

SAP Digital Payments Add-On

REST API



Developer Guide



cybersource
A Visa Solution

© 2023. Cybersource Corporation. All rights reserved.

Cybersource Corporation (Cybersource) furnishes this document and the software described in this document under the applicable agreement between the reader of this document (You) and Cybersource (Agreement). You may use this document and/or software only in accordance with the terms of the Agreement. Except as expressly set forth in the Agreement, the information contained in this document is subject to change without notice and therefore should not be interpreted in any way as a guarantee or warranty by Cybersource. Cybersource assumes no responsibility or liability for any errors that may appear in this document. The copyrighted software that accompanies this document is licensed to You for use only in strict accordance with the Agreement. You should read the Agreement carefully before using the software. Except as permitted by the Agreement, You may not reproduce any part of this document, store this document in a retrieval system, or transmit this document, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written consent of Cybersource.

Restricted Rights Legends

For Government or defense agencies: Use, duplication, or disclosure by the Government or defense agencies is subject to restrictions as set forth the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

For civilian agencies: Use, reproduction, or disclosure is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in Cybersource Corporation's standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States.

Trademarks

Authorize.Net, eCheck.Net, and The Power of Payment are registered trademarks of Cybersource Corporation. Cybersource and Cybersource Decision Manager are trademarks and/or service marks of Cybersource Corporation. Visa, Visa International, Cybersource, the Visa logo, the Cybersource logo, and 3-D Secure are the registered trademarks of Visa International in the United States and other countries. All other trademarks, service marks, registered marks, or registered service marks are the property of their respective owners.

Version: 23.01

Contents

- Recent Revisions to This Document..... 5**
- About This Guide..... 6**
- Introduction..... 7**
 - Requirements for Integrating SAP DPA with Cybersource..... 8
 - New-Merchant Onboarding..... 9
 - Existing-Merchant Onboarding..... 9
 - Activating Cybersource as SAP PSP..... 9
 - Offboarding..... 10
 - Supported Adapter Functionality..... 11
- Digital Payment Add-On Services..... 13**
- Payment Card Processing Services..... 14**
 - Token Preparation..... 15
 - Required Fields for Preparing a Token..... 15
 - Preparing a Token..... 15
 - Example: Preparing a Token..... 16
 - Token Preparation with Authorization..... 17
 - Required Fields for Preparing a Token with Authorization..... 17
 - Preparing a Token with Authorization..... 18
 - Example: Preparing a Token with Authorization..... 19
 - Payment Authorization..... 20
 - Required Fields for Authorizing a Payment..... 20
 - Authorizing a Payment..... 20
 - Example: Authorizing a Payment..... 21
 - Payment Authorization Cancellation..... 22
 - Required Fields for Canceling a Payment Authorization 22
 - Canceling an Authorization..... 22
 - Example: Canceling a Payment Authorization..... 23
 - Payment Reauthorization..... 24
 - Required Fields for Reauthorizing a Payment..... 24
 - Reauthorizing a Payment..... 24
 - Example: Reauthorizing a Payment..... 25
 - Charge..... 26

Required Fields for Performing a Charge.....	26
Performing a Charge.....	26
Example: Performing a Charge.....	27
Sale.....	28
Required Fields for Performing a Sale.....	28
Performing a Sale.....	28
Example: Performing a Sale.....	29
Payment Credit.....	30
Required Fields for Performing a Payment Credit.....	30
Performing a Payment Credit.....	30
Example: Performing a Payment Credit.....	31
Blind Payment Credit.....	32
Required Fields for Performing a Blind Payment Credit.....	32
Performing a Blind Payment Credit.....	32
Example: Performing a Blind Payment Credit.....	33
Alternative Payment Processing Services.....	34
Payment Authorization Preparation.....	36
Required Fields for Preparing a Payment Authorization.....	36
Preparing a Payment Authorization.....	36
Example: Preparing a Payment Authorization.....	37
Sale Preparation.....	38
Required Fields for Preparing a Sale.....	38
Preparing a Sale.....	38
Example: Preparing a Sale.....	39
Charge.....	40
Required Fields for Performing a Charge.....	40
Performing a Charge.....	40
Example: Charge.....	41
Payment Credit.....	42
Required Fields for Performing a Payment Credit.....	42
Performing a Payment Credit.....	42
Example: Performing a Payment Credit.....	43

Recent Revisions to This Document

23.01

Added notes on request and retry delays:

- [Requirements for Integrating SAP DPA with Cybersource \(on page 8\)](#)
- [Token Preparation \(on page 15\)](#)
- [Token Preparation with Authorization \(on page 17\)](#)
- [Payment Authorization Preparation \(on page 36\)](#)
- [Sale Preparation \(on page 38\)](#)

21.01

SAP Digital Payment Add-On Integration with Cybersource

Initial release of integration guide.

