

Telekom Multi Payment Service Centre

integration guide of the DirectPayment API

Version 5
Stand 05/11/2020
Status final

public

table of contents

1	overview.....	4
2	service architecture.....	5
2.1	base URLs.....	5
2.2	service authentication	5
2.3	disclaimer.....	5
3	direct payment without couponing	6
4	direct payment with couponing.....	7
5	direct registration	8
6	API documentation	9
6.1	direct payment (without couponing)	9
6.1.1	payment methods.....	9
6.1.2	reserve.....	9
6.1.3	update	10
6.1.4	claim.....	11
6.1.5	cancel.....	12
6.2	direct payment (with couponing).....	12
6.2.1	initialize	13
6.2.2	session details.....	13
6.2.3	couponing	14
6.2.4	reserve (of an initialized transaction)	15
6.3	direct register.....	15
6.3.1	register	15
6.3.2	update	16
6.4	error response	17
6.5	payment method details	17
6.5.1	payment method: creditcard_computop.....	17
6.5.2	payment method: mobilebill_mbe4.....	17
6.5.3	payment method: paypal.....	17
6.5.4	payment method: telephonecard.....	17
6.5.5	payment method: fixedlinebill_croatia	18
6.5.6	payment method: fixedlinebill_telekom.....	18
6.5.7	payment method: mobilebill_telekom.....	18
6.6	other structures	18
6.6.1	couponing constraint.....	18
6.6.2	couponing data.....	18
6.6.3	customer information.....	18
6.6.4	discount details.....	18
6.6.5	ipdr basket item	19
6.6.6	ipdr settlement.....	19
6.6.7	ipdr transaction.....	19
6.6.8	input assistance	19
6.6.9	line item	20
6.6.10	mobile xtc	20
6.6.11	payment service data (payment)	20
6.6.12	payment service data (register)	21
6.6.13	payment service data list	21
6.6.14	payment service item data	21
6.6.15	payment service data info	21
6.6.16	payment service data info detail.....	22

6.6.17	shipping address	22
6.7	states and transitions	22
7	table of abbreviation	24

1 overview

The Telekom Multi Payment Service Centre ("MPS") is offering a set of interfaces a business partner can use to enable payment and billing functionality in its online service (e.g. e-commerce portal). Therefore it supports a set of public and Telekom-internal payment methods, that all can be used in the same technical way. This is the payment gateway functionality of MPS.

MPS offers an optional user interface, where an end customer ("user") can select his preferred payment method during the checkout process. The business partner can customize this user interface by design, displayed or preselected payment methods. However, the business partner service can skip this dialog by submitting a payment method in the service call.

The process of the payment consists of two phases: The first is called "reservation". In this phase a check is made whether the payment method can be used. For this check the user will be redirected to the website of the payment method provider, on which the user needs to authenticate himself and agree to the payment request. In the second phase, the so-called claim, the transaction is finalized and the requested amount will be collected.

This 2-phase process has the following advantage for the business partners: After the first phase the business partner can be sure, that the end customer is able to pay and already authorized the payment. However, the transaction is not finally committed. If the business partner has problems to deliver the products to the customer, the business partner can still cancel the transaction. A more cost-intensive refund process is not necessary. The 2-phase process also supports use cases like reservation during the online payment process, claim at a later point, when the product is ready for shipment..

The 2-phase process is supported for all payment methods.

However, if the business partner needs to refund, the MPS offers an interface to do this too. Please note, not all payment methods offer a refund functionality.

Furthermore, MPS offers payment transactions without customer interactions too. These payments are useful in the subscription use case and are called "billings". To use this kind of payments the business partner needs to get a payment agreement of the customer during the checkout process.

This document describes the direct payment API. The direct payment API can be used in applications, that not support html clients. This payment variation is also known as "SOA payment".

All supported payment methods for direct payment are listed below.

payment method ¹	value	subscription	comments
credit card ²	creditcard_computop	yes	all popular credit card brands are supported. payment service provider: Computop
mobile payment (Germany)	mobilebill_mbe4	no	all German mobile network operators (MNOs) and some resellers are supported restriction for online services of the Telekom Deutschland GmbH: only customers of Vodafone or Telefónica (E-Plus, O2) are supported payment service provider: mbe4 GmbH
paypal ³	paypal	yes	payment service provider: PayPal
Telefonkarte Comfort	telephonecard	no	prepaid card, can be purchased in Telekom Shops
invoice by Deutsche Telekom			
... fixed line Croatia	fixedlinebill_croatia	yes	only direct registration
... fixed line Germany	fixedlinebill_telekom	yes	
... mobile (via SDP)	mobilebill_sdp	yes	deprecated
... mobile (via XTC)	mobilebill_xtc	yes	

