

# **Számlázz.hu Agent**

## **PHP dokumentáció**

### **v2.9**

## **SZÁMLA és NYUGTA**

KBOSS.hu Kft. - <https://www.szamlazz.hu>

E-mail: [fejlesztes@szamlazz.hu](mailto:fejlesztes@szamlazz.hu)

2020.07.14.

|   |           |
|---|-----------|
| <b>1 Bevezető</b>                       | <b>3</b>  |
| <b>2 Mappaszerkezet</b>                 | <b>3</b>  |
| <b>3 Alapok</b>                         | <b>4</b>  |
| 3.1 Karakterkódolás                     | 4         |
| 3.2 Inicializálás                       | 5         |
| 3.2.1 Alapértelmezett beállítások       | 5         |
| 3.2.1.1 Naplózási szintek (logLevel)    | 6         |
| 3.2.1.2 Naplózási fájlok                | 6         |
| 3.2.1.3 Naplózási e-mail                | 6         |
| 3.2.1.4 PDF dokumentum a válaszban      | 6         |
| 3.2.1.5 Cookie és Session kezelés       | 7         |
| 3.2.1.6 Tanúsítvány használata          | 7         |
| 3.2.1.7 Agent hívás módszerei           | 7         |
| 3.3 Hibák és kivételek kezelése         | 8         |
| <b>4 Példák az API használathoz</b>     | <b>9</b>  |
| 4.1 Számlával kapcsolatos példák        | 9         |
| 4.2 Nyugtával kapcsolatos példák        | 10        |
| 4.3 Adózóval kapcsolatos példák         | 10        |
| <b>5 Számla Agent kérés elküldése</b>   | <b>11</b> |
| 5.1 Számla generálás                    | 11        |
| 5.1.1 Elvárt, és opcionális bemenetek   | 12        |
| 5.2 Sztornó számla generálás            | 15        |
| 5.2.1 Elvárt, és opcionális bemenetek   | 16        |
| 5.3 Jóváírások rögzítése                | 17        |
| 5.3.1 Elvárt, és opcionális bemenetek   | 17        |
| 5.4 Számla adatok lekérdezése           | 18        |
| 5.4.1 Elvárt, és opcionális bemenetek   | 18        |
| 5.5 Számla letöltése PDF dokumentumként | 19        |
| 5.5.1 Elvárt, és opcionális bemenetek   | 19        |
| 5.6 Díjbekérő törlése                   | 20        |
| 5.6.1 Elvárt, és opcionális bemenetek   | 20        |
| 5.7 Nyugta generálás                    | 21        |
| 5.7.1 Elvárt, és opcionális bemenetek   | 22        |
| 5.8 Sztornó nyugta generálás            | 23        |
| 5.8.1 Elvárt, és opcionális bemenetek   | 23        |
| 5.9 Nyugta kiküldése                    | 24        |
| 5.9.1 Elvárt, és opcionális bemenetek   | 25        |
| 5.10 Nyugta adatok lekérdezése          | 25        |
| 5.10.1 Elvárt, és opcionális bemenetek  | 25        |

|   |           |
|---|-----------|
| 5.11 Nyugta adatok letöltése PDF dokumentumként | 26        |
| 5.11.1 Elvárt, és opcionális bemenetek          | 26        |
| <b>6 Számla Agent válasz feldolgozása</b>       | <b>27</b> |

## 1 Bevezető

A **Számla Agent PHP API** (továbbiakban Agent) feladata a Számla Agent rendszerével történő adatkommunikáció egyszerűbbé tétele a végfelhasználók számára. Nem szükséges a felhasználó számára az xml generálási metódusok, és a php alapú távoli adatkapcsolati funkciók használatának ismerete, php osztályunk elvégzi ezt a munkát egy egyszerű eljárásívási metodika mögé bújtatva.

Ez a dokumentáció a [Számla Agent dokumentációja](#) (v3.71) alapján készült. Minden egyes funkció megvalósításához külön példát találsz az "examples" könyvtárban.

Az egyes példák reprezentálják, hogy hogyan hozd létre vagy kérdezd le a különböző adatokat az Agent használatával.

Minden példa az alábbi lépések szerint épül fel:

1. Agent beállítása: számlázz.hu fiók adatok megadása (felhasználónév és jelszó)
2. Az elküldendő adatok létrehozása, beállítása (pl. számla, nyugta stb.)
3. A kérés elküldése a Számlázz.hu felé
4. A válasz fogadása és feldolgozása (sikeres és sikertelen esetek esetén is)

## 2 Mappaszerkezet

A Számla Agent PHP API mappaszerkezete a következők szerint épül fel:

|               |   |
|---------------|---|
| ./attachments | az API kérésekhez csatolt fájlok (pl. számlamellékletek).                         |
| ./cert        | az API használatához szükséges tanúsítványok.                                     |
| ./docs        | az API használatát leíró dokumentációk.   |
| ./examples    | az API használatához tartozó példák (autoloader helye).                           |
| ./logs        | az API naplózása (napi bontásban).  |
| ./pdf         | az API által létrehozott vagy lekérdezett PDF típusú fájlok.                      |
| ./src         | az API osztálykönyvtára, amely tartalmazza az API működéséhez szükséges fájlokat. |
| ./vendor      | 3rd party eszközök és kiegészítések gyűjtőhelye.                                  |
| ./xmls        | az API által létrehozott (request) vagy lekérdezett (response) XML típusú fájlok. |

## 3 Alapok

A különböző példák használatához hívjuk meg a csomagban található **../examples/autoload.php** fájlt, amely automatikusan betölti azokat az osztályokat, amelyekre az Agent kérések elküldésénél szükségünk lesz. Ezt csupán egy alkalommal szükséges megtennünk, abszolút vagy akár relatív elérési útvonal megadásával, az igényünk szerint.

```
require_once(__DIR__ . '/../examples/autoload.php');
```

Az API a 'SzamlaAgent' névtéren (namespace) keresztül érhető el. A példáknál láthatod a helyes használatot. A fejlesztői környezetek (IDE) automatikusan kiegészítik az importálandó osztályok névtereit, ellenkező esetben ezeket neked kell pótolnod a következőképpen:

```
use \SzamlaAgent\SzamlaAgentAPI,  
    \SzamlaAgent\Buyer,  
    \SzamlaAgent\Item\InvoiceItem,  
    \SzamlaAgent\Document\Invoice\Invoice;
```

A fenti példa egy papír- vagy e-számla kiállításához használt osztályokat importálja.

