

EasyCard Next Generation API v1 - *Checkout Page (Redirect Page)*

The simple way to collect payments hosted by EasyCard.

Dear customer, The purchase process on an e-commerce site includes three steps: searching for products to buy, filling in delivery details and making a purchase on a payment page. The most sensitive in the process is the payment page.

One option is to enter the payment information on the site itself, but then the entire site, software and servers, must meet the PCI standard and go through a long and expensive certificate process. The more recommended option is to use the Redirect interface, the user is redirected to a payment page located on a remote server and is part of the website of a company certified by the PCI standard. This ensures that the payment page has passed testing and certification as part of that company's certification. This saves a lot of time in developing the clearing interface, Makes it easier for the site to get approval from the credit card companies and most importantly gives peace of mind that the credit data is secure and not stored at all on the trading site.

EasyCard's Redirect debit page is included in the company's PCI certification, PCI DSS Level 1 and is secured according to the strongest standards of credit card companies and ensures secure execution of credit charges.

Environments

API addresses

Environment	API	Address
Sandbox	Transactions API	https://ecng-transactions.azurewebsites.net
	Metadata API	https://ecng-profile.azurewebsites.net
	Identity Server	https://ecng-identity.azurewebsites.net
Stage	Transactions API	https://api-stage.e-c.co.il
	Metadata API	https://merchant-stage.e-c.co.il
	Identity Server	https://identity-stage.e-c.co.il
Live	Transactions API	https://api.e-c.co.il
	Metadata API	https://merchant.e-c.co.il
	Identity Server	https://identity.e-c.co.il

Please use <api_address>/swagger to get full API description

Please request our developers team for Postman examples if needed


Checkout Page addresses

Environment	Address
Sandbox	https://ecng-checkout.azurewebsites.net
Stage	https://checkout-stage.e-c.co.il
Live	https://checkout.e-c.co.il

Checkout Page - common information

There are two options to generate Checkout Page:

- classic style with query string which contains all deal parameters (which can be used first of all to support legacy systems)
- Payment-Intent based Checkout Page

 Bob's Enterprise English ▾

Goods and services from Test Wen Store: My Product

Transaction type

Regular Deal ▾

Total Sum: 100.00₪

Card Number

.....

Expiration

../..

CVV

...

Card Owner Name

John Smith

Email

vit.muromsky@gmail.com

National ID

National ID

Phone

Phone

Authorization Code (optional)

Authorization Code

Cancel payment

Pay with EasyCard

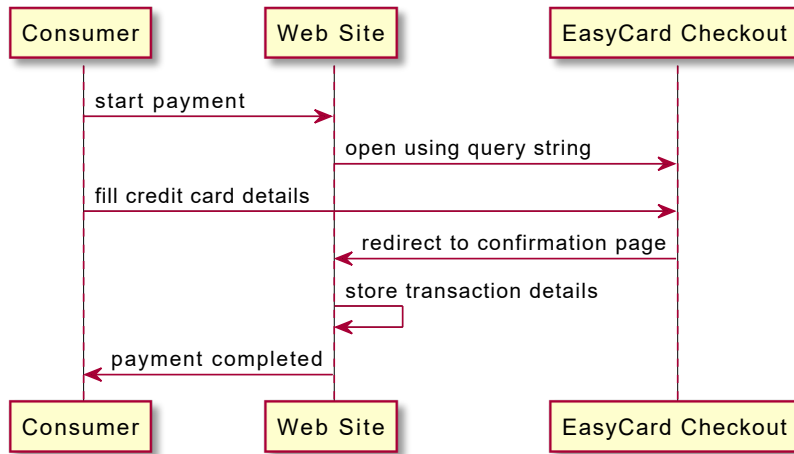
© 2021 - EasyCard

Please note that under the hood Checkout Page generates regular payment transaction, so all documentation and settings related to transactions are relevant also for transactions generated by Checkout Page - please see corresponding documentation.

