

WEB ESIR 1.3 API

IT Creator SLTS d.o.o.

Table of Contents

| | |
|---|-----------|
| Endpoints | 3 |
| Authentication | 3 |
| Sign in with email / password | 3 |
| Exchange a refresh token for an ID token | 4 |
| Status | 7 |
| Invoice | 13 |
| Create Invoice | 13 |
| Refund - Normal and Training | 19 |
| Invoice Examples | 20 |
| Normal Sale | 20 |
| Normal Refund | 23 |
| Copy Sale | 25 |
| Training Sale | 27 |
| Advance transactions | 31 |
| New transaction | 31 |
| Refund transaction | 32 |
| Add additional invoice to the transaction | 33 |
| Close transaction | 34 |
| Advance transaction price change | 34 |
| Invoice receipt | 38 |
| Buyer id | 40 |

Endpoints

The API endpoints are different for each country. In the following table you can find the domain for the endpoints. Please replace COUNTRY_DOMAIN with domain matching the country that the software is being developed for.

| Country | Domain |
|-----------------------------|---|
| Serbia | europe-central2-esir-44ade.cloudfunctions.net |
| Bosnia and Herzegovina (RS) | api.fissoft.bh |

Authentication

Sign in with email / password

You can sign in a user with an email and password by issuing an HTTP POST request to the Auth verifyPassword endpoint.

Method: POST

Content-Type: application/json

Endpoint: https://COUNTRY_DOMAIN/v1/accounts:signInWithPassword?key=[API_KEY]

Request Body Payload

| Property Name | Type | Description |
|-------------------|---------|--|
| email | string | The email the user is signing in with. |
| password | string | The password for the account. |
| returnSecureToken | boolean | Whether or not to return an ID and refresh token. Should always be true. |

Response Payload

| Property Name | Type | Description |
|---------------|--------|--|
| idToken | string | An Auth ID token for the authenticated user. |
| email | string | The email for the authenticated user. |
| refreshToken | string | An Auth refresh token for the authenticated user. |
| expiresIn | string | The number of seconds in which the ID token expires. |

| | | |
|------------|---------|---|
| localId | string | The uid of the authenticated user. |
| registered | boolean | Whether the email is for an existing account. |

Sample request

```
curl 'https://COUNTRY_DOMAIN/v1/accounts:signInWithPassword?key=[API_KEY]' \
-H 'Content-Type: application/json' \
--data-binary '{"email":"user@example.com","password":"[PASSWORD]","returnSecureToken":true}'
```

In the example above, you would replace **[API_KEY]** with the Web API Key provided by IT Creator, **[user@example.com]** with the user's email and **[PASSWORD]** with the user's password.

A successful request is indicated by a 200 OK HTTP status code. The response contains the ID token and refresh token associated with the existing email/password account.

Sample response

```
{
  "localId": "ZY1rJK0eYLg...",
  "email": "[user@example.com]",
  "displayName": "",
  "idToken": "[ID_TOKEN]",
  "registered": true,
  "refreshToken": "[REFRESH_TOKEN]",
  "expiresIn": "3600"
}
```

Common error codes

- EMAIL_NOT_FOUND: There is no user record corresponding to this identifier. The user may have been deleted.
- INVALID_PASSWORD: The password is invalid or the user does not have a password.
- USER_DISABLED: The user account has been disabled by an administrator.

Exchange a refresh token for an ID token

You can refresh an ID token by issuing an HTTP POST request to the `securetoken.googleapis.com` endpoint.

The user must be authenticated, and the received ID token must be used as bearer token in the HTTP Authorization header.

Method: POST

Content-Type: application/x-www-form-urlencoded

Endpoint:

`https://COUNTRY_DOMAIN/v1/token?key=[API_KEY]`

Request Body Payload

| Property Name | Type | Description |
|---------------|--------|---|
| grant_type | string | The refresh token's grant type, always "refresh_token". |
| refresh_token | string | An Auth refresh token. |

Response Payload

| Property Name | Type | Description |
|---------------|--------|--|
| expires_in | string | The number of seconds in which the ID token expires. |
| token_type | string | The type of the refresh token, always "Bearer". |
| refresh_token | string | The Auth refresh token provided in the request or a new refresh token. |
| id_token | string | An Auth ID token. |
| user_id | string | The uid corresponding to the provided ID token. |
| project_id | string | Your project ID. |

Sample request

```
curl 'https://securetoken.googleapis.com/v1/token?key=[API_KEY]' \
-H 'Content-Type: application/x-www-form-urlencoded' \
--data 'grant_type=refresh_token&refresh_token=[REFRESH_TOKEN]'
```

In the example above, you would replace **[API_KEY]** with the Web API Key provided by IT Creator, **[REFRESH_TOKEN]** with the refresh token received previously.

A successful request is indicated by a 200 OK HTTP status code. The response contains the new ID token and refresh token.

Sample response

```
{
  "expires_in": "3600",
  "token_type": "Bearer",
  "refresh_token": "[REFRESH_TOKEN]",
  "id_token": "[ID_TOKEN]",
  "user_id": "tRcfmLH7o2XrNELi...",
```

```
"project_id": "1234567890"  
}
```

Common error codes

- **TOKEN_EXPIRED:** The user's credential is no longer valid. The user must sign in again.
- **USER_DISABLED:** The user account has been disabled by an administrator.
- **USER_NOT_FOUND:** The user corresponding to the refresh token was not found. It is likely the user was deleted.
- **API key not valid.** Please pass a valid API key. (invalid API key provided)
- **INVALID_REFRESH_TOKEN:** An invalid refresh token is provided.
- **Invalid JSON payload received.** Unknown name \"refresh_tokens\": Cannot bind query parameter. Field 'refresh_tokens' could not be found in request message.
- **INVALID_GRANT_TYPE:** the grant type specified is invalid.
- **MISSING_REFRESH_TOKEN:** no refresh token provided.

Status

This service is used to get status information from SDC.

The user must be authenticated, and the received ID token must be used as bearer token in the HTTP Authorization header.

Method: GET

Content-Type: application/x-www-form-urlencoded

Endpoint:

`https://COUNTRY_DOMAIN/v1/taxcore:status?key=[API_KEY]`

Header: PAC: <PAC_value> (PAC value for the provided client authentication certificate)

Response Body

JSON formatted data in accordance with the below table.

| Field | Description | Example |
|--------------------|---|--|
| sdcDateTime | Current Local Date and Time in ISO 8601 format | 2017-08-30T11:53:05+13:00 |
| supportedLanguages | List of language-culture strings supported by SDC and intercepted with languages supported by specific environment, as per Get Environment Parameters | en-US; sr-Cyrl-RS |
| uid | UID of the client certificate (V-SDC) or inserted smart card (E-SDC) | TK7SV2AY |
| taxCoreApi | Root URL of the TaxCore.API targeted by SDC | https://api.sandbox.taxcore.dti.rs:443/ |
| currentTaxRates | An object of currently active tax rates | An object containing validFrom date, group Id and the array of currently active tax categories |
| allTaxRates | An array of all tax rates | An array containing validFrom date, group Id and tax categories for each group |

Model

```
GetStatusResponse {  
  sdcDateTime (string, optional),  
  ...  
}
```

```

supportedLanguages (Array[string], optional),
uid (string, optional),
taxCoreApi (string, optional),
currentTaxRates (TaxRateGroup, optional),
allTaxRates (Array[TaxRateGroup], optional)
}

```

Example for V-SDC

```

{
  "sdcDateTime": "2021-07-14T18:21:11.3320963+02:00",
  "supportedLanguages": [
    "en-US",
    "sr-Cyrl-RS"
  ],
  "uid": "TK7SV2AY",
  "taxCoreApi": "https://api.sandbox.taxcore.dti.rs:443/",
  "currentTaxRates": {
    "validFrom": "2020-08-03T12:56:56",
    "groupId": 2,
    "taxCategories": [
      {
        "name": "ECAL",
        "categoryType": 0,
        "taxRates": [
          {
            "rate": 5,
            "label": "D"
          },
          {
            "rate": 11,
            "label": "F"
          }
        ]
      },
      {
        "name": "N-TAX",
        "categoryType": 0,
        "taxRates": [
          {
            "rate": 0,
            "label": "N"
          }
        ]
      },
      {
        "name": "PBL",
        "categoryType": 2,
        "taxRates": [
          {
            "rate": 0.5,

