

🔧 Setup and Configuration

1. Install Adyen's Python Library

Adyen provides an official Python library to simplify API interactions.

```
bash
CopyEdit
pip install Adyen
```

2. Initialize the Client

Set up the Adyen client with your API key and environment.

```
python
CopyEdit
import Adyen

adyen = Adyen.Adyen()
adyen.client.xapikey = 'YOUR_API_KEY'
adyen.client.platform = 'test' # Use 'live' for production
adyen.client.merchant_account = 'YOUR_MERCHANT_ACCOUNT'
```

💳 Payment Processing

1. Create a Payment Session

Initiate a payment session to start the checkout process.[GitHub](#)+1[Adyen Docs](#)+1

```
python
CopyEdit
request = {
    "amount": {"currency": "EUR", "value": 1000},
    "reference": "Your order number",
    "returnUrl": "https://your-company.com/checkout?shopperOrder=12xy",
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.checkout.sessions(request)
```

2. Make a Payment

Process a payment using the collected payment method details.

```
python
CopyEdit
request = {
    "amount": {"currency": "EUR", "value": 1000},
    "reference": "Your order number",
    "paymentMethod": {
        "type": "scheme",
        "encryptedCardNumber": "test_4111111111111111",
    },
}
```

```

        "encryptedExpiryMonth": "test_03",
        "encryptedExpiryYear": "test_2030",
        "encryptedSecurityCode": "test_737"
    },
    "returnUrl": "https://your-company.com/checkout?shopperOrder=12xy",
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.checkout.payments(request)

```

3. Submit Additional Payment Details

Handle additional actions like 3D Secure authentication.

```

python
CopyEdit
request = {
    "details": {"threeds2.fingerprint": "fingerprint_value"},
    "paymentData": "Ab02b4c0!BQABAgCW5sx..." # From previous response
}

response = adyen.checkout.payments_details(request)

```

Payment Modifications

1. Capture a Payment

Capture funds for an authorized payment.[Adyen Docs](#)¹[Adyen Docs](#)¹

```

python
CopyEdit
request = {
    "amount": {"currency": "EUR", "value": 1000},
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.payment.captures('pspReference', request)

```

2. Refund a Payment

Refund a captured payment.

```

python
CopyEdit
request = {
    "amount": {"currency": "EUR", "value": 500},
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.payment.refunds('pspReference', request)

```

3. Cancel a Payment

Cancel an authorized payment before capture.

```
python
CopyEdit
request = {
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.payment.cancels('pspReference', request)
```

Webhooks Integration

Adyen uses webhooks to notify about payment events.[Pipedream+2GitHub+2Adyen Docs+2](#)

1. Set Up Webhook Endpoint

Create an endpoint to receive webhook notifications.[Adyen Docs+2GitHub+2Adyen Docs+2](#)

```
python
CopyEdit
from flask import Flask, request

app = Flask(__name__)

@app.route('/webhooks/notifications', methods=['POST'])
def webhook():
    notification = request.json
    # Process the notification
    return '[accepted]'
```

2. Verify HMAC Signature

Ensure the notification's authenticity by verifying the HMAC signature.[GitHub](#)

```
python
CopyEdit
import hmac
import hashlib
import base64

def verify_hmac(notification, hmac_key):
    data = notification['data']
    expected_signature = notification['additionalData']['hmacSignature']
    message = ':'.join([data['pspReference'], data['originalReference'],
data['merchantAccountCode'], str(data['amount']['value']),
data['amount']['currency'], data['eventCode'], data['success']])
    hmac_calculated = base64.b64encode(hmac.new(base64.b64decode(hmac_key),
message.encode('utf-8'), hashlib.sha256).digest()).decode()
    return hmac.compare_digest(hmac_calculated, expected_signature)
```

☐ Testing and Utilities

- API Explorer: <https://docs.adyen.com/api-explorer/GitHubAdyen Docs>

- **Postman Collections:** <https://medium.com/adyen/effortless-api-testing-with-the-adyen-postman-collections-52fd3ce3f6a7>
- **Test Card Numbers:** <https://docs.adyen.com/development-resources/test-cards/test-card-numbers>

🔧 Setup and Configuration

1. Install Adyen's Python Library

Adyen provides an official Python library to simplify API interactions.

```
bash
CopyEdit
pip install Adyen
```

2. Initialize the Client

Set up the Adyen client with your API key and environment.

```
python
CopyEdit
import Adyen

adyen = Adyen.Adyen()
adyen.client.xapikey = 'YOUR_API_KEY'
adyen.client.platform = 'test' # Use 'live' for production
adyen.client.merchant_account = 'YOUR_MERCHANT_ACCOUNT'
```

💳 Payment Processing

1. Create a Payment Session

Initiate a payment session to start the checkout process. [GitHub+1Adyen Docs+1](#)

```
python
CopyEdit
request = {
    "amount": {"currency": "EUR", "value": 1000},
    "reference": "Your order number",
    "returnUrl": "https://your-company.com/checkout?shopperOrder=12xy",
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.checkout.sessions(request)
```

2. Make a Payment

Process a payment using the collected payment method details.

```
python
CopyEdit
```

```

request = {
    "amount": {"currency": "EUR", "value": 1000},
    "reference": "Your order number",
    "paymentMethod": {
        "type": "scheme",
        "encryptedCardNumber": "test_4111111111111111",
        "encryptedExpiryMonth": "test_03",
        "encryptedExpiryYear": "test_2030",
        "encryptedSecurityCode": "test_737"
    },
    "returnUrl": "https://your-company.com/checkout?shopperOrder=12xy",
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.checkout.payments(request)

```

3. Submit Additional Payment Details

Handle additional actions like 3D Secure authentication.

```

python
CopyEdit
request = {
    "details": {"threads2.fingerprint": "fingerprint_value"},
    "paymentData": "Ab02b4c0!BQABAgCW5sx..." # From previous response
}

response = adyen.checkout.payments_details(request)

```

Payment Modifications

1. Capture a Payment

Capture funds for an authorized payment.[Adyen Docs+1](#)[Adyen Docs+1](#)

```

python
CopyEdit
request = {
    "amount": {"currency": "EUR", "value": 1000},
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.payment.captures('pspReference', request)

```

2. Refund a Payment

Refund a captured payment.

```

python
CopyEdit
request = {
    "amount": {"currency": "EUR", "value": 500},
    "merchantAccount": adyen.client.merchant_account
}

```

```
response = adyen.payment.refunds('pspReference', request)
```

3. Cancel a Payment

Cancel an authorized payment before capture.

```
python
CopyEdit
request = {
    "merchantAccount": adyen.client.merchant_account
}

response = adyen.payment.cancels('pspReference', request)
```

Webhooks Integration

Adyen uses webhooks to notify about payment events.[Pipedream+2GitHub+2Adyen Docs+2](#)

