst bei einer späteren Status-Abfrage keine eindeutige Zuordnung erfolgen kann.

### 7.2. Antwort der Status-Schnittstelle

Als Antwort liefert die Status-Schnittstelle ein XML-Dokument zurück:

```
<?xml version='1.0' encoding='UTF-8'>
<response>
  <process_id> 123456 </process_id>
  <result> ok </result>
</response>
```

Erläuterung der möglichen Rückgabewerte des Tags „result“:

- “ok” wenn mit Bezahlung alles in Ordnung
- “problem” wenn es bei Bezahlung ein Problem gibt
- “nomatch” wenn pid nicht gefunden wurde
- “double” wenn die übergebene pid bei SEPA.net mehrmals existiert
- “error” wenn Schnittstelle fehlerhaft aufgerufen wurde

## 8. Support

SEPA.net stellt im Downloadbereich der Hauptseite [www.sepa.net](http://www.sepa.net) einzelne beispielhafte Funktionen zur Einbindung von SEPA.net in verschiedenen Systemumgebungen bereit.

Bei Fragen wenden Sie sich bitte an: [support@sepa.net](mailto:support@sepa.net)

## 9. Anhang: Reseller-Programm

### Generierung fertiger Kauf-Buttons als Reseller

Hersteller von HTML-Editoren, Shop-Systemen oder CMS können die in der Einleitung genannten „fertigen“ Kauf-Buttons ganz einfach selbst generieren und Ihren Kunden als Reseller zur Verfügung stellen. Dabei wird von SEPA.net eine Umsatz-Provision an den Reseller gezahlt, wodurch der Reseller an jedem Verkaufs-Prozess der vermittelten Kunden partizipiert. Hierzu wird in den vom Reseller generierten Kauf-Buttons eine Reseller-ID codiert, womit die über den Button getätigten Bezahlprozesse dem Reseller zugeordnet werden können.

Hier der HTML-Code eines fertigen Kauf-Buttons:

```
<a href="https://www.sepa.net/sicher-bezahlen/"
onclick="document.minipay_a=this;document.minipay_reference='XXX';document.
minipay_reseller='XXX';if(typeof(minipay_load)=='undefined'){minipay_js=doc
ument.createElement('script');minipay_js.type='text/javascript'
;minipay_js.src= https://payment.sepa.net/capp/javascripts/minipay';
document.getElementsByTagName('head')[0].appendChild(minipay_js);}else{
minipay_open_reference(document.minipay_a,document.minipay_reference,docume
nt.minipay_reseller);}return false;"></a>
```

Folgende Werte müssen vom Reseller angepasst werden:

- **document.minipay\_reseller:**  
ID des Resellers. Wird bei Abschluss des Reseller-Vertrags vergeben.
- **document.minipay\_reference:**  
Vom Reseller generierte unique ID des jeweiligen Kauf-Buttons. Es wird empfohlen, die ID aus folgenden Werten zusammensetzen: Reseller-ID, Timestamp, Zufallszahl

**Beispiel:** 15\_1274259388\_345