```



```

        "label": "P"
      }
    ],
    "orderId": 3
  },
  {
    "name": "STT",
    "categoryType": 0,
    "taxRates": [
      {
        "rate": 6,
        "label": "E"
      }
    ],
    "orderId": 4
  },
  {
    "name": "VAT",
    "categoryType": 0,
    "taxRates": [
      {
        "rate": 9,
        "label": "A"
      },
      {
        "rate": 0,
        "label": "B"
      }
    ],
    "orderId": 5
  },
  {
    "name": "VAT-EXCL",
    "categoryType": 1,
    "taxRates": [
      {
        "rate": 0,
        "label": "C"
      }
    ],
    "orderId": 6
  }
]
},
"allTaxRates": [
  {
    "validFrom": "2020-08-02T21:04:00",
    "groupId": 1,
    "taxCategories": [
      {
        "name": "ECAL",
        "categoryType": 0,
        "taxRates": [
          {

```

```

        "rate": 5,
        "label": "D"
    },
    {
        "rate": 11,
        "label": "F"
    }
],
"orderId": 1
},
{
    "name": "N-TAX",
    "categoryType": 0,
    "taxRates": [
        {
            "rate": 0,
            "label": "N"
        }
    ],
    "orderId": 2
},
{
    "name": "PBL",
    "categoryType": 2,
    "taxRates": [
        {
            "rate": 0.2,
            "label": "P"
        }
    ],
    "orderId": 3
},
{
    "name": "STT",
    "categoryType": 0,
    "taxRates": [
        {
            "rate": 6,
            "label": "E"
        }
    ],
    "orderId": 4
},
{
    "name": "VAT",
    "categoryType": 0,
    "taxRates": [
        {
            "rate": 9,
            "label": "A"
        },
        {
            "rate": 0,
            "label": "B"
        }
    ]
}

```

```

        }
      ],
      "orderId": 5
    },
    {
      "name": "VAT-EXCL",
      "categoryType": 1,
      "taxRates": [
        {
          "rate": 0,
          "label": "C"
        }
      ],
      "orderId": 6
    }
  ]
},
{
  "validFrom": "2022-01-01T00:00:00Z",
  "groupId": 3,
  "taxCategories": [
    {
      "name": "ECAL",
      "categoryType": 0,
      "taxRates": [
        {
          "rate": 10,
          "label": "F"
        }
      ],
      "orderId": 5
    },
    {
      "name": "PB",
      "categoryType": 2,
      "taxRates": [
        {
          "rate": 0.1,
          "label": "P"
        }
      ],
      "orderId": 6
    },
    {
      "name": "STT",
      "categoryType": 0,
      "taxRates": [
        {
          "rate": 1,
          "label": "E"
        }
      ],
      "orderId": 2
    }
  ],

```

```
{
  {
    {
      "name": "VAT",
      "categoryType": 0,
      "taxRates": [
        {
          "rate": 9,
          "label": "A"
        }
      ],
      "orderId": 1
    }
  ]
}
```

Invoice

Create Invoice

This section contains the description of the Create Invoice command.

This command enables applying a digital signature on the transaction data received from the taxpayer's invoicing system. It provides non-repudiation of the signed invoice, i.e. the taxpayer cannot deny the content of the invoice nor that the invoice was signed using his/her secure element.

The user must be authenticated, and the received ID token must be used as bearer token in the HTTP Authorization header.

Method: POST

Content-Type: application/x-www-form-urlencoded

Endpoint:

https://COUNTRY_DOMAIN/v1/taxcore:invoices?key=[API_KEY]

Header:

PAC: <PAC_value> (PAC value for the provided client authentication certificate)

Content-Type: 'application/json'

Request Body Payload

For all prices use two decimal representations, and for quantities three decimals.

| Field | Optional/Mandatory | Description |
|--------------------|--------------------|---|
| dateAndTimeOfIssue | Optional | Current Local Date and Time in ISO 8601 format. This is an optional element on the API level. However, tax legislation in certain jurisdictions might mandate including it on invoices - so make sure you become familiarized with the requirements in each jurisdiction. |
| invoiceType | Mandatory | Invoice Type enumeration value: 0 - Normal, 1 - Proforma, 2 - Copy, 3 - Training, 4 - Advance |
| transactionType | Mandatory | Transaction Type enumeration value: 0 - Sale, 1 - Refund (for refund please refer to Refund - Normal and Training and Normal Refund) |

| | | |
|------------------------|--|---|
| payment | Mandatory | List of Payments for the invoice, where each Payment defines it's method and amount. In cases when the cashier needs to give change to the customer, the amount submitted in the payment field must be the actual amount that stays in the cash register (not the total amount that was given by the customer to the cashier). |
| cashier | Mandatory | Cashier's identification. |
| buyerId | Optional | Unique identification of the buyer/customer. It is mandatory only if there is a legal obligation for buyer identification (in both B2C and B2B transactions); otherwise, it's optional. Please refer to the section Buyer id . |
| buyerCostCenterId | Optional | Cost Center ID provided by the buyer to the cashier or an identification of the invoice purpose. It is mandatory only if there is a legal obligation for invoice purpose identification (in both B2C and B2B transactions); otherwise, it's optional. |
| invoiceNumber | Optional | Invoice number generated by a POS or the software version of the accredited POS. This is an optional element on the API level. However, tax legislation in certain jurisdictions might mandate including it on invoices. Make sure you become familiarized with specific jurisdiction requirements regarding its use and content. |
| referentDocumentNumber | Mandatory for all Copies and Refunds and some Normal Sale or Advance Sale invoices | Mandatory only in case Invoice Type is Refund, Copy or Advance Sale connected to an Advance Sale (other jurisdiction-specific rules may apply). In all cases, this field must contain the SDC Invoice Number of the previously issued invoice. In any other case this field is optional. Must be in the requestedBy-signedBy-Ordinal_Number format. Unicode MaxLength: 50 |
| referentDocumentDT | Optional | SDC date and time (local date and time in ISO 8601 format) of the document referenced in the referentDocumentNumber field. It is used to calculate taxes on the date of issue of the original document that is refunded or copied. If it is not provided in the request, SDC uses the currently active tax rates. |

| | | |
|-----------|-----------|---|
| items (n) | Mandatory | Each invoice contains at least one Item in Items collection (E-SDC should support minimum 250, recommended up to 500) |
| options | Optional | <p>Key/value collection defines the output of E-SDC invoice fiscalization, to optimize resources.</p> <p>Key: omitQRCodeGen Value: "1" to omit QR Code generation by E-SDC and "0" to generate and return QR code to POS. QR code <u>must not</u> be submitted to the tax authority as part of an audit package.</p> <p>Key: omitTextualRepresentation Value: "1" to omit generation of textual representation by E-SDC and "0" to generate return textual representation to POS. Textual representation of an invoice (journal) <u>must</u> be submitted to the tax authority as part of an audit package.</p> |

Item:

Each Item represents one line item on the invoice.

| Field | Optional/Mandatory | Description |
|-----------|--------------------|---|
| gtin | Optional | Global Trade Item Number (GTIN) is an identifier for trade items, incorporating the ISBN, ISSN, ISMN, IAN (which includes the European Article Number and Japanese Article Number) and some Universal Product Codes, into a universal number space. |
| name | Mandatory | Human-readable name of the product or service. |
| quantity | Mandatory | The quantity of an item, with a maximum of 3 decimals. Example: 2 (pieces), 0.100 (grams). |
| unitPrice | Mandatory | Unit price of the line item. It does not take part in tax calculation. End user price. |
| labels | Mandatory | The array of labels. Each Label represents one of the Tax Rates applied on the invoice item. Tax Items are calculated based on totalAmount and applied labels as described in the Calculate Taxes section. This field is |

| | | |
|-------------|-----------|--|
| | | mandatory (i.e. the caller must submit a non-empty collection) for each item, even when the price is 0.00. |
| totalAmount | Mandatory | Gross price for the line item. End user price. |

Payment:

| Field | Optional/Mandatory | Description |
|-------------|--------------------|--|
| amount | Mandatory | Decimal amount of the payment |
| paymentType | Mandatory | Payment Type enumeration value: 0 - Other, 1 - Cash, 2 - Card, 3 - Check, 4 - Wire Transfer, 5 - Voucher, 6 - Mobile Money |

