Interface Description
Button and iFrame Integration

Version 3.5
20.08.2019

www.sepa.net

Author:
SEPA.net, a service of the
GRÜN Digital Invest GmbH

Pascalstaße 6
52076 Aachen
Germany

Table of Contents

1. Introduction

This document describes the technical integration of the SEPA.net payment solution. It enables buyers to initiate a fast and secure purchase from a website. A valid online vendor account is required in order to be able to use the interface. An account can be applied for on www.sepa.net, free of charge.

SEPA.net is suited especially for the sales of digital content and fee-based services. For simplicity reasons, this document will refer subsequently to digital contents and services to as "products".

SEPA.net is suited equally for all sizes of websites, be it a basic static web site or a big portal with wide-spread dynamic content. For rather little online vendors, SEPA.net provides a very convenient approach to integrate the solution by offering a ready purchase button for the sale of files.

With the first click, the button can be configured (determining which file should be sold and for which price). Through the following link, you can download a ready purchase button:

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

Developers of HTML editors, shop solutions or CMS can easily generate the ready SEPA.net purchase button themselves and can offer it for their customers (see appendix).

Should SEPA.net be used for a greater or dynamic number of products, it is recommendable to integrate it with a self-created purchase button or iFrames. This method of integration is described in the present document.

2. Fundamental operating principle

2.1. Method of Integration: Purchase Button or iFrame

The buyer interface can be integrated into the online vendors' website using two methods:

- Purchase button: The buyer interface from SEPA.net opens when clicking the "SEPA.net" purchase button. For this, the website solely needs to integrate the SEPA.net purchase button. When clicking the button, a new division or layer will be added to the website automatically, which will be placed over the content of the website. Through the minimal size of the SEPA.net buyer interface (approx. 290 x 272 pixels), the original page stays almost completely visible. Nevertheless, all necessary information for SEPA.net will be displayed for all buyers.

- iFrame: The buyer interface from SEPA.net is opened within the website of the online vendor from the beginning. For this, the buyer interface will be permanently integrated into the online vendor's site within a iFrame with the measures of 370 x 260 pixel.

- 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. Individual adjustment of the payment dialogue

The texts within the SEPA.net window are phrased generic in order to be suitable to numerous products. Upon request, these can be customized in order to suit a specific product (see parameter "template" in chapter 4.3).

2.3. What happens after the payment process?

If a buyer has completed a payment process through SEPA.net, an email will be sent subsequently. This email contains a link, which allows the buyer to retrieve the paid product (e.g. a download of a content, an activation of service or subscription, etc.).

2.3.1. Verification of the authorization

The link refers to special download area of SEPA.net, which examines the authorization of the buyer to retrieve the product. The system will examine the following factors:

- Did the buyer really purchase through SEPA.net?
- Is the request to retrieve the product in the valid time period (can be customized by the online vendor)?
- Did the buyer exceed the maximum permitted number of downloads (can be customized by the online vendor)?

2.3.2. Delivery of the product

When the authorization is successfully verified, a delivery URL will be provided, which the online vendor specifies and through which the purchase button communicates to SEPA.net. the returned content will be sent to the browser of the online vendor.

The request of the delivery URL can occur in two different ways. Which of the two methods is used can be defined through a parameter within the purchase button:

- Proxy: With this method, not the online vendor will initiate the request, but rather a specific proxy server from SEPA.net and provides the returned content one-to-one to the online vendor. Thus, the buyer does not receive a distinct access to the delivery URL, but requests the content from a proxy server of SEPA.net, which is equipped with additional security measure. Through this approach it is secured that a buyer is not able to retrieve the delivery URL and the product multiple times (e.g. the buyer might give the link to friends or might post it in a forum).

- Forwarding: With this method, the browser of the buyer will be redirected to the delivery or activation URL of the provider, after the authentication (siehe 2.3.1) is verified. The delivery URL is visible in the browser address bar. The online vendors have to ensure themselves that the delivery URL will not be used multiple times or by third parties. SEPA.net supports the online vendor by attaching an additional security parameter to the delivery URL, which enables the online vendor to verify the authentication autonomously („PermissionCheck", see chapter 5).