Checkout Page url for live environment is <https://checkout.e-c.co.il>

Classic Checkout Page

The general behavior is next:



Checkout page can be opened inside iframe

Query string parameters

Name	In	Type	Required	Description
Description	query	string	true	deal description
Name	query	string	true	consumer name
NationalID	query	string	false	consumer national id
Email	query	string	false	consumer's email
Phone	query	string	false	consumer's phone
Currency	query	string	false	currency in ISO format, like ILS, USD etc.
Amount	query	decimal	false	deal amount, can be omitted - see description below
UserAmount	query	boolean	false	use this flag if consumer can change payment amount
RedirectUrl	query	string	false	url on your website which will accept response from EasyCard
ApiKey	query	string	true	Shared api key - see description below
IssueInvoice	query	boolean	false	invoice will be generated by EasyCard if "true"
AllowPinPad	query	boolean	false	Pinpad device payment can be used within Checkout page

Sample query string:

```

https://checkout.e-c.co.il/?
Description=Goods%20and%20services%20from%20Test%20Web%20Store%3A%20My%20Product&Name=John%20Smith&Email=testemail@gmail.com&Currency=ILS&Amount=99.99&RedirectUrl=https%3A%2F%2Fecng-testwebstore.azurewebsites.net%3A443%2FPaymentResult%3FInternalOrderID%3D123456&ApiKey=XXXXXXXX&IssueInvoice=True&AllowPinPad=False&UserAmount=False
  
```

You can use simple **Test Web Store** (demo web page) which demonstrate query string redirect: <https://ecng-testwebstore.azurewebsites.net/> If you need to specify any additional parameters, like **OrderID** or some security key in your system, you can simple add it to **RedirectUrl**

Configuration in EasyCard Merchant's Portal

On EasyCard Merchant's portal <https://merchant.e-c.co.il> there are few settings related to Checkout Page. To open the settings page you need to press on your user name link:

The screenshot shows the 'My profile' page in the EasyCard Merchant's Portal. The header includes the user name 'Vlad's terminal 29/07', the EasyCard logo, and buttons for 'Charge', 'Request', and 'Refund'. The left sidebar contains navigation links: Dashboard, Transactions, Customers, Items, Invoicing, Payment Requests, and Billing Deals. The main content area shows the user's profile information, including the app version and terminal settings. The terminal settings section includes a dropdown for 'Language' (set to English), a dropdown for 'Terminal' (set to Vlad's terminal 29/07), and a dropdown for 'Currency' (set to ₪). Below this, there is a 'Terminal Settings' section with a table showing the terminal's label, creation date, and status. The table has columns for 'Label', 'Created', and 'Status'. The first row shows 'Vlad's terminal 29/07', 'Friday, 3 April 2020 10:47', and 'Pending Approval'. Below the table, there are three toggle switches: 'Prevent Double Transactions' (disabled), 'Billing' (enabled), and 'Sms Notification' (disabled).

Redirect urls

Then you need to find **Checkout Redirect URLs** section:

The screenshot shows the 'Checkout Redirect URLs' section in the EasyCard Merchant's Portal. The section has a header 'Checkout' and a sub-header 'Checkout Redirect URLs'. Below the sub-header, there is a table with two rows of redirect URLs. The first row has a label 'Redirect URL #1' and a value 'https://ecng-testwebstore.azurewebsites.net'. The second row has a label 'Redirect URL #2' and a value 'https://localhost:3000'. To the right of the table, there is a green 'ADD' button. Above the table, there is a toggle switch for 'Issue Invoice' (enabled) and a link 'Link to CSS'.

Please add base address of your website here. You can use several base addresses. **RedirectUrl1** in query string described above should belongs to one of the configured base urls.

Custom logo and stylesheet

You can specify custom logo image and custom css file. Please use **Checkout** section and **Payment Request** section:


Payment Request

Default Request Subject
Payment request from Coca Cola
30 / 128

Default Refund Request Subject
0 / 128

From Address
no-reply@cc.co.il

Merchant Logo



Billing

Notification emails
Separate Emails with Commas
0 / 512

☒ Create Recurrent Payments Automatically


Checkout

Custom CSS
[Link to CSS](#)

☒ Issue Invoice

Checkout Redirect URLs

☐ Extended response for checkout transaction



Legacy redirect url response

In **Checkout Redirect URLs** section there is a flag **Extended response for checkout transaction** enables additional query string parameters which will be added to **RedirectUrl** by EasyCard. In this mode NextGen EasyCard will mimic EasyCard classic behavior.