Model

```

InvoiceRequest {
  dateAndTimeOfIssue (string, optional),
  cashier (string) Unicode MaxLength:50,
  buyerId (string, optional) ASCII MaxLength:20,
  buyerCostCenterId (string, optional) Unicode MaxLength:50,
  invoiceType (string) = ['Normal', 'ProForma', 'Copy', 'Training', 'Advance'] (int) =
[0,1,2,3,4],
  transactionType (string) = ['Sale', 'Refund'] (int) = [0,1],
  payment (Array[Payment]),
  invoiceNumber (string, optional) Unicode MaxLength:60,
  referentDocumentNumber (string, optional) Unicode MaxLength: 50,
  referentDocumentDT (string, optional),
  options (inline_model, optional),
  items (Array[Item])
}
Payment {
  amount (number),
  paymentType (string) = ['Other', 'Cash', 'Card', 'Check', 'WireTransfer', 'Voucher',
'MobileMoney'] (int) = [0,1,2,3,4,5,6]
}
inline_model {
omitQRCodeGen (string, optional) = ["0", "1"],
omitTextualRepresentation (string, optional) = ["0", "1"]
}
Item {
  gtin (string, optional) MinLength:8 MaxLength:14,
  name (string) Unicode MaxLength:2048,
  quantity (number) Decimal(14,3) MinValue:0.001,
  labels (Array[string]) MinLength:1,
  unitPrice (number) Decimal(14,2),

```



```
totalAmount (number) Decimal(14,2)
}
```

Response Payload

Headers

The following HTTP headers shall be returned in response

Data Fields

| Field | Description |
|-------------------------|---|
| requestedBy | UID of client's Secure Element digital certificate. |
| signedBy | UID of SDC's Secure Element digital certificate. |
| sdcdatetime | Local date and time in ISO 8601 format provided by E-SDC. |
| invoiceCounter | Invoice Counter in format transactionTypeCounter/totalCounter invoiceCounterExtension For Example: 14/17NS |
| invoiceCounterExtension | First letters of Invoice Type and Transaction Type of the invoice. NS for Normal Sale, CR – Copy Refund, TS – Training Sale, etc. |
| invoiceNumber | SDC Invoice Number in format requestedBy-signedBy-totalCounter |
| verificationUrl | VerificationURL generated in the <i>Create Verification URL</i> process |
| verificationQRCode | Base64 encoded byte array of GIF image created in the Create QR Code process |
| journal | Textual Representation of the invoice created in the <i>Create a Textual Representation of an Invoice (Receipt)</i> process |
| totalCounter | Total number of invoices signed by Secure Element. Returned by <i>Sign Invoice</i> APDU command |
| transactionTypeCounter | Total number of invoices for a requested type. Returned by <i>Sign Invoice</i> APDU command |
| totalAmount | Sum of all Items – total payable by the customer |
| encryptedInternalData | Base64 encoded byte array returned by <i>Sign Invoice</i> APDU command |
| signature | Base64 encoded byte array returned by <i>Sign Invoice</i> APDU command |
| taxItems | Array of TaxItem entities |
| businessName | Taxpayer Business Name obtained from digital certificate subject field |
| locationName | Location Name obtained from digital certificate subject field |

| | |
|---------------------|--|
| address | Street address obtained from digital certificate subject field |
| tin | Tax Identification Number obtained from digital certificate subject field |
| district | District obtained from digital certificate subject field |
| taxGroupRevision | Revision of taxes used in the calculation |
| mrc | <p>Manufacturer Registration Code is mandatory for audit packages sent to the tax authority database, but it's optional for invoice response sent to POS. It always has the format MakeCode-SoftwareVersionCode-DeviceSerialNumber - all 3 elements of MRC are mandatory.</p> <p>Explanation: MakeCode - unique 2 characters received from the tax authority during accreditation - case sensitive. SoftwareVersionCode - unique 4 characters received from the tax authority during accreditation - case sensitive. DeviceSerialNumber - serial number assigned by the manufacturer for each E-SDC installation - case sensitive (ASCII, max 32 characters: a-z, A-Z, 0-9).</p> |
| messages (optional) | Custom human-readable message that shall be printed or displayed by POS. |
| PosId | The POS identifier. Required to print/show in journal as "ESIR broj" |

TaxItem

TaxItem represents tax liability on the invoice per one tax category.

| Field | Description |
|--------------|---|
| label | Tax Label (A, F, G, N, P...) |
| categoryName | Tax Category Name (e.g. VAT, Consumption) |
| categoryType | Tax Category Type (0 - Tax on net, 1 - Tax on total, 2 - Amount per quantity) |
| rate | Tax rate percentage for Label (i.e. 12.50%) |
| amount | Tax amount calculated by E-SDC during invoice fiscalization |

Model

```
InvoiceResult {
  requestedBy (string, optional),
  sdcDateTime (string),
  invoiceCounter (string, read only),
  invoiceCounterExtension (string, optional),
  invoiceNumber (string, read only),
  taxItems (Array[TaxItem], optional),
}
```

```

verificationUrl (string, optional),
verificationQRCode (string, optional),
journal (string, optional),
messages (string, optional),
signedBy (string),
encryptedInternalData (string),
signature (string, optional),
totalCounter (integer, optional),
transactionTypeCounter (integer, optional),
totalAmount (number, optional),
taxGroupRevision (integer, optional),
businessName (string, optional),
tin (string, optional),
locationName (string, optional),
address (string, optional),
district (string, optional),
mrc (string, optional)
}
TaxItem {
categoryType (integer, optional),
label (string),
amount (number),
rate (number),
categoryName (string)
}

```

Refund Full - Normal and Training

The refund action is a specific *Create Invoice* action, used to issue an invoice with transaction type: "Refund" (1). Please refer to the previous section [Create Invoice](#) and the example section [Normal Refund](#).

Similar to Normal-Sale and Training-Sale invoices, the API allows to issue Normal-Refund and Training-Refund invoices. This operation is only allowed for already issued Normal-Sale and Training-Sale invoices.

Request Body Payload

| Field | Optional/Mandatory | Description |
|-------------------|--------------------|---|
| transactionType | Mandatory | Must be 1 - Refund |
| cashier | Mandatory | Cashier's identification. |
| buyerId | Mandatory | Unique identification of the buyer/customer. Please refer to the section Buyer id . |
| buyerCostCenterId | Optional | Cost Center ID provided by the buyer to the cashier or |

| | | |
|------------------------|-----------|---|
| d | | an identification of the invoice purpose. It is mandatory only if there is a legal obligation for invoice purpose identification (in both B2C and B2B transactions); otherwise, it's optional. |
| referentDocumentNumber | Mandatory | This field must contain the SDC Invoice Number of the previously issued invoice. Must be in the requestedBy-signedBy-Ordinal_Number format. Unicode MaxLength: 50 |
| options | Optional | <p>Key/value collection defines the output of E-SDC invoice fiscalization, to optimize resources.</p> <p>Key: omitQRCodeGen Value: "1" to omit QR Code generation by E-SDC and "0" to generate and return QR code to POS. QR code <u>must not</u> be submitted to the tax authority as part of an audit package.</p> <p>Key: omitTextualRepresentation Value: "1" to omit generation of textual representation by E-SDC and "0" to generate return textual representation to POS. Textual representation of an invoice (journal) <u>must</u> be submitted to the tax authority as part of an audit package.</p> |

Refund Partial - Normal and Training

It is possible to create a partial refund invoice by adding "items" and "payment" data to the request. The request should contain only items for a refund, and they have to match the items from the original invoice. The *name* and *unitPrice* must be equal to the items on the invoice that is partially refunded, and the *quantity* must be less or equal to the item quantity from the original invoice.

Refund old fiscal invoices

If the *referentDocumentNumber* is prefixed with "XXXXXXXX-" and followed by the following pattern: {8-character-id}-{invoice-number}, it will be interpreted as a refund of an old fiscal invoice. It must contain all other fields defined in section [Create Invoice](#).

For Serbia the 8-character-id is "XXXXXXXX".