1. Set Up Webhook Endpoint

Create an endpoint to receive webhook notifications.[Adyen Docs+2GitHub+2Adyen Docs+2](#)

```
python
CopyEdit
from flask import Flask, request

app = Flask(__name__)

@app.route('/webhooks/notifications', methods=['POST'])
def webhook():
    notification = request.json
    # Process the notification
    return '[accepted]'
```

2. Verify HMAC Signature

Ensure the notification's authenticity by verifying the HMAC signature.[GitHub](#)

```
python
CopyEdit
import hmac
import hashlib
import base64

def verify_hmac(notification, hmac_key):
    data = notification['data']
    expected_signature = notification['additionalData']['hmacSignature']
    message = ':'.join([data['pspReference'], data['originalReference'],
data['merchantAccountCode'], str(data['amount']['value']),
data['amount']['currency'], data['eventCode'], data['success']])
    hmac_calculated = base64.b64encode(hmac.new(base64.b64decode(hmac_key),
message.encode('utf-8'), hashlib.sha256).digest()).decode()
    return hmac.compare_digest(hmac_calculated, expected_signature)
```

□ Testing and Utilities

- **API Explorer:** <https://docs.adyen.com/api-explorer/GitHubAdyen Docs>
- **Postman Collections:** <https://medium.com/adyen/effortless-api-testing-with-the-adyen-postman-collections-52fd3ce3f6a7>
- **Test Card Numbers:** <https://docs.adyen.com/development-resources/test-cards/test-card-numbers>

#Python code for create payment session :

Adyen Python API Library v13.2.0

```
import Adyen
```

```
adyen = Adyen.Adyen()
```

```
adyen.client.xapikey = "ADYEN_API_KEY"
```

```
# For the LIVE environment, also include your liveEndpointUrlPrefix.
```

```
adyen.client.platform = "test" # The environment to use library in.
```

```
# Create the request object(s)
```

```
json_request = {
```

```
    "merchantAccount": "YOUR_MERCHANT_ACCOUNT",
```

```
    "amount": {
```

```
        "value": 100,
```

```
        "currency": "EUR"
```

```
    },
```

```
    "returnUrl": "https://your-company.com/checkout?shopperOrder=12xy..",
```

```
    "reference": "YOUR_PAYMENT_REFERENCE",
```

```
    "countryCode": "NL"
```

```
}
```

```
# Send the request
```

```
result = adyen.checkout.payments_api.sessions(request=json_request, idempotency_key="UUID")
```

#Json response payment session created success

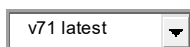
```
{
  "amount": {
    "currency": "EUR",
    "value": 100
  },
  "countryCode": "NL",
  "expiresAt": "2022-01-11T13:53:18+01:00",
  "id": "CS451F2AB1ED897A94",
  "merchantAccount": "YOUR_MERCHANT_ACCOUNT",
  "reference": "YOUR_PAYMENT_REFERENCE",
  "returnUrl": "https://your-company.com/checkout?shopperOrder=12xy..",
  "sessionData": "Ab02b4c0!BQABAgBfYI29..."
}
```

#Get list of all payment methods

Get a list of available payment methods

POST/paymentMethods

API Version



Retrieves the list of available payment methods for the transaction, based on the transaction information like amount, country, and currency.

Endpoint destination URL

<https://checkout-test.adyen.com/v71/paymentMethods>

Click to copy

Header Parameters



Required



Optional

Idempotency-Keystring

A unique identifier for the message with a maximum of 64 characters (we recommend a UUID).

Request Parameters

☒ Required

☒ Optional

additionalDataobject

This field contains additional data, which may be required for a particular payment request.

The **additionalData** object consists of entries, each of which includes the key and value.

Select additionalData

▼

allowedPaymentMethodsarray[string]

List of payment methods to be presented to the shopper. To refer to payment methods, use their payment method type.

Example: **"allowedPaymentMethods":["ideal","giropay"]**

amountobject

The amount information for the transaction (in minor units). For BIN or card verification requests, set amount to 0 (zero).

Show children

blockedPaymentMethodsarray[string]

List of payment methods to be hidden from the shopper. To refer to payment methods, use their payment method type.

Example: **"blockedPaymentMethods":["ideal","giropay"]**

channelstring

The platform where a payment transaction takes place. This field can be used for filtering out payment methods that are only available on specific platforms. Possible values:

- iOS
- Android
- Web

countryCodestring

The shopper's country code.

merchantAccountstringRequired

The merchant account identifier, with which you want to process the transaction.

orderobject

The order information required for partial payments.

Show children

shopperConversionIdstring

Max length: **256**

A unique ID that can be used to associate **/paymentMethods** and **/payments** requests with the same shopper transaction, offering insights into conversion rates.

shopperLocalestring

The combination of a language code and a country code to specify the language to be used in the payment.

shopperReferencestring

Required for recurring payments. Your reference to uniquely identify this shopper, for example user ID or account ID. The value is case-sensitive and must be at least three characters.

Your reference must not include personally identifiable information (PII) such as name or email address.

splitCardFundingSourcesboolean

Boolean value indicating whether the card payment method should be split into separate debit and credit options.

storestring

Min length: 1Max length: 16

Required for Adyen for Platforms integrations if you are a platform model. This is your reference (on balance platform) or the storeReference (in the classic integration) for the ecommerce or point-of-sale store that is processing the payment.

storeFiltrationModestring

Specifies how payment methods should be filtered based on the 'store' parameter:

- 'exclusive': Only payment methods belonging to the specified 'store' are returned.
- 'inclusive': Payment methods from the 'store' and those not associated with any other store are returned.

Response parameters

After submitting a call, you receive a response message to inform you that your request was received and processed.

Depending on the HTTP status code of the response message, it is helpful to build some logic to handle any errors that a request or the system may return.

HTTP Responses

- *200 - OK*

The request has succeeded.

Show more

- *400 - Bad Request*

A problem reading or understanding the request.

Show more

- *401 - Unauthorized*

Authentication required.

Show more

- *403 - Forbidden*

Insufficient permissions to process the request.

Show more

- *422 - Unprocessable Entity*

A request validation error.

Show more

- *500 - Internal Server Error*

The server could not process the request.

