

Telekom Multi Payment Service Centre

integration guide of the Checkout API

Version 18
Stand 05/17/2021
Status final

public

table of contents

1	overview.....	5
2	service architecture.....	7
2.1	base URLs.....	7
2.2	service authentication.....	7
2.3	disclaimer.....	7
3	onetime payment.....	8
4	recurring payment.....	9
4.1	recurring register.....	9
4.2	recurring payment	10
4.3	recurring charge (two phase).....	11
4.4	recurring charge (one phase).....	11
5	refund	12
6	API documentation	13
6.1	onetime payment	13
6.1.1	payment methods.....	13
6.1.2	reserve.....	13
6.1.3	result.....	15
6.1.4	claim.....	16
6.1.5	cancel.....	16
6.1.6	session details.....	17
6.2	recurring payment	18
6.2.1	payment methods.....	18
6.2.2	register	18
6.2.3	register result	20
6.2.4	register session details	21
6.2.5	payment.....	23
6.2.6	payment result	25
6.2.7	payment claim.....	26
6.2.8	payment cancel	27
6.2.9	payment session details	28
6.2.10	charge reserve	28
6.2.11	charge claim.....	30
6.2.12	charge cancel	31
6.2.13	charge reserveAndClaim.....	31
6.2.14	charge session details	32
6.3	refund	33
6.3.1	refund on basket level.....	33
6.3.2	refund on item level.....	34
6.4	error response	35
6.5	payment method details	36
6.5.1	payment method: alipay_wirecard	36
6.5.2	payment method: alipay_computop.....	36
6.5.3	payment method: amazon	36
6.5.4	payment method: applepay_computop.....	36
6.5.5	payment method: banktransfer_sofort.....	36
6.5.6	payment method: bcinvoice_telekom.....	37
6.5.7	payment method: creditcard_computop.....	37
6.5.8	payment method: directdebit_mrnxnet	37

6.5.9	payment method: eps	37
6.5.10	payment method: loyalty_milesandmore.....	37
6.5.11	payment method: mobilebill, mobilebill_mbe4	37
6.5.12	payment method: paypal.....	38
6.5.13	payment method: paysafecard	38
6.5.14	payment method: telephonecard.....	38
6.5.15	payment method: fixedlinebill_croatia	38
6.5.16	payment method: fixedlinebill_telekom.....	38
6.5.17	payment method: mobilebill_sdp, mobilebill_xtc	38
6.6	other structures	38
6.6.1	address	38
6.6.2	bc invoice	39
6.6.3	contact data	39
6.6.4	couponing constraint.....	39
6.6.5	couponing data.....	39
6.6.6	customer information.....	40
6.6.7	debitor account data.....	40
6.6.8	ipdr basket item	40
6.6.9	ipdr settlement	41
6.6.10	ipdr transaction.....	41
6.6.11	input assistance	41
6.6.12	invoice data	41
6.6.13	line item	41
6.6.14	mobile xtc	42
6.6.15	payment service item data	42
6.6.16	service period.....	42
6.6.17	subscription info	43
6.6.18	transaction details	43
6.6.19	ui details.....	43
6.7	states and transitions.....	44
7	theme customization	46
7.1	API documentation	46
7.1.1	Check current themes	46
7.1.2	Upload a new theme	47
7.1.3	Download an existing theme	48
7.1.4	Delete an existing theme	48
7.2	Theme development	49
8	privacy information	50
8.1	payment method: alipay_wirecard	50
8.2	payment method: alipay_computop	50
8.3	payment method: amazon	50
8.4	payment method: applepay_computop	50
8.5	payment method: banktransfer_sofort	50
8.6	payment method: bcinvoice_telekom	51
8.7	payment method: creditcard_computop	51
8.8	payment method: directdebit_nexnet.....	51
8.9	payment method: eps.....	51
8.10	payment method: fixedlinebill_croatia	51
8.11	payment method: fixedlinebill_telekom	51
8.12	payment method: loyalty_milesandmore	52

8.13	payment method: mobilebill.....	52
8.14	payment method: mobilebill_mbe	52
8.15	payment method: mobilebill_sdp.....	52
8.16	payment method: mobilebill_xtc	52
8.17	payment method: paypal	52
8.18	payment method: paysafecard	53
8.19	payment methode: telephonecard	53
9	table of abbreviation	54

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 flex payment 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.

All supported payment methods are listed below.

payment method ¹	value	refund	partial refund ²	flex payments ³	comments
alipay	alipay_wirecard	yes	yes	no	payment service provider: Alipay (via Wirecard) deprecated
alipay	alipay_computop	yes	yes	no	payment service provider: Alipay (via Computop)
amazon pay apple pay	amazon applepay_computop	yes yes	yes yes	yes not yet	payment service provider: Amazon payment service provider: Apple (via Computop)
bank transfer	banktransfer_sofort	no	no	no	available for customers in Germany only
					payment service provider: SOFORT Überweisung
credit card	creditcard_computop	yes	yes	yes	all popular credit card brands are supported.
direct debit	directdebit_mrnexnet	no	no	yes	payment service provider: Computop available for customers in Germany only.
eps	eps	no	no	no	payment service provider: mr.nexnet
miles & more	loyalty_milesandmore	yes	no	yes	payment service provider: STUZZA GmbH only available for business partners of Deutsche Lufthansa
mobile payment (Germany)	mobilebill	no	no	no	payment service provider: Miles&More GmbH a convenience method for using the payment methods “mobilebill_mbe4” and “mobilebill_xtc”.

mobile payment (Germany)	mobilebill_mbe4	yes	no	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 paysafecard	paypal paysafecard	yes no	yes no	yes no	payment service provider: PayPal at the moment not available for online services of Deutsche Telekom
Telefonkarte Comfort	telephonecard	yes	no	no	prepaid card, can be purchased in Telekom Shops

invoice by Deutsche Telekom					
... for business customers	bcinvoice_telekom	no	no	yes	for business customers (of T-Systems), only recurring
... fixed line Croatia	fixedlinebill_croatia	no	no	yes	only recurring charge
... fixed line Germany	fixedlinebill_telekom	no	no	yes	
... mobile (via SDP)	mobilebill_sdp	no	no	yes	deprecated
... mobile (via XTC)	mobilebill_xtc	no	no	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. You cannot refund more than the original transaction.
3. "Flex payment" was called "subscription" in the past.

2 service architecture

The interface is a REST interface with following standards:

- HTTP/1.1 as transport protocol
- JSON as content data structure
- UTF8 as character set of JSON
- OAuth2 as service authentication

The responses of the REST calls will be synchronous.

2.1 base URLs

Authentication Test&Production: <https://global.telekom.com>

Checkout API Test: <https://api-payment0.ver.sul.t-online.de/pbs-checkout-api>

Checkout API Production: <https://api.payment.telekom.com/pbs-checkout-api>

2.2 service authentication

The MPS uses standard OAuth2 authentication to secure its interface. The service needs to get authenticated by the identity provider of the MPS: the Global Customer Platform (GCP). The GCP will provide an access token to the service, which is valid for a certain time. This access token allows the service to use the interface of the MPS.

The following sample shows the communication of the service with the GCP:

Request
<pre>POST /gcp-web-api/oauth HTTP/1.1 Host: global.telekom.com Content-Type: application/x-www-form-urlencoded Authorization: Basic {{basic_auth}}</pre>
Response
<pre>{ "access_token": "eyJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJnY3AiLCJhdWQiOiJodH-RwczoVl3NhbmrRib3guaw50ZXJuZXQtYnVzaW5lc3Mtc3VpdGUuZGUvcGJzLXNldHR-sZW11bnQtYXBpIiwiWF0IjoxNDc0MDA4NDU0LCJuYmYiOjE0NzQwMDgzOTQsImV4cCI6MTQ3NDAzNzI1NCwianRpIjoiY2U4NzBkNDEtOWRmNS00YWZ1LWFlMmUtMTFmYjk3MDg0NmVjIiwiHR0cDovL2djcC50ZWxla29tLmR1L2F4c2NoZW1hL3Rlbm-FudCI6InBhcnRuZXJBIn0.hj0goGguC8p052rpfH8DT6hiNAXnFPNd5RrLJGN1o7o", "token_type": "bearer", "expires_in": 28800, "scope": "QN5SQWI8" }</pre>

Note: For additional information please check the API documentation of the GCP.

2.3 disclaimer

Due to the permanent further development of the payment API it is possible that additional fields will be added to the JSON structures of different operations. Therefor connected services need to be able to handle currently unknown fields.

Current JSON libraries enable this flexible handling of input data. E.g. the Jackson Java library can handle these kind of fields with

- the global directive `DeserializationConfig.Feature.FAIL_ON_UNKNOWN_PROPERTIES` or
- the annotation `@JsonIgnoreProperties(ignoreUnknown = true)`.

3 onetime payment

Onetime payment is the standard use case for most payment processes of a service.