2.4. Demo or production mode

In order to test the operating principle of SEPA.net unproblematically during the integration, it can be operated in demo mode. Within the demo mode no real payments will be triggered. The demo mode can be activated through a specific parameter (see chapter 4.3). If the demo mode is active, there will be a blinking notice on the down left side and the fields do not have to be filled out. SEPA.net will always start in production mode.

2.5. Defining prefilled fields in the payment dialogue

The fields in the SEPA.net payment dialogue can be prefilled from the online vendor, if the data has been already retrieved through a different source. The specific data will be transferred as "data parameter" (see e.g. the parameter "customer_email " in chapter 4.3).

3. Integration of SEPA.net into the website

3.1. Integration through the purchase button

In order to implement SEPA.net through the purchase button on a website, the following changes have to be conducted previously:

- Referencing of a JavaScript file (provides essential SEPA.net functions)
- Purchase button within a website (triggers the SEPA.net payment dialogue)

3.1.1. Referencing of a JavaScript file:

If the SEPA.net purchase button should be used on a website, there needs to be a implemented JavaScript file in a specific position (head section). For this, the following code snippet needs to be used (no line break; it will be created due to space issues).

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

In the above displayed code snippet, the JavaScript file will be implemented through the "https" protocol. Please ensure for the best possible browser compatibility that the specific protocol of the site will be used on which the JavaScript file will be implemented (possibly "http").

3.1.2. The structure of the purchase button

In all cases, where a SEPA.net payment process shall be started within a website, there needs to be a SEPA.net button implemented. Through the purchase all data will be transferred encoded, which are necessary for the sales process (e.g. name or price of the article). The purchase button consists of a linked graphic. In the link, there is a JavaScript request integrated with a permanently defined structure. A purchase button may look like the following (no line breaks; they will be created due to space issues):

Example

```
<a href="https://www.sepa.net/sicher-bezahlen/"
onclick="minipay_open(this,'1000','S0D24wAFeNdLdLzRpPlVPjpacyg0gaXVLaXo00wX
32dqoeLrgErt+H42GqVkhGGipWuqKtn3OFBJX6eqpu8cI7xXzSQwKUgej6viOtPTQJVZTbwlsHM
I9xmdwq4yFzhfqhPDqAzhKcOsYX2ZX/PWXG25YprueSMZvlag9zqodAE5+VUuNqSPGWQDZq7C9+
hi3+IU8JTrVtPJF51d8KtIepOPmhUcR37I3CyKOiQW4s3aK86vKUmfKMhwojxWNMWMRtPANnxFL
nagbzJjBJAApl2/cOhpCxBpoB15dJ7M3jVr3Zhk5p50ePPEz1TcnDWYVc3rVXz7xdQwbI+NWzjT
8/cmV3Iz8tuztaFa8FrTtKA+kWJ70wW+dRvu3BKpApF9aCLDQ/PaVTJTDkMqfvsXTE5v1s0yB8P
VLmy3ZVSj1aaT6J3eRoesrWqav/2UIP4UgPAityPnPfyrAGQ/CWBrkyYd1PCb5uLqQM4GpK6v08
s=');return false;"><img src="IhrButtonPfad" border="0"></a>
```

Each purchase button is valid for one product since all necessary files for the product are coded in it. With one purchase button, any number of payment processes with any number of buyers can be executed.

A purchase button is constructed like the following:

```
<a href="https://www.sepa.net/sicher-bezahlen/" "
onclick="minipay_open( Anker, 'Kundennummer', 'Data' );return false;">
<img src="IhrButtonPfad" border="0"></a>
```

SEPA.net will be opened through a JavaScript function request, which will be executed with the event "onclick". The "href", which is contained in the link, will only be requested if the user has deactivated the JavaScript.

Like it is explained in the box above, the function request "minipay_open" has three parameters: "anchor", "customer ID" and "data". These will be explained further in chapter 4 "Interface parameter".