**A példát megtalálod az API csomagban** a következő helyen:

```
./examples/document/invoice/create_invoice_with_default_data.php
```

A Számla Agent PHP API használatához szükséges minimum PHP verzió: 5.6.0.

### 3.1 Karakterkódolás

Az Agent kérések és válaszok alapértelmezetten **utf-8** karakterkódolást használnak.

## 3.2 Inicializálás

Az Agent kommunikáció előkészítéséhez hozzunk létre egy Agent példányt:

```
$agent = SzamlaAgentAPI::create('agentApiKey');
```

A Számla Agent kulcsot ('apiKey') abban a Számlázz.hu fiókban kell létrehoznod, amellyel az Agent-et használni szeretnéd (figyelj arra, hogy ez olyan felhasználó legyen, aki csak egyetlen számlázási fiókhoz fér hozzá). A fentebbi metódus alkalmazásával az Agent kommunikációját biztosító PHP osztály létrejön az alapértelmezett beállításokkal együtt.

Erről bővebben itt olvashatsz:

[https://www.szamlazz.hu/blog/2019/07/szamla\\_agent\\_kulcsok/](https://www.szamlazz.hu/blog/2019/07/szamla_agent_kulcsok/)

- sikeres válasz (response) esetén az adatok tartalmazni fogják a létrejött bizonylatot (1 példányban) PDF formátumban.
- a naplózási szint alapértelmezetten DEBUG, amelyet megváltoztathatsz például csak hibanaplózásra a következőképpen:

```
$agent = SzamlaAgentAPI::create('apiKey', true, Log::LOG_LEVEL_ERROR);
```

vagy külön paraméterezve:

```
$agent->setLogLevel(Log::LOG_LEVEL_ERROR);
```

- Ha szeretnél e-mailben értesülni arról, hogy ha valami hiba történik az Agent működése közben, akkor ezt a következőképpen tudod beállítani:

```
$agent->setLogEmail('example@example.com');
```

### 3.2.1 Alapértelmezett beállítások

Az alábbiakban bemutatjuk, hogy milyen beállításokat érhetsz el.

Fontos, hogy a PHP a "logs", "xmls", "pdf" könyvtárakat képes legyen írni.

### 3.2.1.1 Naplózási szintek (LogLevel)

Az alkalmazott logolás „szintjét” határozza meg.

- **0** - nincs naplózás: a megadott érték inaktívvá teszi a naplózást.
- **1** - hibák: csakis 'error' típusú eseményt rögzít, illetve igény szerint emailt küld erről.
- **2** - figyelmeztetések: a hibanaplózáson felül rögzítésre kerülnek a figyelmeztetések is, amelyek nem befolyásolják feltétlenül az üzemszerű működést.
- **3** - debug mód: a fentiek felül a funkciók üzemszerű működéséről is folyamatos tájékoztatást kapunk.

```
0: LOG_LEVEL_OFF - nincs naplózás  
1: LOG_LEVEL_ERROR - hibák naplózása  
2: LOG_LEVEL_WARN - hibák és figyelmeztetések naplózása  
3: LOG_LEVEL_DEBUG - minden típus naplózása (fejlesztői mód)
```

### 3.2.1.2 Naplózási fájlok

Minden naplófájl alapértelmezett beállítások szerint a ./logs könyvtárba generálódik és napi bontásban tartalmazza a naplózott eseményeket, hibákat. A naplózás az alábbi módokon kapcsolható ki:

```
$agent = SzamlaAgentAPI::create('agentApiKey', true, Log::LOG_LEVEL_OFF);
```

### 3.2.1.3 Naplózási e-mail

Amennyiben kitöltjük, a honlap adminisztrátora e-mailt kap az error szintű eseményekről az itt megadott email címére. Ha nem töltjük ki, akkor a funkció inaktív marad.

```
$agent->setLogEmail('example@example.com');
```

### 3.2.1.4 PDF dokumentum a válaszban

Beállíthatjuk, hogy az egyes eljárshívások válaszában (ahol alkalmazható), megkapjuk-e a bizonylatot pdf formátumban. Alapértelmezett beállítás szerint minden válasz tartalmazni fogja a pdf-et, ezért ezt csak akkor kell beállítanunk, ha nem szeretnénk megkapni azt. Ezt az alábbi módokon tehetjük meg:

```
$agent = SzamlaAgentAPI::create('apiKey', false);
```

vagy külön paraméterezve:

```
$agent->setDownloadPdf(false);
```

Az elmentett pdf fájl minden esetben a generált számla sorszáma szerint kerül elnevezésre.

### 3.2.1.5 Cookie és Session kezelés

Biztonsági, és erőforrás takarékosági szempontból egyaránt fontos, hogy az egyes eljárás hívások során a kliens egy felhasználónak látszódjon az Agent távoli rendszere felől. A Számla Agent API példányosításával az API automatikusan beállítja az Agent kérésekhez szükséges session sütit, amely biztosítja, hogy a további kérések során a számlazz.hu rendszere egy munkamenetben kezelje a beérkező kéréseket. Ez a gyakorlatban azt jelenti, hogy minden egyes Számla Agent API példányhoz külön süti fájl fog tartozni, ami biztosítja, hogy a példányokhoz kapcsolódó számlázási fiókok felé elküldött kérések külön session-t használjanak.

