

EasyCard Invoice API

Authentication

- Authentication is based on tokens, authentication token must be added to the `X-AUTH-TOKEN` request header with every request
- Authentication token can be obtained in 2 ways:
 - Using customer panel
 - Using the API (described below)
- If the token is invalid, the HTTP 403 (Forbidden) is returned

Login using the API

In order to obtain the authentication token using the API make an HTTP POST request with a valid request body (JSON) to the endpoint: `/api/v1/login`.

Request body:

```
{
  "email" : "USER_EMAIL",
  "password": "USER_PASSWORD"
}
```

The token will be returned in `X-AUTH-TOKEN` response header.

Customer Management Functions

To use endpoints listed below authentication (`X-AUTH-TOKEN`) is required.

Get user info

- Endpoint: `/api/v1/user`
- Method: `GET`
- Request body: empty
- Success response: HTTP 200 (ok)
- Example response body:

```
{
  "id": 2,
  "email": "test001@wp.pl",
  "taxId": "123456",
  "name": "Maia Sharpowa",
  "address": {
    "street": "Lęborska",
    "streetNumber": "3b, pok 4.25.2",
    "city": null,
    "postalCode": null,
    "country": null,
    "countryCode": null
  },
  "phoneNumber": "444 555 666",
  "logoUrl": "http://www.thebloodycure.com/wp-content/uploads/2016/02/FaceBook-Logo-200x200.png",
  "accountEnabled": true,
  "roles": [
    "USER"
  ]
}
```

Patch user (update user data)

- Endpoint: `/api/v1/user`
- Method: `PATCH`

- Available fields to PATCH:

- name (string)
- taxId (string)
- street (string)
- streetNumber (string)
- city (string)
- postalCode (string)
- country (string)
- countryCode (string)
- phoneNumber (string)

- Example request body:

```
{
  "name": "test",
  "taxId": "123456"
}
```

- Success response: HTTP 200 (ok)
- Response body: empty

Upload user logo

- Endpoint: `/api/v1/user/logo`
- Method: `POST`
- **Required request header:** `Content-Type = multipart/form-data` (boundary part might be required as well)
- Required request param: `file`
- Accepted file formats: `.png, .jpg, .jpeg`
- Success response: HTTP 200 (ok)
- Example response body:

```
{
  "logoUrl": "https://s3-eu-west-1.amazonaws.com/invoicesystem-logos-stage/1/customer_logo"
}
```

Change user password

- Endpoint: `/api/v1/user/change-password`
- Method: `POST`
- Required request fields:
- `oldPassword` (string)
- `newPassword` (string)
- Example request body:

```
{
  "oldPassword": "OLD_PASSWORD",
  "newPassword": "NEW_PASSWORD"
}
```

- **Note:** New password must be at least 8 characters long
- Success response: HTTP 200 (ok)
- Response body: empty
- Possible errors:
- HTTP 400 (bad request) - `oldPassword` is incorrect

Get user certificate info

- Endpoint: `/api/v1/user/certificate`
- Method: `GET`
- Request body: empty
- Success response: HTTP 200 (ok)
- Example response body:

```
{
  "hasCertificate": true,
  "lastUpdated": "2016-05-16T15:36:38.000Z"
}
```

Upload certificate

- Endpoint: `/api/v1/user/certificate`
- Method: `POST`
- **Required request header:** `Content-Type = multipart/form-data` (boundary part might be required as well)
- Required request param: `file`
- Accepted file formats: `.p12, .pfx`
- Success response: `HTTP 200 (ok)`
- Response body: empty

Generate certificate

- Endpoint: `/api/v1/user/generate-certificate`
- Method: `POST`
- Available request fields:
 - `firstName` (string)
 - `lastName` (string)
 - `organizationalUnit` (string)
 - `organization` (string)
 - `city` (string)
 - `province` (string)
 - `countryCode` (string)
- **Note:** at least 1 field is required (doesn't matter which one)
- Example request body:

```
{
  "firstName": "Bright",
  "lastName": "Inventions"
}
```

- Success response: `HTTP 200 (ok)`
- Example response body:

```
{
  "keyStorePassword": "73j6q1eocso9rqr4hopih11rf7"
}
```

- **Caution** - server does **not** store the user key store password, user must store it on his own

Delete certificate

- Endpoint: `/api/v1/user/certificate`
- Method: `DELETE`
- Request body: empty
- Response body: empty
- **Info:** This will fail if user doesn't have the certificate.
- **Caution:** Deleted certificate can't be reverted!