Checkout Redirect URLs

☐ Extended response for checkout transaction

You can use **Test Web Store** to check all parameters which will be present in this mode.

In case if **Extended response for checkout transaction** is disabled, only EasyCard **TransactionID** will be present in response, then you can get transaction details via API (see below).

Shared API Key

You can get **ApiKey** for simple Checkout Page using **Show Shared Key** button in **Checkout** section or generate ne key using **Reset Shared Key** button:

☒ Use Quick Charge by Default

Min Installments
 Max Installments

Min Credit Installments
 Max Credit Installments

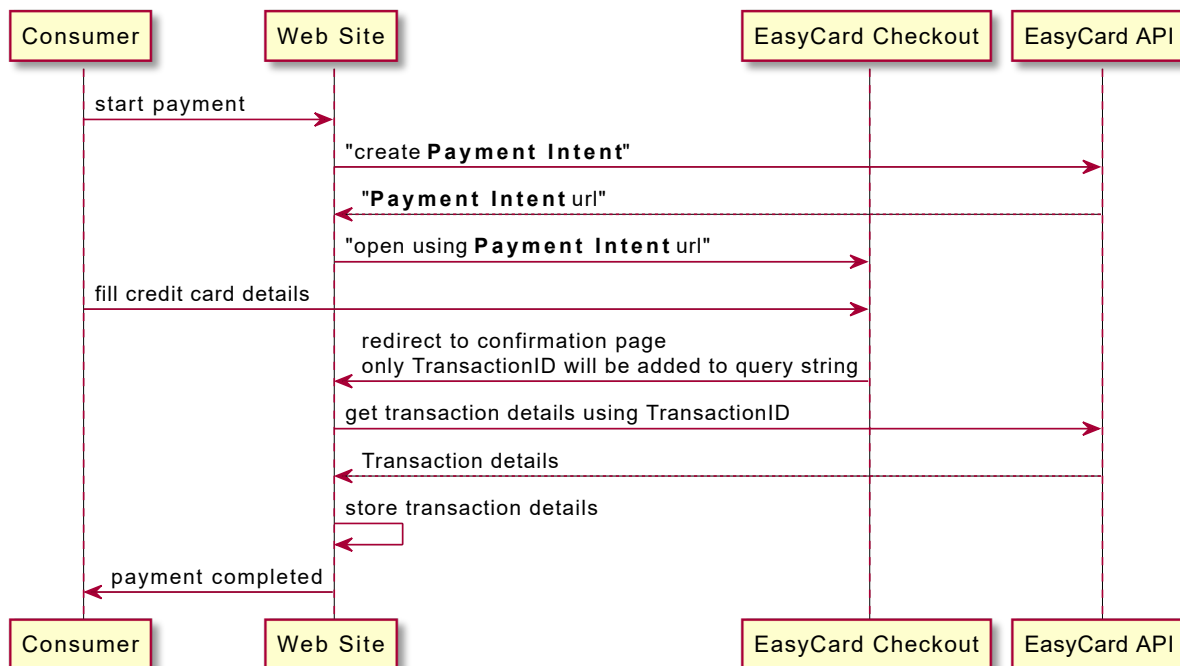
Shared Api Key

Shared Api Key

Private Api Key

Advanced Checkout Page

EasyCard Next Generation also provides more secured and solid way to collect Checkout transactions using Payment Intent API:



The general idea is to keep all information required for Checkout Page securely in structured way (which is more convenient than query string). You can use **Create Payment Intent API method** to create it. This method returns short url which you can use to redirect your consumer to Checkout Page (so, in other words this method creates payment link).

As soon consumer will enter credit card details and pressed **Pay With EasyCard** button, Checkout Page will be redirected to the link, specified in **Create Payment Intent** parameters. EasyCard **TransactionID** will be added to redirect url.

Then you can use this **TransactionID** to get payment transaction details from the API - please use **Get Transaction Details** method.

Please find out API description in section below.

Please note, all request should be secured with Authorization header **Bearer {{access_token}}** - see **Authentication** section below to get information about how to obtain this token.