Ha mégis meg szeretnénk változtatni ennek útvonalát, az alábbiak szerint módosíthatjuk:

```
$agent->setCookieFileName('uj-cookie-fajlnév.txt');
```

### 3.2.1.6 Tanúsítvány használata

Egyes szerver konfigurációk nem rendelkeznek a távoli https alapú adatkapcsolatok létesítéséhez, és a biztonsági ellenőrzéshez szükséges tanúsítvány csomaggal. Ezen problémát kiküszöbölendő, hogy szükséges egy root certificate egyedi megadása. Az Agent alapértelmezetten a saját tanúsítványát használja, de megadható egyéni tanúsítvány is az alábbi módon:

```
$agent->setKeychain('fájl-útiránya');
```

### 3.2.1.7 Agent hívás módszerei

Az Agent API elsődlegesen a legmodernebb **CURL** metódus használatával kommunikál. Bizonyos szerver konfigurációkon azonban előfordul, hogy a CURL telepítés hiányos vagy nem kerültek telepítésre a szükséges tanúsítvány csomagok, így ezekben az esetekben az eljáráshívás bármely visszajelzése hibába ütközhet. Ezt kiküszöbölendő integráltuk a fentebb említett tanúsítvány automatikus importálását. Bizonyos esetekben ez a megoldás is problémába ütközhet, például rendszerszintű tiltás esetén. Ilyen esetekben az adatkommunikáció egy régebbi **'file\_get\_contents'** típusú híváson bonyolítható le. Az Agent automatikusan határozza meg a hívás módszerét, így ezt külön nem kell beállítanunk, de ha mégis meg szeretnénk tenni, akkor azt következőképpen tehetjük:

```
$agent->setCallMethod(SzamlaAgentRequest::CALL_METHOD_CURL);
```

Az Agent hívás módszerei az alábbiak lehetnek:

1. **Natív (legacy) beállítás esetén** a esetén a PHP csak a **'file\_get\_contents'** hívásokkal kommunikál a távoli szerverrel. Ez a módszer akkor alkalmazandó, amikor tudjuk, hogy a CURL lekérések hibába fognak ütközni, és nem várható ezen probléma megoldása.
2. **CURL beállítás esetén** a rendszer kizárólagosan CURL metódus alkalmazásával kommunikál, hiba esetén nem próbálkozik mással.

3. **Automatikus beállítás esetén** (alapértelmezett) az osztály először CURL hívás megkísérlésével próbálkozik, mely sikertelensége esetén az eredményt az adott munkafolyamatra eltárolva visszavált a 'file\_get\_contents' módszerére.

```
1: CALL_METHOD_LEGACY - natív
2: CALL_METHOD_CURL   - CURL
3: CALL_METHOD_AUTO   - automatikus
```

**Vigyázat!** Számos informatikai fórumon tévesen azt állítják, hogy a hibába ütköző CURL eljárás hívások kijavíthatók a távoli szerver SSL hitelesítésének deaktiválásával. Ez a megoldási javaslat semmilyen esetben sem követendő, hiszen biztonsági kockázatot hordoz magában. Az ilyen módon folytatott kommunikációért, és a kommunikáció során keletkezett esetleges károkért az Agent fejlesztői és üzemeltetői nem vonhatóak felelősségre!

### 3.3 Hibák és kivételek kezelése

Ha a Számla Agent használata közben bármilyen hiba vagy kivétel keletkezik, arról az Agent egy egyéni kivételt dob (SzamlaAgentException), amelyet egy try-catch kivétel kezeléssel el tudunk kapni az alábbi módon:

```
try {
    $agent = SzamlaAgentAPI::create('agentApiKey');
    ...
    ...
    if ($result->isSuccess()) {
        ...
    }
} catch (SzamlaAgentException $e) {
    $agent->logError($e->getMessage());
} catch (\Exception $e) {
    $agent->logError($e->getMessage());
}
```

Minden hibáról napló bejegyzés készül a ./logs/{current-date} fájlba. Ahhoz, hogy minden hibát naplózzon az Agent, a naplózási szintet fejlesztői módba érdemes állítanunk:

```
$agent = SzamlaAgentAPI::create('apiKey', true, Log::LOG_LEVEL_DEBUG);
```

vagy külön paraméterezve:

```
$agent->setLogLevel(Log::LOG_LEVEL_DEBUG);
```



## 4 Példák az API használathoz

Az API számos előre elkészített példát tartalmaz, amelyeket felhasználva elkészítheted, vagy lekérdezheted a már létrehozott bizonylataidat.

Az alábbi táblázat felsorolja a csomagban található példákat csoportokra bontva:

### 4.1 Számlával kapcsolatos példák

| Funkció elnevezése                             | SzamlaAgent metódus       | Példafájl útvonala   |
|--|---------------------------|--|
| normál számla létrehozása                      | generateInvoice           | ./examples/document/invoice/<br>create_invoice_with_default_data.php |
| normál számla létrehozása<br>egyedi adatokkal  | generateInvoice           | ./examples/document/invoice/<br>create_invoice_with_custom_data.php  |
| normál számla létrehozása<br>fuvarlevéllel     | generateInvoice           | ./examples/document/invoice/<br>create_invoice_with_waybill.php      |
| normál számla létrehozása<br>fájl melléklettel | generateInvoice           | ./examples/document/invoice/<br>create_invoice_with_attachment.php   |
| díjbekérő létrehozása                          | generateProforma          | ./examples/document/create_proforma.php                              |
| díjbekérő törlése                              | deleteProforma            | ./examples/document/delete_proforma.php                              |
| szállítólevél létrehozása                      | generateDeliveryNote      | ./examples/document/create_delivery_note.php                         |
| előlegszámla létrehozása                       | generatePrePaymentInvoice | ./examples/document/invoice/<br>create_prepayment_invoice.php        |
| vég számla létrehozása                         | generateFinalInvoice      | ./examples/document/invoice/<br>create_finalpayment_invoice.php      |
| helyesbítő számla létrehozása                  | generateCorrectiveInvoice | ./examples/document/invoice/<br>create_corrective_invoice.php        |
| sztorzó számla létrehozása                     | generateReverseInvoice    | ./examples/document/invoice/<br>create_reverse_invoice.php           |
| számla adatok lekérdezése                      | getInvoiceData            | ./examples/document/invoice/<br>get_invoice_data.php                 |
| számla lekérdezése<br>PDF formátumban          | getInvoicePdf             | ./examples/document/invoice/<br>get_invoice_pdf.php                  |
| számla jóváírások rögzítése                    | payInvoice                | ./examples/document/invoice/pay_invoice.php                          |

## 4.2 Nyugtával kapcsolatos példák

| Funkció elnevezése                    | SzamlaAgent metódus    | Példafájl útvonala   |
|---------------------------------------|------------------------|--|
| nyugta létrehozása                    | generateReceipt        | ./examples/document/receipt/<br>create_receipt_with_default_data.php |
| nyugta létrehozás<br>egyedi adatokkal | generateReceipt        | ./examples/document/receipt/<br>create_receipt_with_custom_data.php  |
| sztorító nyugta létrehozása           | generateReverseReceipt | ./examples/document/receipt/<br>create_reverse_receipt.php           |
| nyugta adatok lekérdezése             | getReceiptData         | ./examples/document/receipt/<br>get_receipt_data.php                 |
| nyugta lekérdezése PDF<br>formátumban | getReceiptPdf          | ./examples/document/receipt/<br>get_receipt_pdf.php                  |
| nyugta elküldése                      | sendReceipt            | ./examples/document/receipt/<br>send_receipt.php                     |

## 4.3 Adózóval kapcsolatos példák

| Funkció elnevezése          | SzamlaAgent metódus | Példafájl útvonala          |
|-----------------------------|---------------------|-----------------------------|
| adózó adatainak lekérdezése | getTaxPayer         | ./examples/get_taxpayer.php |

## 5 Számla Agent kérés elküldése

### 5.1 Számla generálás

Egy normál számla generálásának teljes példáját a következő fájl szemlélteti:

*./examples/document/invoice/create\_invoice\_with\_custom\_data.php*

A számla generálásához a következő lépések szükségesek:

1. **Papír számla létrehozása** (átutalással fizetendő magyar nyelvű (Ft) számla kiállítása mai keltezési és teljesítési dátummal, +8 nap fizetési határidővel, üres számla előtaggal).

```
$invoice = new Invoice(Invoice::INVOICE_TYPE_P_INVOICE);
```

2. **Egyedi fejléc** adatok hozzáadása a számlához (pl. számlakép sablon)

```
$invoice->getHeader()->setInvoiceTemplate(Invoice::INVOICE_TEMPLATE_8CM);
```

3. **Eladói** adatok hozzáadása a számlához

```
$invoice->setSeller(new Seller('OBER', '11111111-22222222-33333333'));
```

4. **Vevő** adatainak hozzáadása (kötelezően kitöltendő adatokkal)

```
$buyer = new Buyer('Kovacs Bt.', '2030', 'Érd', 'Tarnoki street 23.');
```

- a. Vevői adatok hozzáadása a számlához

```
$invoice->setBuyer($buyer);
```

5. **Számla tétel** összeállítása alapértelmezett adatokkal  
(1 db tétel 27%-os ÁFA tartalommal)

```
$item = new InvoiceItem('Eladó tétel 1', 10000.0);
```

- a. Tétel nettó értékének beállítása

```
$item->setNetPrice(10000.0);
```

- b. Tétel ÁFA értékének beállítása

```
$item->setVatAmount(2700.0);
```

- c. Tétel bruttó értékének beállítása

```
$item->setGrossAmount(12700.0);
```

- d. Tétel adatainak hozzáadása a számlához

```
$invoice->addItem($item);
```

6. **Számla elkészítése** (generálás)

```
$result = $agent->generateInvoice($invoice);
```

### 5.1.1 Elvárt, és opcionális bemenetek

| Számla fejléc adatok    |                 |         |          |                 |
|-------------------------|-----------------|---------|----------|-----------------|
| megnevezés              | mező            | típus   | kötelező | alapérték       |
| típus                   | invoiceType     | int     | igen     | 1               |
| keltezés dátum          | issueDate       | date    | igen     | mai nap         |
| teljesítés dátum        | fulfillment     | date    | igen     | mai nap         |
| fizetési határidő       | paymentDue      | date    | igen     | keltezés +8 nap |
| fizetési mód            | paymentMethod   | string  | igen     | átutalás        |
| pénznem                 | currency        | string  | igen     | Ft              |
| nyelv                   | language        | string  | igen     | hu              |
| megjegyzés              | comment         | string  |          |                 |
| bank név                | exchangeBank    | string  |          |                 |
| banki árfolyam          | exchangeRate    | string  |          |                 |
| rendelésszám            | orderNumber     | string  |          |                 |
| díjbekérőszám           | proformaNumber  | string  |          |                 |
| logo                    | extraLogo       | string  |          |                 |
| előtag                  | prefix          | string  |          |                 |
| fizetendő korrekció     | correctionToPay | double  |          |                 |
| kifizettség             | paid            | boolean |          | false           |
| ÁFA árrés               | profitVat       | boolean |          | false           |
| számla sablon           | invoiceTemplate | int     |          |                 |
| Helyesbített számlaszám | correctedNumber | string  |          |                 |

