Payolution GmbH
Columbusplatz 7-8
1120 Wien - Österreich
integration-vie@unzer.com

UNZER PAY LATER SOLUTIONS

TECHNICAL INTRODUCTION

Version 2.0

Release History

Release	Description	Date	Changes
1.0.0	Initial Version	29-Apr-2011	-
1.0.1	Minor Changes	19-May-2011	-
1.0.2	Address change	24-Nov-2011	Address change
1.1.0	ELV	01-March-2012	Add payment method ELV
1.2.1	Complete rework	08-November-2012	
1.2.3	Web shop / invoice requirements	15-March-2013	
1.3.0	Plugin / ERP requirements	28-March-2013	
1.3.1	reworked ASYNC description	26-Aug-2013	reworked ASYNC description, added result codes, added claims mgmt. system info
1.3.2	reauthorize and reversal	26-Aug-2013	Added reauthorize and reversal operation
1.3.3 – 1.3.6	Minor changes	03-March-2015	
1.4	Major changes	26-July-2016	New structure, added ELV
1.5	Major changes	19-January-2017	Updated SSL-certificates and IP addresses
1.6	Minor changes	08-February-2017	Updated IP addresses to port
1.7	Minor changes	12-June-2017	Updated URLs for API-Gateway and removed BIP-Credentials
1.8	Major changes	24-October-2017	Addded Fraudprevention to Documentaiton, removed BCC-Email
1.9	Minor changes	08-October-2018	Updated logo, wording
1.10	Minor changes	06-June-2019	Removed chapter SSL certificates
2.0	Minor changes	19-November-2020	Unzer Rebranding 1.0
2.0	Minor changes	20-April-2021	Updated address

Payolution GmbH
Columbusplatz 7-8
1120 Wien - Österreich
integration-vie@unzer.com

1. PREFACE

Independent from the chosen integration option the following documents will provide a pragmatic access to the topic:

- This technical introduction:
  contains all essential information to get going. Requirements regarding Web shop, ERP System and invoice generation are followed by a general process description for the Unzer Pay Later solutions: Test accounts for integration, all URLs to access the various parts of the system, test plans and procedures and finally a description of the going live process.

In connection with the chosen integration option for sending transactions to the CTPE the following documents will add the required level of detail for the respective interface:

- XML Interface contains the complete reference specification of all request parameters and response codes which can be used with the XML integration interface.

- Integration Examples:
  programming language specific installation guidelines and archives which contain a lot of source code and shop integration examples to speed up implementation. Currently the following languages are provided: Java, PHP and .NET

If you encounter any issues after reading the documentation mentioned above, feel free to contact your integration manager or integration-vie@unzer.com.

2. DISCLAIMER

Unzer Pay Later solutions Documentation - Introduction

CONTENT

Unzer Pay Later solutions Technical Introduction

3. OVERVIEW

The integration of the Unzer Pay Later solutions requires an adaption of the checkout process in the web shop. Since the shipment of goods and the payment process are linked, an adaption of the ERP / order fulfillment system is needed, as well. For invoice payment, the invoice sent to the customer requires specific adjustments. Those requirements are listed in the following paragraphs:

3.1. WEB SHOP REQUIREMENTS

- A different shipping / billing address should hide the payment option
- Settings for minimum / maximum basket size
- Limiting on certain countries (AT/ DE/ CH/ NL)
- Exclude gift certificates
- Input fields for the date of birth
- A check on the customer age (> 18 years)
- PreCheck request after payment method selection
- PreAuthorization request after order submission via https POST or XML
- No order confirmations for declined PreAuthorization
- Storage of Transaction Reference & Unique ID together with the order
- Implementation of fraudprevention

3.2. ERP SYSTEM REQUIREMENT

- Changes of orders before shipment trigger ReAuthorize requests
- Full cancellation before shipment trigger Reversal requests
- Shipment of good trigger Capture requests
- Transmission of invoice id
- Transmission of shipment tracking number
- Return announcements by customers trigger Refund Announcement requests
- Returned goods trigger Refund requests
- Multiple capture and refunds are possible
- Exchange of goods with a higher total amount trigger Refund + PreAuthorization requests
- Capture and refund requests can have different amounts
- All requests contain tax amounts