Show more

```
#Get Available payment methods
```

```
# Adyen Python API Library v13.2.0
```

```
import Adyen
```

```
adyen = Adyen.Adyen()
```

```
adyen.client.xapikey = "ADYEN_API_KEY"
```

```
# For the LIVE environment, also include your liveEndpointUrlPrefix.
```

```
adyen.client.platform = "test" # The environment to use library in.
```

```
# Create the request object(s)
```

```
json_request = {
```

```
    "merchantAccount": "YOUR_MERCHANT_ACCOUNT"
```

```
}
```

```
# Send the request
```

```
result = adyen.checkout.payments_api.payment_methods(request=json_request,  
idempotency_key="UUID")
```

Json response for list of available payment methods

```
{
```

```
    "paymentMethods": [
```

```
        {
```

```
            "name": "ACH Direct Debit",
```

```
            "type": "ach"
```

```
        },
```

```
        {
```

```
            "name": "Adyen Voucher",
```

```
            "type": "adyen_test_voucher"
```

```
        },
```

```
    ]
```

```
"name": "AfterPay Invoice",  
"type": "afterpay_default"  
},  
{  
  "name": "AfterPay DirectDebit",  
  "type": "afterpay_directdebit"  
},  
{  
  "name": "Afterpay",  
  "type": "afterpaytouch"  
},  
{  
  "name": "Cards",  
  "type": "scheme"  
},  
{  
  "name": "AliPay",  
  "type": "alipay"  
},  
{  
  "name": "AliPay HK",  
  "type": "alipay_hk"  
},  
{  
  "name": "AliPay",  
  "type": "alipay_wap"  
},  
{  
  "name": "Android Pay",  
  "type": "androidpay"  
},
```

```
{
  "name": "Apple Pay",
  "type": "applepay"
},
{
  "name": "Credit Card via AsiaPay",
  "type": "asiapay"
},
{
  "name": "China UnionPay",
  "type": "asiapay_unionpay"
},
{
  "name": "Baby Gift Card",
  "type": "babygiftcard"
},
{
  "name": "Baloto",
  "type": "baloto"
},
{
  "name": "BancNet",
  "type": "bancnet"
},
{
  "name": "Bank Transfer (BG)",
  "type": "bankTransfer_BG"
},
{
  "name": "Bank Transfer (CH)",
  "type": "bankTransfer_CH"
```

```
},
{
  "name": "Bank Transfer (DE)",
  "type": "bankTransfer_DE"
},
{
  "name": "Bank Transfer (FI)",
  "type": "bankTransfer_FI"
},
{
  "name": "Bank Transfer (GB)",
  "type": "bankTransfer_GB"
},
{
  "name": "Bank Transfer (HU)",
  "type": "bankTransfer_HU"
},
{
  "name": "SEPA Bank Transfer",
  "type": "bankTransfer_IBAN"
},
{
  "name": "Bank Transfer (IE)",
  "type": "bankTransfer_IE"
},
{
  "name": "Electronic Bank Transfer (MX)",
  "type": "bankTransfer_MX_linked"
},
{
  "name": "Bank Transfer (MX)",
```

```
"type": "bankTransfer_MX_offline"
},
{
  "name": "Bank Transfer (NL)",
  "type": "bankTransfer_NL"
},
{
  "name": "Bank Transfer (PL)",
  "type": "bankTransfer_PL"
},
{
  "name": "Bank Transfer (SE)",
  "type": "bankTransfer_SE"
},
{
  "name": "Bank Transfer (US)",
  "type": "bankTransfer_US"
},
{
  "name": "Payconiq by Bancontact",
  "type": "bcmc_mobile"
},
{
  "name": "Bijenkorf Cadeaukaart",
  "type": "bijcadeaukaart"
},
{
  "name": "99Bill",
  "type": "bill99"
},
{
```

```
"name": "Online Banking India",  
"type": "billdesk_online"  
},  
{  
  "name": "UPI",  
  "type": "billdesk_upi"  
},  
{  
  "name": "Wallets India",  
  "type": "billdesk_wallet"  
},  
{  
  "name": "Blik",  
  "type": "blik"  
},  
{  
  "name": "Bloemen Giftcard",  
  "type": "bloemengiftcard"  
},  
{  
  "name": "Boekenbon Giftcard",  
  "type": "boekenbon"  
},  
{  
  "name": "Boleto",  
  "type": "boleto"  
},  
{  
  "name": "Boleto Bancario",  
  "type": "boletobancario_santander"  
},
```



```
{
  "name": "Bradesco",
  "type": "bradesco"
},
{
  "name": "Cash-Ticket",
  "type": "cashticket"
},
{
  "name": "CashU",
  "type": "cashu"
},
{
  "name": "CCAvenue",
  "type": "ccavenue"
},
{
  "name": "Mula Checkout",
  "type": "cellulant"
},
{
  "name": "Chasin Giftcard",
  "type": "chasingiftcard"
},
{
  "name": "Clearpay",
  "type": "clearpay"
},
{
  "name": "ClickandBuy",
  "type": "clickandbuy"
}
```

```
},
{
  "name": "Paiement en 3 fois par Cartes Bancaires",
  "type": "cofinoga_3xcb"
},
{
  "name": "Costes Giftcard",
  "type": "costesgiftcard"
},
{
  "name": "custom_settlement",
  "type": "custom_settlement"
},
{
  "name": "DANA",
  "type": "dana"
},
{
  "name": "DineroMail",
  "type": "dineromail"
},
{
  "name": "Online bank transfer.",
  "type": "directEbanking"
},
{
  "name": "Direct Debit Brazil - Banco do Brazil",
  "type": "directdebit_BR_bancodobrasil"
},
{
  "name": "Direct Debit Brazil - Bradesco",
```

```
"type": "directdebit_BR_bradesco"
},
{
  "name": "Direct Debit Brazil - Caixa Economica Federal",
  "type": "directdebit_BR_caixa"
},
{
  "name": "Direct Debit Brazil - HSBC",
  "type": "directdebit_BR_hsbc"
},
{
  "name": "Direct Debit Brazil - Itau",
  "type": "directdebit_BR_itau"
},
{
  "name": "Direct Debit Brazil - Santander",
  "type": "directdebit_BR_santander"
},
{
  "name": "BACS Direct Debit",
  "type": "directdebit_GB"
},
{
  "name": "Alfamart",
  "type": "doku_alfamart"
},
{
  "name": "BCA Bank Transfer",
  "type": "doku_bca_va"
},
{
```

```
"name": "BNI VA",  
"type": "doku_bni_va"  
},  
{  
  "name": "BRI VA",  
  "type": "doku_bri_va"  
},  
{  
  "name": "CIMB VA",  
  "type": "doku_cimb_va"  
},  
{  
  "name": "Danamon VA",  
  "type": "doku_danamon_va"  
},  
{  
  "name": "Indomaret",  
  "type": "doku_indomaret"  
},  
{  
  "name": "Mandiri VA",  
  "type": "doku_mandiri_va"  
},  
{  
  "name": "OVO",  
  "type": "doku_ovo"  
},  
{  
  "name": "Bank Transfer",  
  "type": "doku_permata_lite_atm"  
},
```

```
{
  "name": "DOKU wallet",
  "type": "doku_wallet"
},
{
  "name": "Local Polish Payment Methods",
  "type": "dotpay"
},
{
  "name": "Dragonpay Prepaid Credits",
  "type": "dragonpay_credits"
},
{
  "name": "Online Banking",
  "type": "dragonpay_ebanking"
},
{
  "name": "GCash",
  "type": "dragonpay_gcash"
},
{
  "name": "Over The Counter Banks",