For BIH the 8-character-id represents the identifier of fiscal module: "IBFM"

Invoice Examples

Normal Sale

This is an example of Normal Sale Invoice

Request Header

PAC: 'HGDFK'

Request Body

```
{
  "dateAndTimeOfIssue": "2020-12-08T08:55:23.286Z",
  "cashier": "123456789",
  "buyerId": "RS34564565",
  "buyerCostCenterId": "567546",
  "invoiceType": "Normal",
  "transactionType": "Sale",
  "payment": [
    {
      "amount": 70.00,
      "paymentType": "Cash"
    }
  ],
  "invoiceNumber": "POS2017/998",
  "referentDocumentNumber": "",
  "options": {
    "omitQRCodeGen": "1",
    "omitTextualRepresentation": "0"
  },
  "items": [
    {
      "name": "Sport-100 Helmet, Blue",
      "quantity": 2,
      "unitPrice": 34.23,
      "labels": [
        "A"
      ],
      "totalAmount": 68.46
    }
  ]
}
```

Response Body

```
{
  "requestedBy": "WHKV8WJH",
```

```

"sdcDateTime": "2020-12-09T01:51:43.1847161+12:00",
"invoiceCounter": "6/16NS",
"invoiceCounterExtension": "NS",
"invoiceNumber": "WHKV8WJH-GFB38T00-16",
"taxItems": [
  {
    "categoryType": 0,
    "label": "A",
    "amount": 5.6527,
    "rate": 9,
    "categoryName": "VAT"
  }
],
"verificationUrl":
"https://frontendui.test.taxcore.dti.rs/v/?vl=AldIS1Y4V0pIR0ZCMzhUTzAQAAAAABgAAADhyCgAAAAA
AAAAABdkKfRIAAAABqij704lyBt75rU5DJ4m3jgOpsTwrIGH1%2BPj2xpA2vddwxGAfP4gTWhqCZ5upW
5yMziBUla%2FzbVUhbtAHDeevopB2Spjgn5lOr97ZXWzG7Jai8G%2B%2Fu74X248IKY5%2BYJ4ABtFEh
CaQNgN4QXU%2FU2DpLsuyjOiaz8AoZFLaReEBiZFiMWrtUtZtTioQn7pQB6u%2FH2bYWaTGXq5sJ07
vRLK9r99YlyvweCgPLmnmOuzz363mA3TIPX%2Bov49x869TIC%2BACsqCW5SzLZax0nlNBheSNXD7
vw03p7K%2BgBs31NsrBwBx4DdEAN3YMq2%2F6db2j9tiu6Vkf09WMxIsBtXlyVP6kYE6eOSCgZaz%2F
eCVzeLQHvD02zWBBeOxFnQhFRiY1jNmaEbHi%2B7j910%2FvXd47ui6qvKtYp%2BpahbbRZULW9kmq
xyxblzxdNxBt1tOPFKx8A5smljVvc1Hn8lu6bt2lRnEespnmJha81okRxs1V2eTlCW%2FU3%2B9P9GC23
6XAVjLURtJDiHS3RB58tyWgyBi10TI7p8xKfcN7JH%2B%2FJ%2FyiM6%2Bn9UCzMcwYmddwTcun%2B
TofbN747so2onz2ZADSRoYjfaEWgzl7GwrjrEx6YwGba4HtzwP4aqH8DWldiul6ziC4jnvZN8jvwK9JNB
mz9YA1c6%2BmllloiXxvhBcGICyRwb16oLg%3D%3D",
"verificationQRCode": null,
"journal": "===== FISCAL INVOICE =====\r\n
TIN:                HIHKLKJ\r\n
Company:             Fiskalizacija doo\r\n
Store:               Fiskalizacija doo\r\n
Address:             Milinka Milinkovica 44\r\n
District:            Vinca\r\n
Cashier TIN:         123456789\r\n
POS Number:          POS2017/998\r\n
POS Time:            08/12/2020 08:55:23\r\n
-----NORMAL SALE-----\r\n
Items\r\n
=====
Name  Price    Qty.    Total\r\n
Sport-100 Helmet, Blue (A)          \r\n
    34.23     2     68.46\r\n
-----\r\n
Total Purchase:          68.46\r\n
-----\r\n
Cash:                    70.00\r\n
=====
Label  Name  Rate    Tax\r\n
A      VAT  9.00%   5.65\r\n

```

```

-----\r\n
Total Tax:          5.65\r\n
=====
SDC Time:          09/12/2020 01:51:43\r\n
SDC Invoice No:     WHKV8WJH-GFB38T00-16\r\n
Invoice Counter:    6/16NS\r\n
=====
===== END OF FISCAL INVOICE =====\r\n",
"messages": "Success",
"signedBy": "GFB38T00",
"encryptedInternalData":
"aoo+9OCMgbe+a10QyeJt44DqbE8KyBh9fj49saQNr3XcMRgHz+IE1oagmebqVucjM4gVCGv821VIW7
QBw3nr6KQdkqY4J+ZTq/e2V1sxuyWovBvv7u+F9uPJSmOfmCeAAbRRlQmkDYDeEF1P1Ng6S7Lsozo
ms/AKGRS2kXhAYmRYjFq7VLWbU4qEJ+6UAervx9m2Fmkxl6ubCTu70Syva/fWCMr8HgoDy5p5jrs89+
t5gN05T1/qL+PcfOvUyAvgArKgluUsy2WsdJ5TQYXkjVw+78NN6eyvoAbN9TbKwcAceA3RADd2DKtv+
nW9o/bYrulZH9PVjMSLabVyMIT+pA==",
"signature":
"YE6eOSCgZaz/eCVzeLQHvD02zWBBE0xFnQhFRiY1jNmaEbHi+7j910/vXd47ui6qvKtYp+pahbbRZUL
W9kmqxyxblzxhdNxH1tOPFKx8A5smljVvc1Hn8lu6bt2IRnEespmjha81okRxs1V2eTlCw/U3+9P9GC2
36XAVjLURtJDiHS3RB58tyWgyBi10TI7p8xKfcN7JH+/J/yiM6+n9UCzMcwYmddwTcun+TofbN747so2
onz2ZADSRoYjfaEWgzl7GwrjrEx6YwGba4HtzwP4aqH8DWldiul6ziC4jnvZN8jvwK9JNBmz9YA1c6+mll
oiXxvhBcGICyRwb16oLg==",
"totalCount": 16,
"transactionTypeCounter": 6,
"totalAmount": 68.46,
"taxGroupRevision": 2,
"businessName": "Shtelovanje doo",
"tin": "108010101",
"locationName": "Shtelovanje doo",
"address": "Milinka Milinkovica 44",
"address": "Vinca",
"mrc": "99-0100-GFB38T00"
}

```

Normal Refund

Request Header

PAC: 'HGDFK'

Request Body

```

{
  "cashier": "SomeName",
  "buyerId": "10:1234567890123",
  "transactionType": "Refund",
  "referentDocumentNumber": "YTSS6KKG-HJHR9E00-305",
  "options": {
    "omitQRCodeGen": "1" ,