3.3. INVOICE REQUIREMENTS

- Automatically generated by web shop or ERP System
- Bank details (IBAN, BIC, Transaction reference)
- No other bank details, such as the merchant account

Unzer Pay Later solutions Technical Introduction

3.4. PAYMENT PROCESSING SYSTEMS:

Besides web shop and ERP System, following additional systems are involved:

CORE TRANSACTION PROCESSING ENGINE

The core transaction processing engine (CTPE) is the central part of the platform where all transactions are cleared. It is built to process massive concurrent transaction loads in a high-availability environment. The technical web shop and ERP integration connects the shop system with the CTPE via XML responses.

In addition, there is a mapping of XML Tags to HTTP POST parameters.

UNZER PAY LATER SOLUTIONS REST-API

The Unzer Pay Later solutions REST-API system is the core component for risk checks and all customer related activities. The following requests are processed directly here via XML requests or by CSV-batch: Refund Announcement, Add Data, Calculation.

Payolution GmbH
Columbusplatz 7-8
1120 Wien - Österreich
integration-vie@unzer.com

5. PROCESS DESCRIPTION

A transaction with Unzer Pay Later solutions can have following operations:

Operation or _Transaction State_	Example XML	Description	Cashflow relevant
CALCULATION	(CL)	retrieves all information like e.g. possible durations for the transmitted amount (only for installment)	
PRE_CHECK	(PC)	customer risk check after payment method is selected	
PRE_AUTHORIZATION _PreAuth OK_	(PA)	customer risk check and acceptance by Unzer Pay Later solutions	
RE AUTHORIZATION	(PZ)	changes a previously authorized amount	
REVERSAL	(RV)	fully reverts the Pre-Authorization	
CAPTURE _Captured_	(CP)	actual claims acquisition by the merchant, payout and invoicing starts	x
REFUND_ANNOUNCEMENT	(RFA)	delays the Unzer Pay Later solutions dunning process to avoid overlaps with refunds	
REFUND	(RF)	returns by the merchant, fully or partial	x
ADD_DATA	(AD)	Generic request to pass additional IDs or events to Unzer Pay Later solutions	

Each state transition is either initiated automatically by the web shop / ERP system or manually by the merchant.

5.1. REQUEST DESCRIPTIONS:

CALCULATION (CL)

The Calculation request is only needed for installment integration and it gathers all necessary installment information for the transmitted amount: Durations, payment dates, interest fee, total amount, standard credit information, and some more.

The PreAuth must be referenced to this request.

PRECHECK (PC)

This operation is not part of the payment process. Like credit card check, it is used to pre-check customer data immediately after the payment method selection step in the checkout. This way customer receives direct feedback before finishing the order, avoiding irritation.

The PreCheck request contains customer data as well as transactional details. An extensive is provided in the XML interface description (Chapter Analysis)

PREAUTHORIZATION (PA)

The PreAuth request is the initial request of a payment with Unzer Pay Later solutions. It contains payment details, information about the customer and the shopping basket. Processing the request includes a decisive risk check. Within a few seconds a response is returned:

Declined: The amount was not authorized and the customer should be routed back to the payment method selection.
Accepted: The transaction amount is reserved for Capture and the web shop order can be finished.

REAUTHORIZATION (PZ)

The ReAuth request changes a previously pre-authorized amount of a payment with Unzer Pay Later solutions. It contains payment details, but NO customer information. The customer information from the referenced PA request is used for another risk check. Within a few seconds a response is returned:

Declined: The new amount was not accepted. The originally pre-authorized amount is still valid.
Accepted: The previous amount is updated and the new one is reserved for Capture.

CAPTURE (CP)

A CP request finalizes the payment process and fixes the payment details, such as the maximum amount. It is possible to reduce the captured amount compared to the pre-authorized amount. The CP triggers the payout towards the merchant as well as dunning towards the customer.