  "type": "dragonpay_otc_banking"
},
{
  "name": "OTC non-Bank via Dragonpay",
  "type": "dragonpay_otc_non_banking"
},
{
  "name": "Convenience Stores",
  "type": "dragonpay_otc_philippines"
```

```
},
{
  "name": "7/11",
  "type": "dragonpay_sevenelevan"
},
{
  "name": "eagleeye_voucher",
  "type": "eagleeye_voucher"
},
{
  "name": "Finnish E-Banking",
  "type": "ebanking_FI"
},
{
  "name": "Pay-easy ATM",
  "type": "econtext_atm"
},
{
  "name": "Online Banking",
  "type": "econtext_online"
},
{
  "name": "7-Eleven",
  "type": "econtext_seven_eleven"
},
{
  "name": "Convenience Stores",
  "type": "econtext_stores"
},
{
  "name": "eft_directdebit_CA",
```

```
"type": "eft_directdebit_CA"
},
{
  "name": "Lastschrift (ELV)",
  "type": "elv"
},
{
  "name": "Bank Payment",
  "type": "entercash"
},
{
  "name": "Nationale Entertainment Card",
  "type": "entertainmentcard"
},
{
  "name": "EPS",
  "type": "eps"
},
{
  "name": "Expert Cadeaukaart",
  "type": "expertgiftcard"
},
{
  "name": "3x Oney",
  "type": "facilypay_3x"
},
{
  "name": "4x Oney",
  "type": "facilypay_4x"
},
{
```

```
"name": "Fashioncheque",  
"type": "fashioncheque"  
},  
{  
  "name": "Fawry",  
  "type": "fawry"  
},  
{  
  "name": "FijnCadeau",  
  "type": "fijncadeau"  
},  
{  
  "name": "Fleurop Bloemenbon",  
  "type": "fleuropbloemenbon"  
},  
{  
  "name": "Fonq Giftcard",  
  "type": "fonqgiftcard"  
},  
{  
  "name": "Gall & Gall",  
  "type": "gallgall"  
},  
{  
  "name": "GCash",  
  "type": "gcash"  
},  
{  
  "name": "Generic GiftCard",  
  "type": "genericgiftcard"  
},
```



```
{
  "name": "GiftFor2",
  "type": "giftfor2card"
},
{
  "name": "Givex",
  "type": "givex"
},
{
  "name": "Globe GCash",
  "type": "globegcash"
},
{
  "name": "Goldsmiths Card",
  "type": "goldsmithscard"
},
{
  "name": "GoPay Wallet",
  "type": "gopay_wallet"
},
{
  "name": "OVO",
  "type": "grabpay_ID"
},
{
  "name": "GrabPay",
  "type": "grabpay_PH"
},
{
  "name": "GrabPay",
  "type": "grabpay_SG"
```

```
},
{
  "name": "Hallmark Card",
  "type": "hallmarkcard"
},
{
  "name": "HDFC",
  "type": "hdfc"
},
{
  "name": "Hunkemoller Member Card",
  "type": "hmclub"
},
{
  "name": "Hunkemoller Lingerie Card",
  "type": "hmlingerie"
},
{
  "name": "iDEAL",
  "type": "ideal"
},
{
  "name": "igive",
  "type": "igive"
},
{
  "name": "Korean Account Transfer (IniPay)",
  "type": "iniciisIniPay_accounttransfer"
},
{
  "name": "Korean Credit Cards (IniPay)",
```

```
"type": "iniciisIniPay_creditcard"
},
{
  "name": "Korean Mobile Phone (IniPay)",
  "type": "iniciisIniPay_mobilephone"
},
{
  "name": "Korean Virtual Account (IniPay)",
  "type": "iniciisIniPay_virtualaccount"
},
{
  "name": "Korean Account Transfer (Mobile)",
  "type": "iniciisMobile_accounttransfer"
},
{
  "name": "Korean Credit Cards (Mobile)",
  "type": "iniciisMobile_creditcard"
},
{
  "name": "Korean Mobile Phone (Mobile)",
  "type": "iniciisMobile_mobilephone"
},
{
  "name": "Korean Virtual Account (Mobile)",
  "type": "iniciisMobile_virtualaccount"
},
{
  "name": "Korean Credit Cards",
  "type": "iniciis_creditcard"
},
{
```

```
"name": "Interac® Online",  
"type": "interac"  
},  
{  
  "name": "Instant EFT",  
  "type": "ipay"  
},  
{  
  "name": "iPay88",  
  "type": "ipay88"  
},  
{  
  "name": "isracard",  
  "type": "isracard"  
},  
{  
  "name": "Phone Payment",  
  "type": "ivr"  
},  
{  
  "name": "Landline phone",  
  "type": "ivrLandline"  
},  
{  
  "name": "Mobile phone",  
  "type": "ivrMobile"  
},  
{  
  "name": "Kado Wereld",  
  "type": "kadowereld"  
},
```

```
{
  "name": "KakaoPay",
  "type": "kakaopay"
},
{
  "name": "Karen Millen Card",
  "type": "karenmillen"
},
{
  "name": "Karen Millen GiftCard",
  "type": "karenmillengiftcard"
},
{
  "name": "Bank Transfer",
  "type": "kcp_banktransfer"
},
{
  "name": "Korea-issued cards",
  "type": "kcp_creditcard"
},
{
  "name": "PayCo",
  "type": "kcp_payco"
},
{
  "name": "Naver Pay",
  "type": "kcp_naverpay"
},
{
  "name": "Virtual Account via KCP",
  "type": "kcp_va"
```

```
},  
{  
  "name": "Pay later with Klarna.",  
  "type": "klarna"  
},  
{  
  "name": "Pay over time with Klarna.",  
  "type": "klarna_account"  
},  
{  
  "name": "Buy Now, Pay Later with Billie",  
  "type": "klarna_b2b"  
},  
{  
  "name": "Pay now with Klarna.",  
  "type": "klarna_paynow"  
},  
{  
  "name": "Leisure Card",  
  "type": "leisurecard"  
},  
{  
  "name": "China Credit Card",  
  "type": "lianlianpay_creditcard"  
},  
{  
  "name": "China Debit Card",  
  "type": "lianlianpay_debitcard"  
},  
{  
  "name": "China Online Banking - Credit Card",
```

```
"type": "lianlianpay_ebanking_credit"
},
{
  "name": "China Online Banking - Debit Card",
  "type": "lianlianpay_ebanking_debit"
},
{
  "name": "China Online Banking - Enterprise",
  "type": "lianlianpay_ebanking_enterprise"
},
{
  "name": "Loods5 Cadeaukaart",
  "type": "loods5giftcard"
},
{
  "name": "Loods5 Tegoedbon",
  "type": "loods5prepaidcard"
},
{
  "name": "Love2Shop GiftCard",
  "type": "love2shop"
},
{
  "name": "mada",
  "type": "mada"
},
{
  "name": "Mappin & Webb Card",
  "type": "mappinwebbcard"
},
{
```

```
"name": "MB WAY",  
"type": "mbway"  
},  
{  
  "name": "Amazon Pay",  
  "type": "amazonpay"  
},  
{  
  "name": "Mercado Pago",  
  "type": "mercadopago"  
},  
{  
  "name": "MobilePay",  
  "type": "mobilepay"  
},  
{  
  "name": "AliPay via Razer Merchant Services",  
  "type": "molpay_alipay"  
},  
{  
  "name": "7-Eleven",  
  "type": "molpay_cash"  
},  
{  
  "name": "CIMB Virtual Account",  
  "type": "molpay_cimb_va"  
},  
{  
  "name": "Malaysia E-Banking via Razer Merchant Services",  
  "type": "molpay_ebanking_MY"  
},
```



```
{
  "name": "Vietnam E-Banking",
  "type": "molpay_ebanking_VN"
