Előszó

Ez a dokumentáció, az onlineszámla adatszolgáltatás API-nak való megfelelőséggel kapcsolatos információkat tartalmaz.

Ez az onlineszamla adatszolgáltatás V2.0 verziójának megfelelően dokumentál.

A dokumentációban 🔧 Fix Me! tagekkel felhívom a figyelmet azokra a pontokra ahol az implementáció a lehetséges legbővebb változathoz képest kevesebbet nyújt. Erre természetesen meg van a lehetőség, hiszen a NAV által felkínált lehetőségek közül számos nem kötelező.

Indoklásképpen a következő fő változatok lehetségesek

- O: Nem kötelező implementálni
  - O1: Az eVir-ben az adott feature nincs biztosítva, így a(z adat)szolgáltatás során nem merül fel a használata
  - O2: Nem élünk a(z adat)szolgáltatás lehetőségével
  - O3: Az implementáció időzítése alapján nem fejeződött be. (bármit is jelentsen ez :)

1. Bevezető

- [ ] Implementálandó v. Implementálható
- [.] Implementálás alatt
- [t] Implementált, de tesztelni kell 🔧 Fix Me!
- [X] Implementált, tesztelt

- [t] manageAnnulment
- [t] manageInvoice
  - [.] nav_onlineszamla_data
- [ ] queryInvoiceChainDigest
- [X] queryInvoiceCheck
- [X] queryInvoiceData
- [ ] queryInvoiceDigest
- [ ] queryTransactionList
- [t] queryTransactionStatus
- [X] queryTaxpayer
- [X] tokenExchange

1.1 folyamat

A rendszer egy technikai azonosítóval dolgozik.

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

A jövőben lehetséges implementáció, hogy tömeges számlázáskor egy tokent igényel az egyszerre kiállított számlákhoz (trigger) ezért a tömeges számlázásnál nem emelhető meg 100 fölé az egyszerre kiállított számlák száma.

Pillanatnyilag a tömeges számlázáskor egyesével igényelt tokenekkel küldi be a számlákat.

Ha a számlák ismételt beküldése esetén (átmeneti technikai sikertelenség esetén) az újraküldésekhez már

számlánként egyenként igényel tokent.

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 valaszt feldolgoz
- `process_result` ertelmezi HTTP es hibakod szinten Fix Me!
- `process_response_xml` csak parsol

1.4.1. result

- 1) nem használtuk Fix Me!
- 2) `errorCode` Fix Me! ujrairni amit kell
- 3) message behozása a hibakezelésbe Fix Me!

1.5.1 requestSignature számolás manageInvoice és manageAnnulment

Új kötelező implementáció

- manageInvoice tesztelve, működik
- manageAnnulment Opcionális Fix Me! O3

1.5.2. requestSignature számolás egyébként

Kötelezően implementálva

1.5.3 UTC

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

1.6. A szolgáltatás technikai leírása

1.6.1. általános adatok

Fix Me! context root verzio nélkül???

1.6.2.

Mint 1. pontban

1.6.3. http header

`onlineszamla→calc_header`

1.6.4. http statusok

lásd 1.4 ...

1.6.5. tömörítés

Opcionális. Nem tömörítünk.

1.6.6. timeout

`onlineszamla→http_ua_request`

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

default 5 sec.

timeout eseten sajat response kodunk van:

504 Client-side timeout Fix Me! átvezetni a többi mellé a doksiban

1.6.7. szerveridő, NTP

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

1.6.8. karbantartási mód

Fix Me! ezt implementalni kellhet

1.6.9. verziókezelés

logikus

1.7. req resp elementek

1.4. szerint

1.8.1. manageAnnulment

Implementálva Fix Me! Tesztelni kell majd

1.8.2. manageInvoice

Tömörítést nem használunk

Implementált Fix Me! tesztelés alatt

A beküldött számla adattartalmára vonatkozó információk a nav_onlineszamla_data oldalon találhatóak.

1.8.3. queryInvoiceChainDigest

Fix Me! 2020-02 új

1.8.4. queryInvoiceCheck

Implementálva Tesztelve

Fix Me! batch feldolgozás és vevő oldali lekérdezés nélkül

Fix Me! használja-e valaki (close_trigger pl?) lásd: invoice_data Fix Me! !!!

1.8.5. queryInvoiceData

Implementálva. Tesztelve

Fix Me! Használata még változás alatt van

Fix Me! batch feldolgozás és vevő oldali lekérdezés nélkül

1.8.6. queryInvoiceDigest

Fix Me! Implementálandó

Adatszolgáltatási kötelezettség teljesítését nem akadályozza

1.8.7. queryTransactionList

Fix Me! 2020-02 új

1.8.8. queryTransactionStatus

Implementálva, close_trigger hívja, mikor kell. Elvileg tesztelve, és működik

1.8.9. queryTaxPayer

Implementálva, tesztelve

Implementált 2.0-s változások: V2-20-20

- cím adatok kinyerése HQ-ra partnerek modulban
- Ez tesztelve is van v_2 esetére

További 2.0-s változások (nem implementálva) Fix Me!

- taxpayerShortName
- vatGroupMembership

címadatokon belül új mezők:

- region
- lotNumber

Különösen nincsenek implementálva Fix Me!

1.8.10. tokenExchange

Implementálva Tesztelve

From:
https://doc.evir.hu/ - **eVIR tudásbázis**

Permanent link:
https://doc.evir.hu/doku.php/spec:nav_onlineszamla_api

Last update: **2020/03/04 18:24**