| Számla eladói adatok |               |        |          |           |
|----------------------|---------------|--------|----------|-----------|
| megnevezés           | mező          | típus  | kötelező | alapérték |
| bank                 | bank          | string |          |           |
| bankszámlaszám       | bankAccount   | string |          |           |
| aláíró neve          | signatoryName | string |          |           |
| e-mail válaszcím     | emailReplyTo  | string |          |           |
| e-mail tárgy         | emailSubject  | string |          |           |
| e-mail tartalma      | emailContent  | string |          |           |

| Számla vevői adatok |               |            |          |           |
|---------------------|---------------|------------|----------|-----------|
| megnevezés          | mező          | típus      | kötelező | alapérték |
| azonosító           | id            | string     |          |           |
| név                 | name          | string     | igen     |           |
| ország              | country       | string     |          |           |
| irányítószám        | zipCode       | string     | igen     |           |
| város               | city          | string     | igen     |           |
| cím                 | address       | string     | igen     |           |
| e-mail cím          | email         | string     |          |           |
| küldünk-e e-mail    | sendEmail     | boolean    |          | true      |
| adószám             | taxNumber     | string     |          |           |
| EU adószám          | taxNumberEU   | string     |          |           |
| postázási név       | postalName    | string     |          |           |
| postázási ország    | postalCountry | string     |          |           |
| postázási irsz.     | postalZip     | string     |          |           |
| postázási város     | postalCity    | string     |          |           |
| postázási cím       | postalAddress | string     |          |           |
| főkönyvi adatok     | ledgerData    | LedgerData |          |           |
| aláíró neve         | signatoryName | string     |          |           |
| telefonszám         | phone         | string     |          |           |
| megjegyzés          | comment       | string     |          |           |

| Számla vevői főkönyvi adatok |                       |         |          |           |
|------------------------------|-----------------------|---------|----------|-----------|
| megnevezés                   | mező                  | típus   | kötelező | alapérték |
| vevő azonosító               | buyerId               | string  |          |           |
| könyvelés dátum              | bookingDate           | date    | igen     |           |
| főkönyvi szám                | buyerLedgerNumber     | string  |          |           |
| folyamatos teljesítés        | continuedFulfillment  | boolean | igen     | false     |
| elszámolási időszak kezdete  | settlementPeriodStart | date    |          |           |
| elszámolási időszak vége     | settlementPeriodEnd   | date    |          |           |

| Számlatétel adatok |                 |         |          |           |
|--------------------|-----------------|---------|----------|-----------|
| megnevezés         | mező            | típus   | kötelező | alapérték |
| azonosító          | id              | string  |          |           |
| megnevezés         | name            | string  | igen     |           |
| mennyiség          | quantity        | double  | igen     |           |
| mennyiségi egység  | quantityUnit    | string  | igen     |           |
| nettó egységár     | netUnitPrice    | double  | igen     |           |
| ÁFA kulcs          | vat             | string  | igen     |           |
| árrés ÁFA alap     | priceGapVatBase | double  |          |           |
| nettó érték        | netPrice        | double  | igen     |           |
| ÁFA érték          | vatAmount       | double  | igen     |           |
| bruttó érték       | grossAmount     | double  | igen     |           |
| megjegyzés         | comment         | comment |          |           |

| Számlatétel főkönyvi adatok              |                         |        |          |           |
|--|-------------------------|--------|----------|-----------|
| megnevezés                               | mező                    | típus  | kötelező | alapérték |
| Árbevétel főkönyvi szám                  | revenueLedgerNumber     | string |          |           |
| ÁFA főkönyvi szám                        | vatLedgerNumber         | string |          |           |
| Gazdasági esemény típus                  | economicEventType       | string |          |           |
| ÁFA gazdasági esemény típus              | vatEconomicEventTy<br>e | string |          |           |
| Számla tétel elszámolási időszak kezdete | settlementPeriodStart   | date   |          |           |
| Számla tétel elszámolási időszak vége    | settlementPeriodEnd     | date   |          |           |

## 5.2 Sztornó számla generálás

A sztornó számla generálásának példáját a következő fájl szemlélteti:  
*./examples/document/invoice/create\_reverse\_invoice.php*

A sztornó számla generálásához a következő lépések szükségesek:

1. **Új sztornó számla** létrehozása egyedi adatokkal

```
$invoice = new ReverseInvoice(ReverseInvoice::INVOICE_TYPE_P_INVOICE);
```

2. **Számla fejléc** lekérdezése

```
$header = $invoice->getHeader();
```

3. Számlaszám beállítása

```
$header->setInvoiceNumber('TESZT-001');
```

4. Számla kiállítás és teljesítés dátumának beállítása

```
$header->setIssueDate('2019-03-19');  
$header->setFulfillment('2019-03-28');
```

5. **Eladó** létrehozása

```
$seller = new Seller();
```

6. Eladó e-mail adatok beállítása

```
$seller->setEmailReplyTo('hello@evulon.hu');  
$seller->setEmailSubject('Számla értesítő');  
$seller->setEmailContent('Fizesse ki a számlát, különben a mindenkori banki...');
```

7. Eladó hozzáadása a számlához

```
$invoice->setSeller($seller);
```

8. **Vevő** létrehozása

```
$buyer = new Buyer();
```

9. Vevő e-mail címének beállítása

```
$buyer->setEmail('vevoneve@example.org');
```

10. Vevő hozzáadása a számlához

```
$invoice->setBuyer($buyer);
```

11. **Sztornó számla elkészítése**

```
$result = $agent->generateReverseInvoice($invoice);
```