3.2. Integration through iFrame

If SEPA.net should be implemented through "iFrame" in the site of the online vendor, there needs to be a "iFrame" with the dimension of 370x220 pixels on the desired position. Please deactivate scrolling in the iFrame.

Example:

```
<iframe src="https://payment.sepa.net/capp/payments/start?
oid=1000&layout=extended&data=S5kxhQACbMR%2FbgqxYEEOfOZfvHZWs5d5kwL9r%2FlXj
U7jv8ninIIXOv76h%2B7MzokNATItzCdkKkkOl0TE7WPKCyybE8EPMPiPM8bnQvi7IuNqa5To4n
6i1bYdXlppahxYCJQQre0FKX4yGxEsP9ippbqE6asTh7ZCkxZx%2BXl70qEA5%2Fsf61zs5XVdv
E0Wzz1ePWQK6k8%2FIDHeMlOvDReb5cplso4LD4%2FfkJTPNt%2BmpwoYA4XzbpW6g5rtFPeWtj
fw9HMxUPx1wKfIE3JdlSwZ61IDx20XbN5kXcMRTyIUg%2FAs0Rdo%2F8BGl9wOmM%2BeJcsZB%2
BMxD5YqCyqxXBAxXmIWx7ygsSQz%2Be7%2Bz2m8uc25x0x5kzYyluSq0Mda4zSwIx0SDCt3mT7v
ia%2BGAR%2BVXgSPqarO7vEdO5mkOivWXFVxnftn0de8DFuNNUg"
style="width:370px;height:220px" scrolling="no">
</iframe>
```

Like shown in the example above, the iFrame of the URL refers to: https://payment.sepa.net//capp/payments/start/.

At the same time, the following GET parameter will be transferred, from which "oid" and "data" will be explained further in chapter 4 "Interface parameter":

- oid (Your SEPA.net customer ID)
- layout (iFrame integration needs always the following string: „extended")
- data (contains all necessary information for the payment process)

1. Please be aware of that the purchase button needs to be removed when the product is not available and cannot be sold.

4. The interface parameters

4.1. „anker":

When using the purchase button integration method, this parameter determines through which HTML element of the website the SEPA.net window should be opened. In the most cases, SEPA.net is expected to be opened above the link, which the buyer has just clicked on. In this case, one should use "this" instead of the parameter of "Anker" (without using the quotation marks). Alternatively, one can use a different element of the HTML website, e.g.: document.getElementById('test')

4.2. „oid":

The customer id will be sent via email after the registration as a SEPA.net online vendor. It is valid for all payment processes, which are linked to a legal entity.

4.3. „data":

The data parameter is an encoded string, in which all data of a respective product is contained. The encryption of the data parameter takes place with the encryption algorithm "xtea" (Extended Tiny Encryption Algorithm), which is very secure and is easily integrable into many system environments since it is popular and widespread. As encryption code, one should use the "security string", which was provided during registration together with the customer ID.

The parameter data may look like the following if it is not encrypted:

```
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
```

Through the example, it becomes visible that the parameter "Data" consists of a range of "sub-parameter", which are noted according the GET parameter transfer:

- Nameofthesubparameter = Valueofthesubparameter
- The separation of each sub-parameter occurs through a „&"
- The value of each sub-parameter is URL encoded

A list of all sub-parameters can be found in the following table below. Finally, the **whole** string will be encrypted using the encryption algorithm "xtea" (Extended Tiny Encryption Algorithm). As encryption key, one should make use of the "security string".