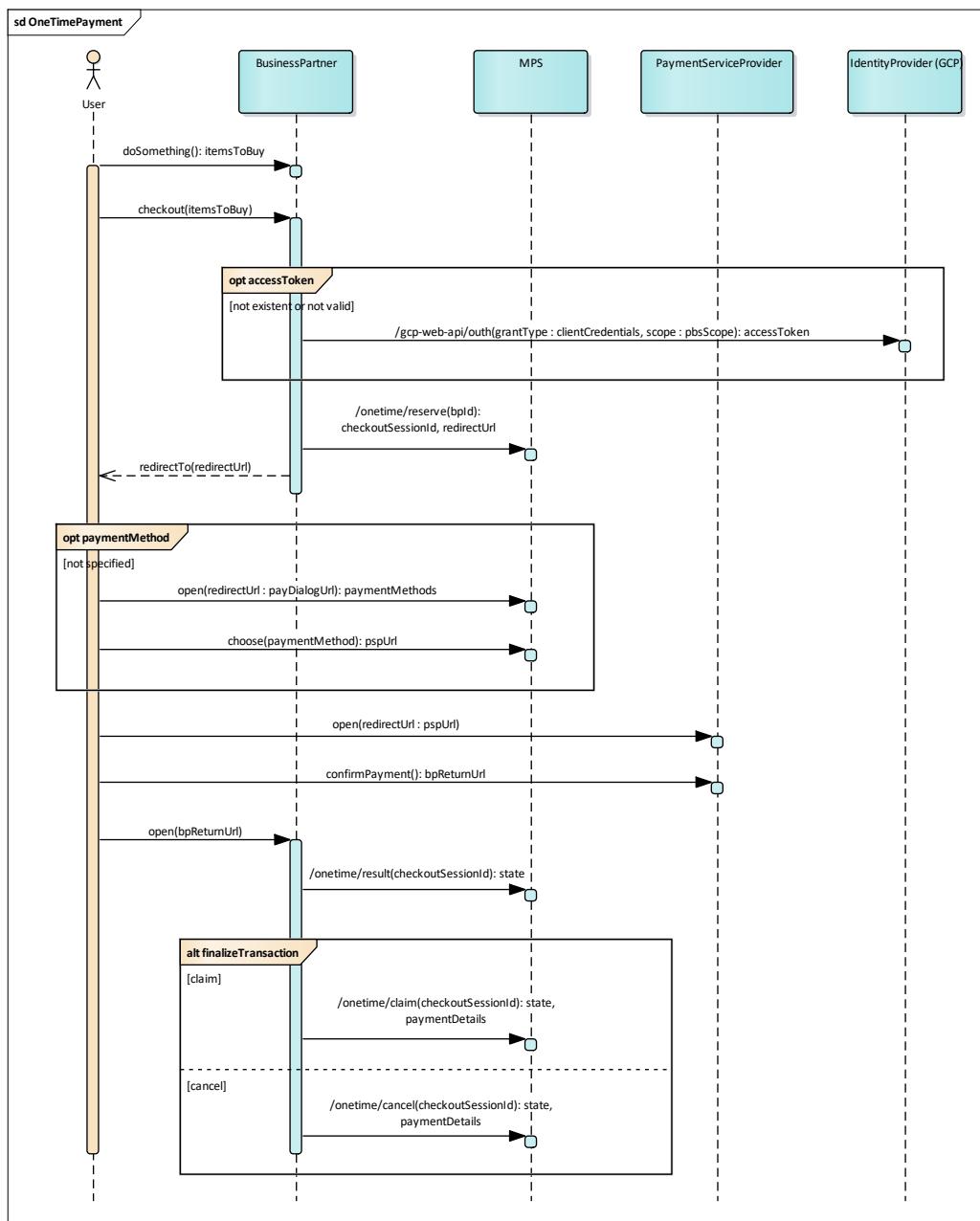
The customer needs to interact with the payment service provider to authorise a payment.

To start the workflow the calls the onetime payment interface of the MPS checkout module. In return the service gets a session id and a redirect URL.

If the services doesn't provide a payment method the customer will be redirected to the pay dialog URL of the MPS checkout module, where the customer can choose its preferred payment procedure.

If the payment method is known the customer will be redirected to the payment service provider, where he can authorise the payment.

The customer will be redirected to the service, where the service can check the result of the authorisation and claim the reserved amount.



4 recurring payment

If a subscription is involved the recurring payment workflow is needed. This workflow contains two phases.

In the first phase the customer needs to interact with the payment service provider to authorise the service to collect recurring payments. Two workflows supports the first phase: "recurring register" and "recurring payment".

In the second phase the service can collect the recurring payments without interactions of the customer. Two workflows supports the second phase: "recurring charge (one phase)" and "recurring charge (two phase)".

4.1 recurring register

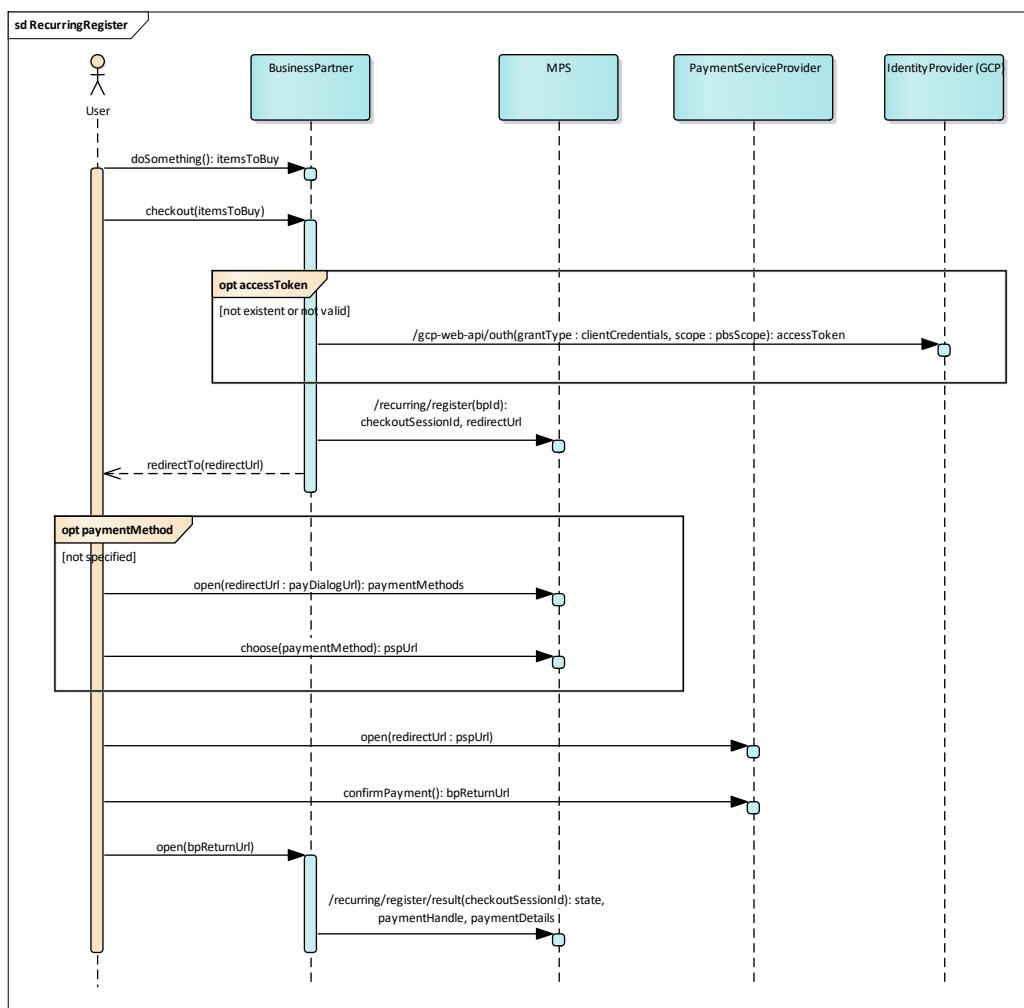
With the workflow "recurring register" the customer can authorise a service to collect recurring payments. In this workflow the account of the customer will not be charged.

To start the workflow the calls the recurring register interface of the MPS checkout module. In return the service gets a session id and a redirect URL.

If the services doesn't provide a payment method the customer will be redirected to the pay dialog URL of the MPS checkout module, where the customer can choose its preferred payment procedure.

If the payment method is known the customer will be redirected to the payment service provider, where he can authorise the service to collect recurring payments.

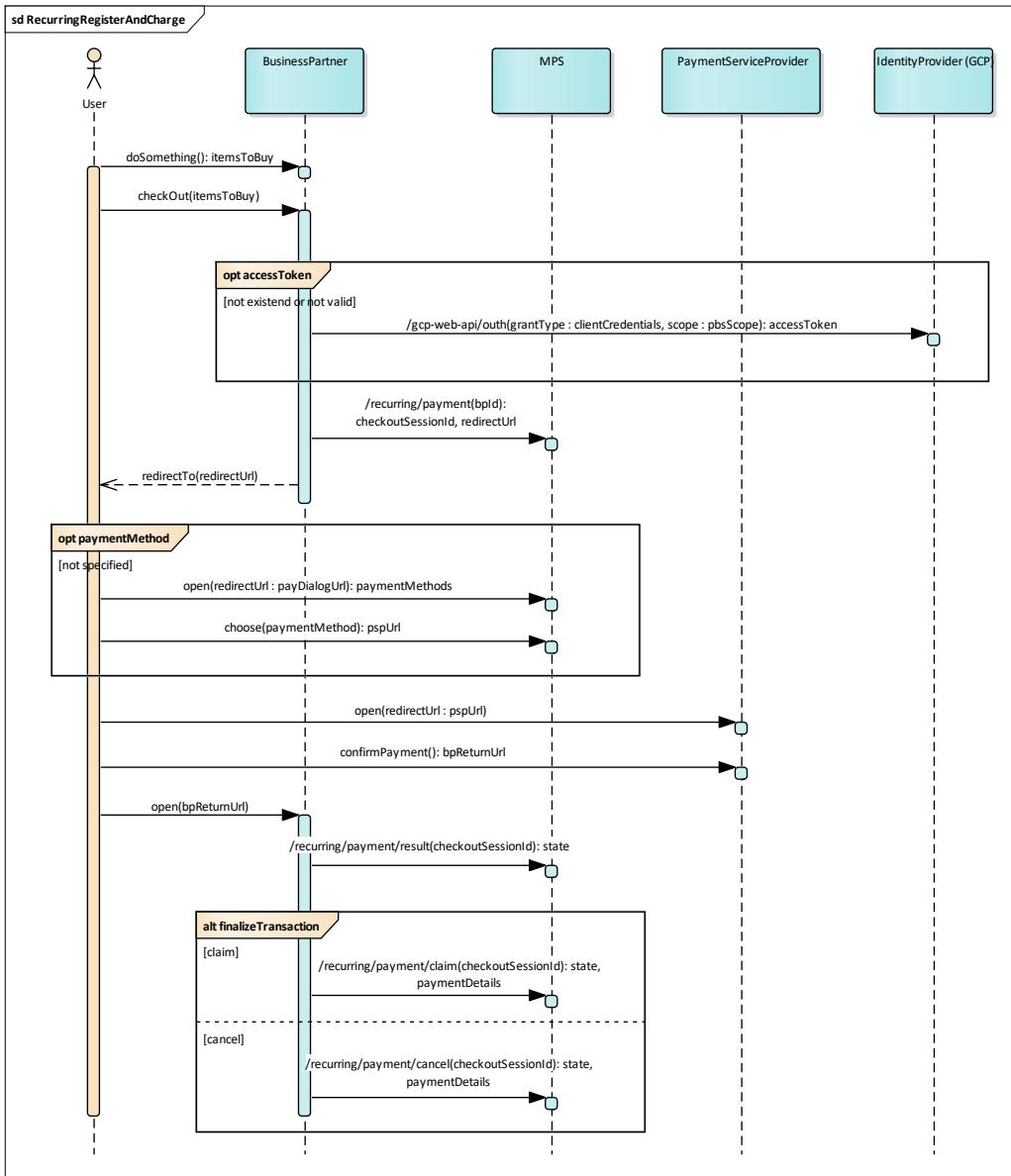
The customer will be redirected to the service, where the service can check the result of the authorisation. The service will be getting a payment handle, which he needs to collect recurring payments via the MPS.