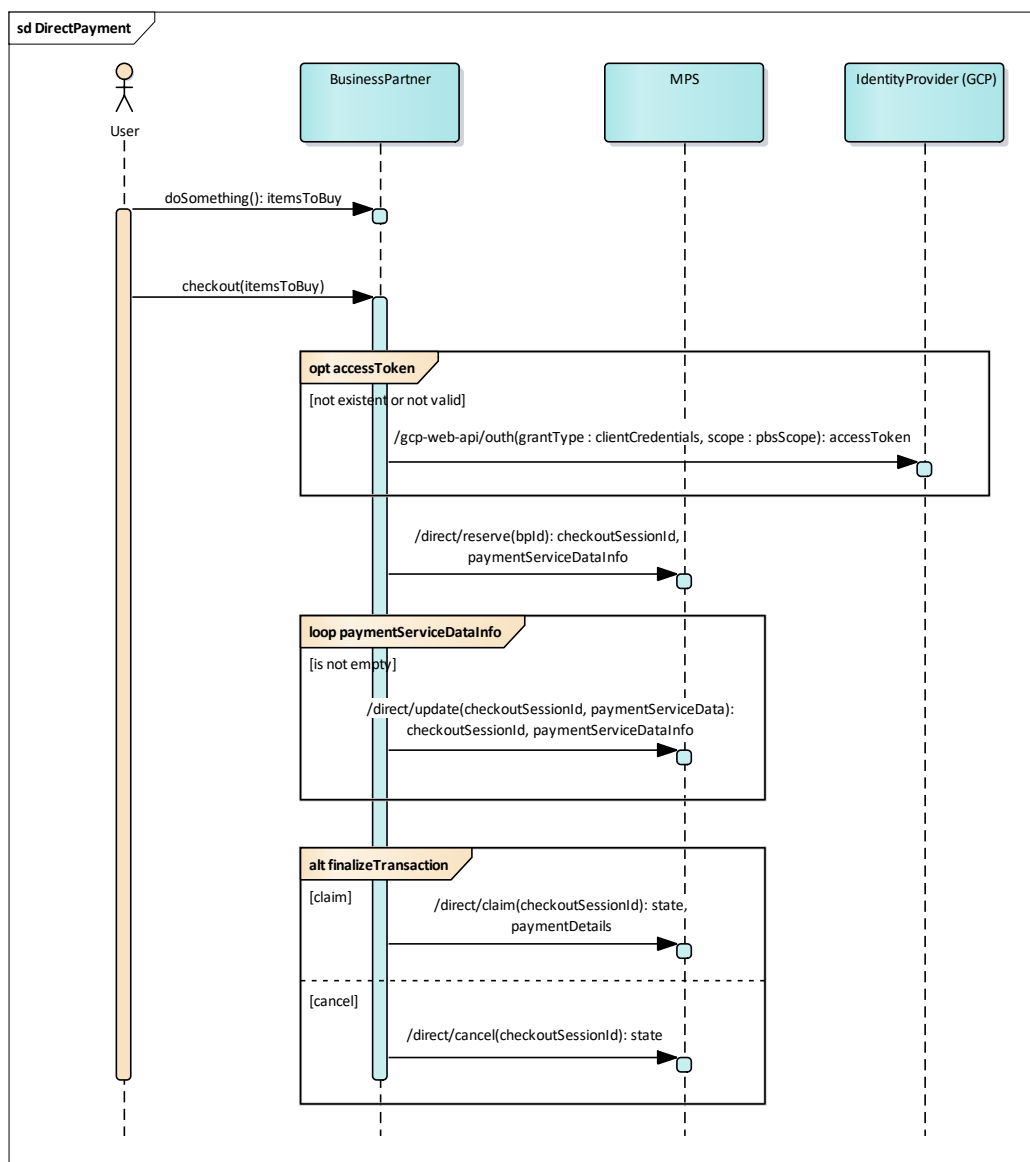
Notes:

1. If you're using your own user interface for the payment method selection and you grant a hundred per cent discount to a customer, use "voucher" as value for the payment method in the interface object instead of the original payment methods name.
2. To use direct payment with credit cards, the customer needs to register first. You will get an input assistance you can use for direct payment.
3. To use direct payment with PayPal, the customer needs to register first. You will get an input assistance you can use for direct payment.

3 direct payment without couponing

The workflow “direct payment without couponing” can be used in applications, if the application or system doesn’t support html clients. The application needs to provide all required payment data to finalize the payment process.

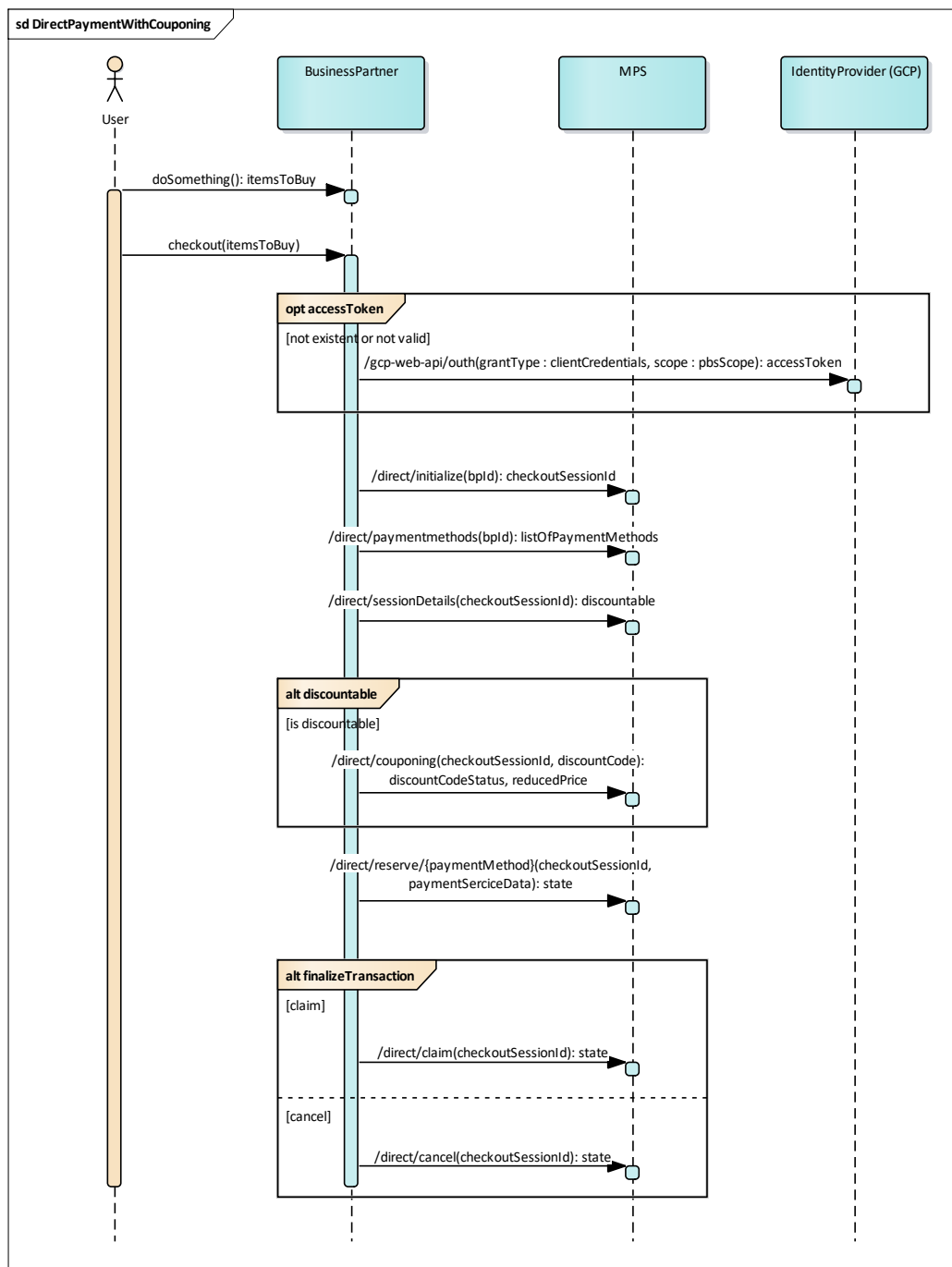
Note: Not all payment methods support the direct payment workflow. And some of the payment methods needs to use in the web payment flow before the service can use them in the direct payment workflow to get the required input assistances.