},
{
  "name": "Malaysia E-Banking",
  "type": "molpay_ebanking_fpx_MY"
},
{
  "name": "eNETS Debit",
  "type": "molpay_enetsd"
},
{
  "name": "epay",
  "type": "molpay_epay"
},
{
  "name": "Esapay",
  "type": "molpay_esapay"
},
{
  "name": "MyClear FPX",
  "type": "molpay_fpx"
},
{
  "name": "Maybank2u",
  "type": "molpay_maybank2u"
},
{
  "name": "Nganluong",
  "type": "molpay_nganluong"
```

```
},  
{  
  "name": "Convenience Stores Thailand",  
  "type": "molpay_paysbuy"  
},  
{  
  "name": "MOLPoints",  
  "type": "molpay_points"  
},  
{  
  "name": "RHB Now",  
  "type": "molpay_rhb"  
},  
{  
  "name": "SAM by SingPost",  
  "type": "molpay_singpost"  
},  
{  
  "name": "MOLWallet",  
  "type": "molpay_wallet"  
},  
{  
  "name": "MoMo ATM",  
  "type": "momo_atm"  
},  
{  
  "name": "Momo Wallet",  
  "type": "momo_wallet"  
},  
{  
  "name": "Moneybookers",
```

```
"type": "moneybookers"
},
{
  "name": "Multibanco",
  "type": "multibanco"
},
{
  "name": "De Nationale Musicalcard",
  "type": "musicalcard"
},
{
  "name": "Nationale Bioscoopbon",
  "type": "nationalebioscoopbon"
},
{
  "name": "Nationale Tuinbon",
  "type": "nationaletuinbon"
},
{
  "name": "Nationale Verwen Cadeaubon",
  "type": "nationaleverwencadeaubon"
},
{
  "name": "BankAxess",
  "type": "netaxept_bankaxess"
},
{
  "name": "NETELLER",
  "type": "neteller"
},
{
```

```
"name": "Onebip",
"type": "onebip"
},
{
  "name": "One Two Three",
  "type": "onetwothree"
},
{
  "name": "Online Banking PL",
  "type": "onlineBanking_PL"
},
{
  "name": "Online banking",
  "type": "openbanking_UK"
},
{
  "name": "Oxxo",
  "type": "oxxo"
},
{
  "name": "Pathe Giftcard",
  "type": "pathegiftcard"
},
{
  "name": "PayBright",
  "type": "paybright"
},
{
  "name": "Maya Wallet",
  "type": "paymaya_wallet"
},
}
```

```
{
  "name": "PayPal",
  "type": "paypal"
},
{
  "name": "Paysafecard",
  "type": "paysafecard"
},
{
  "name": "Payshop",
  "type": "payshop"
},
{
  "name": "PayD AMT via Paythru",
  "type": "paythru_amt"
},
{
  "name": "EFT via Paythru",
  "type": "paythru_eft"
},
{
  "name": "PayTM",
  "type": "paytm"
},
{
  "name": "PayU UPI",
  "type": "payu_IN_upi"
},
{
  "name": "EFT Pro via PayU",
  "type": "payu_ZA_eftpro"
}
```

```
},  
{  
  "name": "Google Pay",  
  "type": "paywithgoogle"  
},  
{  
  "name": "pix",  
  "type": "pix"  
},  
{  
  "name": "Plastix",  
  "type": "plastix"  
},  
{  
  "name": "Pluim",  
  "type": "pluimgiftcard"  
},  
{  
  "name": "Podium Card",  
  "type": "podiumcard"  
},  
{  
  "name": "POLi",  
  "type": "poli"  
},  
{  
  "name": "PPS",  
  "type": "pps"  
},  
{  
  "name": "Primera Cadeaukaart",
```

```
"type": "primeracadeaucard"
},
{
  "name": "Illicado Gift Card",
  "type": "prosodie_illicado"
},
{
  "name": "PSE",
  "type": "pse"
},
{
  "name": "Qivi Wallet",
  "type": "qiviwallet"
},
{
  "name": "RatePay Invoice",
  "type": "ratepay"
},
{
  "name": "RatePay Direct Debit",
  "type": "ratepay_directdebit"
},
{
  "name": "Rituals Giftcard",
  "type": "rituals"
},
{
  "name": "Rob Peetoom Giftcard",
  "type": "robpeetoomgiftcard"
},
{
```

```
"name": "SafetyPay",  
"type": "safetypay"  
},  
{  
  "name": "SafetyPay Cash",  
  "type": "safetypay_cash"  
},  
{  
  "name": "Shoes&Accessories Cadeau",  
  "type": "sagiftcard"  
},  
{  
  "name": "Score Giftcard",  
  "type": "scoregiftcard"  
},  
{  
  "name": "SEB Direktbetalning",  
  "type": "sebdirectpayment"  
},  
{  
  "name": "SEPA Direct Debit",  
  "type": "sepadirectdebit"  
},  
{  
  "name": "7-Eleven",  
  "type": "seveneleven"  
},  
{  
  "name": "Premium SMS",  
  "type": "sms"  
},
```



```
{
  "name": "SVS",
  "type": "svs"
},
{
  "name": "Swish",
  "type": "swish"
},
{
  "name": "TCS Test GiftCard",
  "type": "tcstestgiftcard"
},
{
  "name": "TenPay",
  "type": "tenpay"
},
{
  "name": "The Sting Giftcard",
  "type": "thestinggiftcard"
},
{
  "name": "TrueMoney",
  "type": "truemoney"
},
{
  "name": "Trustly",
  "type": "trustly"
},
{
  "name": "Online Banking by Trustpay",
  "type": "trustpay"
}
```

```
},  
  
{  
  "name": "TWINT",  
  "type": "twint"  
},  
  
{  
  "name": "Ukash",  
  "type": "ukash"  
},  
  
{  
  "name": "UnionPay",  
  "type": "unionpay"  
},  
  
{  
  "name": "UPI Collect",  
  "type": "upi_collect"  
},  
  
{  
  "name": "Valuelink",  
  "type": "valuelink"  
},  
  
{  
  "name": "V&D Cadeaukaart",  
  "type": "vdcadeaukaart"  
},  
  
{  
  "name": "Vipps",  
  "type": "vipps"  
},  
  
{  
  "name": "Visa Checkout",
```

```
"type": "visacheckout"
},
{
  "name": "VVV Cadeaubon",
  "type": "vvvcadeaubon"
},
{
  "name": "VVV Giftcard",
  "type": "vvvgiftcard"
},
{
  "name": "Webshop Giftcard",
  "type": "webshopgiftcard"
},
{
  "name": "WeChat Pay",
  "type": "wechatpayMiniProgram"
},
{
  "name": "WeChat Pay",
  "type": "wechatpayQR"
},
{
  "name": "WeChat Pay",
  "type": "wechatpayWeb"
},
{
  "name": "WE Fashion Giftcard",
  "type": "wefashiongiftcard"
},
{
```

```
"name": "Western Union",  
"type": "westernunion"  
},  
{  
  "name": "Winkel Cheque",  
  "type": "winkelcheque"  
},  
{  
  "name": "WOS Card",  
  "type": "woscard"  
},  
{  
  "name": "Alfa-Click",  
  "type": "yandex_alfaclick"  
},  
{  
  "name": "Pay using bank card",  
  "type": "yandex_bank_card"  
},  
{  
  "name": "Cash terminals",  
  "type": "yandex_cash"  
},  
{  
  "name": "Pay using installments",  
  "type": "yandex_installments"  
},  
{  
  "name": "YooMoney",  
  "type": "yandex_money"  
},
```

```

{
  "name": "Promsvyazbank",
  "type": "yandex_promsvyazbank"
},
{
  "name": "SberPay",
  "type": "yandex_sberbank"
},
{
  "name": "WebMoney",
  "type": "yandex_webmoney"
},
{
  "name": "Your Gift",
  "type": "yourgift"
},
{
  "name": "Zip",
  "type": "zip"
}
]
}