Declined: The amount could not be captured. Usually the pre-authorization is expired or there was a technical error in the request.
Accepted: The amount will be included in the next payout cycle

The typical timing of the CP request is:
- For installment payments immediately triggered after the PA response arrives. The payment of the goods is decoupled of the actual shipping date. Unzer Pay Later solutions can automatically trigger the Capture after the order acceptance, but this can also be made by the merchants' system.

- For invoice & direct debit transactions defined by the shipping date / date of the invoice. Hence, the CP request should be sent the very day the goods are shipped.

Successful PA requests expire after a certain time frame, per default 21 days. Hence CP requests need to be performed within this period. It is possible to capture the full amount in multiple steps (multiple partial captures). In order to increase the pre-authorized amount a PZ request is required before the capture.

REVERSAL (RV)

A reversal cancels a pre-authorized order and blocks it for future captures. Since the Unzer Pay Later solutions risk engine considers the number of PA per customer, the reversal operation allows earlier re-orders and automated cancellation of orders before shipping of goods. Note that no amounts are considered. If a part of the order is canceled, the remainder should be captured. Reversals are only allowed before CP.

REFUND ANNOUNCEMENTS (RFA)

A refund announcement request should be sent when customers announce a refund. Typical events are customer phone calls or printing return stickers (RMA). They help to prevent unjustified dunning of customers.

REFUNDS (RF)

A refund request must be sent in case of goods returned by the customer. The amount is deducted from the next merchant payout. Refunds can also be used for invoice reduction or settlement of faulty payments into the merchant account.