4 direct payment with couponing

The workflow “direct payment with couponing” can be used in applications, if the application or system doesn’t support html clients. The application needs to provide all required payment data to finalize the payment process.

Note: Not all payment methods support the direct payment workflow. And some of the payment methods needs to use in the web payment flow before the service can use them in the direct payment workflow to get the required input assistances.



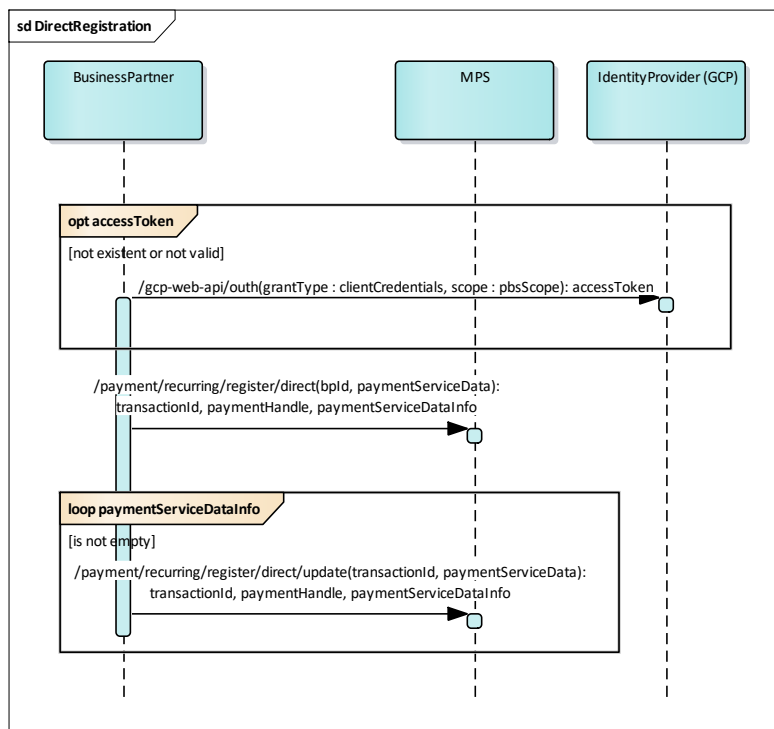
5 direct registration

The direct registration can only be used for payment methods of Deutsche Telekom. The workflow allows a service agent to register a payment method in the name of a customer.

Note: This operation is only accessible via the payment API.

Endpoint for the payment API Test: <https://api-payment0.ver.sul.t-online.de/pbs-payment-api>

Endpoint for the payment API Production: <https://api.payment.telekom.com/pbs-payment-api>



6 API documentation

6.1 direct payment (without couponing)

6.1.1 payment methods

The service can check which payment methods are valid for his payment. The service can use this call if the service renders the payment method selection page by itself.

The service will get all available payment methods.

operation

POST /direct/paymentmethods/

request

field name	type	length	multiplic- ity	comments
businessPartnerConfigId	string	50	1	Id of the configuration of the business partner. The Id will be provided by MPS.
currency	string	3	1	Currency of the purchase Based on ISO-4217 For purchases with Miles&More use "XYL".
grossAmount	number		1	gross amount of the product

response

field name	type	length	multipli- city	comments
availablePaymentMethods	string		0..*	List of the available payment methods.

error messages

http status	message	description
400	FAILED_VALIDATION	request validation failed
400	INVALID_REQUEST	client errors (e.g. media type not supported)
404	CONFIGURATION_NOT_FOUND	business partner configuration not found
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

6.1.2 reserve

The service can start the direct payment process by calling the operation "reserve".