4.3.1. Listing of all sub-parameter from „data":

Parameter Title	Obligatory	Format	Description
sec	yes	String (32)	Security string (provided from SEPA.net)
shopDomain	yes	String (50)	Domain through which the product will be sold. Appears subsequently e.g. on the bill of the buyer.
offerType	yes	'sale' || 'service'	When selling digital content: „sale"; When selling services: „service"
productName	yes	String (100)	How ist he product titled towards customer? Appears in the payment dialogue and on the bill of the buyer.
productId	no	String (50)	Optional identifier of the product (appears on the monthly online vendor billing)
price	yes	Decimal (5,2)	Gross price of the product in Euro. Must be within the permitted price range.
tax	yes	Decimal (5,2)	Included VAT rate (%) in „price"
url	yes	String (1000)	Which delivery URL should be requested by SEPA.net or the buyer after the payment process? The purchased product is available at offerType="sale". Valid URL or IP.
useProxy	no	0 || 1	Should SEPA.net request for protection purposes of the sold product, the above mentioned URL via proxy and thus deliver the content to the buyer in that manner? (1=Yes)?
username	no	String (50)	If useProxy=1 and the URL is protected by a password (HTACCESS).
Password	no	String (50)	Password, which belongs to the „username"
fileType	no	String (4)	File type of the content (e.g. "pdf")
fileName	no	String (100)	Optional file name of the download via Proxy
testmode	no	0 || 1	Should SEPA.net be started in test mode in order to test it (1=Yes)? If yes, no real payments will be triggered and the fields in the dialogue do not have to be filled out.
maxDownloadNumbers	no	Integer	How often can a buyer initiate a download after the payment process. Minimum=1, Maximum=10, Default=10

maxDownloadDuration	no	Integer	How long can a buyer initiate the download after the payment process (in minutes). Minimum=1, Maximum=43200, Default=43200
template	no	String (20)	Name of the desired template for the fine adjustment of the payment dialogue
processId	no	String (100)	The generated ID from your system, which enables requesting the status through XML interface at a later stage.
eventUrl	no	String (900)	URL, which should be retrieved when changing the status of a payment.
customer_email	no	String (50)	Email address of the buyer (for prefilling)
customer_phone	no	String (50)	Mobile phone number of the buyer (see above)
customer_name	no	String (50)	First and last name of the buyer (see above)
customer_account	no	String (50)	Account number of the buyer (see above)
customer_bankcode	no	String (50)	Bank sort code (BIC) of the buyer (see above)
reseller	no	Integer (4)	Related reseller ID
billing	no	0 || 1	Should SEPA.net send a bill via email on behalf of the online vendor to the buyer (Default=1)
url_afterDialog	no	String (255)	Optional URL in order to trigger, whether the payment dialogue of the buyer was running through completely. Will be triggered unnoticedly from the buyer upon client side, on the completion page of the payment dialogue (before the buyer will make use of the link, which has been provided via email).

A library for PHP or ASP can be downloaded from our download area for the coding of the data parameter in XTEA. From there one may access also examples for the iFrame integration.

5. Checking if the buyer has paid when forwarding

Should the buyer retrieve the purchased product directly from the website of the online vendor (thus via "redirect" and without using the SEPA.net proxy server), the online vendor needs to ensure autonomously that only entitled buyers, who have initiated a purchase, can retrieve the product. SEPA.net supports online vendors through an optional parameter, which can be integrated into the delivery URL from SEPA.net (PermissionCheck). Through the evaluation of this parameter, an online vendor can verify autonomously if the user has the necessary authentification.

5.1. Activating the PermissionCheck

SEPA.net inserts the parameter "PermissionCheck" optionally in the defined delivery URL from the online vendor. If he buyer is directed to delivery URL after the payment process, the online vendor is able to readout and to check the parameter.

The online vendor can define autonomously on which position within the delivery URL the parameter "PermissionCheck" should be included subsequently from SEPA.net. For this the online vendor needs to stipulate in the delivery URL the string "PermissionCheck" as a place holder on the desired position. It will be replaced by SEPA.net with the value of the PermissionCheck.

Examples of URL with the place holder:

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

Before transferring the buyer during requesting the content, the string "PermissionCheck" will be replaced from SEPA.net with the actual value.

