**Kazalo**

[1. Uvod 2](#_Toc43365712)

[2. Funkcionalnosti servisa 2](#_Toc43365713)

[3. Definicija vmesnika servisa 2](#_Toc43365714)

[3.1. Splošni parametri klicev 2](#_Toc43365715)

[Avtentikacija 2](#_Toc43365716)

[Tip kodiranja vhodnih parametrov 3](#_Toc43365717)

[Tip kodiranja izhodnih parametrov 3](#_Toc43365718)

[3.2. Prijava 3](#_Toc43365719)

[Metoda /api/Auth 3](#_Toc43365720)

[3.3. Podatki o upravičencih 5](#_Toc43365721)

[Metoda /api/Voucher/Balance 5](#_Toc43365722)

[3.4. Delo z dokumenti 6](#_Toc43365723)

[Metoda /api/Document/Validate 6](#_Toc43365724)

[Metoda /api/Document/Deposit 8](#_Toc43365725)

[Metoda /api/Document/Download/{edpDocumentNumber} 9](#_Toc43365726)

[3.5. Generični odgovori 12](#_Toc43365727)

[400 Bad request 12](#_Toc43365728)

[4. Delo z dokumenti 12](#_Toc43365729)

[4.1. Rezervacija vavčerjev 12](#_Toc43365730)

[4.2. Unovčitev vavčerjev 13](#_Toc43365731)

[4.3. Originali in popravki predhodno oddanega dokumenta 13](#_Toc43365732)

[4.4. Storno predhodno oddanega dokumenta 14](#_Toc43365733)

[5. Možen način uporabe 15](#_Toc43365734)

[6. Priloge 15](#_Toc43365735)

# Uvod

Unovčevanje turističnih bonov je v sistemu eDavki trenutno možno preko spletnega portala. Pričujoči dokument specificira spletni servis, ki bo omogočil integracijo funkcionalnosti za turistične bone s hotelskimi zalednimi sistemi in računovodskimi programi.

Spletni servis bo imel REST vmesnik, ki bo predvidoma podpiral podatke, kodirane v XML in JSON.

Funkcionalnosti servisa

Spletni servis bo imel naslednje funkcionalnosti:

* Prijava. Omogočena bo prijava z istimi uporabniškimi računi, ki jih uporabniki uporabljajo za spletni portal.
* Stanje sredstev na vavčerju za upravičenca, določenega s podatki: EMŠO, priimek, ime.
* Oddaja dokumenta. Z oddajo ustreznega vrste dokumenta bo mogoče:
  + Rezervirati enega ali več vavčerjev.
  + Unovčiti enega ali več vavčerjev. (v tem dokumentu se tudi navede številka fiskalnega računa).
  + Popraviti predhodno oddan dokument. Popravek je dovoljen samo za dokumente, s katerimi je bila opravljena rezervacija.
  + Stornirati predhodno oddan dokument. Stornirati je mogoče tako rezervacije kot unovčitve.

Definicija vmesnika servisa

V tem poglavju so opisane posamezne metode servisa, združene po sklopih funkcionalnosti. Parametri XML in JSON klicev imajo enaka imena, razlikujejo pa se v veliki/mali začetnici, npr. **AccommodationEstablishmentId** v XML in **accommodationEstablishmentId** v JSON. Imena objektov se ne razlikujejo. Točna imena so razvidna iz primerov klicev metod.

## Splošni parametri klicev

Pri vseh klicih servisa se uporablja nekaj splošnih parametrov, ki se pošiljajo v glavi zahtevka.

### Avtentikacija

Za dostop do vseh funkcionalnosti servisa (razen za prijavo samo) se mora uporabnik avtenticirati. To stori tako, da v klicu metode pošlje avtentikacijski žeton, ki ga pridobi pri prijavi (glej podpoglavje 3.2). Žeton je potrebno poslati v glavi zahtevka v polju Authorization. Vrednost v polju mora imeti naslednjo obliko:

Bearer {žeton}

Primer celotnega polja:

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..w8\_TRViFOTh99IbdE8tqaRLsq4tkbOQZB\_Xi5tBumpM

### Tip kodiranja vhodnih parametrov

Tip kodiranja vhodnih parametrov je definiran v glavi zahtevka v polju Content-Type. Predvidoma podprti tipi bodo »application/xml« ali »text/xml« za XML ter »application/json« za JSON.