operation

POST /direct/reserve

request

field name	type	length	multiplic- ity	comments
businessPartnerConfigId	string	50	1	Id of the configuration of the business partner. The Id will be provided by MPS.
currency	string	3	1	Currency of the purchase Based on ISO- 4217 For purchases with Miles&More use "XYL".
customerInformation	customer information		0..1	additional information about the customer can be used for specific payment methods only
description	string	127	1	Description of the basket

discountDetails	discount details		0..1	Information about the discount of a basket. If settlement is active and more than one item in the basket, the request will be rejected. Cannot be used, if "paymentMethod" is empty. "couponingData" and "discountDetails" cannot be used together. For more details see chapter 6.6.
lineItems	line item		0..*	List of items in the basket. For more details see chapter 6.6.
paymentMethod	string	255	1	The stated payment method will be used.
paymentServiceData	payment service data (payment)		1	Contains the payment information to fulfil the payment process. For more details see chapter 6.6.
settlementData	ipdr settlement		0..1	Contains the settlement information of the basket. The items of the list need to match with the items of the basket ("lineItems"). For details see chapter 6.6.

response

field name	type	length	multi- city	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentServiceDataInfo	payment service data info		0..1	... For more details see chapter 6.6
paymentServiceDataInfoDetail	payment service data info detail		0..1	... For more details see chapter 6.6
state	state	255	0..1	session state. For more details see chapter 6.7

error messages

http status	message	description
400	FAILED_VALIDATION	request validation failed
400	INVALID_REQUEST	client errors (e.g. media type not supported)
400	PAYMENT_TRANSACTION_IN_INVALID_STATE	payment transaction needed to be settled is in ERROR state
400	SETTLEMENT_ITEM_INDEX_MISMATCH	payment transaction indexes do not match the ones of the settlement transaction
400	SETTLEMENT_TRANSACTION_ALREADY_EXISTS	settlement transaction already exist for the current payment
400	COUPONING_TRANSACTION_IN_INVALID_STATE	couponing transaction is not in the required state
403	AUTHORIZATION_ERROR	authorization challenges exceeded or user not allowed
403	CUSTOMER_COUNTRY_NOT_ALLOWED	customers country is restricted
403	TENANT_MISMATCH	the tenant of the payment transaction doesn't match the one of the settlement
404	CONFIGURATION_NOT_FOUND	business partner configuration not found
404	SETTLEMENT_CONFIGURATION_NOT_FOUND	settlement configuration not found
404	PAYMENT_TRANSACTION_NOT_FOUND	payment transaction wasn't found by the settlement api
404	COUPONING_TRANSACTION_NOT_FOUND	couponing transaction could not be found
500	RESERVATION_ERROR	reservation process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

6.1.3 update

The service can update the direct payment process with additional information (e.g. a pin) by calling the operation "update".

operation

POST /direct/update

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentServiceData	payment service data (payment)		1	Contains the payment information to fulfil the payment process. For more details see chapter 6.6.

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentServiceDataInfo	payment service data info		0..1	... For more details see chapter 6.6
paymentServiceDataInfoDetail	payment service data info detail		0..1	... For more details see chapter 6.6
state	state	255	0..1	session state. For more details see chapter 6.7

error messages

http status	message	description
400	FAILED_VALIDATION	request validation failed
400	INVALID_REQUEST	client errors (e.g. media type not supported)
400	TRANSACTION_TIMED_OUT	payment transaction expired
400	TRANSACTION_INVALID_STATE	payment transaction is in an invalid state
403	CUSTOMER_COUNTRY_NOT_ALLOWED	customers country is restricted
403	AUTHORIZATION_ERROR	authorization challenges exceeded or user not allowed
404	SESSION_NOT_FOUND	checkout session not found
404	TRANSACTION_NOT_FOUND	payment transaction not found
404	CONFIGURATION_NOT_FOUND	business partner configuration not found
409	TRANSACTION_IN_PROGRESS	the current checkout session is already being processed
500	RESERVATION_ERROR	reservation process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

6.1.4 claim

The service can collect the reserved amount by calling the operation "claim".

operation

POST /direct/claim

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
couponingTransactionId	string	40	0..1	Id of the couponing transaction, the payment transaction based upon.
effectiveTotal	number		0..1	the collected amount
paymentMethod	string	255	1	used payment method
paymentMethodDetails	payment method de- tails		1	Structure of the payment method details. Each payment method used to have a different one. For more see chapter 6.5.

transactionAlreadyClaimed	boolean		0..1	if true, the transaction was already claimed.
paymentTransactionId	string	40	0..1	Id of the payment transaction.
settlementTransactionId	string	40	0..1	Id of the settlement transaction.
state	state	255	0..1	session state. For more details see chapter 6.7

error messages

http status	message	description		
400	FAILED_VALIDATION	request validation failed		
400	INVALID_REQUEST	client errors (e.g. media type not supported)		
400	TRANSACTION_INVALID_STATE	payment transaction has an invalid state		
400	TRANSACTION_TIMED_OUT	payment or PSP transaction expired		
400	SETTLEMENT_TRANSACTION_IN_INVALID_STATE	settlement transaction has an invalid state for finalization		
404	SESSION_NOT_FOUND	checkout session not found		
404	TRANSACTION_NOT_FOUND	payment transaction not found		
404	CONFIGURATION_NOT_FOUND	business partner configuration not found		
404	SETTLEMENT_TRANSACTION_NOT_FOUND	settlement transaction not found		
409	TRANSACTION_IN_PROGRESS	the current checkout session is already being processed		
500	CLAIM_ERROR	claim process failed		
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)		

6.1.5 cancel

The service can release the reserved amount by calling the operation “cancel”.