Declined: the amount could not be refunded. Usually the transaction is too old or there was a technical error in the request.
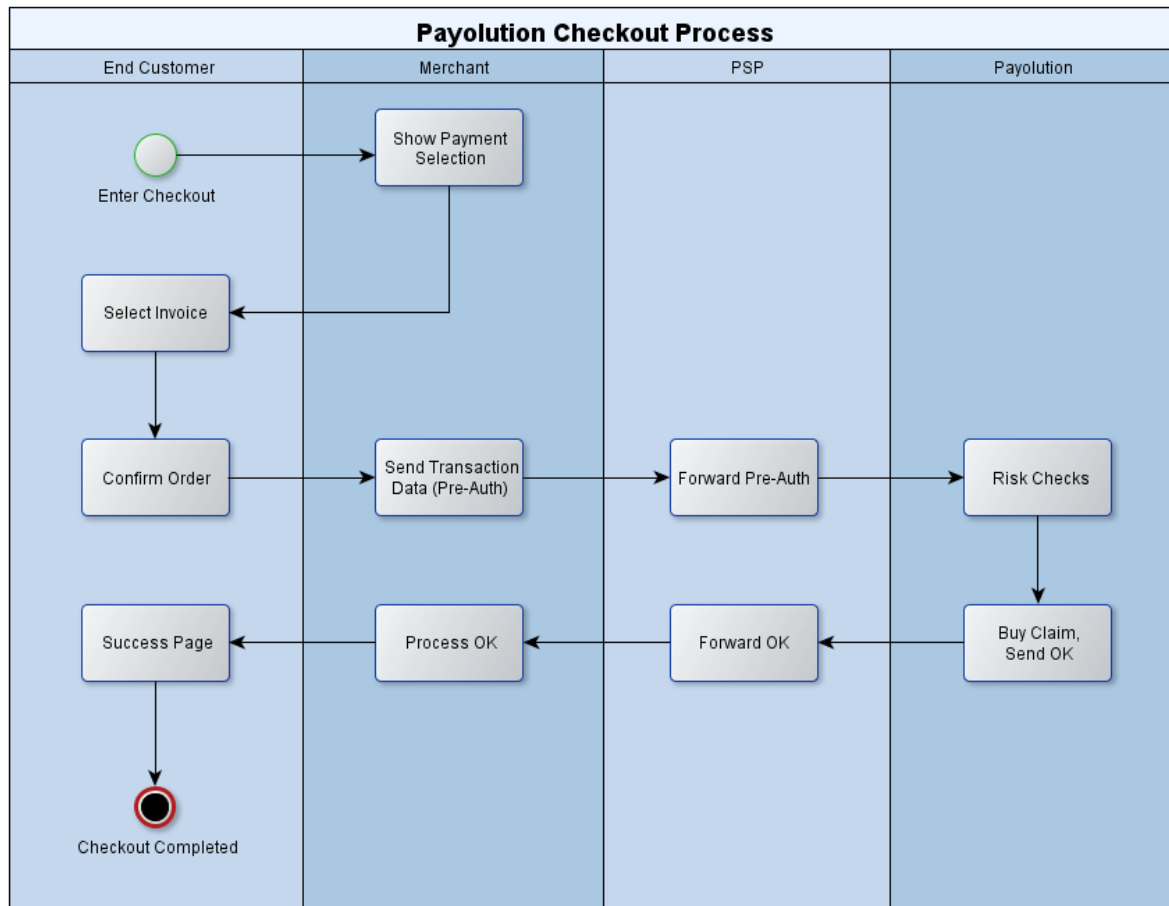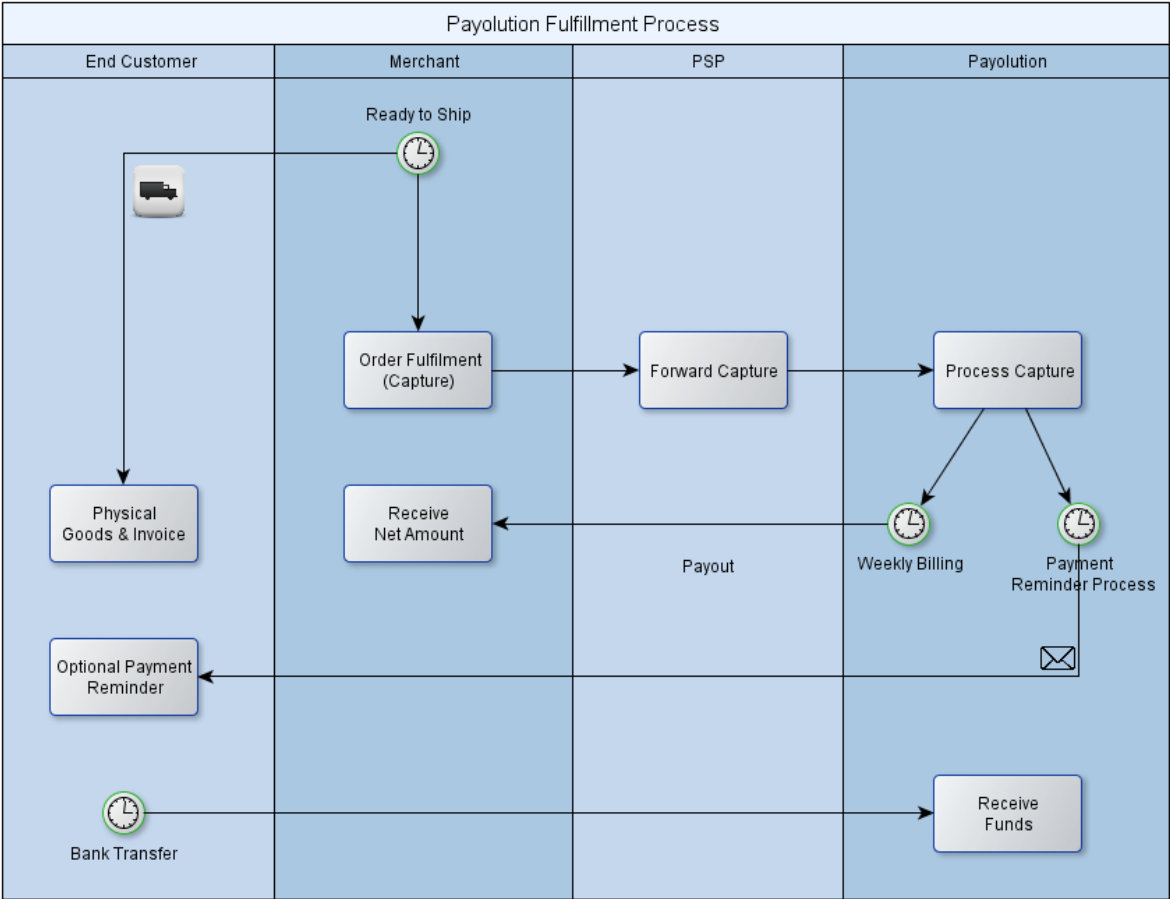**Accepted**: the amount will be deducted in the next payout cycle. If applicable, customers receive a reimbursement.