You can simple check how this functionality works by using **Payment Link** feature on Merchant's Portal:

Vlad's terminal 29/07

Payment Request Settings

Due Date



Refund



Allow PinPad



Installment Transaction



Allow User Amount

Customer Name

Vlad Volkov

Customer Email

vit.muromsky@gmail.com

11 / 50

Customer Phone

12345678

Customer Address

Test Strasse

Deal Description

0 / 250

Invoice Type

Invoice With Payment Info >

Invoice Subject

Invoice

Additional Email to CC

Separate Emails with Commas

PAYMENT REQUEST

CREATE PAYMENT LINK

EasyCard Next Generation API v1 - *Transactions API (sections related to Checkout Page)*

Please check details in full API description using swagger <https://api.e-c.co.il/swagger>

Live transactions api address is <https://api.e-c.co.il>

Authentication

Authentication/Authorization type

- Bearer (oauth2)
 - Parameter Name: **Authorization**, in: header. Standard Authorization header using the Bearer scheme. Example: `bearer {access_token}`

Get Access Token

POST `{{IdentityServerAddress}}/connect/token`

Live Identity Server Address is <https://identity.e-c.co.il>

Headers

Name	Value
Content-Type	application/x-www-form-urlencoded

Body parameter (x-www-form-urlencoded)

```
<?php

$ch = curl_init();

curl_setopt($ch, CURLOPT_URL, "https://identity.e-c.co.il/api/token");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS,

"client_id=terminal&grant_type=terminal_rest_api&authorizationKey=hL5pXeI96df0wxCb....IWYcmhJH2bye1gg=="
);
curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/x-www-form-urlencoded'));

// receive server response ...
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

$server_output = curl_exec ($ch);

curl_close ($ch);

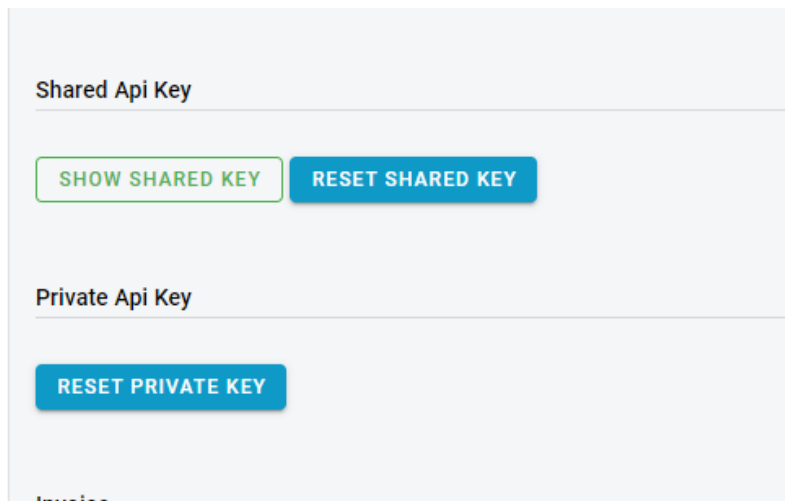
// further processing ....
if ($server_output == "OK") { ... } else { ... }

?>
```

Name	Value
client_id	terminal

Name	Value
grant_type	terminal_rest_api
authorizationKey	{{TerminalApiKey}}

You can get **TerminalApiKey** by using **Reset Private Key** button on merchant's portal:



Please note - you need to store given key securely. You cannot get active value from the merchant's portal - you only can reset current value (new key will be generated and old key will be no longer valid)

Response

```
{
  "access_token": "eyJhbGciOiJSUzI1NiImtpZCI6Ij.....q7MQ8jfwBQ_JKJif_7lTuiVhOGWA",
  "expires_in": 86400,
  "token_type": "Bearer",
  "scope": "transactions_api"
}
```

Name	Type	Description
access_token	string	Value which should be added to each API request in Authorization header with Bearer prefix Bearer {{access_token}}
expires_in	number	number of seconds before token will be expires
access_token	string	always Bearer
scope	string	API scope

Create Payment Intent (*payment link*)