About This Guide

This section describes the audience and purpose of this guide, its use of conventions, and links to further information.

Audience and Purpose

This guide provides information to help acquirers, technology partners, and independent software vendors to integrate the SAP digital payments add-on (DPA) with Cybersource.

The main function of the SAP DPA is to promote secure, end-to-end digital payment processing.

Related Documentation

[Credit card developer guides](#)

[Token Management Service guide](#)

[SAP Digital Payments Administration guide](#)

[SAP Digital Payments User guide](#)

Refer to the Technical Documentation Portal for complete technical documentation:

<https://docs.cybersource.com/en/index.html>

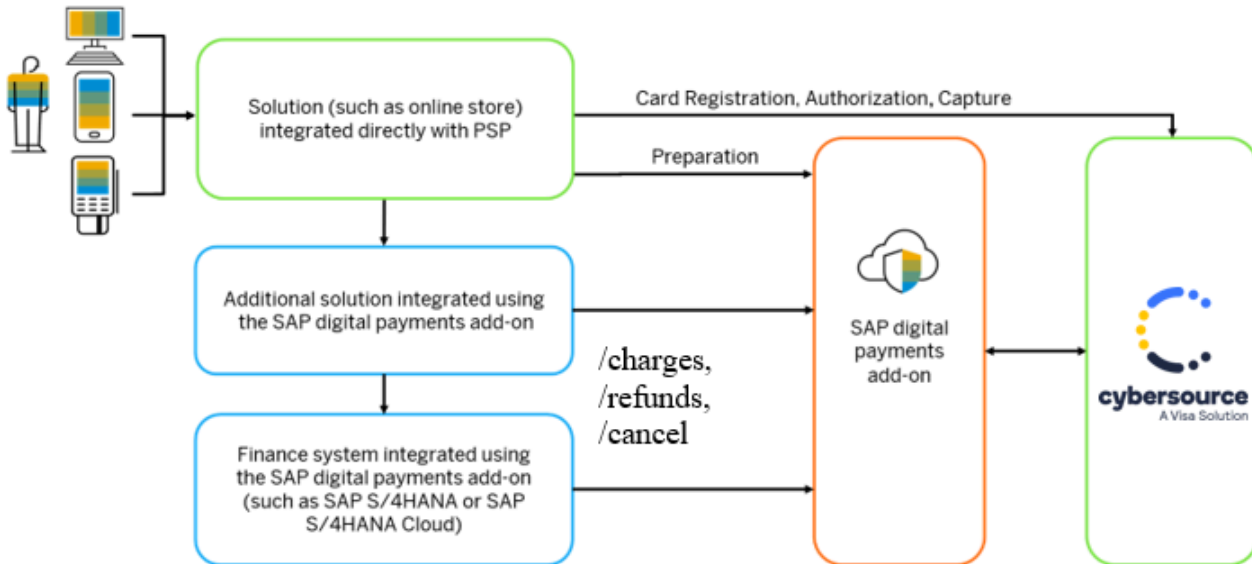
Customer Support

For support information about any service, visit the Cybersource Support Center:

<http://www.cybersource.com/support>

Introduction

The SAP digital payments add-on (DPA) enables you to connect SAP and non-SAP consumer applications with non-SAP payment service providers (PSPs). The SAP DPA solution is an alternative to current custom PSP integrations and provides ready-to-use PSP connectivity:



The SAP DPA consists of two distinct elements: the DPA core and the DPA adapter. The DPA core is found within SAP architecture and the SAP DPA adapter is within Cybersource. The initial authorization and tokenization are run directly through Cybersource. Following authorization and tokenization, all calls coming through DPA go to the SAP DPA core. The SAP DPA core then communicates with the SAP DPA adapter.

Requirements for Integrating SAP DPA with Cybersource

You must meet these requirements in order to use Cybersource services through SAP DPA.

- Your Cybersource account must be configured for tokenization.
- You must run the authorization and tokenization directly through Cybersource.
- Alternative payments are supported through the Cybersource alternative payment method platform. If you are using legacy alternative payment methods such as PayPal or bank transfer, you will need to migrate to newer solutions.