5.2. Evaluation of the PermissionCheck

The can check autonomously if the buyer has the necessary authentification in order to retrieve the requested content. For this purpose, following encoded values (sub-parameter) will be handed over in the PermissionCheck through SEPA.net:

- ip: current IP of the buyer (when clicking the delivery URL in the email)
- url: delivery URL, with a not replaced PermissionCheck
- productId: ID of the product, if handed over as data sub-parameter
- time: Unix timestamp of the creation of the PermissionCheck
- sec: md5 hash value of the security strings of the online vendor

Example of a decrypted PermissionCheck:

```
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
```

As the example shows, each sub-parameter is noted in URLs according the GET parameter handover (just like the structure of the "data" parameter, see chapter 4.3).

- Nameofthesubparameter = Valueofthesubparameter
- The separation of each sub-parameter occurs through a „&"
- The value of each sub-parameter is URL encoded

The PermissionCheck will be handed over encrypted. The encryption occurs also here with the encryption algorithm "xtea" (Extended Tiny Encryption Algorithm). As encryption key one should make use of the "security string", which is provided during the registration process together with the customer ID from SEPA.net.

6. Receiving notification when experiencing payment issues

Through an event interface, it is possible to communicate to an online vendor, if issues regarding payments arise (e.g. the buyer objects a direct debit) or if a payment issue is solved (e.g. when a buyer has paid his debt). An online vendor is able to block a buyer until the outstanding amount is paid.

SEPA.net communicates these changes in the status of a payment through a HTTP request to the online vendor's system. The URL, which is requested by SEPA.net, can be defined by the online vendor autonomously (see parameter "eventURL" chapter 4.3.1). At the same time, the online vendor will see a place holder before "status", which will be replaced by SEPA.net during requesting the transfer of the status change.

Example of an online vendor URL for SEPA.net events:

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

As soon as a status change appears at a respective payment process, SEPA.net will request the above-mentioned URL. For this the place holder "status" will be replaced by these possible values:

- "problem" when there is an issue with the payment (e.g. disagreement)
- "ok" when the issue is solved (e.g. through a bank transfer)

It is recommended to include protection against misuse in the URL, for example through a GET parameter transferred password.

7. Request status of a payment process subsequently

Through a status interface an online vendor is able to check the status of a previous payment process. With this method, an online vendor is able to request in real-time if a direct debit payment is not successful e.g. due to an insufficient account balance or due to a customer objecting the direct debit.

7.1. Requesting the status interface

The status interface can be requested like following:

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

As visible in the example, the following parameters are handed over via the GET method:

- **oid**: SEPA.net customer ID of the online vendor
- **sec**: md5 hash value of the security string of the online vendor
- **process_id**: The process ID of the online vendor, which is handed over during the payment process of the buyer through the interface.

7.2. Response of the status interface

As a response, the status interface returns a XML document:

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

Explanation of the possible return value from the "result" tag:

- "ok", when the payment occurs trouble-free
- "problem", when there is a problem experienced with the payment
- "nomatch", when the pid could not be found
- "double", when the transferred pid already exists multiple times
- "error", when the interface was requested incorrectly

8. Support

SEPA.net provides in the download area of the website www.sepa.net individual and exemplary functions to integrate SEPA.net in various system environment.

If you have any further question, please do not hesitate to contact: support@sepa.net

9. Appendix: Reseller Program

Generating a ready purchase button as a reseller

Makers of HTML editors, shop systems or CMS are able to easily generate a ready purchase button and can provide it to resellers. During this approach, a sales commission will be paid to the reseller, whereby a reseller benefits from each payment process of the acquired buyers. For this, a reseller ID will be coded into the generated purchase button from the reseller, whereby all sales that are initiated through this button can be traced back to the reseller.

The HTML code of a ready purchase button:

```
<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;"><img border=0
src="http://localhost/minipay/capp/img/buttons/bgjk.png"></a>
```

Following values need to be adjusted:

- **document.minipay_reseller:**
  ID of the resellers. Will be provided when closing a reseller contract.
- **document.minipay_reference:**
  From the reseller generated unique ID of the respective purchase button. It is recommended to assemble the ID from the following values:
  Reseller ID, timestamp, random number

  Example: 15_1274259388_345