```

```

    "omitTextualRepresentation" : "0"
  }
}

```

Response Body

```

{
  "requestedBy": "YTSS6KKG",
  "sdcDateTime": "2021-07-15T16:20:55.6856444+02:00",
  "invoiceCounter": "30/319NR",
  "invoiceCounterExtension": "NR",
  "invoiceNumber": "YTSS6KKG-HJHR9E00-319",
  "taxItems": [
    {
      "categoryType": 0,
      "label": "A",
      "amount": 5.6527,
      "rate": 9,
      "categoryName": "VAT"
    }
  ],
  "verificationUrl":
    "https://ft8.test.taxcore.dti.rs/v/?v1=A11UU1M2S0tHSEpIUj1FTzA%2FAQAAHgAAADhyCgAAAAAABAB
    eqqKuAUAAQpSUzM0NTY0NTY1C6LKBVy3PlWQeDOEPJFcvl1xbpbxR8r6Rw%2Fo%2FkkU%2FPI54u2NYvPLXSfRad%
    2BkuAOX3BBuaiB%2Fu8p0kF%2B09QJXMy7CoH12pFVdyoKcwQYwFMACpxD0KEp5veRVgXVYenjtPO2oX16Xat4v5j
    1tgU9zx5yGs5P2M6pL8VW3mDk0Cp%2FE3ApLuiTzw9DG5yo9z6fMn7qNww17KvDnESrnj83bIcXLTfQbC4YbwwnT1
    hx9ewZ%2ButQLvB4ZFB6ybWCWF%2BnbEnJ10hmLEMMpiPVdhJ1Tu%2B01nYj2UiQJ3IvarfYBmONQuTdx4swQz92
    RNb0uImyHAG3UEGQlag6ExJKAr3LgDbecUlRw9SsybqKYfqnFZATjRvLhMbehZloA7bzI69g8mocZ%2FmX1wWjICX
    kgx%2FfQYgBgOJfQ0TG%2Fp6SuUv9MaFhJCPpASwCFWlJvkUOwAc%2B7TbtS86ieqbqVSs2e1t9jZMK%2BJOm05ZB
    eRCKwrcm8La213v6JjKnMEGKYEVpANHYpcRLao1buRbB7DW%2FmW1r1N%2BwNlaMyWnhgx5pE2%2BK1590hHHTulw
    06mD8T43QLMNq6YYUELuSoNM01jjawBjMvdy0QW1Gw8ZbH%2B2kf8VZNmmI9Avumfh%2FPGJXv5bixPjBj7szfx
    MzYZjW8Vct5kDOrnQbo4hnMZqMskRDwPsDdST2RtK87byrdV%2B5b7VbpH",
  "verificationQRCode": null,
  "journal": "===== FISCAL INVOICE =====\r\n
  TIN: RS543456345\r\n
  Company: PKBB Livada\r\n
  Store: PKBB Livada\r\n
  Address: Leskovacki drum bb\r\n
  District: Lestane\r\n
  Cashier TIN: 123456789\r\n
  Buyer TIN: RS34564565\r\n
  Buyers Cost Center: 567546\r\n
  POS Number: POS2017/998\r\n
  POS Time: 08/12/2020 08:55:23\r\n
  Ref No: YTSS6KKG-HJHR9E00-305\r\n
  Ref DT: 15/07/2021 15:57:49\r\n
  -----NORMAL REFUND-----\r\n
  Items\r\n
  =====\r\n
  Name Price Qty. Total\r\n
  Sport-100 Helmet, Blue (A) \r\n
  34.23 2 -68.46\r\n
  -----\r\n

```



```

Total Purchase:                68.46\r\n
-----\r\n
Cash:                          70.00\r\n
===== \r\n
Label      Name      Rate      Tax\r\n
A          VAT      9.00%      5.65\r\n
----- \r\n
Total Tax:                        5.65\r\n
===== \r\n
SDC Time:          15/07/2021 16:20:55\r\n
SDC Invoice No:     YTSS6KKG-HJHR9E00-319\r\n
Invoice Counter:   30/319NR\r\n
===== \r\n
===== END OF FISCAL INVOICE =====\r\n",
  "messages": "Success",
  "signedBy": "HJHR9E00",
  "encryptedInternalData":
"C6LKBVy3PlWQeD0EPJFcvl1xbpbxR8r6Rw/o/kkU/PI54u2NYvPlXSfRad+kuAOX3BbuaiB/u8p0kF+09QJXMy7C
oH12pFVdyoKcwQYwfMACpxD0KEp5veRVgXVYenjtP02oX16Xat4v5j1tgU9zx5yGs5P2M6pL8VW3mDk0Cp/E3ApLu
iTzw9DG5yo9z6fMn7qNww17KvDnESrnj83bIcXLTfQbC4YbwwnT1hx9ewZ+utQLvB4ZFB6ybWCWF+nbEnJ10hmLEM
MpiPVdhJ1Tu+01nYj2UiqJ3IvarfYBmONQuTdx4swQz92RNb0uImyhAG3UEGQlag6ExJKAr3LgA==",
  "signature":
"Nt5xSWtfd1LJuoPh+qd9kBONG8uExt6FmWgDtvMjr2Dyahxn+ZfXBaMgJeSDH99BgZsY419DRMb+npK5S/0xowEk
I+kBLAIVaUm+RQ7ABz7tNu1LzqJ5uptVKzZ7W32Nkwr4k6Y7lkF5EIrCtybwtrbXe/omMqcwQYpgRWkA0dilxEtqj
Vu5FsHsNb+ZbWuU37A2VozJaeGDHmkTb4rXn3SEcd06XDTqYPxPjdAsw2rphhQQu5Kg0w7W00NpYGOYxV3LRBbUbD
xlsf7aR/xVk2aYj0C+6Z+H88Yle/luLE+MGPuzN/EzNhmNbxVy3mQM6udBujiGcxmoyyREPA+wNw==",
  "totalCounter": 319,
  "transactionTypeCounter": 30,
  "totalAmount": 68.46,
  "taxGroupRevision": 3,
  "businessName": "PKBB Livada",
  "tin": "RS543456345",
  "locationName": "PKBB Livada",
  "address": "Leskovacki drum bb",
  "district": "Lestane",
  "mrc": "99-3000-HJHR9E00"
}

```

Copy Sale

Request Header

PAC: 'HGDFK'

Request Body

```

{
  "dateAndTimeOfIssue": "2020-12-08T08:55:23.286Z",
  "cashier": "123456789",
  "buyerId": "RS34564565",
  "buyerCostCenterId": "567546",
  "invoiceType": "Copy",
  "transactionType": "Sale",

```

```

"payment": [
  {
    "amount": 70.00,
    "paymentType": "Cash"
  }
],
"invoiceNumber": "POS2017/998",
"referentDocumentNumber": "YTSS6KKG-HJHR9E00-305",
"referentDocumentDT": "2021-07-15T15:57:49.7423334+02:00",
"options": {
  "omitQRCodeGen" : "1" ,
  "omitTextualRepresentation" : "0"
},
"items": [
  {
    "name": "Sport-100 Helmet, Blue",
    "quantity": 2,
    "unitPrice": 34.23,
    "labels": [
      "A"
    ],
    "totalAmount": 68.46
  }
]
}

```

Response Body

```

{
  "requestedBy": "YTSS6KKG",
  "sdcDateTime": "2021-07-15T16:03:35.173426+02:00",
  "invoiceCounter": "31/307CS",
  "invoiceCounterExtension": "CS",
  "invoiceNumber": "YTSS6KKG-HJHR9E00-307",
  "taxItems": [
    {
      "categoryType": 0,
      "label": "A",
      "amount": 5.6527,
      "rate": 9,
      "categoryName": "VAT"
    }
  ],
  "verificationUrl":
  "https://ft8.test.taxcore.dti.rs/v/?v1=AllUU1M2S0tHSEpIUjlFTzAzAQAaHwAAADhyCgAAAAAAAAABeq
  p614UCAApSUzM0NTY0NTY1dBgBvpa%2F5Y0qh0LYDcXTsMw9IAY40%2BpggnDtEv%2FUa1pvEa6JU4LtQ%2FIakGM
  N3BG7y7NKOBBXNXuXn0AQylsZp0WV7Kitqh4Isng6b33qYtxRdPm%2FfSLjjRapHi8UmadQ9Av6Q9MSaThEpgw5A7
  yn7vQOpWws91j%2BCnS62%2BTZWcgeIPGvWRrUXfjC%2BYATsOC5sZUAtwxu0gf8iXCEwrFdnvK9060fvggRK7q%
  2FERntkqVwayw3VIXaG%2FokKf6zgDm8mZ3xvHgHJSKnpIVUC0xI%2BMT83XG4o1Vib%2BCFBz2yLnpXwzNGLi0DZ
  N8UuzEGKsparT1XV%2FDhkz721q5bCrKwTls70qs%2F03L1rBu1LC91NUHb3Hze%2Bef9%2B8f0q0Eh4R%2F9z23L
  4k%2BNfbi5x7VCI%2FVGg1%2Brhs54hKLlvxKX4gQU2B6btR6T1dQDWkic2qdtUdTgSg4rJOYla301aEmuNn2DBY
  01mnuNx3yVZkz6DT%2FsB36XSVQm2MV1Yj8mVr2ENGtH2BJIqQ9%2BdfKoQ36KkBvOxj%2BjBftD0dsVnPY0%2BUr