Important: After you run any authorization or tokenization request directly through Cybersource, Cybersource recommends that you implement a retry policy before you request the `prepare` call. For example, you could implement a policy to attempt retries every 0.5 seconds after the initial `prepare` call. This delay provides enough time for the data to reach all available data sources utilized by our Transaction Search functionality and ensures that SAP receives consistent data. Cybersource is working to improve or remove any need for a delay in the future.

New-Merchant Onboarding

The sequence below describes how a new connection between the SAP DPA and Cybersource is established.

1. SAP onboards you to its system and assigns you a tenant ID.
2. The Cybersource Developer Center generates a merchant ID (MID) for testing.
3. You must contact customer support to configure services on the MID.
4. You must share the tenant ID and MID with the Cybersource support team (GDLSAPDPA@visa.com) for MID-to-tenant ID mapping.
5. Cybersource ensures that you are a registered user of the MID.
6. Cybersource notifies you that boarding is complete.
7. [Activate Cybersource as your SAP PSP \(on page 9\)](#).

Existing-Merchant Onboarding

The sequence below describes how a connection between the SAP DPA and Cybersource is established when you already have a Cybersource MID.

1. SAP onboards you to its system and assigns you a tenant ID.
2. You must share the tenant ID and MID with the Cybersource support team (GDLSAPDPA@visa.com) for MID-to-tenant ID mapping.
3. Cybersource ensures that you are a registered user of the MID.
4. Cybersource notifies you that boarding is complete.
5. [Activate Cybersource as your SAP PSP \(on page 9\)](#).

Activating Cybersource as SAP PSP

To activate Cybersource as your PSP within SAP:

1. Navigate to <https://{{baseUrl}}/pspStatus/index.html>. Select **Cybersource** and click **Activate**. "Active" status indicates a successful activation of Cybersource as PSP in SAP.
2. Navigate to <https://{{baseUrl}}/pspDetermination/index.html> and ensure that the requested MIDs appear under the Merchant column.
3. Set up transaction routing as required. Blank columns are interpreted as though you've selected all options and could affect the routing of the transaction.

Offboarding

The sequence below describes how to remove the MID from MID-to-tenant mapping.

1. You must contact the Cybersource support team via email (GDLSAPDPA@visa.com).
2. Cybersource ensures that you are a registered user of the MID.
3. Cybersource notifies you that offboarding is complete.

Supported Adapter Functionality

The SAP DPA supports payment cards and alternative payments. The supported capabilities and the corresponding SAP DPA API endpoints are described below.

Payment Cards

The SAP DPA supports payments made using payment cards.

cc_authorize

/v1/authorizations

Authorizes a payment using a registered payment card.

cc_blind_credit

/v1/refunds

Credits a payment card without reference to a previous payment transaction, for example in the case of a credit memo without reference to a specific invoice.

cc_cancel_authorization

/v1/authorizations/cancel

Cancels (voids) the supplied authorizations.

cc_charge

/v1/charges

When an authorization is supplied, the charge settles the authorization.

cc_direct_capture

/v1/charges

Directly charges an amount to a registered payment card without first getting an authorization.

cc_prepare_authorization

/v1/tokens/preparepaymentcardauthorizationforfurtherusage

Prepares externally created payment card authorizations for further use.

cc_prepare_card

/v1/tokens/preparecardforfurtherusage

Prepares externally registered cards for further use.

cc_reauthorize_v2

/v2/reauthorizations

Refreshes a payment card payment authorization that was generated previously but has expired.

cc_refund

v1/refunds

Credits the amount supplied in reference to the original payment.

Alternative Payments

The SAP DPA supports what are referred to as *alternative payments*. Alternative payments are payments made using methods other than payment cards.

ep_charge

/v1/charges

When an alternative payment authorization is supplied, the charge settles the authorization. This capability is currently supported for PayPal and Klarna. For more information, see the [PayPal Express Checkout Using Alternative Payment Services](#) guide.

ep_prepare_authorization

/v1/tokens/prepareauthorizationforfurtherusage

Prepares externally created alternative payment authorizations for further use.

ep_prepare_direct_capture

/v1/tokens/preparedirectcaptureforfurtherusage

Prepares externally settled payment for an alternative payment service. This capability is currently supported for PayPal, Klarna, Sofort, and iDEAL.

ep_refund

/v1/refunds

Credits the settled payment for an alternative payment service. This capability is currently supported for PayPal, Klarna, Sofort, and iDEAL. For more information, see the [PayPal Express Checkout Using Alternative Payment Services](#) and [Online Bank Transfers](#) guides.