Primer celotnega polja:

Content-Type: application/xml

### Tip kodiranja izhodnih parametrov

Tip kodiranja izhodnih parametrov je definiran v glavi zahtevka v polju accept. Predvidoma podprti tipi bodo »application/xml« ali »text/xml« za XML ter »application/json« za JSON.

Primer celotnega polja:

accept: application/xml

## Prijava

### Metoda /api/Auth

* Opis: metoda omogoča prijavo uporabnika. Ob uspešni prijavi se izda avtentikacijski žeton, ki ga je potrebno uporabiti v vseh naslednjih klicih vmesnika. Žeton ima omejen čas veljavnosti. Po poteku le-te je potrebno pridobiti nov žeton.
* HTTP metoda: POST
* Vhodni parametri:

**LoginRequest{**

|  |  |
| --- | --- |
|  |  |
| **password** | **string**  **Uporabniško geslo.** |
| **taxNumber** | **integer($int32)**  **Davčna številka ponudnika.** |
| **username** | **string**  **Uporabniško geslo.** |
| **accommodationEstablishmentId** | **string**  **Identifikacijska oznaka nastanitvenega objekta. Ujemati se mora z identifikacijsko oznako v registru AJPES.** |

**}**

* Predvidene HTTP kode odgovora:
  + 200 OK: klic je uspel.
  + 400 Bad Request: napačni vhodni podatki (npr. neveljavna identifikacijska oznaka nastanitvenega objekta)
  + 401 Unauthorized: napačni podatki za prijavo (npr. napačno geslo ali neobstoječe uporabniško ime)
* Izhodni parametri:
  + Status odgovora 200 OK:

**LoginResponse{**

|  |  |
| --- | --- |
|  |  |
| **expires** | **string($date-time)Datum in čas poteka žetona.** |
| **token** | **stringŽeton za avtentikacijo.** |
| **loginId** | **stringIdentifikacijska oznaka prijave. Uporabiti jo je mogoče za revizijsko sled.** |

**}**

* + Status odgovora 400 Bad Request: glej podpoglavje 3.5.
  + Status odgovora 401 Unauthorized: ne vrača vsebine
* Primeri:
  + Vhodni parametri XML:

<LoginRequest>

<Password>geslo123</Password>

<TaxNumber>92137318</TaxNumber>

<Username>hans</Username>

<AccommodationEstablishmentId>106574</AccommodationEstablishmentId>

</LoginRequest>

* + Vhodni parametri JSON:

{

"password": "geslo123",

"taxNumber": 92137318,

"username": "hans",

"accommodationEstablishmentId": "106574"

}

* + Odgovor XML za HTTP status 200 OK:

<LoginResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Expires>2020-06-12T14:01:29.1061537+02:00</Expires>

<Token>eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..lN\_dc9VZRxUUWFSLBHWih1l3oCkovfT0DiTdCBCXlq4</Token>

<LoginId>332</LoginId>

</LoginResponse>

* + Odgovor JSON za HTTP status 200 OK:

{

"expires": "2020-06-12T14:23:57.2481556+02:00",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..YYHz4Sq0\_FAQ3Jz6e\_2knUhxsu\_qnT2o87BC2iJgVzc",

"loginId": "332"

}

## Podatki o upravičencih

### Metoda /api/Voucher/Balance

* Opis: metoda vrača podatke o stanju sredstev na vavčerju za upravičenca.
* HTTP metoda: POST
* Vhodni parametri:

**VoucherBalanceRequest{**

|  |  |
| --- | --- |
|  |  |
| **emso** | **string**  **EMŠO upravičenca.** |
| **name** | **string**  **Ime upravičenca.** |
| **surname** | **string**  **Priimek upravičenca.** |

**}**

* Predvidene HTTP kode odgovora:
  + 200 OK: klic je uspel.
  + 400 Bad Request: napačni vhodni podatki (npr. neveljavna kombinacija EMŠO – ime - priimek)
  + 401 Unauthorized: neveljaven žeton
* Izhodni parametri:
  + Status odgovora 200 OK:

**VoucherBalanceResponse{**

|  |  |
| --- | --- |
|  |  |
| **availableAmount** | **number($double)Vrednost vavčerja, ki je še na voljo.** |
| **emso** | **stringEMŠO upravičenca** |
| **totalValue** | **number($double)Vrednost vavčerja (200 EUR za odrasle, 50 EUR za otroke).** |

**}**

* + Status odgovora 400 Bad Request: glej podpoglavje 3.5.
  + Status odgovora 401 Unauthorized: ne vrača vsebine
* Primeri:
  + Vhodni parametri XML:

<VoucherBalanceRequest>

<Emso>0101001500008</Emso>

<Name>Ime\_12345678</Name>

<Surname>Priimek\_12345678</Surname>

</VoucherBalanceRequest>

* + Vhodni parametri JSON:

{

"emso": "0101001500008",

"name": "Ime\_12345678",

"surname": "Priimek\_12345678"

}

* + Odgovor XML za HTTP status 200 OK:

<VoucherBalanceResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<AvailableAmount>0.0100</AvailableAmount>

<Emso>0101001500008</Emso>

<TotalValue>200.0</TotalValue>

</VoucherBalanceResponse>

* + Odgovor JSON za HTTP status 200 OK:

{

"availableAmount": 0.01,

"emso": "0101001500008",

"totalValue": 200

}

## Delo z dokumenti

### Metoda /api/Document/Validate

* Opis: metoda omogoča preverjanje pravilnosti dokumenta. Odgovor vsebuje seznam napak in njihove tipe: kritična (0 – Critical), opozorilo (1 – NonCritical), predizpolnjeni podatki spremenjeni (2 – PrefilledMismatch) (ta tip je sicer definiran, vendar je rezerviran za spletni portal).
* HTTP metoda: POST
* Vhodni parametri:

**ValidateDocumentRequest{**

|  |  |
| --- | --- |
| ***:*** |  |
| **documentXml** | **string($byte)**  **Tekst XML dokumenta, kodiran po standardu Base64. Tekst mora biti v formatu UTF-8.** |

**}**

* Predvidene HTTP kode odgovora:
  + 200 OK: klic je uspel.
  + 400 Bad Request: napačni vhodni podatki (npr. tekst ni veljavni XML)
  + 401 Unauthorized: neveljaven žeton
* Izhodni parametri:
  + Status odgovora 200 OK:

**ValidateDocumentResponse{**

|  |  |
| --- | --- |
|  |  |
| **errors** | **[Lista napak. Lahko je prazna.**  **DocumentError{**   |  |  | | --- | --- | |  |  | | **message** | **stringTekst napake.** | | **type** | **integer($int32) za JSON oz. string za XMLTip napake. V JSON dobimo eno izmed vrednosti 0, 1, 2; v XML pa eno izmed vrednosti Critical, NonCritical, PrefilledMismatch.** |   **}]** |
| **numberCritical** | **integer($int32)Število kritičnih napak.** |
| **numberNonCritical** | **integer($int32)Število opozoril.** |
| **numberPrefilledMismatch** | **integer($int32)Število napak tipa** »**predizpolnjeni podatki spremenjeni«.** |
| **totalNumber** | **integer($int32)Skupno število napak.** |

**}**

* + Status odgovora 400 Bad Request: glej podpoglavje 3.5.
  + Status odgovora 401 Unauthorized: ne vrača vsebine
* Primeri:
  + Vhodni parametri XML:

<ValidateDocumentRequest>

<DocumentXml></DocumentXml>

</ValidateDocumentRequest>

* + Vhodni parametri JSON:

{

"documentXml": ""

}

* + Odgovor XML za HTTP status 200 OK:

<ValidateDocumentResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Errors>

<DocumentError>

<Message>Podatek v polju 'Vrednost' je previsok. Najvišja dovoljena vrednost je 0,01 EUR. Vrstica 1.</Message>

<Type>Critical</Type>

</DocumentError>

<DocumentError>

<Message>Podatki o nočitvah osebe Ime\_12345678 Priimek\_12345678 se ne ujemajo z evidenco nočitev AJPES. Vrstica 1.</Message>

<Type>NonCritical</Type>

</DocumentError>

</Errors>

<NumberCritical>1</NumberCritical>

<NumberNonCritical>1</NumberNonCritical>

<NumberPrefilledMismatch>0</NumberPrefilledMismatch>

<TotalNumber>2</TotalNumber>

</ValidateDocumentResponse>

* + Odgovor JSON za HTTP status 200 OK:

{

"errors": [

{

"message": "Podatek v polju 'Vrednost' je previsok. Najvišja dovoljena vrednost je 0,01 EUR. Vrstica 1.",

"type": 0

},

{

"message": "Podatki o nočitvah osebe Ime\_12345678 Priimek\_12345678 se ne ujemajo z evidenco nočitev AJPES. Vrstica 1.",