Code samples

```
<?php
require 'vendor/autoload.php';

$headers = array(
```

```
'Content-Type' => 'application/json',
'Accept' => 'application/json',
);

$client = new \GuzzleHttp\Client();

// Define array of request body.
$request_body = array();

try {
    $response = $client->request('POST', '/api/paymentIntent', array(
        'headers' => $headers,
        'json' => $request_body,
    ));
    print_r($response->getBody()->getContents());
}
catch (\GuzzleHttp\Exception\BadResponseException $e) {
    // handle exception or api errors.
    print_r($e->getMessage());
}

// ...
```

POST /api/paymentIntent

Create payment link to Checkout Page (Auth policies: terminal_or_merchant_frontend_or_admin)

Body parameter

```
{
  "dealDetails": {
    "dealReference": "123456",
    "dealDescription": "some product pack: 3kg.",
    "consumerEmail": "email@example.com",
    "consumerPhone": "555-765",
    "consumerID": "7263d33c-7e4a-4043-9594-0f4b804fee03"
  },
  "currency": "USD",
  "paymentRequestAmount": 1200,
  "dueDate": "2021-12-31T00:00:00",
  "installmentDetails": {
    "numberOfPayments": 6,
    "initialPaymentAmount": 200,
    "totalAmount": 1200,
    "installmentPaymentAmount": 200
  },
  "isRefund": false,
  "redirectUrl": "https://ecng-testwebstore.azurewebsites.net/PaymentResult?
MyOrderID=123456&MySecurityCode=45678912345",
  "userAmount": true,
  "extension": {
    "customPropertyInMySystem": "MyCustomValue"
  }
}
```

Parameters

Name	In	Type	Required	Description
body	body	PaymentRequestCreate	false	Payment parameters

Example responses

201 Response

```
{
  "message": "Payment Request Created",
  "status": "success",
  "entityUID": "572da646-aadc-4c8a-9f73-8a183ed22f3f",
  "additionalData": {
    "url": "https://checkout.e-c.co.il/i?r=4xgxjJrpKhOgdJa01YhT9fMy7AHdUi1XmWra1F91bbnB0nTGB1vw%3d%3d"
  }
}
```

400 Response

```
{
  "message": "Validation Errors",
  "status": "error",
  "correlationId": "a01373f0-db4a-4433-a728-0508a6546062",
  "errors": [
    {
      "code": "paymentRequestAmount",
      "description": "Could not convert string to decimal: 1200$. Path 'paymentRequestAmount', line 10, position 33."
    }
  ]
}
```

404 Response

```
{
  "message": "Entity Not Found",
  "status": "error",
  "correlationId": "2dbc9714-7e50-4ab0-bd84-e135c039938d",
  "entityType": "Consumer"
}
```

Responses

Status	Meaning	Description	Schema
201	Created	Success	OperationResponse
400	Bad Request	Bad Request	OperationResponse
401	Unauthorized	Unauthorized	None
403	Forbidden	Forbidden	None
404	Not Found	Not Found	OperationResponse

To perform this operation, you must be authenticated by means of one of the following methods: None (Scopes: terminal_or_merchant_frontend_or_admin)

Get Transaction details

Please note, you need to find **transactionID** in the query string parameters in redirect Url (see above)

Code samples

```
<?php

require 'vendor/autoload.php';

$headers = array(
    'Accept' => 'application/json',
);

$client = new \GuzzleHttp\Client();

// Define array of request body.
$request_body = array();

try {
    $response = $client->request('GET','/api/transactions/{transactionID}', array(
        'headers' => $headers,
        'json' => $request_body,
    ))
    );
    print_r($response->getBody()->getContents());
}
catch (\GuzzleHttp\Exception\BadResponseException $e) {
    // handle exception or api errors.
    print_r($e->getMessage());
}

// ...
```

GET /api/transactions/{transactionID}

Get payment transaction details (Auth policies: terminal_or_merchant_frontend_or_admin)

Parameters

Name	In	Type	Required	Description
transactionID	path	string(uuid)	true	Payment transaction UUID

