

Mindenek előtt ez az onlineszamla adatszolgáltatás V2.0 verziójának megfelelően dokumentál.

1. Bevezető

- Tervezés alatt (NAV) * [] Implementálandó v. Implementálható * [.] Implementálás alatt * [t]

Implementált, de tesztelni kell



* [X] Implementált, tesztelt * [t]

manageAnnulment * [t] manageInvoice * [.] nav_onlineszamla_data * [X] queryInvoiceCheck *
[X] queryInvoiceData * [] queryInvoiceDigest * _ queryServiceMetrics

- [t] queryTransactionStatus
- [X] queryTaxpayer
- [X] tokenExchange

1.1 folyamat

A rendszer egy technikai azonosítóval dolgozik.

A token a számla küldésének kötelezettségekor közvetlenül megigényli, A megigényelt token azonnal megpróbálja felhasználni, nem él az 5 perces limittel.

Tömeges számlázáskor egy token igényel az egyszerre kiállított számlákhoz (trigger) ezért a [tömeges számlázás](#)nál nem emelhető meg 100 fölé az egyszerre kiállított számlák száma.

Ha a számlák ismételt beküldése esetén (átmeneti technikai sikertelenség esetén) az újraküldésekhez már számlánként egyenként igényel token.

1.3.1. request

basic.pm

- 1) requestId az onlineszamla_request_log tábla request_id-jéből jön. Fejlesztői környezetben kaphat prefixet az unicitás biztosítása céljából.
- 2) timestamp: onlineszamla_request_log.request_timestamp
- 3) requestVersion: general_data→work_version alapján
- 4) headerVersion: nem kötelező de megadjuk

1.3.2. user

Kötelezően implementálva

1.3.3. software

Kötelezően implementálva

\$general_data→{software}

1.4. response

onlineszamla→

- `handle_response` minden választ feldolgoz
- `process_result` értelmezi HTTP es hibakod szinten
- `process_response_xml` csak parsol



1.4.1. result

- 1) nem használtuk
- 2) `errorCode` újrainni amit kell
- 3) `message` behozása a hibakezelésbe

1.5.1 requestSignature számolás manageInvoice és manageAnnulment

Új kötelező implementáció V2 tesztelés alatt

1.5.2. requestSignature számolás egyébként

Kötelezően implementálva

1.5.3 UTC

SQL-ből megoldva (UTC-ben is generáljuk, tároljuk)

1.7.1. általános adatok



context root verzio nélkül???

1.7.2.

Mint 1. pontban

1.7.3. http header

`onlineszamla`→`calc_header`

1.7.4. http statusok

lásd 1.4 ...

1.7.5. tömörítés

Opcionális. Nem tömörítünk.

1.7.6. timeout

onlineszamla→http_ua_request

setup-ban allithato az interaktiv es a batch timeout kulon kulon is.

default 5 sec.

timeout eseten saját response kodunk van:

504 Client-side timeout  átvezetni a többi mellé a doksiban

1.7.7. szerveridő, NTP

nem kötelező; oprendszer szintű megoldást itt nem dokumentálok. (Elvileg mindenhol NTP van)

1.7.8. karbantartási mód

 ezt implementálni kellhet

1.7.9. verziókezelés

logikus

1.8. req resp elementek

1.4. szerint

1.9.1. manage Annulment

Implementálva  Tesztelni kell majd

1.9.2. manageInvoice

Tömörítést nem használunk

Implementált  tesztelés alatt

A beküldött számla adattartalmára vonatkozó információk a [nav_onlineszamla_data](#) oldalon találhatóak.

1.9.3. queryInvoiceCheck

Implementálva Tesztelve



batch feldolgozás és vevő oldali lekérdezés nélkül

1.9.4. queryInvoiceData

Implementálva. Tesztelve



Használata még változás alatt van



batch feldolgozás és vevő oldali lekérdezés nélkül

1.9.5. queryInvoiceDigest



Implementálandó

Adatszolgáltatási kötelezettség teljesítését nem akadályozza

1.9.6. queryServiceMetrics



Jelenleg NAV által sem dokumentált



Adatszolgáltatási kötelezettség teljesítését nem akadályozza

1.9.7. queryTransactionStatus



Implementálva, tesztelés, integrálás alatt

1.9.8. queryTaxPayer

Implementálva, tesztelve



2.0-s változások (nem implementálva)

- taxpayerShortName
- vatGroupMembership

címadatokon belül új mezők:

- region
- lotNumber

1.9.9 tokenExchange

Implementálva Tesztelve

From:

<https://doc.evir.hu/> - **eVIR tudásbázis**

Permanent link:

https://doc.evir.hu/doku.php/spec:nav_onlineszamla_api?rev=1575304556

Last update: **2019/12/02 17:35**