### 5.2.1 Elvárt, és opcionális bemenetek

| Számla fejléc adatok |                 |        |          |           |
|----------------------|-----------------|--------|----------|-----------|
| megnevezés           | mező            | típus  | kötelező | alapérték |
| számlaszám           | invoiceNumber   | string | igen     |           |
| kelt dátum           | issueDate       | date   |          |           |
| teljesítés dátum     | fulFillment     | date   |          |           |
| számla sablon        | invoiceTemplate | int    |          |           |

| Számla eladói adatok |              |        |          |           |
|----------------------|--------------|--------|----------|-----------|
| megnevezés           | mező         | típus  | kötelező | alapérték |
| e-mail válaszcím     | emailReplyTo | string |          |           |
| e-mail tárgy         | emailSubject | string |          |           |
| e-mail tartalma      | emailContent | string |          |           |

| Számla vevői adatok |       |        |          |           |
|---------------------|-------|--------|----------|-----------|
| megnevezés          | mező  | típus  | kötelező | alapérték |
| e-mail              | email | string |          |           |



## 5.3 Jóváírások rögzítése

Egy-egy számla befizettként történő megjelölését, vagy részletfizetések hozzáadását a következő példafájl szemlélteti: `./examples/document/invoice/pay_invoice.php`

Egy számlához tartozó jóváírás rögzítéséhez a következő lépések szükségesek:

1. Papír számla inicializálása

```
$invoice = new Invoice(Invoice::INVOICE_TYPE_E_INVOICE);
```

2. Fejléc adatok beállítása

```
$header = new InvoiceHeader();  
$header->setInvoiceNumber('TESZT-2019-001');
```

3. Fejléc adatok hozzáadása a számlához

```
$invoice->setHeader($header);
```

4. Létrehozzuk az új jóváírást

```
$creditNote = new InvoiceCreditNote('2019-03-20', 10000.0,  
Document::PAYMENT_METHOD_BANKCARD, 'TESZT');
```

5. Hozzáadjuk a jóváírást a számlához

```
$invoice->addCreditNote($creditNote);
```

6. Elküldjük a számla jóváírást

```
$result = $agent->payInvoice($invoice);
```

### 5.3.1 Elvárt, és opcionális bemenetek

| Számla adatok  |               |         |          |           |
|----------------|---------------|---------|----------|-----------|
| megnevezés     | mező          | típus   | kötelező | alapérték |
| számlaszám     | invoiceNumber | string  | igen     |           |
| jóváírás módja | additive      | boolean | igen     | true      |

| Jóváírás adatok (kifizetés) |               |        |          |           |
|-----------------------------|---------------|--------|----------|-----------|
| megnevezés                  | mező          | típus  | kötelező | alapérték |
| dátum                       | date          | string | igen     | mai nap   |
| jogcím (fizetési mód)       | paymentMethod | string | igen     | átutalás  |
| összeg                      | amount        | double | igen     | 0.0       |
| leírás                      | description   | string |          |           |

## 5.4 Számla adatok lekérdezése

Egy már létező számla adatai lekérdezhetők. A megvalósítást a következő példafájl szemlélteti: `./examples/document/invoice/get_invoice_data.php`

A számla adatait lekérdezhetjük számlaszám vagy rendelésszám alapján. Rendelésszám alapján való lekérdezés esetén az legutolsó bizonylatot adjuk vissza, amelyiknek ez a rendelésszáma. Utolsó paraméterként megadható, ha szeretnénk a számla PDF-et is letölteni.

Letöltés **számlaszám** alapján:

```
$agent->getInvoiceData('TESZT-001');
```

Letöltés **rendelésszám** alapján (számla PDF-el együtt):

```
$agent->getInvoiceData('TESZT-001', Invoice::FROM_ORDER_NUMBER, true);
```

Az összes számlaadatot a következőképpen kaphatjuk meg:

```
$result->getData();
```

### 5.4.1 Elvárt, és opcionális bemenetek

| Számla adatok |               |         |          |           |
|---------------|---------------|---------|----------|-----------|
| megnevezés    | mező          | típus   | kötelező | alapérték |
| számlaszám    | invoiceNumber | string  | igen     |           |
| rendelésszám  | orderNumber   | string  |          |           |
| pdf letöltés  | downloadPdf   | boolean |          | false     |

## 5.5 Számla letöltése PDF dokumentumként

Egy már létező számla letölthető PDF dokumentumként. A megvalósítást a következő példafájl szemlélteti: `./examples/document/invoice/get_invoice_pdf.php`

A Számla PDF letöltése számlaszám vagy rendelésszám alapján is lehetséges. Rendelésszám alapján való letöltés esetén az legutolsó bizonylatot adjuk vissza, amelyeknek ez a rendelésszáma.

Letöltés **számlaszám** alapján:

```
$agent->getInvoicePdf('TESZT-001');
```

Letöltés **rendelésszám** alapján:

```
$agent->getInvoicePdf('TESZT-001', Invoice::FROM_ORDER_NUMBER);
```

A válasz eredménye a PDF típusú számla, amelyet az alábbi módon tölthetünk le:

```
$result->downloadPdf();
```

### 5.5.1 Elvárt, és opcionális bemenetek

| Számla adatok |               |        |          |           |
|---------------|---------------|--------|----------|-----------|
| megnevezés    | mező          | típus  | kötelező | alapérték |
| számlaszám    | invoiceNumber | string | igen     |           |
| rendelésszám  | orderNumber   | string |          |           |

## 5.6 Díjbekérő törlése

A díjbekérő törlését a következő példafájl szemlélteti:

`./examples/document/delete_proforma.php`

Rendelésszám alapján több díjbekérőt is törölhetünk. Ha valamelyik díjbekérő törlése nem sikerül, akkor rollbackelünk.