"type": 1

}

],

"numberCritical": 1,

"numberNonCritical": 1,

"numberPrefilledMismatch": 0,

"totalNumber": 2

}

### Metoda /api/Document/Deposit

* Opis: metoda omogoča oddajo dokumenta v sistem eDavki. Dokumenta s kritičnimi napakami (tip 0 – Critical) ni mogoče oddati. Dokument z drugimi tipi napak je mogoče oddati pod pogojem, da ima pri klicu metode parameter ignoreNonCriticalErrors vrednost 1.
* HTTP metoda: POST
* Vhodni parametri:

**DepositDocumentRequest{**

|  |  |
| --- | --- |
|  |  |
| **documentXml** | **string($byte)**  **Tekst XML dokumenta, kodiran po standardu Base64. Tekst mora biti v formatu UTF-8.** |
| **ignoreNonCriticalErrors** | **boolean**  **Ali naj se ignorirajo nekritične napake.** |

* **}**
* Predvidene HTTP kode odgovora:
  + 200 OK: klic je uspel.
  + 400 Bad Request: napačni vhodni podatki (npr. tekst ni veljavni XML)
  + 401 Unauthorized: neveljaven žeton
* Izhodni parametri:
  + Status odgovora 200 OK:

**DepositDocumentResponse{**

|  |  |
| --- | --- |
|  |  |
| **edpDocumentNumber** | **string**  **Številka oddanega dokumenta v sistemu eDavki.** |

**}**

* + Status odgovora 400 Bad Request: glej podpoglavje 3.5.
  + Status odgovora 401 Unauthorized: ne vrača vsebine
* Primeri:
  + Vhodni parametri XML:

<DepositDocumentRequest>

<DocumentXml></DocumentXml>

<IgnoreNonCriticalErrors>true</IgnoreNonCriticalErrors>

</DepositDocumentRequest>

* + Vhodni parametri JSON:

{

"documentXml": "",

"ignoreNonCriticalErrors": true

}

* + Odgovor XML za HTTP status 200 OK:

<DepositDocumentResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<EdpDocumentNumber>EDP-92137318-80</EdpDocumentNumber>

</DepositDocumentResponse>

* + Odgovor JSON za HTTP status 200 OK:

{

"edpDocumentNumber": "EDP-92137318-80"

}

### Metoda /api/Document/Download/{edpDocumentNumber}

* Opis: metoda omogoča prenos oddanega dokumenta k uporabniku.
* HTTP metoda: GET
* Vhodni parametri:

|  |  |
| --- | --- |
| **edpDocumentNumber**  **string** | Številka dokumenta v sistemu eDavki. |

* Predvidene HTTP kode odgovora:
  + 200 OK: klic je uspel.
  + 400 Bad Request: napačni vhodni podatki (npr. neveljavna številka dokumenta)
  + 401 Unauthorized: neveljaven žeton
* Izhodni parametri:
  + Status odgovora 200 OK:

**DownloadDocumentResponse{**

|  |  |
| --- | --- |
|  |  |
| **documentXml** | **string($byte)**  **Tekst XML dokumenta, kodiran po standardu Base64.** |
| **edpDocumentNumber** | **string**  **Številka dokumenta v sistemu eDavki.** |

**}**

* + Status odgovora 400 Bad Request: glej podpoglavje 3.5.
  + Status odgovora 401 Unauthorized: ne vrača vsebine
* Primeri:
  + Vhodni parametri:

EDP-92137318-80

* + Odgovor XML za HTTP status 200 OK:

<DownloadDocumentResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<DocumentXml></DocumentXml>

<EdpDocumentNumber>EDP-92137318-80</EdpDocumentNumber>

</DownloadDocumentResponse>

* + Odgovor JSON za HTTP status 200 OK:

{

"documentXml": "",

"edpDocumentNumber": "EDP-92137318-80"

}

## Generični odgovori

### 400 Bad request

Izhodni parametri v primeru, da servis vrne HTTP status 400 Bad request (torej v primeru, da so vhodni parametri klicev napačni):

**ErrorResponse{**

|  |  |
| --- | --- |
|  |  |
| **AdditionalInfo** | **stringDodatne informacije oz. opis napake.** |
| **errorCode** | **stringKoda napake v sistemu eDavki.** |
| **statusCode** | **integer($int32)HTTP status koda napake.** |

**}**

Delo z dokumenti

Z oddajo dokumentov v sistem eDavki lahko rezerviramo ali unovčimo vavčerje in izvršimo storno. Dokumenti lahko vsebujejo podatke za enega ali več vavčerjev. Za vse vavčerje na istem dokumentu se izvrši ista operacija – rezervacija ali unovčitev (glede na izbrano operacijo – glej spodaj).