Get document types

- Endpoint: `/api/v1/document-types`
- Method: `GET`
- Request body: empty
- Success response: `HTTP 200 (ok)`
- Example response body:

```
{
  "INVOICE": "Invoice",
  "INVOICE_WITH_PAYMENT_INFO": "Invoice with payment info",
  "CREDIT_NOTE": "Credit note",
  "PAYMENT_INFO": "Payment info",
  "REFUND_INVOICE": "Refund invoice",
  "REFUND_INVOICE_WITH_PAYMENT_INFO": "Refund invoice with payment info"
}
```

Get payment methods

- **Endpoint:** `/api/v1/payment-methods`
- **Method:** GET
- **Request body:** empty
- **Success response:** HTTP 200 (ok)
- **Example response body:**

```
{
  "CASH": "cash",
  "CHEQUE": "cheque",
  "CREDIT_CARD_REGULAR_CREDIT": "credit card regular credit",
  "CREDIT_CARD_PLUS_30": "credit card plus 30",
  "CREDIT_CARD_INSTANT_BILLING": "credit card instant billing",
  "CREDIT_CARD_CLUB_CREDIT": "credit card club credit",
  "CREDIT_CARD_SUPER_CREDIT": "credit card super credit",
  "CREDIT_CARD_CREDITS": "credit card credits",
  "CREDIT_CARD_PAYMENTS": "credit card payments",
  "CREDIT_CARD_INSTALLMENT_CLUB_DEAL": "credit card instalment club deal"
}
```

Get JSON report

- **Endpoint:** `/api/v1/report`
- **Method:** GET
- **Required request header:** `Accept = application/json`
- **Request params:**
 - `startDate` (date in format: `yyyy-MM-dd`) - **required**
 - `endDate` (date in format: `yyyy-MM-dd`) - **required**
 - `documentType` (string, available `types: INVOICE, INVOICE_WITH_PAYMENT_INFO, CREDIT_NOTE, PAYMENT_INFO, REFUND_INVOICE, REFUND_INVOICE_WITH_PAYMENT_INFO`)
 - `paymentMethod` (string, available `methods: CASH, CHEQUE, CREDIT_CARD_REGULAR_CREDIT, CREDIT_CARD_PLUS_30, CREDIT_CARD_INSTANT_BILLING, CREDIT_CARD_CLUB_CREDIT, CREDIT_CARD_SUPER_CREDIT, CREDIT_CARD_CREDITS, CREDIT_CARD_PAYMENTS, CREDIT_CARD_INSTALLMENT_CLUB_DEAL, NONE`)
 - `documentNumberStart` - (integer, find documents with `documentNumber >=` than this value)
 - `documentNumberEnd` - (integer, find documents with `documentNumber <=` than this value)
- **Request body:** empty
- **Success response:** HTTP 200 (ok)
- **Example response body:**

```
[
  {
    "issueDateTime": "2015-05-16T15:36:38.000Z",
    "customerName": "Bright Inventions",
    "documentType": "INVOICE",
    "documentNumber": 1,
    "totalNetAmount": 11,
    "totalGrossAmount": 12.87,
    "totalPaid": 12.87,
    "paymentMethods": ["CASH"]
  }
]
```

- **Possible errors:**
 - HTTP 400 (bad request) - `startDate` is after `endDate`

Get Xlsx report

- **Endpoint:** `/api/v1/report`
- **Method:** GET
- **Required request header:** `Accept = application/vnd.openxmlformats-officedocument.spreadsheetml.sheet`
- **Request params:**
 - `startDate` (date in format: `yyyy-MM-dd`) - **required**
 - `endDate` (date in format: `yyyy-MM-dd`) - **required**
 - `documentType` (string, available `types: INVOICE, INVOICE_WITH_PAYMENT_INFO, CREDIT_NOTE, PAYMENT_INFO, REFUND_INVOICE, REFUND_INVOICE_WITH_PAYMENT_INFO`)