Digital Payment Add-On Services

Consumer applications like online stores that communicate with the SAP digital payment add-on (DPA) to prepare payment cards and authorizations using Cybersource as the PSP must use a previously generated authorization request ID and subscription ID received from Cybersource. For information on generating authorization IDs and subscription IDs, see the [credit card developer guides](#) and the [Token Management Service](#) guide.

All requests to the SAP DPA core have the same base URL. To call individual capabilities, you must append the resource path to the base URL of <https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com>.

Payment Card Processing Services

The following payment card processing services are available through the SAP DPA:

- Token Preparation
- Token Preparation with Authorization
- Payment Authorization
- Payment Authorization Cancellation
- Payment Reauthorization
- Charge
- Sale
- Payment Credit
- Blind Payment Credit

Token Preparation

To prepare a card token, merchants must ensure that the parameter **PaymentCardContext.RequestId** is set to the Cybersource subscription creation request ID.

Endpoint: `{{baseUrl}}/core/v1/tokens/getforpaymentcard`



Important: After you run any authorization or tokenization request directly through Cybersource, Cybersource recommends that you implement a retry policy before you request the `prepare` call. For example, you could implement a policy to attempt retries every 0.5 seconds after the initial `prepare` call. This delay provides enough time for the data to reach all available data sources utilized by our Transaction Search functionality and ensures that SAP receives consistent data. Cybersource is working to improve or remove any need for a delay in the future.

Required Fields for Preparing a Token

Preparing a payment card token requires these fields:

PaymentCards.MerchantAccount

PaymentCards.PaymentCardContext

PaymentCards.PaymentServiceProvider

PaymentCards.PaytCardByPaytServiceProvider

PaymentCards.RequestId

Preparing a Token

1. Send the request to the `getforpaymentcard` endpoint:

POST

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/tokens/getforpaymentcard
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Preparing a Token

Request

```
{
  "PaymentCards": [
    {
      "PaytCardByPaytServiceProvider": "{{subscriptionID}}",
      "PaymentServiceProvider": "CYBS",
      "MerchantAccount": "{{CYBSmerchantID}}",
      "PaymentCardContext": {
        "RequestId": "{{subscriptionReqID}}"
      }
    }
  ]
}
```

Token Preparation with Authorization

Provided that a card token has been registered, an authorization can be imported into SAP using this service.

Endpoint: `{{baseUrl}}/core/v1/tokens/getforpaymentcardwithauthorization`



Important: After you run any authorization or tokenization request directly through Cybersource, Cybersource recommends that you implement a retry policy before you request the `prepare` call. For example, you could implement a policy to attempt retries every 0.5 seconds after the initial `prepare` call. This delay provides enough time for the data to reach all available data sources utilized by our Transaction Search functionality and ensures that SAP receives consistent data. Cybersource is working to improve or remove any need for a delay in the future.

Related information

[Preparing a Token \(on page 15\)](#)

Required Fields for Preparing a Token with Authorization

Preparing a card token with authorization requires these fields:

PaymentCardsWithAuthorization.AuthorizationCurrency

PaymentCardsWithAuthorization.AuthorizationByPaytSrvcPrvdr

PaymentCardsWithAuthorization.AuthorizedAmountInAuthznCrcy

PaymentCardsWithAuthorization.DigitalPaymentCommerceType

PaymentCardsWithAuthorization.DigitalPaymentSessionType

PaymentCardsWithAuthorization.MerchantAccount

PaymentCardsWithAuthorization.PaymentCardContext

PaymentCardsWithAuthorization.PaymentServiceProvider

PaymentCardsWithAuthorization.PaytCardByPaytServiceProvider

PaymentCardsWithAuthorization.RequestId

Preparing a Token with Authorization

1. Send the request to the `getforpaymentcardwithauthorization` endpoint:

```
POST
```

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/tokens/getforpaymentcardwithauthorization
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Preparing a Token with Authorization

Request

```
{
  "PaymentCardsWithAuthorization": [
    {
      "AuthorizationByPaytSrvcPrvdr": "{{authorizationRequestID}}",
      "PaytCardByPaytServiceProvider": "{{subscriptionID}}",
      "MerchantAccount": "{{CYBSmerchantID}}",
      "PaymentServiceProvider": "CYBS",
      "AuthorizedAmountInAuthznCrcy": "{{amount}}",
      "AuthorizationCurrency": "{{currency}}",
      "DigitalPaymentCommerceType": "MOTO",
      "DigitalPaymentSessionType": "OFFLINE",
      "PaymentCardContext": {
        "RequestId": "{{subscriptionReqID}}"
      }
    }
  ]
}
```