```

```

X0Pxws9DSSLZZTRXs4vLMPqY99C1pR%2F1ZLaTtUu%2F1bnlc8vd3W3t811Srd90C7BvqF1nsCWUMamilHB0sKSzn
Ik%2BBgPFS%2BN%2BUhqrV4aa2Fn8Xc1iQBAsivo6MTj2czc2z24vU8qVhz%2BLpncQXmvNsji",
  "verificationQRCode": null,
  "journal": "===== THIS IS NOT A FISCAL RECEIPT =====\r\n
TIN: RS543456345\r\n
Company: PKBB Livada\r\n
Store: PKBB Livada\r\n
Address: Leskovacki drum bb\r\n
District: Lestane\r\n
Cashier TIN: 123456789\r\n
Buyer TIN: RS34564565\r\n
Buyers Cost Center: 567546\r\n
POS Number: POS2017/998\r\n
POS Time: 08/12/2020 08:55:23\r\n
Ref No: YTSS6KKG-HJHR9E00-305\r\n
Ref DT: 15/07/2021 15:57:49\r\n
-----COPY SALE-----\r\n
Items\r\n
=====
Name Price Qty. Total\r\n
Sport-100 Helmet, Blue (A) \r\n
34.23 2 68.46\r\n
-----
Total Purchase: 68.46\r\n
-----
Cash: 70.00\r\n
=====
THIS IS NOT A FISCAL INVOICE \r\n
=====
Label Name Rate Tax\r\n
A VAT 9.00% 5.65\r\n
-----
Total Tax: 5.65\r\n
=====
SDC Time: 15/07/2021 16:03:35\r\n
SDC Invoice No: YTSS6KKG-HJHR9E00-307\r\n
Invoice Counter: 31/307CS\r\n
=====
===== THIS IS NOT A FISCAL RECEIPT =====\r\n",
  "messages": "Success",
  "signedBy": "HJHR9E00",
  "encryptedInternalData":
"dBgBvpa/5Y0qh01YDcXTsMw9IAY40+pggnDtEv/Ua1pvEa6JU4LtQ/IakGMN3BG7y7NK0BbXNXuXn0AQylsZp0WV
7Kitqh4Isng6b33qYtxRdPm/fSLjjRapHi8UmadQ9Av6Q9MSaThEpwg5A7yn7vQOpWws91j+CnS62+TZWcgeIPGvW
RrUXfjC+YATsOC5sZUatwxuOgf8iXCEwrFdnvK9060fVbBgRK7q/ERNtkqVwayw3VIXaG/okKf6zgDm8mZ3xvHgHJ
SKnpIVUC0xi+Mt83XG4olVib+CFBz2yLnpXwzNGLi0DZN8UuzEGKsparT1XV/Dhkz721q5bCrKwQ==",
  "signature":
"0WzvSqz/TcvWsG6UsL3U1QdvcfN755/37x/So4SHhH/3PbcviT419uLnHtUIj9UaDX6uGzniEouW/EpfiBBTYHpu
1HpPV1ANaSJzap221R10BKDisk5iVrFSVoSa42fYMFjTWae43HfJVmTPoNP+wHfPdJVCbYxWViPyZWvYQ2C0fYEki
pD3518qhDfoqQG87GP6MF+0M52xWc9jT5StfQ/Hcz0NJIIt1lNFezi8sw+pj30KWlH+VktP01S7+VueVzy93dbe3zX
VKt304LsG+oXWewJZQxqaKUcHSwpL0ciT4GA8VL435SGqu/hprYwfxdzWJAECyK+joxOPZzNzbPQ==",
  "totalCounter": 307,
  "transactionTypeCounter": 31,
  "totalAmount": 68.46,

```

```
"taxGroupRevision": 3,
"businessName": "PKBB Livada",
"tin": "RS543456345",
"locationName": "PKBB Livada",
"address": "Leskovacki drum bb",
"district": "Lestane",
"mrc": "99-3000-HJHR9E00"
}
```

Training Sale

Request Header

PAC: 'HGDFK'

Request Body

```
{
  "dateAndTimeOfIssue": "2020-12-08T08:55:23.286Z",
  "cashier": "123456789",
  "buyerId": "RS34564565",
  "buyerCostCenterId": "567546",
  "invoiceType": "Training",
  "transactionType": "Sale",
  "payment": [
    {
      "amount": 70.00,
      "paymentType": "Cash"
    }
  ],
  "invoiceNumber": "POS2017/998",
  "referentDocumentNumber": "YTSS6KKG-HJHR9E00-305",
  "referentDocumentDT": "2021-07-15T15:57:49.7423334+02:00",
  "options": {
    "omitQRCodeGen" : "1" ,
    "omitTextualRepresentation" : "0"
  },
  "items": [
    {
      "name": "Sport-100 Helmet, Blue",
      "quantity": 2,
      "unitPrice": 34.23,
      "labels": [
        "A"
      ],
      "totalAmount": 68.46
    }
  ]
}
```

Response Body

```

{
  "requestedBy": "YTSS6KKG",
  "sdcDateTime": "2021-07-15T16:47:48.3965089+02:00",
  "invoiceCounter": "39/395TS",
  "invoiceCounterExtension": "TS",
  "invoiceNumber": "YTSS6KKG-HJHR9E00-395",
  "taxItems": [
    {
      "categoryType": 0,
      "label": "A",
      "amount": 5.6527,
      "rate": 9,
      "categoryName": "VAT"
    }
  ],
  "verificationUrl":
    "https://ft8.test.taxcore.dti.rs/v/?v1=AllUU1M2S0tHSEpIUj1FTzCLAQAAJwAAADhyCgAAAAAAAAABeq
    qjU6wDAAPsUZM0NTY0NTY1UKsZsoFmKMEW2PLX%2BEx%2FP%2FH8E1355WZPrVU%2BjrJYKjsm1dG0vvWxDVTc5pJ
    kRltQhJQsDu9ZBgthUpNxcoTjNEPuQPnb8urBi%2B3wllneYCshG9B2gS3vD790TyRXzhisPiCSIOpcJ1QZ%2B0cP
    kt2p40%2FL%2FsN93CGJ76L8PLfP2U1VSutMdaAWP4IL%2BF%2Fi2tlv1Mu%2B1SMZ7E%2FhNaVu8cco%2BsjRkLC
    PZB71s2NeH151P4A0%2Fi%2FA4%2Fmm4%2Fis2QmkkMRcbzsfDFm0irXp3fTuDsNP0mr7gM7pM9rppMP9L7PSLn95
    KCS7rYWLDHhH36xRjTxmqP21ff9IxUrqp1eQJBledkGxSSA1Uztj1q7uM1Yrtiimy1u4T4aQPxA%2BgFQ%2FB9UpE
    WQYTUhbv7V6%2BtRrrjiCwT7iNYGpdNjF7DZGDyepAe4XB7RLpFgV1C%2BXaVJ204%2FJbVZC5GdEVo25IeHd%2FN
    0YD2v%2F9J4ldWNo3yhm091VcaPS8G4CEGw9q1mUaif4wIb0RVQB9UNNQWSFszw9A7KiEReHG1k9qqNpQev5aEOfi
    wkLI6UlwdzdyLR008R5eDoShrQR5K3%2FsBG00HvMKLoZEMtTA06X1k0%2Fumpbbg0FtcGhRO24Q46tSHNM7e%2Fc
    aLA%2BNo9%2BrIYHhy2sjByrosa4X%2FVfAFb7hwIbX3xM4vDZRAT7FW8QYLfUjbwrMVanRwPm",
  "verificationQRCode": null,
  "journal": "===== THIS IS NOT A FISCAL RECEIPT =====\r\n
  TIN: RS543456345\r\n
  Company: PKBB Livada\r\n
  Store: PKBB Livada\r\n
  Address: Leskovacki drum bb\r\n
  District: Lestane\r\n
  Cashier TIN: 123456789\r\n
  Buyer TIN: RS34564565\r\n
  Buyers Cost Center: 567546\r\n
  POS Number: POS2017/998\r\n
  POS Time: 08/12/2020 08:55:23\r\n
  -----TRAINING SALE-----\r\n
  Items\r\n
  =====\r\n
  Name Price Qty. Total\r\n
  Sport-100 Helmet, Blue (A) \r\n
  34.23 2 68.46\r\n
  -----\r\n
  Total Purchase: 68.46\r\n
  -----\r\n
  Cash: 70.00\r\n
  =====\r\n
  THIS IS NOT A FISCAL INVOICE \r\n
  =====\r\n
  Label Name Rate Tax\r\n
  A VAT 9.00% 5.65\r\n
  -----\r\n