```

#Start transaction

Start a transaction

POST/payments

API Version

v71 latest

Sends payment parameters (like amount, country, and currency) together with other required input details collected from the shopper. To know more about required parameters for specific payment methods, refer to our [payment method guides](#). The response depends on the [payment flow](#):

- For a direct flow, the response includes a `pspReference` and a `resultCode` with the payment result, for example **Authorised** or **Refused**.

- For a redirect or additional action, the response contains an `action` object.

Endpoint destination URL

<https://checkout-test.adyen.com/v71/payments>

[Click to copy](#)

Header Parameters



Required



Optional

Idempotency-Keystring

A unique identifier for the message with a maximum of 64 characters (we recommend a UUID).

Request Parameters



Required



Optional

`accountInfo`object

Shopper account information for 3D Secure 2.

For 3D Secure 2 transactions, we recommend that you include this object to increase the chances of achieving a frictionless flow.

[Show children](#)

`additionalAmount`object

If you want a [BIN or card verification](#) request to use a non-zero value, assign this value to `additionalAmount` (while the amount must be still set to 0 to trigger BIN or card verification).

Required to be in the same currency as the `amount`.

[Show children](#)

`additionalData`object

This field contains additional data, which may be required for a particular payment request.

The `additionalData` object consists of entries, each of which includes the key and value.

`amount`objectRequired

The amount information for the transaction (in [minor units](#)). For [BIN or card verification](#) requests, set amount to 0 (zero).

[Show children](#)

`applicationInfo`object

Information about your application. For more details, see [Building Adyen solutions](#).

[Show children](#)

`authenticationData`object

Data for 3DS authentication.

[Show children](#)

`bankAccount`object

The details of the bank account, from which the payment should be made.
Either `bankAccount` or `card` field must be provided in a payment request.
Show children

billingAddressobject

The address where to send the invoice.

The `billingAddress` object is required in the following scenarios. Include all of the fields within this object.

- For 3D Secure 2 transactions in all browser-based and mobile implementations.
- For cross-border payouts to and from Canada.

Show children

browserInfoobject

The shopper's browser information.

For 3D Secure, the full object is required for web integrations. For mobile app integrations, include the `userAgent` and `acceptHeader` fields to indicate that your integration can support a redirect in case a payment is routed to 3D Secure 2 redirect.

Show children

captureDelayHoursinteger

The delay between the authorisation and scheduled auto-capture, specified in hours.

channelstring

The platform where a payment transaction takes place. This field is optional for filtering out payment methods that are only available on specific platforms. If this value is not set, then we will try to infer it from the `sdkVersion` or `token`.

Possible values:

- iOS
- Android
- Web

checkoutAttemptIdstring

Max length: **256**

Checkout attempt ID that corresponds to the Id generated by the client SDK for tracking user payment journey.

companyobject

Information regarding the company.

Show children

conversionIdstring Deprecated in version 68

Use `checkoutAttemptId` instead

Conversion ID that corresponds to the Id generated by the client SDK for tracking user payment journey.

countryCodestring

Max length: **100**

The shopper country.

Format: ISO 3166-1 alpha-2 Example: NL or DE

dateOfBirthstring

The shopper's date of birth.

Format ISO-8601: YYYY-MM-DD

dccQuoteobject

The forex quote as returned in the response of the forex service.
Show children

deliverAtstring

The date and time the purchased goods should be delivered.

Format ISO 8601: YYYY-MM-DDThh:mm:ss.sssTZD

Example: 2017-07-17T13:42:40.428+01:00

deliveryAddressobject

The address where the purchased goods should be delivered.
Show children

deliveryDatestring Deprecated in version 70

Use **deliverAt** instead.

The date and time the purchased goods should be delivered.

Format ISO 8601: YYYY-MM-DDThh:mm:ss.sssTZD

Example: 2017-07-17T13:42:40.428+01:00

deviceFingerprintstring

Max length: **5000**

A string containing the shopper's device fingerprint. For more information, refer to Device fingerprinting.

enableOneClickboolean

When true and **shopperReference** is provided, the shopper will be asked if the payment details should be stored for future one-click payments.

enablePayOutboolean

When true and **shopperReference** is provided, the payment details will be tokenized for payouts.

enableRecurringboolean

When true and **shopperReference** is provided, the payment details will be stored for recurring payments where the shopper is not present, such as subscription or automatic top-up payments.

entityTypestring

The type of the entity the payment is processed for.

fraudOffsetinteger

An integer value that is added to the normal fraud score. The value can be either positive or negative.

fundOriginobject

The person or entity funding the money.
Show children

fundRecipientobject

the person or entity receiving the money
Show children

industryUsagestring

The reason for the amount update. Possible values:

- **delayedCharge**
- **noShow**

- **installment**
installmentsobject

Contains installment settings. For more information, refer to [Installments](#).
Show children

lineItemsarray[object]

Price and product information about the purchased items, to be included on the invoice sent to the shopper.
This field is required for 3x 4x Oney, Affirm, Afterpay, Clearpay, Klarna, Ratepay, Riverty, and Zip.
Show children

localizedShopperStatementobject

The **localizedShopperStatement** field lets you use dynamic values for your shopper statement in a local character set. If not supplied, left empty, or for cross-border transactions, **shopperStatement** is used. Adyen currently supports the ja-Kana character set for Visa and Mastercard payments in Japan using Japanese cards. This character set supports:

- UTF-8 based Katakana, capital letters, numbers and special characters.
- Half-width or full-width characters.

mandateobject

The mandate details to initiate recurring transaction.
Show children

mccstring

The merchant category code (MCC) is a four-digit number, which relates to a particular market segment. This code reflects the predominant activity that is conducted by the merchant.

merchantAccountstringRequired

The merchant account identifier, with which you want to process the transaction.

merchantOrderReferencestring

Max length: **1000**

This reference allows linking multiple transactions to each other for reporting purposes (i.e. order auth-rate). The reference should be unique per billing cycle. The same merchant order reference should never be reused after the first authorised attempt. If used, this field should be supplied for all incoming authorisations.