Payment Authorization

Endpoint: `{{baseURL}}/core/v1/authorizations`

Required Fields for Authorizing a Payment

The following fields are required for performing authorizations:

Authorizations.AmountInAuthorizationCurrency

Authorizations.AuthorizationCurrency

Authorizations.DigitalPaymentCommerceType

Authorizations.DigitalPaymentSessionType

Authorizations.PaytCardByDigitalPaymentSrcv

Authorizations.PreauthorizationIsRequested

Authorizing a Payment

1. Send the request to the `authorization` endpoint:

```
POST
```

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/authorization
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Authorizing a Payment

Request

```
{
  "Authorizations": [
    {
      "Source": {
        "Card": {
          "PaytCardByDigitalPaymentSrvc": "{{cardDPToken}}"
        }
      },
      "AmountInAuthorizationCurrency": "{{amount}}",
      "AuthorizationCurrency": "{{currency}}",
      "PreauthorizationIsRequested": false,
      "DigitalPaymentCommerceType": "MOTO",
      "DigitalPaymentSessionType": "OFFLINE"
    }
  ]
}
```

Payment Authorization Cancellation

Endpoint: `{{baseUrl}}/core/v1/authorizations/cancel`

Required Fields for Canceling a Payment Authorization

The following fields are required for canceling a payment authorization:

CancelAuthorizations.AmountInAuthorizationCurrency

CancelAuthorizations.AuthorizationByDigitalPaytSrvc

CancelAuthorizations.AuthorizationCurrency

CancelAuthorizations.PaytCardByDigitalPaymentSrvc

Canceling an Authorization

1. Send the request to the `cancel` endpoint:

```
POST
```

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/authorization/cancel
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Canceling a Payment Authorization

Request

```
{
  "CancelAuthorizations": [
    {
      "Source": {
        "Card": {
          "PaytCardByDigitalPaymentSrvc": "{{cardDPToken}}"
        }
      },
      "AuthorizationByDigitalPaytSrvc": "{{authDPToken}}",
      "AmountInAuthorizationCurrency": "{{amount}}",
      "AuthorizationCurrency": "{{currency}}"
    }
  ]
}
```

Payment Reauthorization

Endpoint: `{{baseUrl}}/core/v2/reauthorizations`

Required Fields for Reauthorizing a Payment

The following fields are required for payment reauthorization:

Reauthorizations.AmountInAuthorizationCurrency

Reauthorizations.AuthorizationByDigitalPaytSrcv

Reauthorizations.PaymentCardAuthznRelationID

Reauthorizations.ReauthorizationReasonCode

Reauthorizing a Payment

1. Send the request to the `reauthorizations` endpoint:

```
POST
```

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v2/reauthorizations
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Reauthorizing a Payment

Request

```
{
  "Reauthorizations": [
    {
      "AuthorizationByDigitalPaytSrvc": "{{authDPToken}}",
      "AmountInAuthorizationCurrency": "{{amount}}",
      "ReauthorizationReasonCode": "LATE_DELIVERY",
      "PaymentCardAuthznRelationID": "201506041511351"
    }
  ]
}
```

Charge

Endpoint: `{{baseUrl}}/core/v1/charges`

Required Fields for Performing a Charge

The following fields are required for performing a charge:

Charges.AmountInPaymentCurrency

Charges.AuthorizationByDigitalPaytSrvc

Charges.PaymentIsToBeCaptured

Charges.PaymentTransactionDescription

Charges.PaytCardByDigitalPaymentSrvc

Charges.ReferenceDocument

Performing a Charge

1. Send the request to the `charges` endpoint:

```
POST
```

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/charges
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Performing a Charge