4.2 recurring payment

With the workflow “recurring payment” the customer can authorise to collect recurring payments. In this workflow the service can charge the account of the customer for its first instalment.

The workflow is in most parts identically to the “recurring register” workflow with two modifications: Instead of the recurring register interface the service will use the recurring payment interface and provide an amount for the first charge, which the service can claim at the end of the workflow.

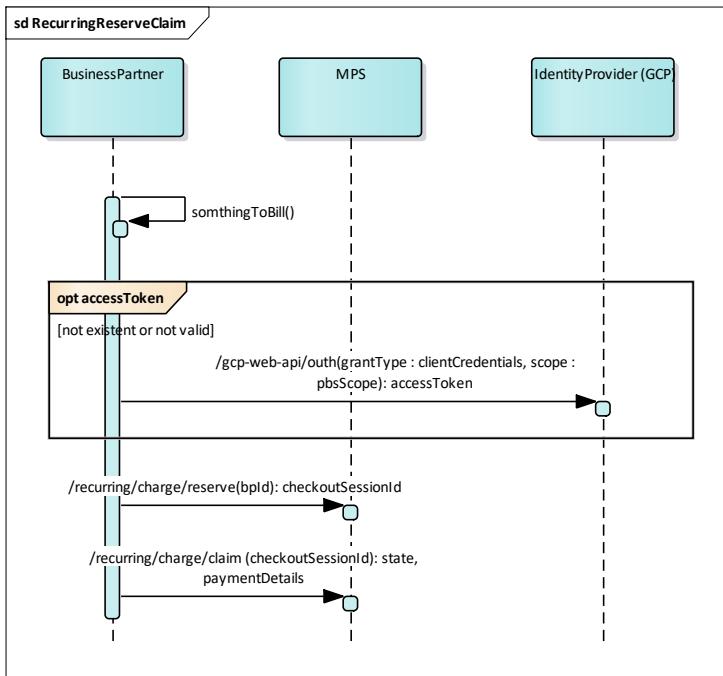


4.3 recurring charge (two phase)

The workflow “recurring charge (two phase)” is one of two workflows to collect recurring payments by a service. There is no interactions by the costumer necessary. With this workflow the service is responsible for the two phase payment process.

In the first phase the service reserves the amount, which he claims in the second phase.

Note: The service needs to provide a valid payment handle, which he gets during the registration process.

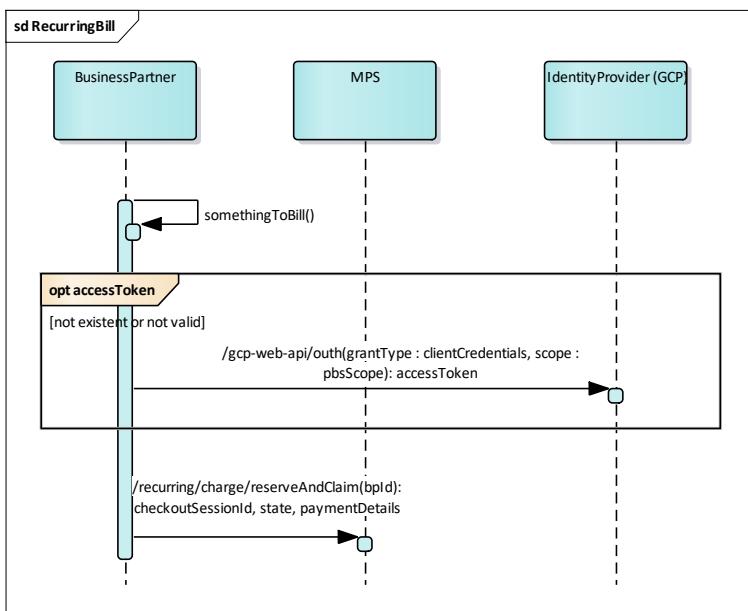


4.4 recurring charge (one phase)

The workflow “recurring charge (one phase)” is the second workflow to collect recurring payments by a service. There is no interactions by the costumer necessary. With this workflow MPS is responsible for the two phase payment process.

For the service is only one call to collect the payment.

Note: The service needs to provide a valid payment handle, which he gets during the registration process.

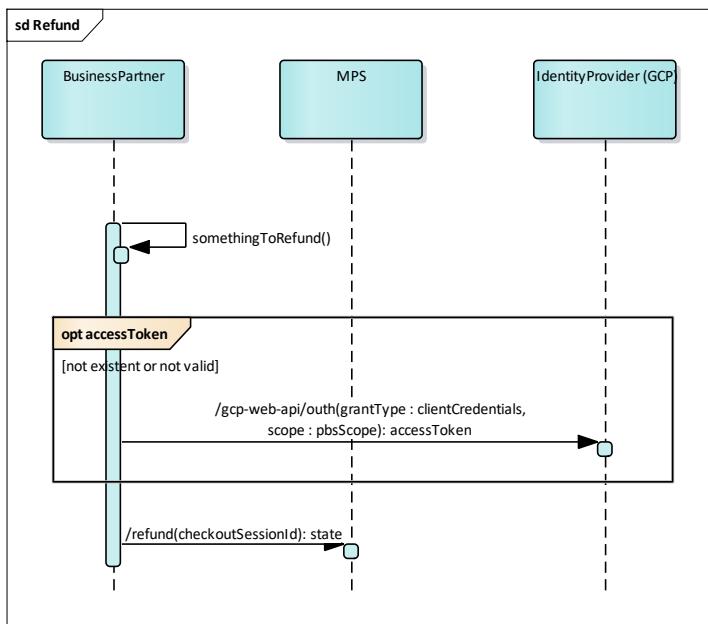


5 refund

MPS provides a refund interface for an easy and automatic way to refunding customers payment.

Note: The service needs to provide the payment transaction id (PayTID) to identify the transaction that needs to be refunded.

Note: Not all payment methods supports a refund with this interface. See chapter 1 for more details.



6 API documentation

6.1 onetime payment

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 /onetime/paymentmethods

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
grossAmount	number		1	For purchases with Miles&More use "XYL". gross amount of the product

response

field name	type	length	multiplicity	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 onetime payment process by calling the operation "reserve".

The service will get an URL, where the service needs to redirect the customer, so the customer can authorise the payment.

operation

POST /onetime/reserve

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 contraints 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.
currency	string	3	1	For more details see chapter 6.6. 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	deprecated 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.
excludedPaymentMethods	string		0..*	List of payment methods, that are excluded from the current purchase. Will be ignored, if no payment method is provided.
issueInputAssistance	boolean		0..1	Set to "true", if the service needs an input assistance of the selected payment method. Will be ignored, if a payment method is provided.
inputAssistances	input assistance		0..*	List of payment methods and their associated input assistance. For more details see chapter 6.6.
lineItems	line item		0..*	List of items in the basket. For more details see chapter 6.6.
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.
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.
preferredPaymentMethod	string	255	0..1	The preferred payment method will be selected in the payment selection screen. Will be ignored, if a payment method is provided.
returnUrl	string	4096	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.
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.
themeName	string	25	0..1	The specified theme will be used for the customization of the payment selection page. Needs to be set, if no payment method is set. For customizing the checkout page see chapter 9
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.

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
redirectUrl	string	4096	0..1	URL of the payment service partner, the user will be redirected to.
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	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)

6.1.3 result

The service can check the result of a reservation by calling the operation “result”.

The state can be “PENDING”, if the customer not complete the authorisation for the payment. If the authorisation has failed or the customer has abort the authorisation process, the service will get an error.

Operation

POST /onetime/result

request

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

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
400	INVALID_REQUEST	client errors (e.g. media type not supported)
400	TRANSACTION_TIMED_OUT	payment transaction expired
400	TRANSACTION_PENDING	payment reservation not authorized yet
400	USER_ABORTED	user aborted the payment
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)

6.1.4 claim

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

operation

POST /onetime/claim

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	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
inputAssistances	input assistance		0..*	list of the payment methods and the associated updated input assistances. For more details see chapter 6.6.
paymentMethod	string		1	used payment method
paymentMethodDetails	payment method details		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 /onetime/cancel

request

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

checkoutSessionId	string	40	1	Id of the checkout session.
response				
field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
inputAssistances	input assistance		0..*	list of the payment methods and the associated updated input assistances. For more details see chapter 6.6.
paymentMethod	string		1	used payment method
paymentMethodDetails	payment method details		0..1	Structure of the payment method details. Each payment method used to have a different one. For more see chapter 6.5.
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)		

6.1.6 session details

This service retrieves information to a session.

operation

POST /onetime/sessionDetails

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	1	Id of the checkout session.
includePaymentTransactionDetails	boolean		0..1	if true, the payment transaction information will be included standard: false

response

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.
creationDate	int64		1	creation time of the transaction format: UNIX timestamp
discountable	boolean		1	if true, the transaction is discountable
modificationDate	int64		1	last modification time of the transaction format: UNIX timestamp
paymentMethod	string	255	1	used payment method.
paymentTransactionId	string	40	0..1	Id of the payment transaction.
paymentTransactionDetails	transaction details		0..1	transaction details. For more information see chapter 6.6

returnUrl	string	4096	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. The URL needs to use a valid TLD. "localhost" and "127.0.0.1" are accepted.
request	onetime re-serve		0..1	original request
sessionId	string	40	1	Id of the checkout session.
sessionState	state	255	1	session state. For more details see chapter 6.7
sessionType	string	25	1	session type
tenant	string	25	1	tenant

error messages

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

6.2 recurring payment

6.2.1 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 /recurring/register/paymentmethods

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".

response

field name	type	length	multiplicity	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)

6.2.2 register

The service can request for a direct debit mandate by calling the operation "recurring register".

The service will get an URL, where the service needs to redirect the customer, so the customer can authorise the request of the direct debit mandate.

operation

MPS-IntegrationGuide Checkout API v18.docx

page 18/55

POST /recurring/register

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
customerInformation	customer information		0..1	additional information about the customer can be used for specific payment methods only
description	string	127	1	Description for the registration.
excludedPaymentMethods	string		0..*	List of payment methods, that are excluded from the current purchase.
locale	locale	10	0..1	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.
paymentMethod	string	255	0..1	The stated payment method will be used. If no payment method is stated, than the pay- ment selection page is shown to the user.
preferredPaymentMethod	string	255	0..1	The preferred payment method will be selected in the payment selection screen. Will be ignored, if a payment method is pro- vided.
returnUrl	string	4096	1	URL of the business partner, the user will be re- directed at the end of the payment process. Depending on the payment method (e.g. Pay- Pal) several parameters may be added to the URL. The URL needs to use a valid TLD. "localhost" and "127.0.0.1" are accepted.
themeName	string	25	0..1	The specified theme will be used for the custom- ization of the payment selection page. Needs to be set, if no payment method is set. For customizing the checkout page see chapter 9
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.