We strongly recommend you send the **merchantOrderReference** value to benefit from linking payment requests when authorisation retries take place. In addition, we recommend you provide **retry.orderAttemptNumber**, **retry.chainAttemptNumber**, and **retry.skipRetry** values in **PaymentRequest.additionalData**.

merchantRiskIndicatorobject

Additional risk fields for 3D Secure 2.

For 3D Secure 2 transactions, we recommend that you include this object to increase the chances of achieving a frictionless flow.

Show children

metadataobject

Metadata consists of entries, each of which includes a key and a value. Limits:

- Maximum 20 key-value pairs per request. When exceeding, the "177" error occurs: "Metadata size exceeds limit".
- Maximum 20 characters per key.
- Maximum 80 characters per value.

mpiDataobject

Authentication data produced by an MPI (Mastercard SecureCode, Visa Secure, or Cartes Bancaires).
Show children

orderobject

The order information required for partial payments.
Show children

orderReferencestring

When you are doing multiple partial (gift card) payments, this is the **pspReference** of the first payment. We use this to link the multiple payments to each other. As your own reference for linking multiple payments, use the **merchantOrderReference** instead.

originstring

Max length: **80**

Required for the 3D Secure 2 **channel** **Web** integration.
Set this parameter to the origin URL of the page that you are loading the 3D Secure Component from.
paymentMethodRequired

The type and required details of a payment method to use.

platformChargebackLogicobject

Defines how to book chargebacks when using [Adyen for Platforms](#).
Show children

recurringExpirystring

Date after which no further authorisations shall be performed. Only for 3D Secure 2.

recurringFrequencystring

Minimum number of days between authorisations. Only for 3D Secure 2.

recurringProcessingModelstring

Defines a recurring payment type. Required when creating a token to store payment details or using stored payment details. Allowed values:

- **Subscription** – A transaction for a fixed or variable amount, which follows a fixed schedule.
- **CardOnFile** – With a card-on-file (CoF) transaction, card details are stored to enable one-click or omnichannel journeys, or simply to streamline the checkout process. Any subscription not following a fixed schedule is also considered a card-on-file transaction.
- **UnscheduledCardOnFile** – An unscheduled card-on-file (UCoF) transaction is a transaction that occurs on a non-fixed schedule and/or have variable amounts. For example, automatic top-ups when a cardholder's balance drops below a certain amount.

redirectFromIssuerMethodstring

Specifies the redirect method (GET or POST) when redirecting back from the issuer.

redirectToIssuerMethodstring

Specifies the redirect method (GET or POST) when redirecting to the issuer.

referencestringRequired

The reference to uniquely identify a payment. This reference is used in all communication with you about the payment status. We recommend using a unique value per payment; however, it is not a requirement. If you need to provide multiple references for a transaction, separate them with hyphens ("-"). Maximum length: 80 characters.

returnUrlstringRequired

The URL to return to in case of a redirection. The format depends on the channel.

- For web, include the protocol `http://` or `https://`. You can also include your own additional query parameters, for example, shopper ID or order reference number. Example: `https://your-company.com/checkout?shopperOrder=12xy`
 - For iOS, use the custom URL for your app. To know more about setting custom URL schemes, refer to the [Apple Developer documentation](#). Example: `my-app://`
 - For Android, use a custom URL handled by an Activity on your app. You can configure it with an [intent filter](#). Example: `my-app://your.package.name`
- If the URL to return to includes non-ASCII characters, like spaces or special letters, URL encode the value.

The URL must not include personally identifiable information (PII), for example name or email address.
riskDataobject

Contains risk data, such as client-side data, used to identify risk for a transaction.

Show children

sessionValiditystring

The date and time until when the session remains valid, in [ISO 8601](#) format.

For example: 2020-07-18T15:42:40.428+01:00

shopperConversionIdstring

Max length: **256**

A unique ID that can be used to associate `/paymentMethods` and `/payments` requests with the same shopper transaction, offering insights into conversion rates.

shopperEmailstring

The shopper's email address. We recommend that you provide this data, as it is used in velocity fraud checks.

For 3D Secure 2 transactions, schemes require `shopperEmail` for all browser-based and mobile implementations.

shopperIPstring

Max length: **1000**

The shopper's IP address. In general, we recommend that you provide this data, as it is used in a number of risk checks (for instance, number of payment attempts or location-based checks).

For 3D Secure 2 transactions, schemes require `shopperIP` for all browser-based implementations. This field is also mandatory for some merchants depending on your business model. For more information, [contact Support](#).

shopperInteractionstring

Specifies the sales channel, through which the shopper gives their card details, and whether the shopper is a returning customer. For the web service API, Adyen assumes Ecommerce shopper interaction by default.

This field has the following possible values:

- **Ecommerce** - Online transactions where the cardholder is present (online). For better authorisation rates, we recommend sending the card security code (CSC) along with the request.
- **ContAuth** - Card on file and/or subscription transactions, where the cardholder is known to the merchant (returning customer). If the shopper is present (online), you can supply also the CSC to improve authorisation (one-click payment).
- **Moto** - Mail-order and telephone-order transactions where the shopper is in contact with the merchant via email or telephone.
- **POS** - Point-of-sale transactions where the shopper is physically present to make a payment using a secure payment terminal.

shopperLocalestring

The combination of a language code and a country code to specify the language to be used in the payment.

shopperNameobject

The shopper's full name.

Show children

shopperReferencestring

Min length: 3Max length: 256

Required for recurring payments. Your reference to uniquely identify this shopper, for example user ID or account ID. Minimum length: 3 characters.

Your reference must not include personally identifiable information (PII), for example name or email address.

shopperStatementstring

Max length: 10000

The text to be shown on the shopper's bank statement. We recommend sending a maximum of 22 characters, otherwise banks might truncate the string. Allowed characters: **a-z, A-Z, 0-9**, spaces, and special characters **., ' _ - ? + * /**.

socialSecurityNumberstring

The shopper's social security number.

splitsarray[object]

An array of objects specifying how to split a payment when using [Adyen for Platforms](#), [Classic Platforms integration](#), or [Issuing](#).