Example responses

200 Response

```
{
  "paymentTransactionID": "b4e3e791-2b73-4359-b582-9efed8f497e8",
  "transactionDate": "2021-11-29T00:00:00",
  "transactionTimestamp": "2021-11-29T17:45:18.3038024Z",
  "status": "awaitingForTransmission",
  "paymentTypeEnum": "card",
  "quickStatus": 0,
  "transactionType": "installments",
  "specialTransactionType": "regularDeal",
  "jDealType": "J4",
  "currency": "USD",
  "cardPresence": "cardNotPresent",
  "numberOfPayments": 6,
  "transactionAmount": 1200,
  "amount": 0,
  "initialPaymentAmount": 200,
  "totalAmount": 1200,
  "installmentPaymentAmount": 200,
```

```
{
  "creditCardDetails": {
    "cardNumber": "123456****1234",
    "cardExpiration": "12/22",
    "cardBrand": "VISA",
    "cardOwnerName": "John Smith",
    "cardOwnerNationalID": "1234567"
  },
  "dealDetails": {
    "dealReference": "123456",
    "dealDescription": "some product pack: 3kg.",
    "consumerEmail": "email@example.com",
    "consumerPhone": "555-765",
    "consumerID": "0f6cc8b7-95bf-4b8b-8be4-fdb6b22b1437"
  },
  "shvaTransactionDetails": {
    "shvaShovarNumber": "123456",
    "shvaDealID": "12345678901234567890"
  },
  "allowTransmission": true,
  "allowTransmissionCancellation": true,
  "vatRate": 0.17,
  "vatTotal": 174.36,
  "netTotal": 1025.64,
  "correlationId": "738579cc-026a-42c5-95f7-f316aba6064f",
  "invoiceID": "6014afae-2a3c-4397-a3cb-03c0e77ade86",
  "issueInvoice": true,
  "documentOrigin": "checkout",
  "paymentRequestID": "e135fa95-0fa2-4e26-a8e2-3a8e852f834c",
  "extension": {
    "customPropertyInMySystem": "MyCustomValue"
  }
}
```

Responses

Status	Meaning	Description	Schema
200	OK	Success	TransactionResponse
401	Unauthorized	Unauthorized	None
403	Forbidden	Forbidden	None

To perform this operation, you must be authenticated by means of one of the following methods: None (Scopes: terminal_or_merchant_frontend_or_admin)

EasyCard Next Generation API v1 - *Transactions API Schemas* (sections related to Checkout Page)

PaymentRequestCreate

Create a link to Checkout Page

Properties

Name	Type	Required	Restrictions	Description
------	------	----------	--------------	-------------

Name	Type	Required	Restrictions	Description
terminalID	string(uuid)!null	false	none	EasyCard Terminal
dealDetails	DealDetails	false	none	none
currency	CurrencyEnum	false	none	ILS
				USD
				EUR
paymentRequestAmount	number(currency)!null	false	none	Deal amount including VAT. This amount will be displayed on Checkout Page. Consumer can override this amount in case if UserAmount flag specified.
dueDate	string(date-time)!null	false	none	Due date of payment link
installmentDetails	InstallmentDetails	false	none	none
invoiceDetails	InvoiceDetails	false	none	none
issueInvoice	boolean!null	false	none	Create document - Invoice, Receipt etc. If omitted, default terminal settings will be used
allowPinPad	boolean!null	false	none	Enables PinPad button on checkout page
vatRate	number(currency)!null	false	none	Deal tax rate. Can be omitted if only PaymentRequestAmount specified - in this case VAT rate from terminal settings will be used
vatTotal	number(currency)!null	false	none	Total deal tax amount. VATTotal = NetTotal * VATRate. Can be omitted if only PaymentRequestAmount specified
netTotal	number(currency)!null	false	none	Deal amount before tax. PaymentRequestAmount = NetTotal + VATTotal. Can be omitted if only PaymentRequestAmount specified
requestSubject	string!null	false	none	You can override default email subject When sending payment link via email
fromAddress	string!null	false	none	You can override "from" address subject When sending payment link via email
isRefund	boolean	false	none	Generate link to Checkout Page to create refund
redirectUrl	string!null	false	none	Url to merchant's web site. Base url should be configured in terminal settings. You can add any details to query string.
userAmount	boolean	false	none	Consumer can override PaymentRequestAmount
cardOwnerNationalID	string!null	false	none	none
extension	object!null	false	none	Any advanced payload which will be stored in EasyCard and then can be obtained using "GetTransaction"
» additionalProperties	JToken	false	none	none