operation

POST /direct/cancel

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentTransactionAlreadyCancelled	boolean		0..1	If true, the payment transaction was already cancelled.
paymentTransactionId	string	40	0..1	Id of the payment transaction.
reservedAmountReleased	boolean		0..1	If true, the reserved amount was released.
state	state	255	0..1	session state. For more details see chapter 6.7

error messages

http status	message	description		
400	FAILED_VALIDATION	request validation failed		
400	INVALID_REQUEST	client errors (e.g. media type not supported)		
400	TRANSACTION_INVALID_STATE	payment transaction has an invalid state		
400	COUPONING_TRANSACTION_INVALID_STATE	couponing transaction is not reserved		
404	SESSION_NOT_FOUND	checkout session not found		
404	TRANSACTION_NOT_FOUND	payment transaction not found		
404	COUPONING_TRANSACTION_NOT_FOUND	couponing transaction to be cancelled not found		
409	TRANSACTION_IN_PROGRESS	the current checkout session is already being processed		
500	CANCEL_ERROR	cancel process failed		
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)		

6.2 direct payment (with couponing)

This chapter contains additional operations, the service need for couponing transactions. The other operations are described in chapter 8.3.

6.2.1 initialize

The service can start the payment process with the initialize request.

operation

POST /direct/initialize/

request

field name	type	length	multiplicity	comments
businessPartnerConfigId	string	50	1	Id of the configuration of the business partner. The Id will be provided by MPS.
couponingData	couponing data		0..1	Couponing constraints of DCS. Cannot be used, if settlement is active and more than one item in the basket. Cannot be used, if "paymentMethod" is set. "couponingData" and "discountDetails" cannot be used together. For more details see chapter 6.6.
currency	string	3	1	Currency of the purchase Based on ISO-4217
customerInformation	customer information		0..1	For purchases with Miles&More use "XYL". additional information about the customer can be used for specific payment methods only
description	string	127	1	Description of the basket
excludedPaymentMethods			0..*	List of payment methods, that are excluded from the current purchase. Will be ignored, if no payment method is provided.
lineItems	line item		0..*	gross amount of the product
paymentServiceDataList	payment service data list		1	Contains the payment information to fulfil the payment process. For more details see chapter 6.6.
settlementData	ipdr settlement		0..1	Contains the settlement information of the basket. The items of the list need to match with the items of the basket ("lineItems"). For details see chapter 6.6.

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
state	state	255	0..1	session state. For more details see chapter 6.7

error messages

http status	message	description
400	FAILED_VALIDATION	request validation failed

6.2.2 session details

The service can get information about the session by calling the "sessionDetails" request.

operation

POST /direct/sessionDetails/

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	1	Id of the checkout session.

response

field name	type	length	multiplicity	comments
businessPartnerConfigId	string	50	0..1	Id of the configuration of the business partner. The Id will be provided by MPS.
couponingTransactionId	string	50	0..1	Id of the couponing transaction.
couponingTransactionState	string		0..1	state of the couponing transaction
creationDate	string		0..1	timestamp of the creation of the transaction
discountCode			0..1	
discountable	boolean		0..1	if true, the transaction is discountable
locale	locale	10	0..1	Defines the language based the locale. Local is a combination of the language code (ISO-639-1) and the country code (ISO 3166-1 alpha-2). Samples: Use "de_DE" for German, "zh_CN" for Simplified Chinese and "zh_TW" for Traditional Chinese. If not set, the default of the browser will be used. If language is not supported, the default of the theme will be used.
modificationDate	string		0..1	timestamp of the last modification of the transaction
paymentMethod	string	255	0..1	The stated payment method will be used. If no payment method is stated, than the payment selection page is shown to the user.
paymentTransactionId	string	50	0..1	Id of the payment transaction
reducedPrice	number		0..1	reduced amount of the basket
request	object		0..1	original request
returnUrl	string	4096	0..1	URL of the business partner, the user will be redirected at the end of the payment process. Depending on the payment method (e.g. PayPal) several parameters may be added to the URL.
sessionId	string	50	0..1	Id of the checkout session
sessionState	state	255	0..1	session state. For more details see chapter 6.7
sessionType	type		0..1	session type
settlementTransactionId	string	50	0..1	Id of the settlement transaction
tenant	string		0..1	business tenant

error messages

http status	message	description
400	FAILED_VALIDATION	request validation failed
404	SESSION_NOT_FOUND	checkout session not found

6.2.3 couponing

The service can check a DCS voucher.

operation

POST /direct/couponing/

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	1	Id of the checkout session.
discountCode	string		1	a DCS discount code

response

field name	type	length	multiplicity	comments
detail	string		0..1	detail of the discount code
discountCodeStatus	string		0..1	status of the discount code
reducedPrice	number		0..1	reduced amount

error messages

http status	message	description
400	FAILED_VALIDATION	request validation failed
404	SESSION_NOT_FOUND	checkout session not found

6.2.4 reserve (of an initialized transaction)

The service can update the checkout session with additional information and start the reservation process.

operation

POST /direct/reserve/{paymentMethod}

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentServiceData	payment service data (payment)		1	Contains the payment information to fulfil the payment process. For more details see chapter 6.6.

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentServiceDataInfo	payment service data info		0..1	... For more details see chapter 6.6
paymentServiceDataInfoDetail	payment service data info detail		0..1	... For more details see chapter 6.6
state	state	255	0..1	session state. For more details see chapter 6.7

error messages

http status	message	description
400	FAILED_VALIDATION	request validation failed
404	SESSION_NOT_FOUND	checkout session not found

6.3 direct register

Note: This operation is only accessible via the payment API.