```

```

Total Tax:                    5.65\r\n
=====
SDC Time:                    15/07/2021 16:47:48\r\n
SDC Invoice No:               YTSS6KKG-HJHR9E00-395\r\n
Invoice Counter:             39/395TS\r\n
=====
===== THIS IS NOT A FISCAL RECEIPT =====\r\n",
  "messages": "Success",
  "signedBy": "HJHR9E00",
  "encryptedInternalData":
"UKsZsoFmKMEW2PLX+Ex/P/H8E1355WZPrVU+jrJYKjsm1dGOvvWxDVTC5pJkRltQhJQsDu9ZBgthUpNxcoTjNEPu
QPnb8urBi+3wllneYCsHG9B2gS3vD790TyRXzhisPiCSI0pcJ1QZ+OcPkt2p40/L/sN93CGJ76L8PLfP2U1VSutMd
aAWP4IL+F/i2tlv1Mu+1SMZ7E/hNaVu8cco+sJrkLCPZB7ls2NeH151P4A0/i/A4/mm4/is2QmkkMRcbzsfDFm0ir
Xp3fTuDsNP0mr7gM7pM9rppMP9L7PSLn95KCS7rYWLdHhH36xRjTxmqP21ff9IxUrqP1eQJBledg==",
  "signature":
"QbFJICVT02OWru4zViu2KKbLW7hPhpA/ED6AVD8H1SkRZBhNQdu/tXr61GuuOILBPuI1gal02MXsNkYPJ6kB7hcH
tEukWBWUL5dpUnbTj8ltVklkZ0RWjbkh4d383RgPa//0niV1Y2jfKGbT3VVxo9LwbgiQbD2rWZRqJ/jAhvRFVAH1Q
01BZIwzPD0DsQIRF4cbWT2qo2lB6/loQ5+LCQsJpSXB3N3ItHTTxH140hKGtBHkrf+wEbTQe8wouhkQy1MA7peWTT
+6altuDQW1waFE7bhDjq1Ic0zt79xosD42j36shgeHLayMHKuixrhf9V8AVvuHAhtffEzi8N1EBA==",
  "totalCounter": 395,
  "transactionTypeCounter": 39,
  "totalAmount": 68.46,
  "taxGroupRevision": 3,
  "businessName": "PKBB Livada",
  "tin": "RS543456345",
  "locationName": "PKBB Livada",
  "address": "Leskovacki drum bb",
  "district": "Lestane",
  "mrc": "99-3000-HJHR9E00"
}

```

Advance transactions

To create an advance invoice, due to regulations in Serbia/BIH, it is required to follow specific rules. In order to avoid any potential errors, the Advance transactions are created. The transaction represents a full historical cycle, beginning with the first advance invoice, any optional following advance invoices, any related optional refunds, and mandatory final invoice. The mandatory invoice is the final step in advance transactions, which concludes the transaction, and no further changes are possible.

New transaction

Every advance transaction begins by creating a new advance invoice. The invoice structure is exactly the same as the regular sale invoice with the addition for advance payment. Please refer to the [Invoice section](#). The *invoiceType* must be 'Advance' or (4).

The user must be authenticated, and the received ID token must be used as bearer token in the HTTP Authorization header.

Method: POST

Content-Type: application/x-www-form-urlencoded

Endpoint:

https://COUNTRY_DOMAIN/v1/taxcore/advanceTransactions/new?key=[API_KEY]

Header:

PAC: <PAC_value> (PAC value for the provided client authentication certificate)

Content-Type: 'application/json'

Request Body Payload

Please refer to the [Invoice section](#) before continuing as the following table explains only differences.

| Field | Optional/Mandatory | Description |
|-------------|--------------------|--|
| invoiceType | Mandatory | Invoice Type enumeration value. Must be: 4 - Advance |
| buyerId | Mandatory | Unique identification of the buyer/customer. It is mandatory only if there is a legal obligation for buyer identification (in both B2C and B2B transactions); otherwise, it's optional. Please refer to the section Buyer id . |

Item:

Each Item represents one line item on the invoice.

| Field | Optional/Mandatory | Description |
|---------------|--------------------|--|
| advanceAmount | Mandatory | Amount of advance payment for particular item. |

Payment:

| Field | Optional/Mandatory | Description |
|--------|--------------------|--|
| amount | Mandatory | Decimal amount of the payment. This is a sum of all advance amounts for items. |

Response Payload

The response is the same as the normal invoice payload. Please refer to the [Invoice section](#).

The *invoiceNumber* in this case represents the advance transaction **reference number**, that will be used when referring to this transaction.

Refund transaction

To refund advance invoice use the following API call:

The user must be authenticated, and the received ID token must be used as bearer token in the HTTP Authorization header.

Method: POST

Content-Type: application/x-www-form-urlencoded

Endpoint:

https://COUNTRY_DOMAIN/v1/taxcore/advanceTransactions/refund?key=[API_KEY]

Header:

PAC: <PAC_value> (PAC value for the provided client authentication certificate)

Content-Type: 'application/json'

Request Body Payload

The payload is similar to invoice payload. Please refer to the [Invoice section](#). The difference is:

1. **referentDocumentNumber**, in this field, enter the invoice number you want to refund, and
2. **buyerId**, this field must contain the customer type and its identification number, separated by ":". Example "**10:123456789**" - In this case "10" is type for legal entity identified with TIN and 123456789 is TIN example.

If you want to refund the entire transaction, pass the invoice number of the first (root) advance bill, otherwise, pass the number of the invoice you want to refund.

Response Payload

The response is the same as the normal invoice payload. Please refer to the [Invoice section](#).

Add additional invoice to the transaction

This method is used to add additional advance invoice to the existing and not closed advance transaction. The invoice structure is exactly the same as the new advance invoice, with addition of the reference to the existing transaction.

The user must be authenticated, and the received ID token must be used as bearer token in the HTTP Authorization header.

Method: POST

Content-Type: application/x-www-form-urlencoded

Endpoint:

https://COUNTRY_DOMAIN/v1/taxcore/advanceTransactions/add?key=[API_KEY]

Header:

PAC: <PAC_value> (PAC value for the provided client authentication certificate)

Content-Type: 'application/json'

Request Body Payload

The payload is the same as the invoice payload. Please refer to the [Invoice section](#).

The *referentDocumentNumber* field must be set to advance transaction **reference number**, received when initial advance invoice was created.

Response Payload

The response is the same as the normal invoice payload. Please refer to the [Invoice section](#).

Close transaction

This method is used to close the advance transaction, effectively creating a refund for all the advance invoices, and issuing final, normal sale invoice. The invoice structure is exactly the same as the new advance invoice, with addition of the reference to the existing transaction.

The user must be authenticated, and the received ID token must be used as bearer token in the HTTP Authorization header.

Method: POST

Content-Type: application/x-www-form-urlencoded

Endpoint:

https://COUNTRY_DOMAIN/v1/taxcore/advanceTransactions/close?key=[API_KEY]

Header:

PAC: <PAC_value> (PAC value for the provided client authentication certificate)

Content-Type: 'application/json'

Request Body Payload

The payload is the same as the invoice payload. Please refer to the [Invoice section](#).

The *referentDocumentNumber* field must be set to advance transaction **reference number**, received when initial advance invoice was created.

Response Payload

The response is the same as the normal invoice payload. Please refer to the [Invoice section](#).

The user is *obligated* by the Law to provide the printed or electronic final invoice to the buyer. All necessary data are provided in the final invoice response.

Advance transaction price change

If the price of the items is changed while the advanced transaction is still open, it is possible to update the price of the item(s). This can be achieved in two steps. First step is to retrieve the item array of the transaction, and the second is to push the updated array with new prices.

The API call for both item array retrieval and price update is the same. To retrieve an existing item array, the user should omit the "items" key from the request body payload, and only set *referentDocumentNumber*. See the examples below.

Note that only price is allowed to change, and nothing else. Users should change only **unitPrice** and **totalAmount**, accordingly. Also, the total invoice price cannot be lowered below the amount already paid by previous advance invoice(s).

For example, if the total price of the invoice is 1000, and total advanced amount is 700, the new total amount for the invoice cannot be lower than 700.

The user must be authenticated, and the received ID token must be used as bearer token in the HTTP Authorization header.

Method: PUT

Content-Type: application/x-www-form-urlencoded

Endpoint:

https://COUNTRY_DOMAIN/v1/taxcore/advanceTransactions/price?key=[API_KEY]

Header:

PAC: <PAC_value> (PAC value for the provided client authentication certificate)

Content-Type: 'application/json'

Request Body Payload

Where the `referentDocumentNumber` is the invoice number of the first issued advance invoice.