TransactionResponse

Payment transaction details

Properties

Name	Type	Required	Restrictions	Description
paymentTransactionID	string(uuid)	false	none	Primary transaction reference (UUID)
transactionDate	string(date-time) null	false	none	Legal transaction day
transactionTimestamp	string(date-time) null	false	none	Date-time when transaction created initially in UTC
initialTransactionID	string(uuid) null	false	none	Reference to initial billing deal
currentDeal	integer(int32) null	false	none	Current deal number (billing)
terminalID	string(uuid) null	false	none	EasyCard terminal UUID
terminalName	string null	false	none	EasyCard terminal name
merchantID	string(uuid) null	false	none	Merchant
status	TransactionStatusEnum	false	none	initial
				confirmedByAggregator
				confirmedByPinpadPreProcessor
				confirmedByProcessor
				awaitingForTransmission
				transmissionInProgress
				transmissionCancelingInProgress
				completed
				awaitingToSelectI5
				transmissionToProcessorFailed
				failedToCommitByAggregator
				failedToConfirmByProcesor
				failedToConfirmByAggregator
				cancelledByMerchant
				rejectedByProcessor
				rejectedByAggregator

Name	Type	Required	Restrictions	Description
paymentTypeEnum	PaymentTypeEnum	false	none	card
				cheque
				cash
				bank
quickStatus	QuickStatusFilterTypeEnum	false	none	Pending
				Completed
				Failed
				Canceled
finalizationStatus	TransactionFinalizationStatusEnum	false	none	AwaitingForTransmission
				initial
				failedToCancelByAggregator
transactionType	TransactionTypeEnum	false	none	canceledByAggregator
				regularDeal
				installments
				credit
specialTransactionType	SpecialTransactionTypeEnum	false	none	immediate
				regularDeal
				initialDeal
jDealType	JDealTypeEnum	false	none	refund
				J4
				J2
transactionJ5ExpiredDate	string(date-time) null	false	none	J5
				Transaction J5 expired date (in general after 1 day)

Name	Type	Required	Restrictions	Description
rejectionReason	RejectionReasonEnum	false	none	unknown
				creditCardIsMerchantsCard
				nationalIdIsMerchantsId
				singleTransactionAmountExceeded
				dailyAmountExceeded
				creditCardDailyUsageExceeded
				refundNotMatchRegularAmount
				refundExceededCollateral
				cardOwnerNationalIdRequired
				authCodeRequired
currency	CurrencyEnum	false	none	ILS
				USD
				EUR
cardPresence	CardPresenceEnum	false	none	cardNotPresent
				regular
numberOfPayments	integer(int32)	false	none	Number Of Installments
transactionAmount	number(double)	false	none	This transaction amount
amount	number(double)	false	none	none
initialPaymentAmount	number(double)	false	none	Initial installment payment
totalAmount	number(double)	false	none	TotalAmount = InitialPaymentAmount + (NumberOfInstallments - 1) * InstallmentPaymentAmount
installmentPaymentAmount	number(double)	false	none	Amount of one instalment payment
creditCardDetails	CreditCardDetails	false	none	Does not store full card number. Used 123456****1234 pattern
creditCardToken	string null	false	none	Stored credit card details token reference
dealDetails	DealDetails	false	none	none
shvaTransactionDetails	any	false	none	Shva details
clearingHouseTransactionDetails	any	false	none	PayDay details
upayTransactionDetails	any	false	none	none
updatedAt	string(date-time) null	false	none	Date-time when transaction status updated