Díjbekérő törlése **számlaszám** alapján:

```
$agent->deleteProforma('D-TEST-001');
```

Díjbekérő törlése **rendelésszám** alapján:

```
$agent->deleteProforma('TEST-001', Proforma::FROM_ORDER_NUMBER);
```

### 5.6.1 Elvárt, és opcionális bemenetek

| Díjbekérő adatok |               |        |          |           |
|------------------|---------------|--------|----------|-----------|
| megnevezés       | mező          | típus  | kötelező | alapérték |
| számlaszám       | invoiceNumber | string |          |           |
| rendelésszám     | orderNumber   | string |          |           |

## 5.7 Nyugta generálás

Egy nyugta generálásának teljes példáját a következő fájl szemlélteti:  
*./examples/document/receipt/create\_receipt\_with\_custom\_data.php*

A nyugta generálásához a következő lépések szükségesek:

1. **Nyugta létrehozása**

(készpénzzel fizetett magyar nyelvű (Ft) nyugta kiállítása)

```
$receipt = new Receipt();
```

2. **Fejléc** inicializálása

```
$receipt->setHeader(new ReceiptHeader());
```

3. **Előtag** beállítása

```
$receipt->getHeader()->setPrefix('NYGTA');
```

4. **Nyugta tétel** összeállítása alapértelmezett adatokkal  
(1 db eladó tétel 27%-os ÁFA tartalommal)

```
$item = new ReceiptItem('Eladó tétel', 10000.0);
```

a. Tétel nettó értékének beállítása

```
$item->setNetPrice(10000.0);
```

b. Tétel ÁFA értékének beállítása

```
$item->setVatAmount(2700.0);
```

c. Tétel bruttó értékének beállítása

```
$item->setGrossAmount(12700.0);
```

d. Tétel adatainak hozzáadása a számlához

```
$receipt->addItem($item);
```

5. **Nyugta elkészítése** (generálás)

```
$result = $agent->generateReceipt($receipt);
```

### 5.7.1 Elvárt, és opcionális bemenetek

| Nyugta fejléc adatok |               |        |          |           |
|----------------------|---------------|--------|----------|-----------|
| megnevezés           | mező          | típus  | kötelező | alapérték |
| hívás azonosító      | callId        | string |          |           |
| előtag               | prefix        | string | igen     |           |
| fizetési mód         | paymentMethod | string | igen     | készpénz  |
| pénznem              | currency      | string | igen     | Ft        |
| bank név             | exchangeBank  | string |          |           |
| banki árfolyam       | exchangeRate  | string |          |           |
| megjegyzés           | comment       | string |          |           |
| pdf sablon           | pdfTemplate   | string |          |           |
| főkönyvi azonosító   | buyerLedgerId | string |          |           |

| Nyugtatétel adatok |              |                   |          |           |
|--------------------|--------------|-------------------|----------|-----------|
| megnevezés         | mező         | típus             | kötelező | alapérték |
| azonosító          | id           | string            |          |           |
| megnevezés         | name         | string            | igen     |           |
| mennyiség          | quantity     | double            | igen     |           |
| mennyiségi egység  | quantityUnit | string            | igen     |           |
| nettó egységár     | netUnitPrice | double            | igen     |           |
| ÁFA kulcs          | vat          | string            | igen     |           |
| nettó érték        | netPrice     | double            | igen     |           |
| ÁFA érték          | vatAmount    | double            | igen     |           |
| bruttó érték       | grossAmount  | double            | igen     |           |
| főkönyvi adatok    | ledgerData   | ReceiptItemLedger |          |           |

| Nyugtatétel főkönyvi adatok |                     |        |          |           |
|-----------------------------|---------------------|--------|----------|-----------|
| megnevezés                  | mező                | típus  | kötelező | alapérték |
| Árbevétel főkönyvi szám     | revenueLedgerNumber | string |          |           |
| ÁFA főkönyvi szám           | vatLedgerNumber     | string |          |           |

| Jóváírás adatok (kifizetés) |               |        |          |           |
|-----------------------------|---------------|--------|----------|-----------|
| megnevezés                  | mező          | típus  | kötelező | alapérték |
| jogcím (fizető eszköz)      | paymentMethod | string | igen     | készpénz  |
| összeg                      | amount        | double | igen     |           |
| leírás                      | description   | string |          |           |

## 5.8 Sztornó nyugta generálás

A sztornó nyugta generálásának példáját a következő fájl szemlélteti:

`./examples/document/receipt/create_reverse_receipt.php`

A sztornó nyugta generálásához a következő lépések szükségesek:

### 12. Sztornó nyugta létrehozása

(paraméterként megadva a sztornózáandó nyugta számát)

```
$receipt = new ReverseReceipt('NYGTA-2019-001');
```

### 13. Sztornó nyugta generálása

```
$result = $agent->generateReverseReceipt($receipt);
```

#### 5.8.1 Elvárt, és opcionális bemenetek

| Nyugta fejléc adatok |               |        |          |           |
|----------------------|---------------|--------|----------|-----------|
| megnevezés           | mező          | típus  | kötelező | alapérték |
| nyugtaszám           | receiptNumber | string | igen     |           |
| pdf sablon           | pdfTemplate   | string |          |           |
| hívás azonosító      | callId        | string |          |           |

## 5.9 Nyugta kiküldése

Egy nyugta értesítő küldésének teljes példáját a következő fájl szemlélteti:  
*./examples/document/receipt/send\_receipt.php*

A nyugta értesítő küldéséhez a következő lépések szükségesek:

1. **Nyugta létrehozása** (nyugtaszám beállításával)

```
$receipt = new Receipt('NYGTA-2019-001');
```