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
redirectUrl	string	4096	0..1	URL of the payment service partner, the user will be redirected to.
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)
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

```
request
POST /pbs-checkout-api/recurring/register HTTP/1.1
Host: pbs0.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
    "businessPartnerConfigId": "3",
    "currency": "EUR",
    "returnUrl": "http://pdp-idm.biz/dump.php",
    "description": "Test:0000 Erteilung einer Einzugsermächtigung für künftige Einkäufe.",
    "excludedPaymentMethods" : ["paysafecard"],
    "uiDetails" : {
        "binding_legal_terms" : "Binding Legal Terms"
    },
    "locale": "en"
}

response
{
    "checkoutSessionId": "CHK-ed1a8c80-c23a-429b-bd9b-139d22128498",
    "state": "INITIALIZED",
    "redirectUrl": "https://pbs0.toon.sul.t-online.de/pbs-checkout-ui/recurring/register/CHK-ed1a8c80-c23a-429b-bd9b-139d22128498?locale=en"
}
```

6.2.3 register result

The service can check the result of the request for a direct debit mandate by calling the operation “recurring register result”.

The state can be “PENDING”, if the customer not complete the authorisation for the payment. If the authorisation has failed or the customer has abort the authorisation process, the service will get an error.

operation

POST /recurring/register/result

request

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

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentHandle	string	4000	0..1	payment handle for recurring payment collections.
paymentMethod	string	255	0..1	only ASCII chars will be used
paymentMethodDetails	payment method details		0..1	selected payment method Structure of the payment method details. Each payment method used to have a different one. For more see chapter 6.5.
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_TIMED_OUT	payment transaction expired
400	TRANSACTION_INVALID_STATE	payment transaction is not in expected state
400	TRANSACTION_PENDING	payment reservation not authorized yet
400	USER_ABORTED	user aborted the payment
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
409	TRANSACTION_IN_PROGRESS	the current checkout session is already being processed
500	REGISTRATION_ERROR	registration process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

```

request
POST /pbs-checkout-api/recurring/register/result HTTP/1.1
Host: pbs0.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
  "checkoutSessionId": "CHK-ed1a8c80-c23a-429b-bd9b-139d22128498"
}

response
{
  "checkoutSessionId": "CHK-ed1a8c80-c23a-429b-bd9b-139d22128498",
  "state": "FINALIZED",
  "paymentTransactionId": "PRR-781e1204-d607-4fac-96f8-4cce032bc85",
  "paymentHandle": "Tk2JpefwSPiMoZiv6Oqp817Y...",
  "paymentMethodDetails": {
    "ccBrand": "VISA",
    "expiryDate": "201702",
    "maskedCreditCardNumber": "*****2222",
    "acquirer": "TESTAQ",
    "referenceNumber": "f1y7rusqxp62itn70u8lvt8ed",
    "accountQualifier": "VISA *****2222"
  },
  "paymentMethod": "computop"
}

```

6.2.4 register session details

This service retrieves information to a session.

operation

POST /recurring/register/sessionDetails

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	1	Id of the checkout session.
includePaymentTransactionDetails	boolean		0..1	if true, the payment transaction information will be included standard: false

response

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.
creationDate	int64		1	creation time of the transaction format: UNIX timestamp
discountable	boolean		1	if true, the transaction is discountable
modificationDate	int64		1	last modification time of the transaction format: UNIX timestamp
paymentMethod	string	255	1	used payment method.
paymentTransactionId	string	40	0..1	Id of the payment transaction.
paymentTransactionDetails	transaction details		0..1	transaction details. For more information see chapter 6.6
request	recurring register		0..1	original request

returnUrl	string	4096	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. The URL needs to use a valid TLD. "localhost" and "127.0.0.1" are accepted.
sessionId	string	40	1	Id of the checkout session.
sessionState	state	255	1	session state. For more details see chapter 6.7
sessionType	string	25	1	session type
tenant	string	25	1	tenant

error messages

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

sample

request

```
POST /pbs-checkout-api/recurring/register/sessionDetails HTTP/1.1
Host: pbs1.toon.sult-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
  "checkoutSessionId": "CHK-59b35b31-2aef-4541-ac0c-4e31c0935ad3",
  "includePaymentTransactionDetails": true
}
```

response

```
{
  "sessionId": "CHK-59b35b31-2aef-4541-ac0c-4e31c0935ad3",
  "sessionType": "RECURRING_REGISTER",
  "sessionState": "FINALIZED",
  "tenant": "congstar",
  "businessPartnerConfigId": "2715",
  "paymentTransactionId": "PRR-159e8628-6cf8-4548-9d85-771ad21f4082",
  "paymentTransactionDetails": {
    "customerInformation": {
      "billingAddress": {
        "firstName": "Hans",
        "lastName": "Mustermann",
        "addressLine1": "Ida-Rhodes-Str. 2",
        "addressLine2": "",
        "state": "Empty",
        "postalCode": "64295",
        "city": "Darmstadt",
        "countryCode": "DE"
      }
    }
  },
  "paymentMethod": "paypal",
  "returnUrl": "http://www.congstar.de",
  "discountable": false,
  "request": {
    "businessPartnerConfigId": "2715",
    "returnUrl": "http://www.congstar.de",
    "description": "Test:0000 Erteilung einer Einzugsermächtigung",
    "currency": "EUR",
    "excludedPaymentMethods": [],
    "uiDetails": {},
    "themeName": "default",
    "paymentMethod": "paypal"
  },
  "modificationDate": 1579253920000,
  "creationDate": 1579253684000
}
```

6.2.5 payment

The service can start the payment process including a request for a direct debit mandate by calling the operation “recurring payment”.

The service will get an URL, where the service needs to redirect the customer, so the customer can authorise the request of the direct debit mandate.

operation

POST /recurring/payment

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
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 recurring payment collection.
excludedPaymentMethods	string array		0..*	List of payment methods, that are excluded from the current purchase.
lineItems	line item		0..*	List of items in the basket. For more details see chapter 6.6.
locale	locale	10	0..1	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.
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.
preferredPaymentMethod	string	255	0..1	The preferred payment method will be selected in the payment selection screen. Will be ignored, if a payment method is provided.
returnUrl	string	4096	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. The URL needs to use a valid TLD. “localhost” and “127.0.0.1” are accepted.
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.
themeName	string	25	0..1	The specified theme will be used for the customization of the payment selection page. Needs to be set, if no payment method is set. For customizing the checkout page see chapter 9
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.

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
redirectUrl	string	4096	0..1	URL of the payment service partner, the user will be redirected to.
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
403	TENANT_MISMATCH	the tenant of the payment transaction doesn't match the one of the settlement
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
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
500	REGISTRATION_ERROR	registration process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

request

```
POST /pbs-checkout-api/recurring/payment HTTP/1.1
Host: pbs1.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
  "businessPartnerConfigId": "2",
  "returnUrl": "http://pdp-idm.biz/dump.php",
  "description": "TEST:0000 Recurring Register mit Einzug",
  "currency": "EUR",
  "excludedPaymentMethods": ["paysafecard"],
  "uiDetails": {
    "binding_legal_terms": "Binding Legal Terms"
  },
  "locale": "en",
  "lineItems": [
    {
      "name": "LineItem_1",
      "description": "Description_1",
      "grossAmount": 0.95,
      "taxRate": 7,
      "quantity": 1,
      "uiDetails": {}
    }
  ],
  "settlementData": {
    "settlementConfigurationId": "1",
    "transactionSettlementData": {
      "country": "DE",
      "targetSystemId": "4711",
      "financialClientId": "4712",
      "salesChannelId": "4713"
    },
    "basketItemsSettlementData": []
  }
}
```

```

        "goodsId": "goodsId_1",
        "goodsType": "digital",
        "goodsPartnerImpactCategory": "Impact_Category_1",
        "subscriptionInfo": "subscription_info_1",
        "voucherId": "",
        "discountReason": ""
    }]
}
}

response
{
  "checkoutSessionId": "CHK-755bb127-a685-4b0a-aae3-e629b6c6b834",
  "state": "INITIALIZED",
  "redirectUrl": "https://pbs1.toon.sul.t-online.de/pbs-checkout-ui/recurring/payment/CHK-755bb127-a685-4b0a-aae3-e629b6c6b834?locale=en"
}

```

6.2.6 payment result

The service can check the result of the payment request including the request for a direct debit mandate by calling the operation “recurring payment result”.

The state can be “PENDING”, if the customer not complete the authorisation for the payment. If the authorisation has failed or the customer has abort the authorisation process, the service will get an error.

operation

POST /recurring/payment/result

request

field name	type	length	multipli-city	comments
checkoutSessionId	string	40	1	Id of the checkout session.

response

field name	type	length	multipli-city	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
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 not in expected state
400	TRANSACTION_PENDING	payment reservation not authorized yet
400	USER_ABORTED	user aborted the payment
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	INVALID_PAYMENTHANDLE	payment handle is invalid

sample call

```

request
POST /pbs-checkout-api/recurring/payment/result HTTP/1.1
Host: pbs1.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
  "checkoutSessionId": "CHK-755bb127-a685-4b0a-aae3-e629b6c6b834"
}

```

```
response
{
    "checkoutSessionId": "CHK-755bb127-a685-4b0a-aae3-e629b6c6b834",
    "state": "RESERVED"
}
```

6.2.7 payment claim

The service can collect the reserved amount by calling the operation “recurring payment claim”.