Request

```
{
  "Charges": [
    {
      "Source": {
        "Card": {
          "PaytCardByDigitalPaymentSrvc": "{{cardDPToken}}",
          "PaymentCardType": "",
          "PaymentCardExpirationMonth": "",
          "PaymentCardExpirationYear": "",
          "PaymentCardMaskedNumber": "",
          "PaymentCardHolderName": ""
        }
      },
      "AmountInPaymentCurrency": "{{amount}}",
      "PaymentCurrency": "{{currency}}",
      "PaymentTransactionDescription": "Charge {{amount}} {{currency}}",
      "PaymentIsToBeCaptured": false,
      "ReferenceDocument": "CHRGTEST011-1",
      "CustomerAccountNumber": "",
      "CompanyCode": "",
      "CustomerCountry": "",
      "PaymentMethod": "",
      "CustomParameterValue": "",
      "Authorization": {
        "AuthorizationByDigitalPaytSrvc": "{{authDPToken}}",
        "AuthorizationByPaytSrvcPrvdr": "",
        "AuthorizationByAcquirer": "",
        "AuthorizedAmountInAuthznCrcy": "",
        "AuthorizationCurrency": "",
        "AuthorizationDateTime": "",
        "AuthorizationExpirationDateTme": "",
        "AuthorizationStatus": "",
        "DetailedAuthorizationStatus": "",
        "AuthorizationStatusName": "",
        "MerchantAlias": ""
      }
    }
  ]
}
```

Sale

In direct capture transactions, capture and authorization occur in one step. Sale requests are similar to charge transactions except that in sale transactions, the **PaymentIsToBeCaptured** field is set to `true`, and specifying authorization detail is not required.

Endpoint: `{{baseUrl}}/core/v1/charges`

Related information

[Performing a Charge \(on page 26\)](#)

Required Fields for Performing a Sale

The following fields are required for performing a sale:

Charges.AmountInPaymentCurrency

Charges.PaymentCurrency

Charges.PaymentIsToBeCaptured

Charges.PaymentTransactionDescription

Charges.PaytCardByDigitalPaymentSrv

Charges.ReferenceDocument

Performing a Sale

1. Send the request to the `charges` endpoint:

```
POST
```

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/charges
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Performing a Sale

Request

```
{
  "Charges": [
    {
      "Source": {
        "Card": {
          "PaytCardByDigitalPaymentSrvc": "{{cardDPToken}}",
          "PaymentCardType": "",
          "PaymentCardExpirationMonth": "",
          "PaymentCardExpirationYear": "",
          "PaymentCardMaskedNumber": "",
          "PaymentCardHolderName": ""
        }
      },
      "AmountInPaymentCurrency": "{{amount}}",
      "PaymentCurrency": "{{currency}}",
      "PaymentTransactionDescription": "Charge {{amount}} {{currency}}",
      "PaymentIsToBeCaptured": true,
      "ReferenceDocument": "CHRGTEST011-1",
      "CustomerAccountNumber": "",
      "CompanyCode": "",
      "CustomerCountry": "",
      "PaymentMethod": "",
      "CustomParameterValue": "",
      "Authorization": {
        "AuthorizationByDigitalPaytSrvc": "",
        "AuthorizationByPaytSrvcPrvdr": "",
        "AuthorizationByAcquirer": "",
        "AuthorizedAmountInAuthznCrcy": "",
        "AuthorizationCurrency": "",
        "AuthorizationDateTime": "",
        "AuthorizationExpirationDateTme": "",
        "AuthorizationStatus": "",
        "DetailedAuthorizationStatus": "",
        "AuthorizationStatusName": "",
        "MerchantAlias": ""
      }
    }
  ]
}
```

Payment Credit

Endpoint: `{{baseURL}}/core/v1/refunds`

Required Fields for Performing a Payment Credit

The following fields are required for performing a payment credit:

Refunds.AmountInRefundCurrency

Refunds.DigitalPaymentTransaction

Refunds.PaymentByPaymentServicePrvdr

Refunds.PaymentTransactionDescription

Refunds.PaytCardByDigitalPaymentSrcv

Refunds.ReferenceDocument

Refunds.RefundCurrency

Performing a Payment Credit

1. Send the request to the `refunds` endpoint:

```
POST
```

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/refunds
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Performing a Payment Credit

Request

```
{
  "Refunds": [
    {
      "DigitalPaymentTransaction": {
        "DigitalPaymentTransaction": "EXAMPLE001"
      },
      "Source": {
        "Card": {
          "PaytCardByDigitalPaymentSrvc": "{{cardDPToken}}",
          "PaymentCardType": "",
          "PaymentCardExpirationMonth": "",
          "PaymentCardExpirationYear": "",
          "PaymentCardMaskedNumber": "",
          "PaymentCardHolderName": ""
        }
      },
      "AmountInRefundCurrency": "{{amount}}",
      "RefundCurrency": "{{currency}}",
      "PaymentTransactionDescription": "Refund {{amount}} {{currency}}",
      "PaymentByPaymentServicePrvdr": "{{captureToken}}",
      "ReferenceDocument": "EXAMPLETEST001"
    }
  ]
}
```

Blind Payment Credit

A blind credit request is similar to a payment credit request. However, you do not set any value corresponding to the **PaymentByPaymentServicePrvdr** field in a blind payment credit request.