Endpoint for the payment API Test: <https://api-payment0.ver.sul.t-online.de/pbs-payment-api>

Endpoint for the payment API Production: <https://api.payment.telekom.com/pbs-payment-api>

6.3.1 register

The service can start the direct registration process by calling the operation “register”.

operation

POST /payment/recurring/register/direct

request

field name	type	length	multiplicity	comments
businessPartnerConfigId	string	50	1	Id of the configuration of the business partner. The Id will be provided by MPS.
currency	string	3	1	Currency of the purchase Based on ISO- 4217 For purchases with Miles&More use “XYL”.
customerInformation	customer infor- mation		0..1	additional information about the customer can be used for specific payment methods only
description	string	127	1	Description of the basket
paymentMethod	string	255	1	The stated payment method will be used.

paymentServiceData	payment service data (register)	1	Contains the payment information to fulfil the payment process. For more details see chapter 6.6.
--------------------	---	---	---

response

field name	type	length	multiplicity	comments
transactionId	string	40	0..1	Id of the register transaction.
paymentHandle	string		0..1	payment handle for recurring charges
paymentMethodDetails	payment method details		0..1	... For more details see chapter 6.6
paymentServiceDataInfo	payment service data info		0..1	... For more details see chapter 6.6
paymentServiceDataInfoDetail	payment service data info detail		0..1	... For more details see chapter 6.6
state	state	255	0..1	session state. For more details see chapter 6.7

error messages

http status	message	description
400	FAILED_VALIDATION	request, pmc or psp validation failed
404	CONFIGURATION_NOT_FOUND	business partner configuration not found
500	REGISTRATION_ERROR	registration process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

6.3.2 update

The service can update the direct payment process with additional information (e.g. a pin) by calling the operation “update”.

operation

POST /payment/recurring/register/direct/update

request

field name	type	length	multiplicity	comments
transactionId	string	40	0..1	Id of the register transaction.
paymentServiceData	payment service data (register)		1	Contains the payment information to fulfil the payment process. For more details see chapter 6.6.

response

field name	type	length	multiplicity	comments
transactionId	string	40	0..1	Id of the register transaction.
paymentHandle	string		0..1	payment handle for recurring charges
paymentMethodDetails	payment method details		0..1	... For more details see chapter 6.6
paymentServiceDataInfo	payment service data info		0..1	... For more details see chapter 6.6
paymentServiceDataInfoDetail	payment service data info detail		0..1	... For more details see chapter 6.6
state	state	255	0..1	session state. For more details see chapter 6.7

error messages

http status	message	description
400	FAILED_VALIDATION	request, pmc or psp validation failed
404	CONFIGURATION_NOT_FOUND	business partner configuration not found
500	REGISTRATION_ERROR	registration process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

6.4 error response

In case of the occurrence of an error, the service will get a dedicated error response. The possible error messages are described on each operation.

response

field name	type	length	multiplicity	comments
timestamp	long		1	milliseconds in UTC since 1/1/1970
status	int		1	HTTP status code
error	string		1	HTTP status message
message	string		1	error reason
path	string		1	requested resource path
subpath	string		0..1	requested subsystem resource path
detail	any		0..1	detailed information for debugging purposes

sample response

```

Response
{
  "timestamp": 1498131785613,
  "status": 404,
  "error": "Not Found",
  "message": "TRANSACTION_NOT_FOUND",
  "path": "/pbs-checkout-api/onetime/result",
  "detail": "Transaction not found or in invalid state"
}

```

6.5 payment method details

The actual structure depends on the payment method.

6.5.1 payment method: creditcard_computop

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	id of the account combination of ccBrand and maskedCreditCardNumber
acquirer	string	255	1	used acquirer
ccBrand	string	255	1	credit card brand
expiryDate	string	10	1	expiry date of credit card
maskedCreditCardNumber	string	20	1	masked credit card number, only the last 4 dig- its are shown
referenceNumber	string	64	1	reference number of the acquirer, needed for match with EPA files

6.5.2 payment method: mobilebill_mbe4

field name	type	length	multiplicity	comments
accountQualifier	string		1	
msisdn	string		1	Mobile Station International Subscriber Direc- tory Number
operatorId	string		1	

6.5.3 payment method: paypal

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "payer"
payer	string	1024	1	e-mail address of the PayPal user
payerId	string	255	1	Id of the PayPal user

6.5.4 payment method: telephonecard

field name	type	length	multiplicity	comments
------------	------	--------	--------------	----------

missingAmount	string	1	collected amount
originalBalance	string	1	original balance
remainingBalance	string	1	new balance

6.5.5 payment method: fixedlinebill_croatia

field name	type	length	multiplicity	comments
accountQualifier	string		1	identically to "serviceld"
serviceld	string		1	Id of the user at Telekom Croatia

6.5.6 payment method: fixedlinebill_telekom

field name	type	length	multiplicity	comments
accountQualifier	string		1	
displayName	string		1	display name of the customer
guid	string		1	global user id of the customer
ip	string		1	IP address

6.5.7 payment method: mobilebill_telekom

field name	type	length	multiplicity	comments
accountQualifier	string		1	
anonymousid	string		1	
msisdn	string		1	Mobile Station International Subscriber Directory Number

6.6 other structures

6.6.1 couponing constraint

For more information see documentation of DCS.

field name	type	length	multiplicity	comments
type	string		0..1	
value	object		0..1	The value will be routed to the discount code system (DCS).