In case of a goods exchange, only the difference in value should be refunded. For more expensive items, a full refund and a new order is required. This can be done within one transaction by sending a PA and CP immediately after the Refund.

ADD DATA (AD)

An AD request passes data to Unzer Pay Later solutions, that was not available at the time of another request. Typical data are Invoice-, Customer-, Order- or Transportation Tracking IDs. The information helps e.g. to prevent unjustified dunning of customers.

5.2.PROCESS DIAGRAMS

5.3. IMPORTANT FIELDS

AMOUNTS

PA, CP and RF requests each contain an amount field. These amounts can be different. However, the maximum amount for CP is set by the PA/PZ, for RF by CP. i.e. you can't capture more than pre-authorized and you can't refund more than captured.

TRANSACTION REFERENCE

The PA response contains the **Unzer Pay Later solutions Transaction reference**, which is a 12-letter code. (e.g. PAYO-LUTI-ONED).

UNIQUE ID

32 character string identifying the transaction (e.g. 40288b162da3e294012db761fd734134). All subsequent requests must contain this reference to match previous requests.

- The PA request is referencing the PC and – in case of installment – the CL request in the analysis section
- The PZ, CP, RV, RF requests are referencing to the PA request.

6. WEB SHOP INTEGRATIONS

A PA request is sent to the gateway as HTTP POST REQUEST (with XML payload or POST parameters). The HTTP response is the result of the risk check (OK / NOK), including the transaction reference. All subsequent requests (CP, RF) are similar and must refer to this transaction reference.

A file with all XML/POST requests examples is included in the Integration Package. There you have an overview about which parameters must or can be transmitted. In the following overview it is showed how the respective payment method can be designed in the shop frontend.

6.1. INVOICE B2C

Invoice implementation in the webshop should look like this:

The customer clicks "Invoice", in case the birthdate is no mandatory field when registring he must fill out the birthdate as Unzer Pay Later solutions need it to perform the riskcheck. In addition, a checkbox must be implemented when the Online shop is certified by Trusted Shop:

Mit der Übermittlung der für die Abwicklung des Rechnungskaufes und einer Identitäts- und Bonitätsprüfung erforderlichen Daten an Unzer Pay Later solutions bin ich einverstanden. Meine Einwilligung kann ich jederzeit mit Wirkung für die Zukunft widerrufen.

By clicking on „Weiter" the PreCheck should be transmitted to Unzer Pay Later solutions. When the result is OK the customer can proceed. By clicking on "Kaufen" later on the PreAuth is made.

 If the result of the riskcheck (either PreCheck or PreAuth) is NOK the following error message should be displayed and the customer should select a different payment method:

Diese Zahlung konnte nicht durchgeführt werden. Dies kann unterschiedliche Gründe haben, wie etwa fehlerhafte Eingabedaten, eine unbekannte Adresse, oder ein vorübergehendes technisches Problem. Bitte überprüfen Sie die angegebenen Daten, oder wählen Sie ein anderes Zahlungsmittel.

6.2. INVOICE B2B

When implementing Invoice for B2B customers the general workflow is like described in 6.1. but instead of birthdate the company name is mandatory:

To increase the acceptance for B2B customers it's reasonable to ask for the UID as well – but only as an optional parameter.

6.3. DIRECT DEBIT

The implementation of direct debit should follow this example:

Apart of the birthdate the customer also needs to fill in his bank account data and on top accept the SEPA-mandate which should be open in a pop-up window or a new tab:

https://payment.payolution.com/payolution-payment/infoport/sepa/mandate.pdf

Unzer Pay Later solutions Technical Introduction

The following workflows regarding OK/NOK results for PreCheck or PreAuth work exactly like described in 6.1.