## Rezervacija vavčerjev

Vavčerje lahko rezerviramo z oddajo dokumenta TB-Unovcitev. Pri tem je pomembno, da ima element /Envelope/body/Data/IsReservation vrednost true.

Primer:

<?xml version="1.0" encoding="utf-8"?>

<Envelope xmlns="http://edavki.durs.si/Documents/Schemas/TB\_Unovcitev\_1.xsd" xmlns:edp="http://edavki.durs.si/Documents/Schemas/EDP-Common-1.xsd" xmlns:fu="http://edavki.durs.si/Documents/Schemas/FU\_Imported\_Types\_1.xsd">

<edp:Header>

<edp:taxpayer>

<edp:taxNumber>92137318</edp:taxNumber>

<edp:taxpayerType>SP</edp:taxpayerType>

<edp:name>MARKET JERCA JERICA GAŠPERIN S.P.</edp:name>

<edp:address1>KAMNJE 005</edp:address1>

<edp:city>KAMNJE</edp:city>

<edp:postNumber>4264</edp:postNumber>

<edp:postName>DejavnostFO Pošta Naziv</edp:postName>

</edp:taxpayer>

<edp:Workflow>

<edp:DocumentWorkflowID>O</edp:DocumentWorkflowID>

<edp:DocumentWorkflowName/>

</edp:Workflow>

<edp:domain>edavki.durs.si</edp:domain>

</edp:Header>

<edp:AttachmentList />

<edp:Signatures />

<body>

<edp:bodyContent />

<Data>

<AccommodationEstablishmentId>106574</AccommodationEstablishmentId>

<AccommodationEstablishmentName>APARTMA KAMNJE</AccommodationEstablishmentName>

<RefundOwnerType>ServiceProviderManual</RefundOwnerType>

<RefundAmountSum>0.01</RefundAmountSum>

**<IsReservation>true</IsReservation>**

<Refunds>

<RefundDetails>

<Id>0</Id>

<IsConfirmator>false</IsConfirmator>

<FirstName>Ime\_12345678</FirstName>

<LastName>Priimek\_12345678</LastName>

<Emso>0101001500008</Emso>

<IsDonor>false</IsDonor>

<AccommodationFrom>2020-11-02</AccommodationFrom>

<AccommodationTo>2020-11-07</AccommodationTo>

<Amount>0.01</Amount>

</RefundDetails>

</Refunds>

</Data>

</body>

</Envelope>

## Unovčitev vavčerjev

Podobno kot za rezervacijo lahko vavčerje unovčimo z oddajo dokumenta TB-Unovcitev. Pri tem je pomembno, da ima element /Envelope/body/Data/IsReservation vrednost false.

## Originali in popravki predhodno oddanega dokumenta

Z oddajo originala dokumenta TB-Unovcitev izvršimo novo rezervacijo oz. unovčitev. Pri tem je pomembno, da ima element /Envelope/edp:Header/edp:Workflow/edp:DocumentWorkflowID vrednost O.

Primer: glej primer v podpoglavju 4.1.

Popravek dokumenta naredimo tako, da oddamo nov dokument TB-Unovcitev. Popravljamo lahko samo dokumente, s katerimi smo izvršili rezervacijo vavčerjev.

Popravek dokumenta pomeni, da se razveljavijo vse rezervacije na dokumentu, ki ga popravljamo.

Pri popravku dokumenta je pomembno naslednje:

* Element /Envelope/edp:Header/edp:Workflow/edp:DocumentWorkflowID mora imeti vrednost P.
* Dokument mora vsebovati element /Envelope/body/Data/CorrectionForDocument, ki mora vsebovati številko dokumenta, ki ga popravljamo.

Primer:

<?xml version="1.0" encoding="utf-8"?>

<Envelope xmlns="http://edavki.durs.si/Documents/Schemas/TB\_Unovcitev\_1.xsd" xmlns:edp="http://edavki.durs.si/Documents/Schemas/EDP-Common-1.xsd" xmlns:fu="http://edavki.durs.si/Documents/Schemas/FU\_Imported\_Types\_1.xsd">