6.6.2 couponing data

For more information see documentation of DCS.

field name	type	length	multiplicity	comments
client	string		1	
transactionConstraints	couponing constraint		0..*	

6.6.3 customer information

Additional information about the customer. Can be used for specific payment methods

field name	type	length	multiplicity	comments
externalIdentifier	string		0..1	identifier of the customer only for payment method „paysafecard“
shippingAddress	shipping address		0..1	shipping address of the customer for real products only used for payment method “paypal”

6.6.4 discount details

field name	type	length	multiplicity	comments
basketReducedTotal	number		1	reduced amount of the basket

description	string	0..1	description for the discount
-------------	--------	------	------------------------------

6.6.5 ipdr basket item

If more than one item in the basket, this information will be transferred to the SAP system in the detail IPDR.

field name	type	length	multiplicity	comments
index	string		0..1	unique index of the item will be used to assign the basket item with the settlement information if the index is not set, the assignment based on the item order of the basket and settlement items in the request if the index is set, the items with the same index will be assigned together
goodsId	string		0..1	unique id of the content
goodsType	string		0..1	classification of the item
targetSystemId	string		0..1	Id of the target SAP system
financialClientId	string		0..1	Id of the financial client in the SAP system
goodsPartnerImpactCategory	string		0..1	Id of the goods partner
salesChannelId	string		0..1	Id of the sales channel
subscriptionInfo	string		0..1	information for the subscription
voucherId	string		0..1	Id of the voucher the user used for the discount
discountReason	string		0..1	Cannot be empty, if discountDetails is set. reason for the discount Cannot be empty, if discountDetails is set.

6.6.6 ipdr settlement

field name	type	length	multiplicity	comments
basketItemsSettlementData	ipdr basket item		0..*	settlement information for a basket if set, "itemSettlementData" needs to be empty
itemSettlementData	ipdr basket item		0..1	settlement information for a single item if set, "basketItemsSettlementData" needs to be empty
settlementConfigurationId	int64		1	Id of the configuration of the settlement configuration of the business partner. The Id will be provided by MPS.
transactionSettlementData	ipdr transaction		1	settlement information for the transaction not specific to a basket

6.6.7 ipdr transaction

These information will be transferred to the SAP system in the head IPDR.

field name	type	length	multiplicity	comments
financialClientId	string		0..1	Id of the financial client in the SAP system
salesChannelId	string		0..1	id of the sales channel
targetSystemId	string		0..1	Id of the target SAP system
country	string		0..1	country code Based on ISO 3166-1 alpha-2
ustId	string		0..1	tax id number

6.6.8 input assistance

field name	type	length	multiplicity	comments
accountQualifier	string		1	dependent of the payment method. For more information see chapter 6.5.
inputAssistance	string		1	input assistant
paymentMethod	string		1	payment method

6.6.9 line item

field name	type	length	multiplicity	comments
description	string	1024	0..1	additional description of the product
grossAmount	number		1	gross amount of the product
index	string		0..1	additional index if set and settlement is required: need to have an item with the identical index in the settlement data
name	string	255	1	name of the product
paymentServiceItemData	payment service item data			additional payment information for specific payment methods
quantity	int32		0..1	quantity of the product, default: 1
tax	number		0..1	deprecated tax of the product For payment method "mobilebill_xtc" this element is mandatory.
taxRate	number		0..1	tax rate of the product
uiDetails	ui details		0..1	A generic structure the business partner can use to customize the payment selection page. For details see chapter 6.6.

sample

```
lineItems": [{
  "name": "Name LineItem 1",
  "description": "Description LineItem_1",
  "grossAmount": 0.22,
  "taxRate": 19,
  "quantity": 1
}],
```

6.6.10 mobile xtc

The element "mobile xtc" contains additional product information for using the payment method "mobile_xtc".

field name	type	length	multiplicity	comments
vasCode	string		1	VAS code of the product

6.6.11 payment service data (payment)

The element "payment service data" is a simple mapping structure of <object, string>. The possible mappings are depending on the payment method.

object	comment
payment method: creditcard_computop	
inputAssistance	To get an input assistance, the user needs to pay with credit card in a web flow and generate an input assistance on that way first.
cvc	credit card verification code
payment method: mobilebill_mbe4	
msisdn	Mobile Station International Subscriber Directory Number. The users mobile phone number.
tan	one time password, the user will get via SMS.
payment method: paypal	
inputAssistance	To get an input assistance, the user needs to pay with credit card in a web flow and generate an input assistance on that way first.
payment method: telephonecard	
pin	pin
payment method: mobilebill_sdp	
msisdn	Mobile Station International Subscriber Directory Number. The users mobile phone number.
otp	one time password, the user will get via SMS.

payment method: mobilebill_xtc

msisdn	Mobile Station International Subscriber Directory Number. The users mobile phone number.
otp	one time password, the user will get via SMS.
forcePinRetrieval	Boolean value

payment method: fixedlinebill_telekom

accessToken	JWT Access Token from Telekom IDM (SAM3) containing UserID etc.
Optional: clientIp	IP-address of the users client.