Endpoint: `{{baseURL}}/core/v1/refunds`

Related information

[Performing a Payment Credit \(on page 30\)](#)

Required Fields for Performing a Blind Payment Credit

The following fields are required for performing a blind payment credit:

Refunds.AmountInRefundCurrency

Refunds.DigitalPaymentTransaction

Refunds.PaymentTransactionDescription

Refunds.PaytCardByDigitalPaymentSrvc

Refunds.ReferenceDocument

Refunds.RefundCurrency

Performing a Blind Payment Credit

1. Send the request to the `refunds` endpoint:

```
POST
```

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/refunds
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Performing a Blind Payment Credit

Request

```
{
  "Refunds": [
    {
      "DigitalPaymentTransaction": {
        "DigitalPaymentTransaction": "EXAMPLE001"
      },
      "Source": {
        "Card": {
          "PaytCardByDigitalPaymentSrvc": "{{cardDPToken}}",
          "PaymentCardType": "",
          "PaymentCardExpirationMonth": "",
          "PaymentCardExpirationYear": "",
          "PaymentCardMaskedNumber": "",
          "PaymentCardHolderName": ""
        }
      },
      "AmountInRefundCurrency": "{{amount}}",
      "RefundCurrency": "{{currency}}",
      "PaymentTransactionDescription": "Blind Credit {{amount}}
      {{currency}}",
      "ReferenceDocument": "EXAMPLETEST001"
    }
  ]
}
```

Alternative Payment Processing Services

Cybersource through the SAP DPA supports a number of alternative payment types. The following alternative payment processing services are available through the SAP DPA:

- Payment Authorization Preparation
- Sale Preparation
- Charge
- Payment Credit

Alternative payment transactions must be at the Cybersource authorization or capture stage (payment type dependant) before integration with the DPA. For each supported alternative payment, you must add the **DigitalPaymentAuthorizationType** field to the transaction request:

Klarna

DigitalPaymentAuthorizationType: [KLA1](#)

Payment Authorization Preparation

Sale Preparation

Charge

Payment Credit

Sofort

DigitalPaymentAuthorizationType: [SOF1](#)

Sale Preparation

Payment Credit

iDEAL

DigitalPaymentAuthorizationType: [IDE1](#)

Sale Preparation

Payment Credit

PayPal

DigitalPaymentAuthorizationType: [DP1P](#)

Payment Authorization Preparation

Sale Preparation

Charge

Payment Credit

All examples use Klarna entries. However, the same logic and request fields are required for each payment type except for the **DigitalPaymentAuthorizationType** field.

Payment Authorization Preparation

Endpoint: `{{baseURL}}/core/v1/tokens/prepareauthorizationforfurtherusage`



Important: After you run any authorization or tokenization request directly through Cybersource, Cybersource recommends that you implement a retry policy before you request the `prepare` call. For example, you could implement a policy to attempt retries every 0.5 seconds after the initial `prepare` call. This delay provides enough time for the data to reach all available data sources utilized by our Transaction Search functionality and ensures that SAP receives consistent data. Cybersource is working to improve or remove any need for a delay in the future.

Required Fields for Preparing a Payment Authorization

The following fields are required for preparing an authorization:

Authorizations.AuthorizationByPaytSrvcPrvdr

Authorizations.AuthorizationCurrency

Authorizations.AuthorizedAmountInAuthznCrcy

Authorizations.DigitalPaymentAuthorizationType

Authorizations.MerchantAccount

Authorizations.PaymentServiceProvider

Preparing a Payment Authorization

1. Send the request to the `preparedirectcaptureforfurtherusage` endpoint:

POST

```
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/tokens/preparedirectcaptureforfurtherusage
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Preparing a Payment Authorization

Request