Name	Type	Required	Restrictions	Description
updateTimestamp	string(byte);null	false	none	Concurrency key
allowTransmission	boolean	false	none	Transaction can be transmitted manually
allowTransmissionCancellation	boolean	false	none	Transaction transmission cannot be canceled manually
billingDealID	string(uuid);null	false	none	Reference to billing schedule which produced this transaction
rejectionMessage	string;null	false	none	Rejection Reason Message (in case of rejected transaction)
vatRate	number(double)	false	none	Deal tax rate
vatTotal	number(double)	false	none	Total deal tax amount. VATTotal = NetTotal * VATRate
netTotal	number(double)	false	none	Deal amount before tax. PaymentRequestAmount = NetTotal + VATTotal
consumerIP	string;null	false	none	We can know it from checkout page activity
merchantIP	string;null	false	none	Merchant's IP
correlationId	string;null	false	none	Request ID
invoiceID	string(uuid);null	false	none	Generated invoice ID
issueInvoice	boolean	false	none	Create document for transaction
documentOrigin	DocumentOriginEnum	false	none	Document origin (primarily payment transaction origin)
				UI (Document created manually by merchant user using Merchant's UI)
				API (Document created via API)
				checkout (Document created by consumer using Checkout Page)
				billing (Document generated based on billing schedule)
				device (Transaction created using pinpad device (or other device))
				paymentRequest (Document created by consumer using Checkout Page with a payment link)
paymentRequestID	string(uuid);null	false	none	Reference to initial payment link creation request
processorResultCode	integer(int32);null	false	none	none

Name	Type	Required	Restrictions	Description
extension	object null	false	none	Any advanced payload which will be stored in EasyCard and then can be obtained using "GetTransaction"
» additionalProperties	JToken	false	none	none

DealDetails

Additional deal information. All these data are not required and used only for merchant's business purposes.

Properties

Name	Type	Required	Restrictions	Description
dealReference	string null	false	none	Deal identifier in merchant's system
dealDescription	string null	false	none	Deal description. In case of generating payment link, these description will be displayed on Checkout Page
consumerEmail	string(email) null	false	none	End-customer Email
consumerName	string null	false	none	End-customer Name
consumerNationalID	string null	false	none	End-customer National Id
consumerPhone	string null	false	none	End-customer Phone
consumerID	string(uuid) null	false	none	End-customer record UUID in EasyCard system
items	[Item] null	false	none	Deal Items ID, Count, Name
consumerAddress	Address	false	none	none
consumerExternalReference	string null	false	none	External system consumer identifier for example RapidOne customer code

InstallmentDetails

Installment payments details

Properties

Name	Type	Required	Restrictions	Description
numberOfPayments	integer(int32)	true	none	Number Of Installments
initialPaymentAmount	number(currency)	true	none	Initial installment payment
totalAmount	number(currency)	true	none	TotalAmount = InitialPaymentAmount + (NumberOfInstallments - 1) * InstallmentPaymentAmount
installmentPaymentAmount	number(currency)	true	none	Amount of each additional payments

InvoiceDetails

Properties

Name	Type	Required	Restrictions	Description
				invoice
				invoiceWithPaymentInfo
invoiceType	InvoiceTypeEnum	false	none	creditNote
				paymentInfo
				refundInvoice
invoiceSubject	string null	false	none	none
sendCCTo	[string] null	false	none	none

Item

Properties

Name	Type	Required	Restrictions	Description
itemID	string(uuid) null	false	none	none
externalReference	string null	false	none	none
itemName	string	true	none	none
sku	string null	false	none	none
price	number(double)	true	none	none
quantity	number(currency) null	false	none	none
amount	number(currency) null	false	none	Row amount
vatRate	number(currency) null	false	none	VAT Rate
vat	number(currency) null	false	none	VAT amount
netAmount	number(currency) null	false	none	Net amount (before VAT)
discount	number(currency) null	false	none	Discount
extension	object null	false	none	none
» additionalProperties	JToken	false	none	none

Address

Properties

Name	Type	Required	Restrictions	Description
countryCode	string null	false	none	none
city	string null	false	none	none
zip	string null	false	none	none
street	string null	false	none	none
house	string null	false	none	none
apartment	string null	false	none	none

CreditCardDetails

Does not store full card number. Used 123456****1234 pattern

Properties

Name	Type	Required	Restrictions	Description
cardNumber	string	true	none	none
cardExpiration	CardExpiration	true	none	none
cardVendor	string null	false	none	none
cardBrand	string null	false	none	none
solek	string null	false	none	none
cardOwnerName	string null	false	none	none
cardOwnerNationalID	string null	false	none	none
cardReaderInput	string null	false	none	none

CardExpiration

Properties

Name	Type	Required	Restrictions	Description
year	integer(int32)	true	none	none
month	integer(int32)	true	none	none
expired	boolean	false	none	none