6.4. INSTALLMENT

The implementation of the installment payment needs a bit more information.
The best-practice workflow should be like this:

When the customer clicks installment payment the CL request should be sent to Unzer Pay Later solutions. In a first step he must then fill out his birthdate and tick the checkbox – like it must be made for all other Unzer Pay Later payment methods too – and by clicking on "Ratenverfügbarkeit prüfen" the PreCheck is sent to Unzer Pay Later solutions.

Payolution GmbH
Columbusplatz 7-8
1120 Wien - Österreich
integration-vie@unzer.com

If the response is NOK the error message should be displayed, if it's OK the Response of the CL request should be displayed to the customer:

The important facts are: possible durations with the respective rate amount, the due dates of the rates, a link to download an example of the credit contract and a summary of the selected rates, the original amount excl. interests, the interest rate, the effective interest rate and the total amount incl. interests.

The example of the credit contract should open in a new tab/window or as a pop-up.
It's important that the real URL of the contract is hidden, that means a special Customer facing URL like e.g. http://www.example.de/kreditvertragsentwurf/?&duration=3 should be used instead of the target URL: https://www.test-payment.payolution.com/payolution-payment/rest/query/customerinfo/pdf?trxId=Tx-jg36bdwqfup&Duration=3

As Unzer Pay Later solutions is collecting the rates via direct debit in DE and AT customers must give their bank account data. In CH the rates must be transferred by the customer.

As for the other payment methods the PreAuth is made when the customer is clicking the buy button and finalizes the order.

Unzer Pay Later solutions Technical Introduction

6.5. FRAUD PREVENTION

In order to successfully implement the current fraudprevention, the following steps are required. Since it is an asynchronous tool, two steps are needed:

- Adaption of webshop pages:
  Generating a unique Session Id and adding a JavaScript-code-snippet to the webshop pages where Unzer Pay Later solutions requests are sent.

- Adding the generated session Id to the XML request:

6.5.1. ADDING THE JS TO THE WEBPAGE

In order to transmit the required information a JS needs to be included on **all** pages of the shop where a request is sent to the Unzer Pay Later solutions systems. Usually this is the payment selection and the final order page.

It is required that the integration of the tool takes place in the <head> section as well as the <body> section for the NoScript implementation.

<head>

      <script type="text/javascript" src="https://h.online-metrix.net/fp/tags.js?org_id=363t8kgq&session_id=**[SessionID]**"></script>

</head>

<body>

      <noscript>

            <iframe style="width: 100px; height: 100px; border: 0; position: absolute; top: -5000px;" src="https://h.online-metrix.net/fp/tags?org_id=363t8kgq&session_id=**[SessionID]**"></iframe>

      </noscript>

</body>

Please note: The org_id must be **"363t8kgq".**

Asynchronous loading

To reduce dependency from an external source, it is recommended to load the code asynchronous with your preferred technology, e.g. AJAX

6.5.2. GENERATING UNIQUE SESSIONID

The Session Id is temporary identifier that is unique to the visitor's **session** and **per event**. It can be up to 128 bytes long.

Only the following characters are allowed for generating the Id:

- upper and lowercase English letters ([a-z], [A-Z])

- digits (0-9)

- underscore (_)

- hyphen (-).

Important: Please ensure that the generated Id is unique for every order and consistent throughout the checkout.

Recommendation for creating the SessionId:

- Use a merchant identifier (URL without domain additions), append an existing session identifier from a cookie, append the date and time in milliseconds to the end of the identifier and then applying a hexadecimal hash to the concatenated value to produce a completely unique Session ID.

- Use a merchant identifier (URL without domain additions), append an existing session identifier from the web application and applying a hexadecimal hash to the value to obfuscate the identifier.

- **Example:** merchantshop_cd-695a7565-979b-4af9

Important: The generated ID **must** be stored temporary for later use in request.

Unzer Pay Later solutions Technical Introduction

6.5.3. REQUEST ADAPTION

To fetch data in the background it is necessary that Unzer Pay Later solutions receives the previously generated and stored SessionId within requests initiated from the storefront.