operation

POST /recurring/payment/claim

request

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

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
effectiveTotal	number		0..1	the collected amount
paymentHandle	string	4000	0..1	payment handle for recurring payment collections. only ASCII chars will be used
paymentMethod	string	255	0..1	selected payment method
paymentMethodDetails	payment method details		0..1	Structure of the payment method details. Each payment method used to have a different one. For more see chapter 6.5.
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
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	CLAIM_ERROR	claim process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

```
request
POST /pbs-checkout-api/recurring/payment/claim HTTP/1.1
Host: pbs1.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
    "checkoutSessionId": "CHK-755bb127-a685-4b0a-aae3-e629b6c6b834"
}

response
{
    "checkoutSessionId": "CHK-755bb127-a685-4b0a-aae3-e629b6c6b834",
    "state": "CLAIMED",
    "settlementTransactionId": "SRC-8b73673a-978b-400c-9e30-a48b19bc3ffd",
    "paymentTransactionId": "PRC-8b73673a-978b-400c-9e30-a48b19bc3ffd",
    "paymentHandle": "UM6sGwVY1D5MiJja4WYWJgJ5unxCUUMFr...",
    "paymentMethodDetails": {
        "payer": "buyer_1328703229_pre@external.telekom.de",
    }
}
```

```

    "payerId": "GRAS92QNR2VFU",
    "accountQualifier": "buyer_1328703229_pre@external.telekom.de"
},
"paymentMethod": "paypal"
}

```

6.2.8 payment cancel

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

operation

POST /recurring/payment/cancel

request

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

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentHandle	string	4000	0..1	payment handle for recurring payment collections.
paymentMethod	string	255	0..1	only ASCII chars will be used
paymentMethodDetails	payment method details		0..1	selected payment method Structure of the payment method details. Each payment method used to have a different one. For more see chapter 6.5.
paymentTransactionAlreadyCancelled	boolean		0..1	If true, the 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.
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	FAILED_VALIDATION	request validation failed
400	TRANSACTION_INVALID_STATE	payment transaction has an invalid state
404	SESSION_NOT_FOUND	checkout session not found
404	TRANSACTION_NOT_FOUND	payment transaction 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)

sample call

```

request
POST /pbs-checkout-api/recurring/payment/cancel HTTP/1.1
Host: pbs1.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
  "checkoutSessionId": "CHK-755bb127-a685-4b0a-aae3-e629b6c6b834"
}

```

response

```

{
  "checkoutSessionId": "CHK-2de1ef1d-6839-4baa-b2e3-2559af5934d9",
  "state": "CANCELLED",
  "paymentTransactionId": "PRC-9c14b7ce-3327-4deb-bc4a-79fcba3fecc",
  "paymentHandle": "siUbxibMwGcq...",
  "paymentMethodDetails": {
    "payer": "buyer_1328703229_pre@external.telekom.de",
    "payerId": "GRAS92QNR2VFU",
  }
}

```

```

        "accountQualifier": "buyer_1328703229_pre@external.telekom.de"
    },
    "reservedAmountReleased": true,
    "paymentMethod": "paypal"
}

```

6.2.9 payment session details

This service retrieves information to a session.

operation

POST /recurring/payment/sessionDetails

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	1	Id of the checkout session.
includePaymentTransactionDetails	boolean		0..1	if true, the payment transaction information will be included standard: false

response

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.
creationDate	int64		1	creation time of the transaction format: UNIX timestamp
discountable	boolean		1	if true, the transaction is discountable
modificationDate	int64		1	last modification time of the transaction format: UNIX timestamp
paymentMethod	string	255	1	used payment method.
paymentTransactionId	string	40	0..1	Id of the payment transaction.
paymentTransactionDetails	transaction details		0..1	transaction details. For more information see chapter 6.6
request	recurring payment		0..1	original request
sessionId	string	40	1	Id of the checkout session.
sessionState	state	255	1	session state. For more details see chapter 6.7
sessionType	string	25	1	session type
tenant	string	25	1	tenant

error messages

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

6.2.10 charge reserve

The service can reserve an amount for recurring payment (subscription) with a direct debit mandate by calling the operation “recurring charge reserve”.

operation

POST /recurring/charge/reserve

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
customerInformation	customer information		0..1	additional information about the customer can be used for specific payment methods only
description	string	127	1	description for the charge
lineItems	line item		0..*	List of items in the basket. For more details see chapter 6.6.
paymentHandle	string	4000	1	payment handle to use for the charge only ASCII chars will be used
paymentMethod	string	255	0..1	payment method to use
settlementData	ipdr settlement		0..1	settlement information for the purchase

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
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	REVOKED_PAYMENTHANDLE	customer revoked payment handle
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
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
500	INVALID_PAYMENTHANDLE	payment handle is invalid
500	RESERVATION_ERROR	reservation process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

```

request
POST /pbs-checkout-api/recurring/charge/reserve HTTP/1.1
Host: pbs1.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
  "businessPartnerConfigId": "2",
  "currency": "EUR",
  "description": "TEST:0000 Recurring Charge über Checkout",
  "paymentMethod": "paypal",
  "paymentHandle" : "VTXruQw3fgaMVd...",

  "lineItems": [
    {
      "name": "name LineLineItem",
      "description": "description LineItem",
      "grossAmount": 1,
      "taxRate": 19,
      "quantity": 2
    }
  ],
  "settlementData": {
    "settlementConfigurationId": "1",

    "transactionSettlementData": {
      "country": "DE",
      "targetSystemId": "4711",
    }
  }
}

```

```

        "financialClientId": "4712",
        "salesChannelId": "4713"
    },
    "basketItemsSettlementData": [
        {
            "goodsId": "goodsId_LineItem_1",
            "goodsType": "digital",
            "goodsPartnerImpactCategory": "GPIC_1",
            "subscriptionInfo": "subinfo_1",
            "voucherId": "",
            "discountReason": ""
        }
    ]
}
}

response
{
    "checkoutSessionId": "CHK-782f56f0-eae6-4270-8b7d-953abfc61c60",
    "state": "RESERVED"
}

```

6.2.11 charge claim

The service can claim the reserved amount for a recurring payment (subscription) by calling the operation “recurring charge claim”.

operation

POST /recurring/charge/claim

request

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

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
effectiveTotal	number		0..1	the collected amount
paymentHandle	string	4000	0..1	used, updated payment handle only ASCII chars will be used
paymentMethod	string	255	0..1	used payment method
paymentMethodDetails	payment method de- tails		0..1	Structure of the payment method details. Each payment method used to have a different one. For more see chapter 6.5.
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
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	CLAIM_ERROR	claim process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

```

request
POST /pbs-checkout-api/recurring/charge/claim HTTP/1.1
Host: pbs1.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache
{

```

```

    "checkoutSessionId": "CHK-782f56f0-eae6-4270-8b7d-953abfc61c60"
}

response
{
  "checkoutSessionId": "CHK-782f56f0-eae6-4270-8b7d-953abfc61c60",
  "state": "CLAIMED",
  "settlementTransactionId": "SRC-748fc27c-e091-45fa-a572-9451e0f3f605",
  "paymentTransactionId": "PRC-748fc27c-e091-45fa-a572-9451e0f3f605",
  "paymentHandle": "VTXruQw3fgaMV...",
  "effectiveTotal": 1.99,
  "paymentMethod": "computop"
}

```

6.2.12 charge cancel

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

operation

POST /recurring/charge/cancel

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	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
404	SESSION_NOT_FOUND	checkout session not found
404	TRANSACTION_NOT_FOUND	payment transaction 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)

6.2.13 charge reserveAndClaim

The service can reserve and claim an amount for a recurring payment (subscription) with a direct debit mandate by calling the operation “recurring charge reserveAndClaim”.

operation

POST /recurring/charge/reserveAndClaim

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
customerInformation	customer information		0..1	additional information about the customer can be used for specific payment methods only

description	string	127	1	description for the charge
lineItems	line item		0..*	List of items in the basket. For more details see chapter 6.6.
paymentHandle	string	4000	1	payment handle to use for the charge only ASCII chars will be used
paymentMethod	string	255	0..1	payment method to use
settlementData	ipdr settlement		0..1	settlement information for the purchase

response

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	0..1	Id of the checkout session.
paymentHandle	string	4000	0..1	updated payment handle only ASCII chars will be used
paymentMethod	string	255	0..1	used payment method
paymentMethodDetails	payment method details		0..1	Structure of the payment method details. Each payment method used to have a different one. For more see chapter 6.5.
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

All error messages from "charge reserve" and "charge claim" can occur. Please see there.

sample call

...

6.2.14 charge session details

This service retrieves information to a session.

operation

POST /recurring/charge/sessionDetails

request

field name	type	length	multiplicity	comments
checkoutSessionId	string	40	1	Id of the checkout session.
includePaymentTransactionDetails	boolean		0..1	if true, the payment transaction information will be included standard: false

response

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.
creationDate	int64		1	creation time of the transaction format: UNIX timestamp
discountable	boolean		1	if true, the transaction is discountable
modificationDate	int64		1	last modification time of the transaction format: UNIX timestamp
paymentMethod	string	255	1	used payment method.
paymentTransactionId	string	40	0..1	Id of the payment transaction.
paymentTransactionDetails	transaction details		0..1	transaction details. For more information see chapter 6.6
request	recurring charge		0..1	original request
sessionId	string	40	1	Id of the checkout session.
sessionState	state	255	1	session state. For more details see chapter 6.7
sessionType	string	25	1	session type

tenant	string	25	1	tenant
--------	--------	----	---	--------

error messages

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

sample

```

request
POST /pbs-checkout-api/recurring/charge/sessionDetails HTTP/1.1
Host: pbs1.toon.sult-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache

{
  "checkoutSessionId": "CHK-59b35b31-2aef-4541-ac0c-4e31c0935ad3",
  "includePaymentTransactionDetails": true
}

response
{
  "sessionId": "CHK-d536bd20-9c48-4340-83b2-ac7a04a73566",
  "sessionType": "RECURRING_CHARGE",
  "sessionState": "CLAIMED",
  "tenant": "congstar",
  "businessPartnerConfigId": "2715",
  "paymentTransactionId": "PRC-542c855f-f611-438f-b03f-aa4b59b643f5",
  "paymentTransactionDetails": {
    "paymentServiceTransactionId": "66557363Y41079317"
  },
  "paymentMethod": "paypal",
  "discountable": false,
  "request": {
    "businessPartnerConfigId": "2715",
    "currency": "EUR",
    "lineItems": [
      {
        "name": "name LineLineItem",
        "description": "description LineItem",
        "grossAmount": 1,
        "taxRate": 19,
        "quantity": 2
      }
    ],
    "description": "Recurring Charge über Checkout",
    "paymentMethod": "paypal",
    "paymentHandle": "sKrWceu51MjxKmYBymP..."
  },
  "modificationDate": 1579256682000,
  "creationDate": 1579256667000
}

```

6.3 refund

6.3.1 refund on basket level

The service can refund a transaction by calling the operation “refund”.

