**Integration checklist for classic paysafecard payments**

Once you have received your Test Data package from paysafecard and have, completed your integration tests, please return this document to the **Partner Integration and Support Team** at [integration@paysafecard.com](mailto:integration@paysafecard.com) so that your integration can be verified.

This document verifies that you have successfully integrated and tested the paysafecard API and that full functionality has been confirmed. After we have verified your integration we will be able to provide you LIVE credentials (Productive Data).

**Important checkpoints:**

* Ensure your payment server uses static IP addresses.
* Ensure that your IP addresses are whitelisted with paysafecard (your staging environment). Please send an email with your MID and payment server IP’s to [integration@paysafecard.com](mailto:integration@paysafecard.com)
* Ensure that the MTID (merchant transaction ID) is always unique for each transaction
* Ensure that the merchantclientid is provided uniquely per customer ID
* If you use e-mail address or username, please encrypt the values.
* Ensure that you send executeDebit API request after successful receipt of the payment notification (pnURL)
* Ensure that you credit the customer account only after a successful executeDebit API response [errorcode = 0 and resultcode = 0]
* Ensure that you send the nokURL, okURL and pnURL in a URL encoded form (percent-encoding)
* Ensure that the paysafecard landing page fits in your iFrame or pop-up window:

width: 600px

height: 750px

* Log all API xml requests in your database, even if they fail
* Our recommendation is to use paysafecard **sample codes**; You can download these from the paysafecard business website ([visit here](https://www.paysafecard.com/en-global/business/downloads/))

* Only use authorized paysafecard brand logo’s; You can download these from paysafecard business website ([visit here](https://www.paysafecard.com/en-global/business/downloads/))
* Use our **Merchant Reporting Tool** to monitor your transactions in real-time
* Never provide unencrypted Productive Data by email.

Please carefully read the SOAP API integration manual.

1. **Acceptance test details**

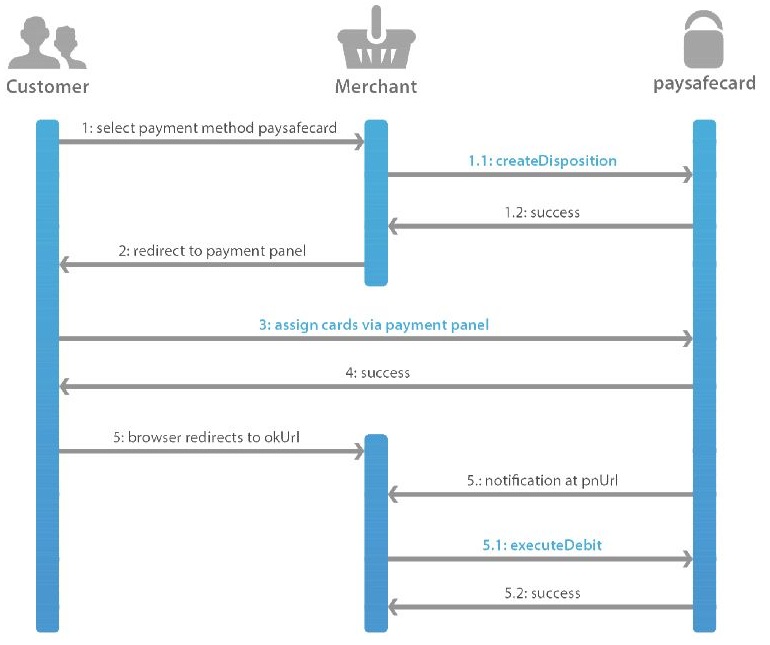
|  |  |
| --- | --- |
| Test started on | *[Please select date]* |
| Test executed by | ***[****Please enter your full name****]*** |
| Test finished on | ***[****Please select date****]*** |

|  |  |  |
| --- | --- | --- |
| Business partner name | *[Please provide with company name]* | |
| MID(s) | If MIDs are sequential (eg ***[100000xxx1]* - *[100000xxx5]***) enter here: | |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | *[]* | | *[]* | | *[]* | | *[]* | | *[]* | | *[]* | | |  | | --- | | *[]* | | *[]* | | *[]* | | *[]* | | *[]* | | *[]* | | | |
| IP address (your payment server) | **TEST SYSTEM** (if existent) | **PRODUCTIVE SYSTEM** |
| |  | | --- | | *[]* | | *[]* | | *[]* | | *[]* | | *[]* | | *[]* | | |  | | --- | | *[]* | | *[]* | | *[]* | | *[]* | | *[]* | | *[]* | |
| List all URLs where paysafecard is used | ***[****Provide incl login details for end customer account(s), to make tests with URLs here****]*** | |
| Technical contact  e-mail | ***[****Please provide with contact details****]*** | |
| Notification e-mail address: | ***[****For notification regarding maintenance work, temporary outages,…****]*** | |
| Phone (in case of emergency) | ***[****Please provide with contact details****]*** | |
| instant messenger contact | ***[****Please provide with contact details****]*** | |

1. **Payment flow verification**

This is the default payment flow according to paysafecard policy.