Show children

storestring

Min length: 1Max length: 64

Required for Adyen for Platforms integrations if you are a platform model. This is your [reference](#) (on [balance platform](#)) or the [storeReference](#) (in the [classic integration](#)) for the ecommerce or point-of-sale store that is processing the payment.

storePaymentMethodboolean

When true and **shopperReference** is provided, the payment details will be stored for future [recurring payments](#).

subMerchantsarray[object]

This field contains additional information on the submerchant, who is onboarded to an acquirer through a payment facilitator or aggregator

Show children

telephoneNumberstring

The shopper's telephone number.

threeDS2RequestDataobject

Request fields for 3D Secure 2. To check if any of the following fields are required for your integration, refer to [Online payments](#) or [Classic integration](#) documentation.

Show children

threeDSAuthenticationOnlyboolean Deprecated in version 69

Use **authenticationData.authenticationOnly** instead.

If set to true, you will only perform the [3D Secure 2 authentication](#), and not the payment authorisation.

`trustedShopper`boolean

Set to true if the payment should be routed to a trusted MID.

Response parameters

After submitting a call, you receive a response message to inform you that your request was received and processed.

Depending on the HTTP status code of the response message, it is helpful to build some logic to handle any errors that a request or the system may return.

HTTP Responses

- *200 - OK*

The request has succeeded.

Show more

- *400 - Bad Request*

A problem reading or understanding the request.

Show more

- *401 - Unauthorized*

Authentication required.

Show more

- *403 - Forbidden*

Insufficient permissions to process the request.

Show more

- *422 - Unprocessable Entity*

A request validation error.

Show more

- *500 - Internal Server Error*

The server could not process the request.

#Make an apple payment

Adyen Python API Library v13.2.0

import Adyen

adyen = Adyen.Adyen()

adyen.client.xapikey = "ADYEN_API_KEY"

For the LIVE environment, also include your liveEndpointUrlPrefix.

adyen.client.platform = "test" # The environment to use library in.

Create the request object(s)

```
json_request = {  
    "amount": {  
        "currency": "USD",  
        "value": 1000  
    },  
    "reference": "Your order number",  
    "paymentMethod": {  
        "type": "applepay",  
        "applePayToken": "VNRWtuNINEWkRCSm1xWndjMDFFbktkQU..."  
    },  
    "returnUrl": "https://your-company.com/...",  
    "merchantAccount": "YOUR_MERCHANT_ACCOUNT"  
}
```

Send the request

```
result = adyen.checkout.payments_api.payments(request=json_request, idempotency_key="UUID")
```

#Submit details of payment

Submit details for a payment

POST/payments/details

API Version

v71 latest

Submits details for a payment created using `/payments`. This step is only needed when no final state has been reached on the `/payments` request, for example when the shopper was redirected to another page to complete the payment.

Endpoint destination URL

<https://checkout-test.adyen.com/v71/payments/details>

[Click to copy](#)

Header Parameters

☒ Required

☒ Optional

Idempotency-Key string

A unique identifier for the message with a maximum of 64 characters (we recommend a UUID).

Request Parameters

☒ Required

☒ Optional

authenticationData object

Data for 3DS authentication.

Show children

details object **Required**

Use this collection to submit the details that were returned as a result of the `/payments` call.

Show children

paymentData string

Max length: 200000

Encoded payment data. For [authorizing a payment after using 3D Secure 2 Authentication-only](#):

If you received `resultCode: AuthenticationNotRequired` in the `/payments` response, use the `threeDSPaymentData` from the same response.

If you received `resultCode: AuthenticationFinished` in the `/payments` response, use the `action.paymentData` from the same response.

`threeDSAuthenticationOnly` boolean **Deprecated in version 69**

Use `authenticationData.authenticationOnly` instead.

Change the `authenticationOnly` indicator originally set in the `/payments` request. Only needs to be set if you want to modify the value set previously.

Response parameters

After submitting a call, you receive a response message to inform you that your request was received and processed.

Depending on the HTTP status code of the response message, it is helpful to build some logic to handle any errors that a request or the system may return.

HTTP Responses

- **200 - OK**

The request has succeeded.

Show more

- **400 - Bad Request**

A problem reading or understanding the request.

Show more

- *401 - Unauthorized*

Authentication required.

Show more

- *403 - Forbidden*

Insufficient permissions to process the request.

Show more

- *422 - Unprocessable Entity*

A request validation error.

Show more

- *500 - Internal Server Error*

The server could not process the request.

Show more

#Submit the redirect result

Adyen Python API Library v13.2.0

import Adyen

adyen = Adyen.Adyen()

adyen.client.xapikey = "ADYEN_API_KEY"

For the LIVE environment, also include your liveEndpointUrlPrefix.

adyen.client.platform = "test" # The environment to use library in.

Create the request object(s)

json_request = {

 "details": {

 "redirectResult": "X6XtfGC3!Y..."

 }

}

Send the request

result = adyen.checkout.payments_api.payments_details(request=json_request,
idempotency_key="UUID")

#Json response for payment authorised

```
{  
  "resultCode": "Authorised",  
  "pspReference": "V4HZ4RBFJGXXGN82"  
}
```

#Get list of all the brand on card with unencrypted card number

Adyen Python API Library v13.2.0

import Adyen

adyen = Adyen.Adyen()

adyen.client.xapikey = "ADYEN_API_KEY"

For the LIVE environment, also include your liveEndpointUrlPrefix.

adyen.client.platform = "test" # The environment to use library in.

Create the request object(s)

```
json_request = {  
  "merchantAccount": "YOUR_MERCHANT_ACCOUNT",  
  "cardNumber": "411111"  
}
```

Send the request

result = adyen.checkout.payments_api.card_details(request=json_request,
idempotency_key="UUID")

#Json response of list of all the brands on the card

```
{  
  "brands": [  
    {  
      "type": "visa",
```

```


    "supported": true
  },
  {
    "type": "cartebancaire",
    "supported": true
  }
],
"fundingSource": "CREDIT",
"isCardCommercial": false,
"issuingCountryCode": "FR"
}

```

Get the result of a payment session

[GET](#)/sessions/{sessionId}

API Version

v71 latest 

Returns the status of the payment session with the `sessionId` and `sessionResult` specified in the path.

Endpoint destination URL

<https://checkout-test.adyen.com/v71/sessions/{sessionId}>

[Click to copy](#)

Query Parameters



Required



Optional

`sessionResult` `string` **Required**

The `sessionResult` value from the Drop-in or Component.

Path Parameters



Required



Optional

`sessionId` `string` **Required**

A unique identifier of the session.

Response parameters

After submitting a call, you receive a response message to inform you that your request was received and processed.

Depending on the HTTP status code of the response message, it is helpful to build some logic to handle any errors that a request or the system may return.

HTTP Responses

- *200 - OK*

The request has succeeded.

Show more

Json response code 200 success

```
{  
  "id": "CS12345678",  
  "status": "completed"  
}
```