The refund can be partial or full. In both cases the service cannot refund more than he has collected once.

operation

POST /refund

request

field name	type	length	multiplicity	comments
paymentTransactionId	string	40	0..1	Id of the payment transaction to be refunded
checkoutSessionId	string	40	0..1	Id of the checkout session.
amount	number		0..1	amount to be refunded if set, the parameter „fullRefund“ needs to be „false“ or nonexistent

fullRefund	boolean	0..1	use for a full refund if set to „true“, the parameter „amount“ needs to be inexistent
description	string	0..1	description of the refund

response

field name	type	length	multiplicity	comments
paymentTransactionId	string	40	1	Id of the refund transaction
refundedAmount	number		0..1	refunded amount
state	state	255	0..1	session state. For more details see chapter 6.7
transactionAlreadyRefunded	boolean		0..1	if true, the transaction was already refunded

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 is not claimed yet
400	SETTLEMENT_TRANSACTION_IN_INVALID_STATE	settlement transaction in invalid state
404	SESSION_NOT_FOUND	checkout session not found
404	TRANSACTION_NOT_FOUND	payment transaction not found
404	SETTLEMENT TRANSACTION NOT FOUND	settlement transaction not found
409	TRANSACTION_IN_PROGRESS	the current checkout session is already being processed
500	REFUND_ERROR	refund process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

request

```
POST /pbs-checkout-api/refund HTTP/1.1
Host: pbs1.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache
{
  "paymentTransactionId" : "PRC-7842f170-333a-4743-b109-36f47c544764",
  "fullRefund" : true}
```

response

```
{
  "state": "REFUNDED",
  "paymentTransactionId": "PRF-31a29d98-7f44-47fc-8d19-d5bbc8ffbf08",
  "checkoutSessionId": "CHK-4dc88921-aa42-448f-bcbe-e6467cecb96c",
  "refundedAmount": 3
}
```

6.3.2 refund on item level

The service can refund a transaction by calling the operation “refund”.

The refund can be partial or full. In both cases the service cannot refund more than he has collected once.

The refund is based on the item index. Only one item can refunded per transactions. If more than one item is to refund, then more than one refund on item level is needed.

operation

POST /refund/items/{itemIndex}

request

field name	type	length	multiplicity	comments
paymentTransactionId	string	40	0..1	Id of the payment transaction to be refunded
checkoutSessionId	string	40	0..1	Id of the checkout session.
amount	number		0..1	amount to be refunded if set, the parameter „fullRefund“ needs to be „false“ or inexistent

fullRefund	boolean	0..1	use for a full refund if set to „true“, the parameter „amount“ needs to be inexistent
description	string	0..1	description of the refund

response

field name	type	length	multiplicity	comments
paymentTransactionId	string	40	1	Id of the refund transaction
refundedAmount	number		0..1	refunded amount
state	state	255	0..1	session state. For more details see chapter 6.7
itemAlreadyRefunded	boolean		0..1	if true, the item was already refunded

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 is not claimed yet
400	SETTLEMENT_TRANSACTION_IN_INVALID_STATE	settlement transaction in invalid state
404	SESSION_NOT_FOUND	checkout session not found
404	TRANSACTION_NOT_FOUND	payment transaction not found
404	SETTLEMENT TRANSACTION NOT FOUND	settlement transaction not found
409	TRANSACTION_IN_PROGRESS	the current checkout session is already being processed
500	REFUND_ERROR	refund process failed
500	TECHNICAL_ERROR	technical error (e.g. while talking to backend system)

sample call

request

```
POST /pbs-checkout-api/refund/items/10 HTTP/1.1
Host: pbs1.toon.sul.t-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Content-Type: application/json
Cache-Control: no-cache
{
  "paymentTransactionId" : "PRC-7842f170-333a-4743-b109-36f47c544764",
  "fullRefund" : true}
```

response

```
{
  "state": "REFUNDED",
  "paymentTransactionId": "PRF-31a29d98-7f44-47fc-8d19-d5bbc8ffbf08",
  "checkoutSessionId": "CHK-4dc88921-aa42-448f-bcbe-e6467cecb96c",
  "refundedAmount": 3
}
```

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 if type a string, then length is 512

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: alipay_wirecard

Alipay doesn't support any additional information, we could deliver with the payment method details. So the details are empty.

6.5.2 payment method: alipay_computop

field name	type	length	multiplicity	comments
referenceNumber	string	64	1	reference number of the acquirer, needed for match with EPA files

6.5.3 payment method: amazon

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "email"
email	string		1	e-mail address of the Amazon pay user
userId	string		1	Id of the Amazon pay user

6.5.4 payment method: applepay_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 format: YYYYMM
maskedCreditCardNumber	string	20	1	masked credit card number, only the last 4 digits are shown
maskedDeviceAccountNumber	string	4	1	the number of the apple device, only the last 4 digits are shown
referenceNumber	string	64	1	reference number of the acquirer, needed for match with EPA files

6.5.5 payment method: banktransfer_sofort

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "iban"
bic	string		1	Bank Identifier Code (also known as ISO-9362 or SWIFT code)
				an Id code for financial and other institutions
countryCode	string		1	country code Based on ISO 3166-1 alpha-2
holder	string		1	name of the holder of the account
iban	string		1	International Bank Account Number Based on ISO-13616:2007
				masked, only the last 3 digits are shown

6.5.6 payment method: bcinvoice_telekom

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "providerContractId"
companyName	string		1	company name
partnerId	string		1	business partner id (customer number)
providerContractId	string		1	contract id of the business customer in the T-Systems SAP system

6.5.7 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
cardHolderName	string	255	1	name on credit card
cardHolderEmail	string	255	1	inserted email address
ccBrand	string	255	1	credit card brand
expiryDate	string	10	1	expiry date of credit card format: YYYYMM
maskedCreditCardNumber	string	20	1	masked credit card number, the first 6 digits (BIN range) and the last 4 digits are shown e.g. 422222XXXXXX2222
pseudoCardNumber	string	20	1	pseudo card number
referenceNumber	string	64	1	reference number of the acquirer, needed for match with EPA files

6.5.8 payment method: directdebit_mrnexnet

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "paymentAccountNumber"
customerReference	string		1	
paymentAccountNumber	string		1	
paymentInstitutNumber	string		1	
paymentReference	string		1	

6.5.9 payment method: eps

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "paymentAccountNumber"
customerAccountId	string		1	IBAN
customerFinancialInstitutionBic	string		1	BIC
customerName	string		1	account owner

6.5.10 payment method: loyalty_milesandmore

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "maskedCardNumber"
maskedCardNumber	string		1	

6.5.11 payment method: mobilebill, mobilebill_mbe4

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "msisdn"
msisdn	string		1	Mobile Station International Subscriber Directory Number
operatorId	string		1	masked, only the last 2 digits are shown

6.5.12 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

6.5.13 payment method: paysafecard

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	empty
serialNumbers	string		1	

6.5.14 payment method: telephonecard

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	empty
missingAmount	string		1	collected amount
originalBalance	string		1	original balance
remainingBalance	string		1	new balance

6.5.15 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

6.5.16 payment method: fixedlinebill_telekom

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "displayUsername"
displayUsername	string		1	display name of the customer
guid	string		1	global user id of the customer
ip	string		1	IP address

6.5.17 payment method: mobilebill_sdp, mobilebill_xtc

field name	type	length	multiplicity	comments
accountQualifier	string	1024	1	identically to "msisdn"
anonymousid	string		1	
msisdn	string		1	Mobile Station International Subscriber Directory Number masked, only the last 3 digits are shown

6.6 other structures

6.6.1 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

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

6.6.2 bc invoice

The element “bc invoice” contains additional product information for using the payment method “bcinvoice_telekom”.

field name	type	length	multiplicity	comments
sapProductId	string		1	id of the product in the T-Systems SAP system

6.6.3 contact data

The element “contact data” contains additional information about the customer for using the payment method “creditcard_computop”.

field name	type	length	multiplicity	comments
email	string		0..1	email address of the customer Will be used to fill out the input field for the email address in the new credit card UI.

6.6.4 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).

6.6.5 couponing data

For more information see documentation of DCS.

field name	type	length	multiplicity	comments
client	string		1	

transactionConstraints	couponing constraint	0..*
------------------------	-------------------------	------

6.6.6 customer information

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

field name	type	length	multiplicity	comments
billingAddress	address		0..1	billing address of the customer only used in responses for payment methods “paypal”
contactData	contact data		0..1	contact data of the customer only used in requests for payment methods “creditcard_computop”
externalIdentifier	string		0..1	identifier of the customer
shippingAddress	address		0..1	only for payment method „paysafecard“ shipping address of the customer for real products only used in requests for payment methods “creditcard_computop” and “paypal”

6.6.7 debtor account data

Additional information about the debtor.

field name	type	length	multiplicity	comments
accountType	string	2	1	type of the account possible values: <ul style="list-style-type: none">• MF (from CARMEN)• FN (from CRM-T)• MH (from Magenta Haus)
accountNumber	string		1	account number

6.6.8 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	int32		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
originalAmount	number		0..1	original amount of the product in case of a discount If set, then voucherId and discountReason needs to be set too.
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
invoiceData	invoice data		0..1	additional information for the invoice only used for bc_invoice
debtorAccountData	debtor ac- count data		0..1	additional information about the debtor account
salesChannelId	string		0..1	Id of the sales channel

servicePeriod	service period	0..1	start and end date of the service only used for bc_invoice
subscriptionInfo	subscription info	0..1	information for the subscription
voucherId	string	0..1	Id of the voucher the user used for the discount Cannot be empty, if originalAmount is set.
discountReason	string	0..1	reason for the discount Cannot be empty, if originalAmount is set.