Please verify that the payment flow is executed as follows:



1. Customer selects paysafecard payment method at your website
   1. Your payment server calls CreateDisposition request
      * If the response is NOT successful show error message to the customer.

[errorcode and resultcode = not equal 0]

* 1. Successful response from paysafecard to your payment server:

[errorcode = 0 and resultcode= 0]

1. Redirection from website to paysafecard landing page
2. Customer enters paysafecard PIN and clicks on pay button at paysafecard landing page
3. paysafecard confirms validation of the PIN
4. pnURL 🡪 paysafecard sends a payment notification to pnURL

5.1) Upon receival of payment notification (pnURL) you call executeDebit to debit the transaction. ONLY upon successful response [errorcode = 0 and resultcode= 0] you should credit customers account

1. okURL 🡪 paysafecard redirects the customer to okURL

5.2.) check with GetSerialNumbers on okURL the state of the transaction

a.) if dispositionState = “S” 🡪 call executeDebit

Only upon successful response [errorcode = 0 and resultcode= 0] from executeDebit credit customers account

b.) if dispositionState = “O” 🡪 only show success message to the customer, do not credit customers account again.

|  |
| --- |
| Factbox: Technical Information |
| The following values are mandatory must be provided in every transaction. Please find below a short description and a sample CreateDisposition xml request:  GREEN: use paysafecard provided values  RED: configure your values  Description:  Username: use your soap username provided in the test data package (\*.txt)  Password: use your soap password provided in the test data package (\*.txt)  MTID: use unique ‘merchant transaction ID’ (MTID)  Amount: use any amount (min. 0.01, max. 1000 EUR or equivalent in a different transaction currency)  Currency: use your configured MID. MID’s are provided in the test data package (\*.txt)  okUrl: use URL where customer should be redirected to after a successful PIN assignment on paysafecard payment panel  nokUrl: use URL where customer should be redirected to after cancelled transaction on paysafecard payment panel  merchantclientid: use an unique customer identifier  pnUrl: use URL which paysafecard will notify you on after a successful PIN assignment.  Sample xml request:  <urn:createDisposition>  <urn:username>psc\_XXXX\_test</urn:username>  <urn:password>Passw0rd</urn:password>  <urn:mtid>1234-56789-10A3B2C3D456-ABDCEF</urn:mtid>  <urn:amount>10.00</urn:amount>  <urn:currency>EUR</urn:currency>  <urn:okUrl>http://www.yourwebsite.com</urn:okUrl>  <urn:nokUrl>http://www.yourwebsite.com</urn:nokUrl>  <urn:merchantclientid>cID\_919191</urn:merchantclientid>  <urn:pnUrl> http://www.merchantpnURL.com </urn:pnUrl>  </urn:createDisposition> |

1. **Payment tests**

To pass the acceptance test and move forward to Productive Environment, it is mandatory to execute payment tests.

**Please execute the tests below and return document to our integration team after successful approval.**

|  |  |  |  |
| --- | --- | --- | --- |
| Test 1a: successful CreateDisposition | | APPROVED |  |
| 1. | Create a transaction and execute the CreateDisposition request. | | |
| **2.** | CreateDisposition **response should contains:**  **resultcode=0**  **errorcode=0** | | |
| **3.** | **Verify from your Database transaction table that transaction is successfully created and registered as CREATED (transactionStatus=R)** | | |
| **4.** | **Enter your used MTID:**  ***[****Please enter successfully created MTID****]*** | | |

* Please note, a transaction stays in status CREATED for 30 minutes.

|  |  |  |  |
| --- | --- | --- | --- |
| Test 1b: failed CreateDisposition | | APPROVED |  |
| 1. | Create a transaction and execute the CreateDisposition request with an already existing MTID. | |  |
| **2.** | CreateDisposition **response should contains:**  **resultcode=1**  **errorcode=2001** | | |
| **3.** | **Verify from your Database transaction table that transaction is already existing and returned error:**  **Transaction (%1/%2) already exists. Please contact your webshop.** | | |
| **4.** | **Display this or similar error message to customer:**  **Transaction could not be initiated due to connection problems.**  **If the problem persists, please contact our support.** | | |
| **5.** | ***[****Please enter created MTID from transaction****]*** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test 2: ****GetCustomerPanel**** | | APPROVED |  |
| 1. | After successful CreateDisposition use getCustomerPanel to be forward to paysafecard landing page. | | |
| **2.** | **Ensure that only GET is used for redirect to paysafecard landing page.** | | |
| **3.** | **Ensure that the paysafecard landing page size fits in your iFrame and enable vertical scrolling**  **width: 600px**  **height: 750px** | | |
| **4.** | **Confirm successful redirect to our payment panel** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test 3: ****Payment notification (pnURL)**** | | APPROVED |  |
| 1. | Ensure your pnURL is reachable from internet  - whitelist paysafecard notification server IP: 194.1.158.23 | | |
| **2.** | **Provide absolute URL for payment notification** | | |
| **3.** | **Verify your HTTP 200 response will be submitted before calling** executeDebit | | |
| **4.** | **Fetch MTID with your database to call** executeDebit | | |
| **5.** | Do not fetch the provided amount from payment notification | | |
| **6.** | Use only standard ports for HTTP (80) and HTTPS (443) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test 4: receive transaction state with GetSerialNumbers | | APPROVED |  |
| 1. | Use GetSerialNumbers before calling executeDebit  if dispositionState="S" call executeDebit to debit the payment | | |
| **2.** | **Verify transaction State on okURL with** GetSerialNumbers  **if dispositionState="S" 🡪 call executeDebit to debit the payment** | | |
| **3.** | **Verify transaction State on okURL with** GetSerialNumbers  **if dispositionState="O" 🡪 display success message** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test 5: executeDebit | | APPROVED |  |
| 1. | If the response from GetSerialNumbers is **"S"** call executeDebit | | |
| **2.** | Call executeDebit after receipt of payment notification (pnURL) | | |
| **3.** | executeDebit is called with the same Amount as provided in createDisposition request | | |
| **4.** | parameter **"***close***"** is set to **"**1**"** | | |
| **5.** | Make sure, executeDebit is called before the end of the disposition time | | |
| **6.** | If response from executeDebitis successful[errorcode=0 and resultcode=0] **🡪** credit customers account | | |