- `paymentMethod` (string, available methods: CASH, CHEQUE, CREDIT_CARD_REGULAR_CREDIT, CREDIT_CARD_PLUS_30, CREDIT_CARD_INSTANT_BILLING, CREDIT_CARD_CLUB_CREDIT, CREDIT_CARD_SUPER_CREDIT, CREDIT_CARD_CREDITS, CREDIT_CARD_PAYMENTS, CREDIT_CARD_INSTALLMENT_CLUB_DEAL, NONE)
- `documentNumberStart` - (integer, find documents with documentNumber >= than this value)
- `documentNumberEnd` - (integer, find documents with documentNumber <= than this value)
- Request body: empty
- Success response: HTTP 200 (ok)
- Response: Xlsx file
- Possible errors:
- HTTP 400 (bad request) - `startDate` is after `endDate`

Get tax report as zip file

- Endpoint: `/api/v1/tax-report`
- Method: GET
- Request params:
 - `startDate` (date in format: yyyy-MM-dd) - **required**
 - `endDate` (date in format: yyyy-MM-dd) - **required**
- Request body: empty
- Success response: HTTP 200 (ok)
- Response: Zip file
- Possible errors:
- HTTP 400 (bad request) - `startDate` is after `endDate`

Get user next document number

- Endpoint: `/api/v1/user/document-settings`
- Method: GET
- Request params:
 - `documentType` (string, available types: INVOICE, INVOICE_WITH_PAYMENT_INFO, CREDIT_NOTE, PAYMENT_INFO, REFUND_INVOICE, REFUND_INVOICE_WITH_PAYMENT_INFO) - **required**
- Request body: empty
- Success response: HTTP 200 (ok)
- Example response body:

```
{
  "documentType": "INVOICE",
  "nextDocumentNumber": 6
}
```

Update user next document number

- Endpoint: `/api/v1/user/document-settings`
- Method: POST
- Required request fields:
 - `documentType` (string, available types: INVOICE, INVOICE_WITH_PAYMENT_INFO, CREDIT_NOTE, PAYMENT_INFO, REFUND_INVOICE, REFUND_INVOICE_WITH_PAYMENT_INFO)
 - `nextDocumentNumber` (integer)
- Example request body:

```
{
  "documentType": "PAYMENT_INFO",
  "nextDocumentNumber": 14
}
```
- Success response: HTTP 200 (ok)
- Response body: empty
- **Important:** you can only increase the next document number

Create new document

Authentication token is required to create the document ([see how to get the token](#)).

To generate document in Hebrew set request header: `Accept-language` to `he`.

In order to create a document make an HTTP POST request with a valid request body (JSON) to the endpoint: `/api/v1/docs`.

Document request body parameters:

Key	Type	Required	Description
transactionDateTime	string	no	Transaction date and time, current date and time will be applied in past.
documentType	string	yes	Available types: <code>INVOICE</code> , <code>INVOICE_WITH_PAYMENT_INFO</code> , <code>CRE</code>
customerName	string	yes	
customerTaxId	string	no	
customerAddress	<code>Address</code> object	no	<code>Address</code> object is described in details below.
customerPhoneNumber	string	no	
customerEmail	string	yes/no	Required if <code>sendEmail = true</code> (default).
currency	string	no	Currency represented in 3 letter ISO format. Default value: <code>IL</code>
description	string	no	
totalAmount	float	yes	Total gross amount (including discount).
totalNetAmount	float	yes	
discountAmount	float	yes	
totalAmountBeforeDiscount	float	yes	Total gross amount (without discount).
taxAmount	float	yes	
taxPercentage	float	no	

totalPaidAmount	float	yes	
rows	Rowarray	yes/no	List (array) of invoice rows. Row object is described in details for documentType = INVOICE, documentType = INVOICE_WITH...
payments	Paymentarray	yes/no	List (array) of invoice payments. Payment object is described for documentType = PAYMENT_INFO and documentType = INVO...
sendEmail	bool	no	Default is true (1). If user sets this field to false (0), he'll no...
sendEmailCopyToOwner	bool	no	Default is false (0).
keyStorePassword	string	yes	User (uploaded or generated via customer panel) key store p...