The ID must be sent at least at PreAuthorization (PA) and additionally at PreCheck if used. To transmit the SessionId, please adapt the request by adding a new "Criterion" in the "Analysis"-Block of the requests:

Example XML:

Example POST:

CRITERION.PAYOLUTION_SESSION_ID=merchantshop_cd-695a7565-979b-4af9

7. CTPE ACCESS & CONNECTIVITY

7.1. TEST SYSTEM

All tests during the implementation of the integration interface and during the lifecycle of the online shop should be done on the test system. The test system has exactly the same functionality as the live system.

You can find the test URL and credentials in the txt file "Testcredentials" which is included in the Integration Package.

Please note:

- The test system generally allows for more services and features (for example the processing of additional payment methods, credit card brands or direct debit countries) than you probably have ordered for your application. Please contact your account manager if you would like to use one of those services also on the live system.

7.2. LIVE SYSTEM

After having successfully processed transactions in test mode you will be provided with fully personalized access data for the live system by your designated integration manager.

It is also possible to send test transactions to the live system in INTEGRATOR_TEST mode. This is mainly used shortly before going-live to do a few tests in order to crosscheck whether the system configuration and access data is correct. Please note: Every transaction to the live system which is captured will be charged with the contractually agreed transaction fee.

The transaction mode needs to be set to LIVE after successful connectivity testing

Unzer Pay Later solutions Technical Introduction

8. TECHNICAL INFRASTRUCTURE

Below is an overview of technical relevant data and setups.

INTEGRATION OF ACCESS DATA (CREDENTIALS)

Please enter credentials into the XML request for SYNC integration as shown below (prefilled with test credentials for invoice):

```
<?xml version="1.0" encoding="UTF-8"?>
<Request version="1.0">
 <Header>
  <Security sender="8a8294182f965dd4012f9b5c54f50169" />
 </Header>
 <Transaction channel="8a82941832d84c500132e875fc0c0648" mode="CONNECTOR_TEST">
  <User pwd="Eg89ttKk" login="8a8294182f965dd4012f9b5c54f7016d" />
```

REQUEST TRANSMISSION

XML requests are sent as HTTPS POST to

https://test-gateway.payolution.com/ctpe/api

POST parameter name: is "load"

Direct HTTPS POSTs need to be sent to:

https://test-gateway.payolution.com/ctpe/post

(i.e. without XML, parameters)

CONTENT TYPE

application/x-www-form-urlencoded;charset=UTF-8

SERVER-TO-SERVER COMMUNICATION

Outbound traffic through port 443 to all IP addresses must be allowed.

9. TEST PROCEDURE

The recommended test procedure is:

1. Start with submitting transactions in CONNECTOR_TEST mode.

2. Send a simple Pre-Authorization (PA) transaction.
   What is required for a successful PRE_AUTH: customer first & last name, date of birth, address, a valid e-mail address
   Due to common trouble with UTF-8 encoding, include a request with an Umlaut, i.e. Ä, Ö, Ü.

3. *What is required for a denied PRE_AUTH: use* *blacklist-test@example.com*

4. Verify whether the transactions have successfully been processed: Check the Request-Response if it contains the following Statement: `"successfully processed in 'Merchant in Connector Test Mode' (000.100.112)"`

5. If everything worked out correctly submit capture (CP) transactions and all other transaction types which are necessary for your business process. (e.g. Refunds (RF)). In this mode the transactions are processed end-to-end and move through the same steps as in LIVE mode.

6. If the transactions have been successfully processed, contact your integration manager for further steps. If the transactions failed, contact him for support regarding the connector setup.

Unzer Pay Later solutions Technical Introduction

υnzer

Payolution GmbH
Columbusplatz 7-8
1120 Wien - Österreich
integration-vie@unzer.com

COMMON PITFALLS

Problems with Ä, Ö, Ü:

Send a request with an Umlaut, i.e. Ä, Ö, Ü. Check the UTF-8 encoding

Wrong credentials: The request fails immediately. Have you received any response? Was there a mix-up of Channel, Sender and Login? Is there a whitespace (blank,...) in the stored credentials?

After confirming the transaction on the payment page, nothing happens in the shop. (ASYNC only)
Is a POST request from a public IP address allowed to the redirect URL of the shop?

Unzer Pay Later solutions Technical Introduction