2. **Vevő** inicializálása

```
$buyer = new Buyer();
```

3. Vevő e-mail címének beállítása (ide megy ki a levél)

```
$buyer->setEmail('vevo@example.com');
```

4. Vevői adatok hozzáadása a nyugtához

```
$receipt->setBuyer($buyer);
```

5. **Eladó** létrehozása

```
$seller = new Seller();
```

6. Eladó e-mail értesítő beállítása

```
$seller->setEmailReplyTo('elado@example.com');  
$seller->setEmailSubject('Email tárgya');  
$seller->setEmailContent('Ez az email szövege...');
```

7. Eladói adatok hozzáadása a nyugtához

```
$receipt->setSeller($seller);
```

8. **Nyugta elküldése**

```
$result = $agent->sendReceipt($receipt);
```



### 5.9.1 Elvárt, és opcionális bemenetek

| Nyugta fejléc adatok |               |        |          |           |
|----------------------|---------------|--------|----------|-----------|
| megnevezés           | mező          | típus  | kötelező | alapérték |
| nyugtaszám           | receiptNumber | string | igen     |           |

| E-mail küldés adatok |              |        |          |           |
|----------------------|--------------|--------|----------|-----------|
| megnevezés           | mező         | típus  | kötelező | alapérték |
| e-mail               | email        | string |          |           |
| e-mail válaszcím     | emailReplyTo | string |          |           |
| e-mail tárgy         | emailSubject | string |          |           |
| e-mail tartalom      | emailContent | string |          |           |

## 5.10 Nyugta adatok lekérdezése

Egy már létező nyugta adatai lekérdezhetők nyugtaszám alapján. A megvalósítást a következő példafájl szemlélteti: *./examples/document/receipt/get\_receipt\_data.php*

Letöltés **nyugtaszám** alapján:

```
$agent->getReceiptData('NYGTA-2019-001');
```

A nyugta adatokat tartalmazó objektum, amelyet használhatunk a további feldolgozáshoz:

```
$result->getDataObj();
```

### 5.10.1 Elvárt, és opcionális bemenetek

| Nyugta fejléc adatok |               |        |          |           |
|----------------------|---------------|--------|----------|-----------|
| megnevezés           | mező          | típus  | kötelező | alapérték |
| nyugtaszám           | receiptNumber | string | igen     |           |
| pdf sablon           | pdfTemplate   | string |          |           |

## 5.11 Nyugta adatok letöltése PDF dokumentumként

Egy már létező nyugta letölthető PDF dokumentumként. A megvalósítást a következő példafájl szemlélteti: `./examples/document/receipt/get_receipt_pdf.php`

Letöltés **nyugtaszám** alapján:

```
$agent->getReceiptPdf('NYGTA-2019-001');
```

A válasz eredménye egy PDF típusú nyugta, amelyet az alábbi módon tölthetünk le:

```
$result->downloadPdf();
```

### 5.11.1 Elvárt, és opcionális bemenetek

| Nyugta fejléc adatok |               |        |          |           |
|----------------------|---------------|--------|----------|-----------|
| megnevezés           | mező          | típus  | kötelező | alapérték |
| nyugtaszám           | receiptNumber | string | igen     |           |
| pdf sablon           | pdfTemplate   | string |          |           |

## 6 Számla Agent válasz feldolgozása

A kérés elküldése után a szerver válaszában megkapjuk a kért művelet végrehajtásának eredményét (sikeres vagy sikertelen), illetve a válasz sémája szerinti adatokat (lásd. a csomagban a 'SzamlaAgentRequest::XML\_SCHEMA\_CREATE\_' kezdetű konstansoknál).

A kérés sikerességét az alábbi módon ellenőrizhetjük:

```
$agent = SzamlaAgentAPI::create('agentApiKey');  
...  
...  
if ($result->isSuccess()) {  
    ...  
}
```

A válaszból lekérdezhetjük a bizonylat sorszámát:

```
echo $result->getDocumentNumber();
```

Az összes adatot a következőképpen kaphatjuk meg:

```
$data = $result->getData();
```

A számla kérésre adott válaszból lekérdezhető adatokat a következő osztály tartalmazza: InvoiceResponse. A nyugta esetében ez a következő: ReceiptResponse.

Ha az adatokat tartalmazó objektumot szeretnénk használni:

```
$data = $result->getDataObj();
```

Ha a válaszban megkapott PDF dokumentumra van szükségünk:

```
$pdfFile = $result->toPdf();
```

Ha a választ XML formátumban szeretnénk megkapni és feldolgozni:

```
$xmlData = $result->toXML();
```

Ha a választ JSON formátumban szeretnénk megkapni és feldolgozni:

```
$jsonData = $result->toJson();
```

A kérés legenerálásának folyamatában a megadott adatok, a Számla Agent elvárt sémájának megfelelően XML dokumentummá konvertálódnak és elmentésre kerülnek a csomag xml mappájába az alábbi elnevezési konvenciók szerint:

{request|response}-{xml-schema}-{text|xml}-{object-name}-{current-date}.xml

pl. response-xmlszamla-text-invoice-20190320123017.xml

## 6.1 Egyedi hibák kezelése

Előfordulhat, hogy egy számla kiállítása sikeres, azonban a számlaértesítőt valamilyen ok miatt nem sikerül kézbesítenie a számlázz.hu rendszerének.

A számla sikeres kiállítását továbbra is a **\$result->isSuccess()** metódussal ellenőrizhetjük. Ez esetben a számlaértesítő sikertelenségét a **\$result->hasInvoiceNotificationSendError()** metódussal kérdezhetjük le. Az értesítő kiküldésének sikertelenségét az API logolja, így a naplófájlban is megtalálható lesz.

Nézzük a konkrét példát a válasz kezelésére:

```
if ($result->isSuccess()) {  
    echo 'A számla sikeresen elkészült. Számlaszám: ' . $result->getDocumentNumber();  
}  
// ha sikertelen az számlaértesítő kézbesítése  
if ($result->hasInvoiceNotificationSendError()) {  
    var_dump($result->getDataObj());  
}
```