6.6.9 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

6.6.10 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

6.6.11 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

6.6.12 invoice data

field name	type	length	multiplicity	comments
amount	number		1	gross amount of one unit
amountTotal	number		1	net amount of all units
quantity	number		1	quantity of the product

sample

```
"invoiceData": {
  "quantity": 3.5,
  "amount": 4.93,
  "amountTotal": 14.504
}
```

6.6.13 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	int32		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		0..1	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
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  
}],
```

output in the payment selection screen

Bestellübersicht	Menge	Preis in EUR
Name LineItem 1 Description LineItem_1	1x	0,22

6.6.14 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

6.6.15 payment service item data

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

field name	type	length	multiplicity	comments
bcinvoice_telekom	bc invoice		1	additional product information for using business customer invoice
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

6.6.16 service period

This element contains additional information for the service period of the product.

field name	type	length	multiplicity	comments
periodStart	int64		1	start time of the period format: UNIX timestamp
periodEnd	int64		1	end time of the period format: UNIX timestamp For onetime payment “periodEnd” must be equal to “periodStart”.

6.6.17 subscription info

This text field contains information about the subscription.

element	type	length	multiplicity	comments
contractId	string	1		contract id
billingSequenceNumber	string	1		individual sequence number of the current billing for each "contractId"
billingTimePlanned	date	1		date time of the current billing format: YYYY-MM-DDThh:mm:ssZ
billingIntervall	int	1		invoicing period in days (28, 29, 30 or 31)
minContractTerm	int	0..1		minimum duration of the contract
minContractTermType	string	0..1		minimum duration of the contract type "M" for month, "Y" for year
feeType	string	1		always "A"
materialnummer	string	0..1		"materialnummer" and "konditionsId" needs to be coordinated with PAS
konditionsId	string	0..1		mandatory, if PAS is used; otherwise optional

sample (listening formatted for better reading)

```
contractId=9876543210
&billingSequenceNumber=7
&billingTimePlanned=2021-01-18T10:07:32Z
&billingInterval=30
&minContractTerm=6
&minContractTermType=M
&feeType=A
&materialnummer=TBS
&konditionsid=123456789
```

6.6.18 transaction details

This element contains information about the payment transaction of a checkout session.

field name	type	length	multiplicity	comments
customerInformation	customer information	0..1		additional information about the customer will be used for specific payment methods only
paymentServiceTransactionId	string	0..1		transaction id from the payment service provider

sample

```
"paymentTransactionDetails": {
  "customerInformation": {
    "billingAddress": {
      "firstName": "Hans",
      "lastName": "Mustermann",
      "addressLine1": "Ida-Rhodes-Str. 2",
      "addressLine2": "",
      "state": "Empty",
      "postalCode": "64295",
      "city": "Darmstadt",
      "countryCode": "DE"
    }
  }
}
```

6.6.19 ui details

If you are using own templates hosted by MPS, you can add dynamic elements in your templates. To fill these elements you can use this simple mapping structure of <object, string>.

You can use this feature for "Buttongesetzttexte" as a sample.

sample

```
"uiDetails" : {
  "sections" : [
    { "section" : "Nutzungsrechte",
      "sectionLines" : [
```

```

        { "line" : "48 Stunden..." },
        { "line" : "1 Stream gleichzeitig..." }
    ],
},
{ "section" : "Nutzungsvoraussetzungen",
"sectionLines" : [
    { "line" : "Windows 7..." },
    { "line" : "Firefox..." },
    { "line" : "gute Augen..." }
]
}
]
}

```

output in the payment selection screen

```

Nutzungsrechte:
•48 Stunden...
•1 Stream gleichzeitig...

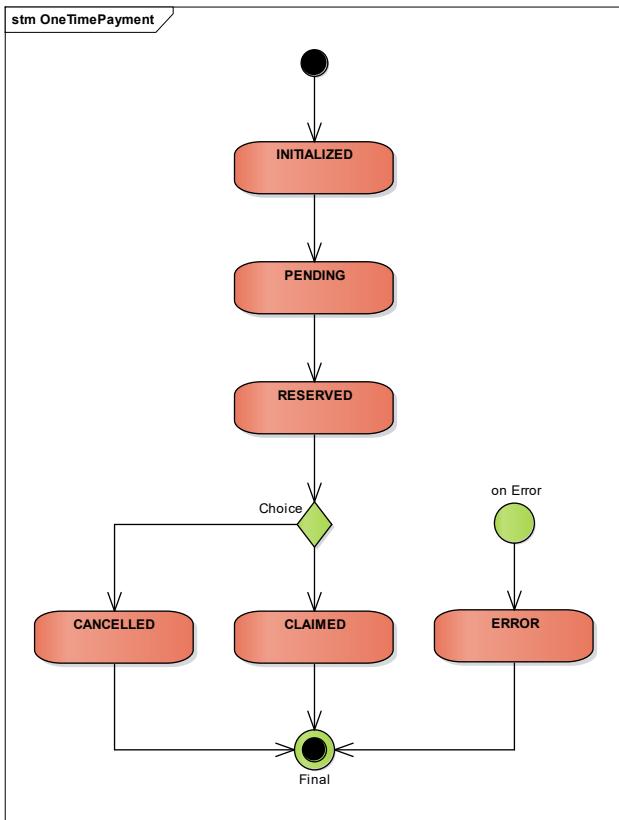
Nutzungsvoraussetzungen:
•Windows 7...
•Firefox...
•gute Augen...

```

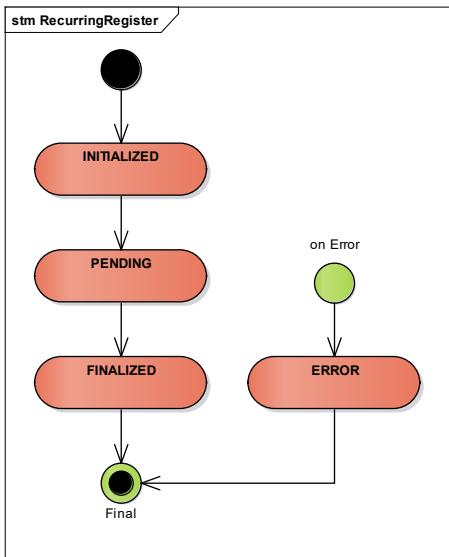
6.7 states and transitions

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.

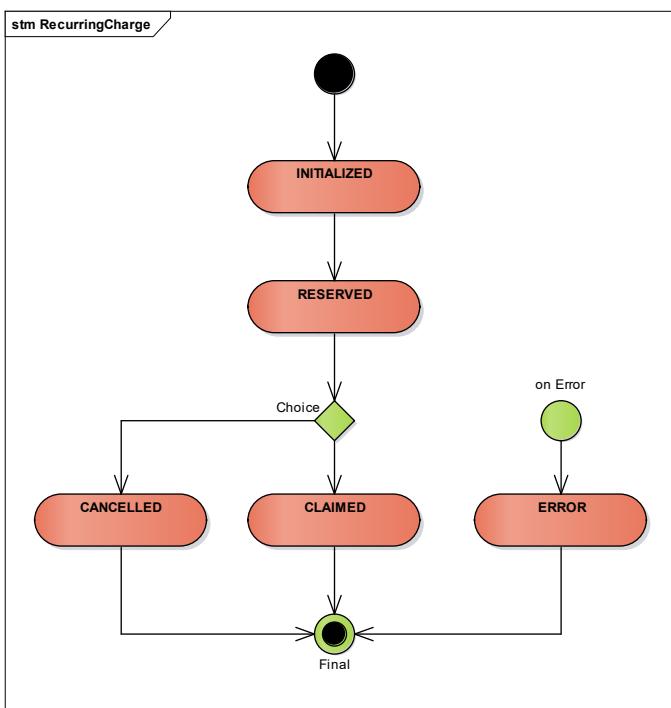
state model for one time payments / direct payments / register and charge transactions



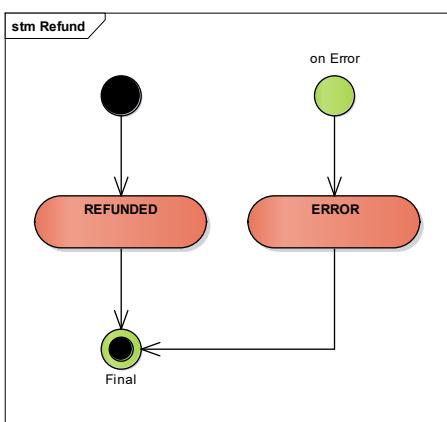
state model for recurring register transactions



state model for recurring bill / recurring charge and claim transactions



state model for refund transactions



7 theme customization

The service can choose if he wants to use his own checkout page or the one from MPS. For the later MPS allows the service to customize this page.

On the checkout page the customer sees his basket and can choose which method he wants to use for the payment. As an example the checkout page from EntertainTV mobil.

Film Kaufen

name LineLineitem_1	1,00 EUR
<small><input checked="" type="checkbox"/> description Lineitem_1</small>	
<hr/>	
Bezahlverfahren	
 Telekom Rechnung ✓ <input type="checkbox"/> Dieses Bezahlverfahren speichern	 Kreditkarte  Telefonkarte Comfort
 PayPal	 Bezahlen per Handy
<hr/>	
Coupon Rabatt Code eingeben	
Nutzungsrechte - 48 Stunden... - Rechte für Versionen: SD, HD	
Nutzungsvoraussetzungen - Windows 7... - Firefox... - gute Augen...	
<hr/> Gesamtsumme 1,00 EUR <hr/>	
Abbrechen Zahlungspflichtig bestellen	

7.1 API documentation

MPS supports the service to customize the checkout page in a self-service approach with an API.

7.1.1 Check current themes

With this operation the service can get all current available template of his tenant.

Operation

GET /configuration/themes

Request

No additional request parameters.

Response

The response contains a list with following fields:

field name	type	length	multiplicity	comments
name	string	25	1	name of the theme Use this name in the field "themeName" of the operations "onetime reserve", "recurring register" and "recurring registration and payment"
tenant	string	25	1	name of the tenant
creationDate	int64		1	creation time of the template format: UNIX timestamp
modificationDate	int64		1	modification time of the template format: UNIX timestamp

sample call

Request

```
GET /pbs-checkout-api/configuration/themes HTTP/1.1
Host: api-payment0.ver.sult-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Cache-Control: no-cache
```

Response

```
[{
  "name": "default",
  "tenant": "entertainMobil",
  "modificationDate": 1516269662000,
  "creationDate": 1516269662000
}, {
  "name": "etvmobil",
  "tenant": "entertainMobil",
  "modificationDate": 1524060375000,
  "creationDate": 1524060375000
}]
```

7.1.2 Upload a new theme

With this operation the service can upload a new template for his tenant.

Operation

POST /configuration/themes

Request

field name	type	length	multiplicity	comments
theme	binary		1	ZIP archive of the new theme
name	string	25	1	name of the theme Use this name in the field "themeName" of the operations "onetime reserve", "recurring register" and "recurring registration and payment"

Response

field name	type	length	multiplicity	comments
name	string	25	1	name of the theme Use this name in the field "themeName" of the operations "onetime reserve", "recurring register" and "recurring registration and payment"
tenant	string	25	1	name of the tenant
creationDate	int64		1	creation time of the template format: UNIX timestamp
modificationDate	int64		1	modification time of the template format: UNIX timestamp

sample call

Request