1. **Optional payment tests**

If you are using Reporting Criteria **(=subId)**. Must be agreed with paysafecard in advance. Please be aware that on the test environment only 3 values are enabled.

|  |  |  |
| --- | --- | --- |
| Test 6: Reporting Criteria (subId) | | |
| subId value provided | Not agreed |  |
| Agreed | Please explain use case:  ***[****Please explain here****]*** |

|  |  |  |  |
| --- | --- | --- | --- |
| Test 6a: subId agreed | | APPROVED |  |
| 1. | Ensure that value in parameter **"**subId**"** is provided in CreateDisposition request | | |
| **2.** | **Ensure that max. provided digits in parameter subId is limited to 8 letters** | | |
| **3.** | **On test environment we have pre-defined values, please see below:**  **subId=1**  **subId=2**  **subId=3** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test 6b: subId not agreed | | APPROVED |  |
| 1. | **As the parameter "subId" is mandatory, please ensure that the value must be left empty.** | | |

1. **Confirmation of tests**

Through submitting this agreement we declare that:

* We have successfully completed the integration of the paysafecard payment API into our webshop.
* We have read and understood the latest version of the SOAP API manual.
* We approved the integration checklist and performed all integration tests from the "iCheck" document successfully.

**FAQ**

**Q: Can I use HTTPS as payment notification URL (pnURL)?**

A: In case you would like to use HTTPS for payment notification URLs (we recommend

using HTTP for payment notification URLs), only certificates created by world-known

CAs (certificate authorities) are supported.

**Q: What happens if I do not call** executeDebit **to finalize the payment?**

A: The transactions is not successful and expires (Status = X). The money is released to customer’s paysafecard and customer can deposit again.

**Q: What is the disposition time window?**

A: It defines the maximum amount of time within which a (paid) disposition can be debited and the Amount can be accounted to the merchant's favour.

**Q: How long is the time periods to call** executeDebit **after receive of payment notification?**

A: This is defined in the disposition Time window which is a part of the contract (Merchant Agreement).

**Q: When I should credit customers account?**

A: Only upon successful executeDebit credit a customer’s account to debit the transaction.

Successful response = [errorcode = 0 and resultcode = 0]

**Q: What happens if I call** executeDebit **more than once on a single transaction?**

A: If you try to call executeDebit on a transactions which is in state ‘O’, you will receive the following response [errorcode = 1 and resultcode = 2003].

**Q: How long is it possible for the customer to enter the PIN on paysafecard landing page?**

A: After creating a transaction (CreateDisposition), the customer has 30 minutes to enter the PIN. After 30 min without successfully PIN assignment the transaction state will be changed from “R” to “X”. Transactions with state X are not successful; the customer can start a new deposit again.

**Q: Where I can find the transaction states?**

A: Please find the link for our documentation ([here](https://www.paysafecard.com/fileadmin/content/website_business/Technische_Daten/API_SPEC_SOAP_CLASSIC.pdf)) (Page 20):

**Q: Where can I find all paysafecard API error codes?**

A: Please find the link for our documentation ([here](https://www.paysafecard.com/fileadmin/content/website_business/Technische_Daten/API_SPEC_SOAP_CLASSIC.pdf)) (Page 26):

**Q: I have finished the integration tests, what are the next steps?**

A: 1.) Please return the completed Ready4Payment document to [integration@paysafecard.com](mailto:integration@paysafecard.com)

2.) paysafecard integration team will test and verify your integration and upon successful verification we will request Productive Data to you (Subject to Commercial Agreements being completed).

3.) Once you have received the Productive Data package, a last real-money test is necessary before we can activate your account for all customers. This simply requires one payment test to be made using the Productive Data that we have provided you.