Address parameters:

Key	Type	Required	De
street	string	no	
streetNumber	string	no	
postalCode	string	no	
country	string	no	
countryCode	string	no	

Row parameters:

Key	Type	Required	Description
sku	string	yes	Stock keeping unit (S...
name	string	no	
price	float	yes	Gross price.
priceNet	float	yes	

taxAmount	float	yes	
quantity	integer	yes	
totalAmount	float	yes	Total gross amount.
totalNetAmount	float	yes	
totalTaxAmount	float	yes	

Payment parameters:

Key	Type	Required	Description
amount	float	yes	Payment amount.
paymentMethod	string	yes	Available methods: CASH, CHEQUE, CREDIT_CARD_REGULAR_CREDIT, CREDIT_CARD_PAYMENTS, CREDIT_CARD_CREDITS, CREDIT_CARD_PAYMENTS. L.
paymentDateTime	string	yes	Date format: yyyy-MM-ddThh:mm. Must be in past.
creditCardType	string	no	For paymentMethod = CREDIT_CARD_* only. Available types: VISA, MASTERCARD, AMERICAN_EXPRESS, DISCOVER, UNIONPAY.
creditCard4LastDigits	string	no	For paymentMethod = CREDIT_CARD_* only. Must be 4 characters.
numberOfPayments	integer	no	For paymentMethod = CREDIT_CARD_CREDITS and paymentMethod = CREDIT_CARD_PAYMENTS only.
amountOfFirstPayment	float	no	For paymentMethod = CREDIT_CARD_PAYMENTS only.
amountOfEachAdditionalPayment	float	no	For paymentMethod = CREDIT_CARD_PAYMENTS only.
chequeBank	string	no	For paymentMethod = CHEQUE only.
chequeBranch	string	no	For paymentMethod = CHEQUE only.
chequeAccount	string	no	For paymentMethod = CHEQUE only.

Example request body:

```
{
  "transactionDateTime": "2015-05-16T15:36",
  "documentType": "INVOICE",
  "customerName": "Bright Inventions",
  "customerTaxId": "123456789",
  "customerAddress": {
    "street": "Lęborska",
    "streetNumber": "3b",
    "city": "Gdańsk",
    "postalCode": "80-386",
    "country": "Poland",
    "countryCode": "PL"
  },
  "customerPhoneNumber": "555 666 777",
  "customerEmail": "mateusz.klimczak+test@brightinventions.pl",
  "description": "This is a test document",
  "totalAmount": 3.68,
  "totalNetAmount": 3.15,
  "taxAmount": 0.53,
  "taxPercentage": 14.4,
  "discountAmount": 1.0,
  "totalAmountBeforeDiscount": 4.68,
  "totalPaidAmount": 12.87,
  "sendEmail": true,
  "rows": [
    {
      "sku": "asd123",
      "name": "Fanta",
      "price": 1.17,
      "priceNet": 1.0,
      "taxAmount": 0.17,
      "quantity": 2,
      "totalAmount": 2.34,
      "totalNetAmount": 2.0,
      "totalTaxAmount": 0.34
    },
    {
      "sku": "qwe321",
      "name": "Cola",
      "price": 2.34,
      "priceNet": 2.0,
      "taxAmount": 0.34,
      "quantity": 1,
      "totalAmount": 2.34,
      "totalNetAmount": 2.0,
      "totalTaxAmount": 0.34
    }
  ],
  "payments": [
    {
      "amount": 12.87,
      "paymentMethod": "CREDIT_CARD_PAYMENTS",
      "paymentDateTime": "2015-05-16",
      "creditCardType": "VISA",
      "creditCard4LastDigits": "1234",
      "numberOfPayments": 2,
      "amountOfFirstPayment": 5.01,
      "amountOfEachAdditionalPayment": 3.93
    }
  ],
  "keyStorePassword": "123456"
}
```

Example response from server:

```
{
  "documentUrl": "https://s3-eu-west-1.amazonaws.com/invoicesystem-invoices-stage/8nacsbb7qbapv2210tgjhp0g3c/invoice_6.pdf",
  "documentCopyUrl": "https://s3-eu-west-1.amazonaws.com/invoicesystem-invoices-stage/8nacsbb7qbapv2210tgjhp0g3c/invoice_6_copy.pdf",
  "documentNumber": 6
}
```

Common errors:

- No certificate has been generated - in order to sign the documents, the certificate must be uploaded or generated via customer panel.
- `sendEmail` is set to `true` (default value) but `customerEmail` is not provided.