```
POST /pbs-checkout-api/configuration/themes HTTP/1.1
Host: api-payment0.ver.sult-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Cache-Control: no-cache
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="theme"; filename="testtheme.zip"
Content-Type: application/x-zip-compressed

-----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="name"

testtheme
-----WebKitFormBoundary7MA4YWxkTrZu0gW--
```

Response

```
{
  "name": "testtheme",
  "tenant": "entertainMobil",
  "modificationDate": 1524650630249,
  "creationDate": 1524650630249
}
```

7.1.3 Download an existing theme

With this operation the service can get all current available template of his tenant.

Operation

GET /configuration/themes/{themeName}

Request

field name	type	length	multiplicity	comments
themeName	string	25	1	name of the theme to download note: The parameter is part of the operation

Response

The response is a binary (zip file).

Note: If you use Postman, use “Send and Download” instead of “Send”.

sample call

Request

```
GET /pbs-checkout-api/configuration/themes/etvmobil HTTP/1.1
Host: api-payment0.ver.sult-online.de
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9...
Cache-Control: no-cache
```

Response

```
PK...
```

7.1.4 Delete an existing theme

With this operation the service can get all current available template of his tenant.

Operation

DELETE /configuration/themes/{themeName}

Request

field name	type	length	multiplicity	comments
themeName	string	25	1	name of the theme to download note: The parameter is part of the operation

Response

The response contains no parameter.

sample call

Request
<pre>DELETE /pbs-checkout-api/configuration/themes/testtheme HTTP/1.1 Host: api-payment0.ver.sult-online.de Authorization: Bearer eyJhbGciOiJIUzI1NiJ9... Cache-Control: no-cache</pre>
Response

7.2 Theme development

The theme needs to be in an archive format (ZIP file) with following structure:

name	type	comments
/resources	folder	folder with the resources of the theme
/resources/css	folder	CSS resources
/resources/fonts	folder	fonts
/resources/img	folder	images
/resources/js	folder	JavaScript
/resources/third-party	folder	third-party add ons
/resources/favicon.ico	file	icon, logo, symbol of the webpage the browser can show
/templates	folder	folder with the template files
/templates/error.html	file	this template will be used when an error occurs
/templates/onetime.html	file	this template will be used for "onetime purchases"
/templates/recurring.html	file	this template will be used for "recurring registration and payment"
/templates/register.html	file	this template will be used for "recurring registration"
/messages.properties	file	contains the default language definitions
/messages_{lang}.properties	file	The default will be used if the requested language is not defined. contains the language definitions for {lang} {lang} is the language code based on ISO-639-1 e.g. "messages_de.properties" for German

PBS is using the {{mustache}} template engine: <https://mustache.github.io/>

With following table contains PBS specific lambdas:

lambda name	comments
ctx	provides modules context path
formatAmount	locale based amount formatting (decimal separator, precision digits) e.g. {{#formatAmount}}{{amount}}{{/formatAmount}} de_DE: 1,10, en_US: 1.10
i18n	localization of message properties e.g. {{#i18n}}msg.property{{/i18n}}
theme	resolves theme resource path e.g. {{#theme}}/resources/css/theme.css{{/theme}}

8 privacy information

With the general data privacy regulation (GDPR) the business partner has the obligation to make available which personal data of a customer is processed and by whom (article 13, GDPR). The following chapter will give an overview of the data, which will be or can be sent to the payment service providers. The business partner can use this information to update their privacy information.

The payment methods are ordered by name.

8.1 payment method: alipay_wirecard

There are no personal data sent to the PSP.

Payment service provider:

- Wirecard AG
Einsteinring 35
85609 Aschheim
Deutschland

8.2 payment method: alipay_computop

There are no personal data sent to the PSP.

Payment service provider:

- Computop GmbH
Schwarzenbergstr. 4
96050 Bamberg
Deutschland

8.3 payment method: amazon

There are no personal data sent to the PSP.

Payment service provider:

- Amazon Payments Europe S.C.A.
38 avenue J.F. Kennedy
L-1855 Luxemburg

8.4 payment method: applepay_computop

There are no personal data sent to the PSP.

Payment service provider:

- Apple, Inc.
1 Infinite Loop
Cupertino, CA 95014
United States
- Computop GmbH
Schwarzenbergstr. 4
96050 Bamberg
Deutschland

8.5 payment method: banktransfer_sofort

There are no personal data sent to the PSP, however the PSP will collect personal data to fulfil the payment process.

Payment service provider:

- Sofort GmbH
Theresienhöhe 12
80339 München
Deutschland

8.6 payment method: bcinvoice_telekom

This payment method can only be used by services of Telekom Deutschland. There are no additionally data privacy information needed.

8.7 payment method: creditcard_computop

The following data is sent to and processed by the PSP, when given by the business partner in the request:

- email address of the customer
(.../customerInformation/contactData/email)

The following data is sent to and processed by the PSP, and collected on the credit card input page:

- for credit card processing
 - credit card number
 - credit card holder name
- for credit card risk management by PSP or acquirer
 - email address of the customer
(the from business partner given email address will displayed and can be changed by the customer)
 - browser info
(acceptHeader, javaScriptEnabled, language, javaEnabled, colorDepth, screenHight and -Width, timeZoneOffset, userAgent)
 - ip address

Payment service provider:

- Computop GmbH
Schwarzenbergstr. 4
96050 Bamberg
Deutschland

8.8 payment method: directdebit_nexnet

There are no personal data sent to the PSP, however the PSP will collect personal data to fulfil the payment process.

Payment service provider:

- nexnet GmbH
Linkstraße 2
10785 Berlin
Deutschland

8.9 payment method: eps

There are no personal data sent to the PSP.

Payment service provider:

- STUZZA Studiengesellschaft für Zusammenarbeit im Zahlungsverkehr GmbH
Frankgasse 10/8
1090 Wien
Austria

8.10 payment method: fixedlinebill_croatia

This payment method can only be used by services of Telekom Deutschland. There are no additionally data privacy information needed.

8.11 payment method: fixedlinebill_telekom

This payment method can only be used by services of Telekom Deutschland. There are no additionally data privacy information needed.

8.12 payment method: loyalty_milesandmore

The following data is sent to and processed by the PSP:

- service card number (Miles&More card number)

Payment service provider:

- Miles & More GmbH
MAC Main Airport Center, Unterschweinstiege 8
60549 Frankfurt/Main
Deutschland

8.13 payment method: mobilebill

This is an amalgam of “mobilebill_mbe” and “mobilebill_xtc”. Please see there.

8.14 payment method: mobilebill_mbe

The following data is sent to and processed by the PSP:

- mobile phone number

Payment service provider:

- mobile business engine GmbH
Niederkasseler Lohweg 175
40547 Düsseldorf
Deutschland

8.15 payment method: mobilebill_sdp

The following data is sent to and processed by the PSP:

- mobile phone number

Payment service provider:

- Telekom Deutschland GmbH

8.16 payment method: mobilebill_xtc

The following data is sent to and processed by the PSP:

- mobile phone number

Payment service provider (depends on business partner configuration):

- Telekom Deutschland GmbH
- T-Mobile Austria GmbH

8.17 payment method: paypal

The following data is sent to and processed by the PSP, when given by the business partner in the request:

- shipping address of the customer
(.../customerInformation/shippingAddress)

Note: It's only useful to use if the business partner sent physical goods.

- customer information for risk management
(account id, account creation date, name, email address, phone number, country code)

Notes:

- The information will only send to PayPal, if the risk management is active. (default configuration: inactive)
- The information can only send to PayPal, if Group Privacy (GPR) gave their okay.

Payment service provider:

- PayPal (Europe) S.à r.l. et Cie, S.C.A.
22-24 Boulevard Royal
L-2449 Luxembourg

8.18 payment method: paysafecard

The following data is sent to and processed by the PSP, when given by the business partner in the request:

- account id of the customer (Money Laundering Act)
(.../customerInformation/externalIdentifier)

Payment service provider:

- Paysafe Prepaid Services Limited
Grand Canal House
Grand Canal Street Upper
Dublin 4, D04 Y7R5
Ireland

8.19 payment methode: telephonecard

There is no personal data collected or processed.

9 table of abbreviation

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

revision history

revision	date	editor	comments
1.0	06/26/2017	Stefan Heitzmann	version 1.0
1.1	06/28/2017	Stefan Heitzmann	added a table of abbreviation
1.2	07/03/2017	Stefan Heitzmann	added structure "customerInformation"
1.3	07/25/2017	Stefan Heitzmann	added attribute "preferredPaymentMethod" in the "reserve" request
1.4	07/31/2017	Stefan Heitzmann	correct comments in settlement for discounted sales
2	09/12/2017	Stefan Heitzmann	added structure "paymentServiceData" on "line item" level
3	10/24/2017	Stefan Heitzmann	added "subscription available" to payment methods
4	02/16/2018	Stefan Heitzmann	specified the valid entries in "returnUrl" added an information, that depending on the payment method (e.g. PayPal) several parameters may be added to the "returnUrl" added several missing informations
5	03/28/2018	Stefan Heitzmann	added payment method "bcinvoice_telekom" added direct register workflow
6	04/25/2018	Stefan Heitzmann	added chapter 7: "theme customization"
7	05/28/2018	Stefan Heitzmann	renaming the recurring payment workflows in chapter 4
8	09/28/2018	Stefan Heitzmann	support of "checkoutSessionId" to start a refund
9	11/28/2018	Stefan Heitzmann	added payment method "amazon pay" marked attribute "tax" in structure "lineItem" as deprecated
10		Stefan Heitzmann	added the "direct payment with couponing" workflow and request parameters added "transactionAlreadyRefunded" in the refund response
12	06/04/2019	Stefan Heitzmann	removed discount details added originalAmount on ipdr basket item added refund on item level
14	09/23/2019	Stefan Heitzmann	changed itemIndex type from string to int32 added "servicePeriod" and "invoiceData" to ipdr basket item these items are only used for "bc_invoice"
15	01/17/2020	Stefan Heitzmann	rename shipping address to address correct sample call for recurring charge reserve added session details for one time and recurring payments
16	04/22/2020	Stefan Heitzmann	added privacy information for every payment method payment handle defined as maximal of 4000 ASCII chars added contact data added new payment method alipay_wirecard , mobilebill and fixed-linebill_croatia
17	10/12/2020	Stefan Heitzmann	changed wording subscriptions to flex payments in chapter overview added debtor account data for additional information about the debtor added missing specifications for some fields
18	05/17/2021	Stefan Heitzmann	added payment method: Apple pay added information about subscription info added privacy information for alipay_computop , applepay_computop , and eps