6.6.12 payment service data (register)

The element “payment service data” is a simple mapping structure of <string, object>. The possible mappings are depending on the payment method.

object **comment****payment method: mobilebill_sdp**

msisdn	Mobile Station International Subscriber Directory Number. The users mobile phone number.
otp	one time password, the user will get via SMS.

payment method: mobilebill_xtc

msisdn	Mobile Station International Subscriber Directory Number. The users mobile phone number.
otp	one time password, the user will get via SMS.
forcePinRetrieval	Boolean value

payment method: fixedlinebill_telekom

guid	Global User ID. The id of the user in the Telekom systems.
clientIp	IP-address of the users client.

payment method: fixedlinebill_croatia

serviceld	ID, with that the payment can be done on a fixed line bill by Telekom Croatia
-----------	---

6.6.13 payment service data list

The element “payment service data list” is a complex mapping structure of <string, <string, object>>. The key is the payment method, the value is the corresponding “payment service data (payment)”.

field name	type	length	multiplicity	comments
“payment method”	payment service data (payment)		0..*	contains the payment service data for a payment method

6.6.14 payment service item data

The element “payment service item data” contains additional product information for specific payment methods.

field name	type	length	multiplicity	comments
fixedlinebill_telekom	fixed line bill		1	additional product information for using landline bill (for future use)
mobile_xtc	mobile xtc		1	additional product information for using XTC

6.6.15 payment service data info

The element “payment service data info” is a simple mapping structure of <object, info>, with “info” as a list of following values:

info	comments
ATTRIBUTE_MISSING	The attribute is missing. Use “update” to send the missing attribute.
ATTRIBUTE_INVALID	The attribute is invalid. Use “update” to send the valid attribute.
INSUFFICIENT_FUNDS	
NOT_FOUND	The contract was not found in the T-Systems SAP system.
TOO_MANY_RETRIES	The attribute was used too often. This info message the service will get, when using a pin or one time password too often.

For possible objects see “payment service data”.

6.6.16 payment service data info detail

This element contains additional information, if payment method “telephoncard” is used.

field name	type	length	multiplicity	comments
originalBalance	number		1	
remainingBalance	number		1	
missingAmount	number		1	

6.6.17 shipping address

This element contains the shipping address of a customer for real products. This element is only useful for the payment method “paypal”

field name	type	length	multiplicity	comments
salutation	string	255	0..1	
firstName	string	255	1	
lastName	string	255	1	
addressLine1	string	1024	1	
addressLine2	string	1024	0..1	
postalCode	string	255	1	
city	string	255	1	
state	string	255	0..1	
countryCode	string	2	1	ISO 3166-1 alpha-2

The shipping address can also be used for shipping to post boxes or post offices.

samples

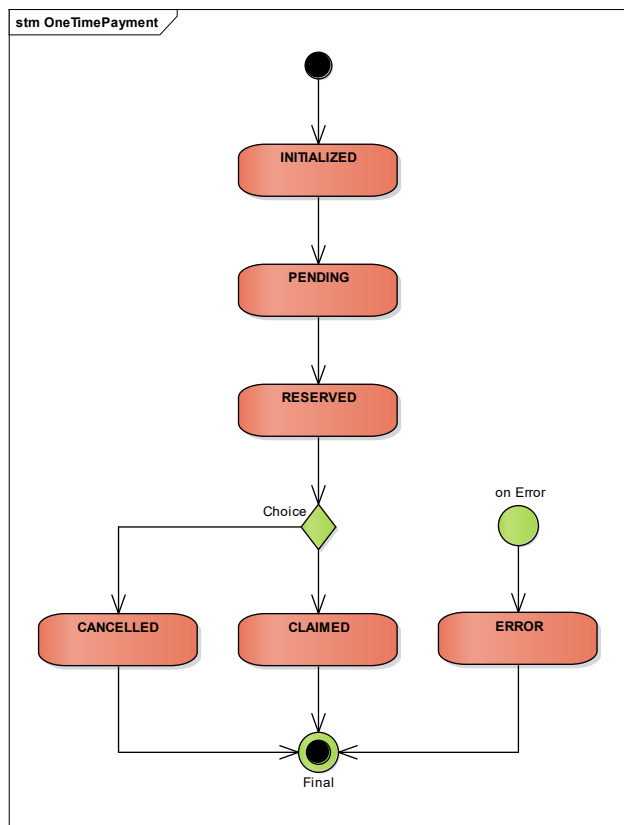
<pre>"shippingAddress" : { "salutation" : "", "firstName" : "Hans", "lastName" : "Mustermann", "addressLine1" : "T-Online-Allee 1", "addressLine2" : "", "postalCode" : "64295", "city" : "Darmstadt", "state" : "", "countryCode" : "DE" }</pre>
<pre>"shippingAddress" : { "salutation" : "", "firstName" : "Hans", "lastName" : "Mustermann", "addressLine1" : "12345678", "addressLine2" : "Packstation 801", "postalCode" : "64295", "city" : "Darmstadt", "state" : "", "countryCode" : "DE" }</pre>
<pre>"shippingAddress" : { "salutation" : "", "firstName" : "Hans", "lastName" : "Mustermann", "addressLine1" : "12345678", "addressLine2" : "Postfiliale 123", "postalCode" : "64295", "city" : "Darmstadt", "state" : "", "countryCode" : "DE" }</pre>

6.7 states and transitions

name	comments
INITIALIZED	The transaction is initialized
PENDING	The transaction is pending: Waiting for interactions of the customer.

RESERVED	The transaction is reserved at the PSP
CLAIMED	The transaction is claimed
CANCELLED	The transaction is cancelled
REFUNDED	The transaction is fully refunded.
FINALIZED	The transaction is finalized.
ERROR	An error has occurred during the transaction.

state model for direct payments



7 table of abbreviation

abbreviation	description
GCP	Global Customer Platform
MPS	Multi Payment Service Centre
PBS	Payment and Billing System
PMC	Payment Method Collector
PSP	Payment Service Provider

revision history

revision	date	editor	comments
1	02/07/2019	Stefan Heitzmann	version 1.0 of this API documentation
2	03/18/2019	Stefan Heitzmann	added payment service data list to the direct initialization request
4	04/22/2020	Stefan Heitzmann	added payment method fixedlinebill_croatia
5	05/11/2020	Stefan Heitzmann	changed field types in payment service data info detail from int64 to number, delete the state field