```
{
  "Authorizations": [
    {
      "AuthorizationByPaytSrvcPrvdr": "{{klarnaAuthorizationID}}",
      "PaymentServiceProvider": "CYBS",
      "MerchantAccount": "{{CYBSmerchantID}}",
      "DigitalPaymentAuthorizationType": "KLA2",
      "AuthorizedAmountInAuthznCrcy": "{{ep_klarna_grandTotalAmount}}",
      "AuthorizationCurrency": "{{ep_klarna_currency}}"
    }
  ]
}
```

Sale Preparation

Endpoint: `{{baseUrl}}/core/v1/tokens/preparedirectcaptureforfurtherusage`



Important: After you run any authorization or tokenization request directly through Cybersource, Cybersource recommends that you implement a retry policy before you request the `prepare` call. For example, you could implement a policy to attempt retries every 0.5 seconds after the initial `prepare` call. This delay provides enough time for the data to reach all available data sources utilized by our Transaction Search functionality and ensures that SAP receives consistent data. Cybersource is working to improve or remove any need for a delay in the future.

Required Fields for Preparing a Sale

The following fields are required for preparing a sale:

- DirectCaptures.AmountInPaymentCurrency**
- DirectCaptures.DigitalPaymentDirectCaptureType**
- DirectCaptures.MerchantAccount**
- DirectCaptures.PaymentByPaymentServicePrvdr**
- DirectCaptures.PaymentCurrency**
- DirectCaptures.PaymentServiceProvider**

Preparing a Sale

1. Send the request to the `preparedirectcaptureforfurtherusage` endpoint:

```
POST
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/tokens/preparedirectcaptureforfurtherusage
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Preparing a Sale

Request

```
{
  "DirectCaptures": [
    {
      "PaymentByPaymentServicePrvdr": "{{klarnaCaptureToken}}",
      "PaymentServiceProvider": "CYBS",
      "DigitalPaymentDirectCaptureType": "KLA1",
      "MerchantAccount": "{{CYBSmerchantID}}",
      "AmountInPaymentCurrency": "{{ep_klarna_grandTotalAmount}}",
      "PaymentCurrency": "{{ep_klarna_currency}}"
    }
  ]
}
```

Charge

Endpoint: `{{baseUrl}}/core/v1/charges`

Required Fields for Performing a Charge

The following fields are required for performing a charge:

Charges.AmountInPaymentCurrency

Charges.AuthorizationByDigitalPaytSrvc

Charges.PaymentCurrency

Charges.PaymentIsToBeCaptured

Charges.PaymentTransactionDescription

Charges.ReferenceDocument

Performing a Charge

1. Send the request to the `charges` endpoint:

```
POST
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/core/v1/charges
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Charge

Request

```
{
  "Charges": [
    {
      "Source": {
      },
      "AmountInPaymentCurrency": "{{ep_klarna_grandTotalAmount}}",
      "PaymentCurrency": "{{ep_klarna_currency}}",
      "PaymentTransactionDescription": "Charge
      {{ep_klarna_grandTotalAmount}} {{ep_klarna_currency}}",
      "PaymentIsToBeCaptured": false,
      "ReferenceDocument": "CHRGTEST011-1",
      "CustomerAccountNumber": "",
      "CompanyCode": "",
      "CustomerCountry": "",
      "PaymentMethod": "",
      "CustomParameterValue": "",
      "Authorization": {
        "AuthorizationByDigitalPaytSrvc": "{{klarnaAuthDPToken}}",
        "AuthorizationByPaytSrvcPrvdr": "",
        "AuthorizationByAcquirer": "",
        "AuthorizedAmountInAuthznCrcy": "",
        "AuthorizationCurrency": "",
        "AuthorizationDateTime": "",
        "AuthorizationExpirationDateTme": "",
        "AuthorizationStatus": "",
        "DetailedAuthorizationStatus": "",
        "AuthorizationStatusName": "",
        "MerchantAlias": ""
      }
    }
  ]
}
```

Payment Credit

Endpoint: `{{baseUrl}}/core/v1/refunds`

Required Fields for Performing a Payment Credit

A payment credit requires these fields:

Refunds.AmountInRefundCurrency

Refunds.DigitalPaymentDirectCaptureType

Refunds.PaymentByDigitalPaymentService

Refunds.ReferenceDocument

Refunds.RefundCurrency

Performing a Payment Credit

1. Send the request to the `refunds` endpoint:

```
POST
https://https://digitalpayments-demo-core.cfapps.eu10.hana.ondemand.com/v1/refunds
```

2. Include the required fields in the request.
3. Check the response message to make sure that the request was successful. A 200-level HTTP response code indicates success. For information about response codes, see [Transaction Response Codes](#).

Example: Performing a Payment Credit

Request

```
{
  "Refunds": [
    {
      "ReferenceDocument": "some info",
      "AmountInRefundCurrency": "{{ep_klarna_grandTotalAmount}}",
      "RefundCurrency": "{{ep_klarna_currency}}",
      "PaymentByDigitalPaymentService": "{{klarnaCaptureDPToken}}",
      "DigitalPaymentDirectCaptureType": "KLA1"
    }
  ]
}
```