Case 1 - retrieve current item array

```
{
  "referentDocumentNumber": "WHKV8WJH-GFB38TO0-16",
}
```

This will retrieve the list of items for transaction with a given referent number.

Case 2 - update prices

```
{
  "referentDocumentNumber": "WHKV8WJH-GFB38TO0-16",
  "items": [
    {
      "unitPrice": 60,
      "uid": "-MiM__jyd8jJy_UFEEJQ",
      "totalAmount": 60,
      "unit": "kom",
      "quantity": 1,
      "vat": "A",
      "name": "Budweiser 0.5 limenka",
      "labels": [
        "A"
      ]
    },
    {
      "unitPrice": 55,
      "uid": "-MiM_C3gFS20J2Jv9aml",
      "totalAmount": 55,
      "unit": "kom",
      "quantity": 1,
      "vat": "A",
      "name": "Kisela voda Knjaz Milos 0.5l",
      "labels": [
        "A"
      ]
    }
  ]
}
```

This will set new prices for both items. The price for both items in this example is changed.

Response Payload

Case 1 - retrieve current item array

```
[
  {
    "unitPrice": 50,
    "uid": "-MiM__jyd8jJy_UFEEJQ",
    "totalAmount": 50,
    "unit": "kom",
    "quantity": 1,
    "vat": "A",
    "name": "Budweiser 0.5 limenka",
    "labels": [
      "A"
    ]
  },
  {
    "unitPrice": 40,
    "uid": "-MiM_C3gFS20J2Jv9aml",
    "totalAmount": 40,
    "unit": "kom",
    "quantity": 1,
    "vat": "A",
    "name": "Kisela voda Knjaz Milos 0.5l",
    "labels": [
      "A"
    ]
  }
]
```

The response contains an array of items with all the data. Nothing should be changed except the **unitPrice** and **totalAmount**.

Case 2 - update prices

After a successful price update the response will reflect the new state.

```
[
  {
    "unitPrice": 60,
    "uid": "-MiM__jyd8jJy_UFEEJQ",
    "totalAmount": 60,
    "unit": "kom",
    "quantity": 1,
    "vat": "A",
```

```
[
  {
    "name": "Budweiser 0.5 limenka",
    "labels": [
      "A"
    ]
  },
  {
    "unitPrice": 55,
    "uid": "-MiM_C3gFS20J2Jv9aml",
    "totalAmount": 55,
    "unit": "kom",
    "quantity": 1,
    "vat": "A",
    "name": "Kisela voda Knjaz Milos 0.5l",
    "labels": [
      "A"
    ]
  }
]
```

Invoice receipt

An invoice receipt, or fiscal invoice is, by definition, a digitally signed acknowledgment that a specified payment has been received or refunded. The mandatory elements of each invoice are shown in the following examples.

Invoicing system

Secure element

===== FISCAL INVOICE =====
RS654321
Premier League DTI
Premier League DTI
Kruzni put 7
Lestane
Cashier: 2-5783
Buyer: RS349802744
Buyers Cost Center: CR02R
Ref No: LLA554DW-LGHFXH00-2
Ref DT: 20.04.2022. 13:08:34
-----NORMAL REFUND-----
Items
=====

| Name | Price | Qty. | Total |
|-----------------------------|-------|------|--------|
| Barn ground coffee 200g (A) | 8,40 | 2 | -16,80 |
| Sunny Way red mug (A) | 12,00 | 1 | -12,00 |
| Total Refund: | | | 28,80 |
| Card: | | | 28,80 |

Sales Data Controller (SDC)

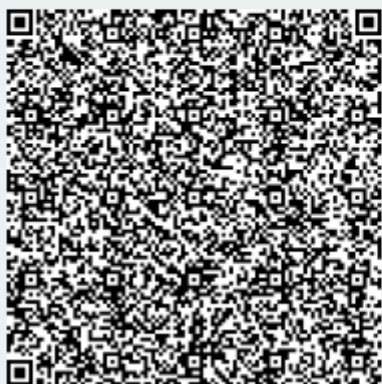
Secure element

=====

Label Name Rate Tax
A VAT 15,00% 3,76

Total Tax: 3,76
=====

SDC Time: 20.04.2022. 13:09:11
SDC Invoice No: BQVWAAR4-NBN68V00-16
Invoice Counter: 3/16/PP
=====



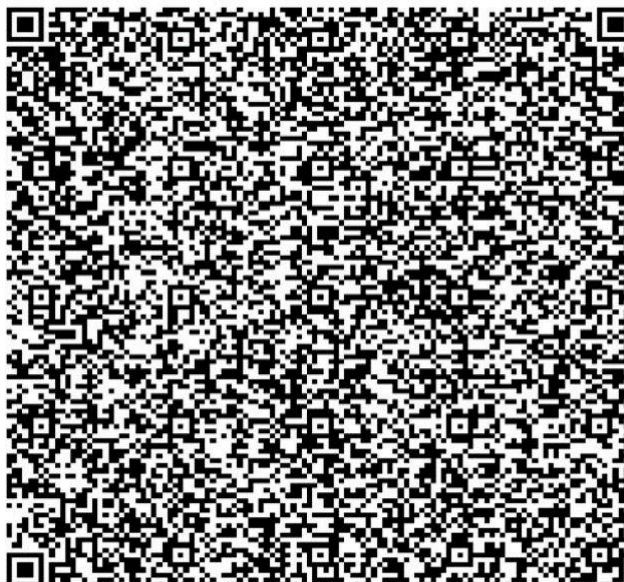
===== END OF FISCAL INVOICE =====

Фискални рачун

| | |
|----------------|---------------------|
| ПИБ: | 123456789 |
| Предузеће | Делта |
| Име | Макси |
| Адреса: | Теслина 8 |
| Место | Београд |
| Касир | 1 |
| ПИБ купца: | 987654321 |
| Податак купца: | xxx |
| ПОС број: | 9280 |
| ПОС време: | xxx |
| Реф. Број | GJDHBB12-GJHBB12-34 |

Продаја

| | | | |
|----------------|----------|---------------------|-------|
| Артикли: | | | |
| Име, цена | Количина | Укупно | |
| Пиво Лав (Ђ) | | | |
| 3.00 | 4.00 | 12.00 | |
| Хлеб домаћи(Е) | | | |
| 1.00 | 2.00 | 2.00 | |
| ЗА УПЛАТУ | | 14.00 | |
| Врста плаћања | | Готовина | |
| Ознака | Назив | Стопа | Порез |
| Ђ | | 20% | 2.00 |
| Е | | 10% | 0.18 |
| Укупно порез | | | 2.18 |
| ПФР време | | 22.09.2019 10 57 | |
| ПФР бр рачуна | | GJDHBB12-GJHBB12-46 | |
| Бројачи | | 35/46NS | |



Крај фискалног рачуна

Buyer id

The invoices with buyer id, will use the following schema based on the code table.

```
"buyerId": "<code>:<documentId>"
```

| code | documentId | description |
|------|---------------|--|
| 10 | PIB | Domestic legal entity identified with TIN (PIB) |
| 11 | JMBG | Domestic individual, identified with JMBG |
| 12 | PIB:JBKJS | Domestic legal entity identified both with PIB and JBKJS |
| 13 | Retiree ID | Individual with retiree id card |
| 14 | PIB | Legal entity - agricultural household, identified with PIB |
| 15 | JMBG | Individual - agricultural household, identified with JMBG |
| 16 | BPG | Individual - agricultural household, identified with BPG |
| 20 | Personal ID | Individual with ID card |
| 21 | Refugee ID | Individual with refugee identification |
| 22 | EBS | Foreign individual who has registered stay in Serbia/BIH |
| 23 | Passport ID | Domestic individual with passport ID |
| 30 | Passport ID | Foreign individual with passport ID |
| 31 | Diplomatic ID | Foreign individual - a diplomat |
| 32 | MKD ID | Foreign individual with N. Macedonian ID |
| 33 | MNE ID | Foreign individual with Montenegrin ID |
| 34 | ALB ID | Foreign individual with Albanian ID |
| 35 | BIH ID | Foreign individual with BIH ID |
| 36 | ID number | Foreign individual with ID card from EU, CHE, NOR or ISL |
| 40 | PIB (TIN) | Foreign Tax Identification number |

Examples:

```
"buyerId": "10:123123123"
```

```
"buyerId": "12:123456789:54321"
```

```
"buyerId": "30:099999999"
```