<edp:Header>

<edp:taxpayer>

<edp:taxNumber>92137318</edp:taxNumber>

<edp:taxpayerType>SP</edp:taxpayerType>

<edp:name>MARKET JERCA JERICA GAŠPERIN S.P.</edp:name>

<edp:address1>KAMNJE 005</edp:address1>

<edp:city>KAMNJE</edp:city>

<edp:postNumber>4264</edp:postNumber>

<edp:postName>DejavnostFO Pošta Naziv</edp:postName>

</edp:taxpayer>

<edp:Workflow>

**<edp:DocumentWorkflowID>P</edp:DocumentWorkflowID>**

<edp:DocumentWorkflowName/>

</edp:Workflow>

<edp:domain>edavki.durs.si</edp:domain>

</edp:Header>

<edp:AttachmentList />

<edp:Signatures />

<body>

<edp:bodyContent />

<Data>

**<CorrectionForDocument>EDP-92137318-80</CorrectionForDocument>**

<AccommodationEstablishmentId>106574</AccommodationEstablishmentId>

<AccommodationEstablishmentName>APARTMA KAMNJE</AccommodationEstablishmentName>

<RefundOwnerType>ServiceProviderManual</RefundOwnerType>

<RefundAmountSum>0.01</RefundAmountSum>

<IsReservation>true</IsReservation>

<Refunds>

<RefundDetails>

<Id>0</Id>

<IsConfirmator>false</IsConfirmator>

<FirstName>Ime\_12345678</FirstName>

<LastName>Priimek\_12345678</LastName>

<Emso>0101001500008</Emso>

<IsDonor>false</IsDonor>

<AccommodationFrom>2020-11-02</AccommodationFrom>

<AccommodationTo>2020-11-07</AccommodationTo>

<Amount>0.01</Amount>

</RefundDetails>

</Refunds>

</Data>

</body>

</Envelope>

Originalni document in njegovi popravki tvorijo verigo. Možno je oddati poljubno število popravkov v verigi. Pri tem je potrebno paziti, da vedno popravljamo zadnji document v verigi – pomembna je pravilna vrednost elementa /Envelope/body/Data/CorrectionForDocument. Primer verige:

EDP-92137318-80 (O – original) -> EDP-92137318-81 (P – popravek) -> EDP-92137318-82 (P – popravek)

## Storno predhodno oddanega dokumenta

Dokument storniramo tako, da oddamo dokument NF-Storno.

Storno dokumenta pomeni, da se razveljavijo vse rezervacije oz. unovčitve na dokumentu, ki ga storniramo.

Pri tem je pomembno, da element /Envelope/body/NF\_Storno/ refDocument vsebuje številko dokumenta, ki ga želimo stornirati.

Storno je vedno zadnji element v verigi. Primer:

EDP-92137318-80 (O – original) -> EDP-92137318-81 (P – popravek) -> EDP-92137318-82 (P – popravek) -> EDP-92137318-83 (strorno)

Primer storno dokumenta:

<?xml version="1.0" encoding="utf-8"?>

<Envelope xmlns="http://edavki.durs.si/Documents/Schemas/NF\_Storno\_1.xsd" xmlns:edp="http://edavki.durs.si/Documents/Schemas/EDP-Common-1.xsd">

<edp:Header>

<edp:taxpayer>

<edp:taxNumber>92137318</edp:taxNumber>

<edp:taxpayerType>SP</edp:taxpayerType>

<edp:name>MARKET JERCA JERICA GAŠPERIN S.P.</edp:name>

<edp:address1>KAMNJE 005</edp:address1>

<edp:city>KAMNJE</edp:city>

</edp:taxpayer>

</edp:Header>

<edp:AttachmentList />

<edp:Signatures />

<body>

<edp:bodyContent />

<NF\_Storno>

<refDocument>

**<DocumentNumber>EDP-92137318-80</DocumentNumber>**

<DepositedDate>2020-06-12T14:39:30.407+02:00</DepositedDate>

<DepositorName>Hans Müller</DepositorName>

<FormCode>TB-Unovcitev</FormCode>

<Reason>Preklic turističnega bona; Napačni podatki.</Reason>

</refDocument>

</NF\_Storno>

</body>

</Envelope>

Možen način uporabe

* Prijava: **/api/Auth**
* Vnos gostov – med vnosom se sproti za vsakega posebej preveri maksimalna višina koriščenja: **/api/Voucher/Balance**
* Preverjanje podatkov kot celote (npr. za celo družino): **/api/Document/Validate**
* Rezervacija: **/api/Document/Deposit** -> <IsReservation>true</IsReservation>
* Izdaja računa
* Unovčitev: **/api/Document/Deposit** -> <IsReservation>false</IsReservation>

Priloge

XML sheme dokumentov:



Swagger datoteka spletnega servisa:

