Опис АРІ Фіскального Сервера (Єдине вікно подання електронної звітності)

Зміст

[Загальні положення 2](#_Toc158570505)

[Поняття і визначення 3](#_Toc158570506)

[Порядок локальної нумерації документів 3](#_Toc158570507)

[Порядок фіскальної нумерації онлайн документів 3](#_Toc158570508)

[Загальний порядок взаємодії ПРРО з Фіскальним Сервером 4](#_Toc158570509)

[Відкриття зміни 4](#_Toc158570510)

[Реєстрація фіскального касового чека 4](#_Toc158570511)

[Реєстрація Z-звіту і закриття зміни 4](#_Toc158570512)

[Офлайн сесія 4](#_Toc158570513)

[Режими онлайн і офлайн 5](#_Toc158570514)

[Тривалість офлайн сесії 5](#_Toc158570515)

[Пакет офлайн документів 5](#_Toc158570516)

[Відкликання останнього онлайн чека 6](#_Toc158570517)

[Сторнування останнього чека 6](#_Toc158570518)

[Структура фіскального номера офлайн документа 6](#_Toc158570519)

[Формування Діапазону та облік фіскальних номерів 6](#_Toc158570520)

[Розрахунок контрольного числа 7](#_Toc158570521)

[Приклад розрахунку контрольного числа 7](#_Toc158570522)

[Розрахунок гешу офлайн документа 8](#_Toc158570523)

[Блокування роботи ПРРО 8](#_Toc158570524)

[Сценарії роботи 8](#_Toc158570525)

[1. Новий екземпляр ПРРО 8](#_Toc158570526)

[2. ПРРО працює з відкритою зміною (Сервер недоступний) 9](#_Toc158570527)

[3. ПРРО працює з закритою зміною (Сервер недоступний) 9](#_Toc158570528)

[4. Перевірка зв’язку з Сервером (ПРРО знаходиться в режимі онлайн) 9](#_Toc158570529)

[5. Перевірка зв’язку з Сервером (ПРРО знаходиться в режимі офлайн) 9](#_Toc158570530)

[6. Скасування реєстрації ПРРО 9](#_Toc158570531)

[Зауваження щодо реалізації 9](#_Toc158570532)

[Класи документів 10](#_Toc158570533)

[Чек. Типи документів 10](#_Toc158570534)

[Чек. Розширені типи документів 11](#_Toc158570535)

[Типи даних запитів документів 12](#_Toc158570536)

[Коди результату обробки запитів документів 12](#_Toc158570537)

[Коди помилок 13](#_Toc158570538)

[Коди HTTP 15](#_Toc158570539)

[Команди 15](#_Toc158570540)

[Загальні параметри 15](#_Toc158570541)

[Запит стану сервера 16](#_Toc158570542)

[Запит XSD схем 16](#_Toc158570543)

[Запит доступних об'єктів 17](#_Toc158570544)

[Запит переліку операторів (касирів) для суб’єкта господарювання 17](#_Toc158570545)

[Запит стану ПРРО 18](#_Toc158570546)

[Запит чека 19](#_Toc158570547)

[Запит чека розширений 20](#_Toc158570548)

[Запит Z-звіту 20](#_Toc158570549)

[Запит Z-звіту розширений 21](#_Toc158570550)

[Запит переліку змін за період 21](#_Toc158570551)

[Запит переліку документів зміни 22](#_Toc158570552)

[Запит підсумків останньої зміни 23](#_Toc158570553)

[Запит відомостей про документ за локальним номером 30](#_Toc158570554)

[Запит надсилання документа 30](#_Toc158570555)

[Запит надсилання пакета офлайн документів 30](#_Toc158570556)

[Видаткові чеки (повернення) 31](#_Toc158570557)

[Сторнування попереднього чека 31](#_Toc158570558)

[Тестові документи 31](#_Toc158570559)

[Порядок засвідчення повідомлень кваліфікованим електронним підписом 31](#_Toc158570560)

[Позначка часу 32](#_Toc158570561)

[Обмеження 32](#_Toc158570562)

[Адреси Фіскального Сервера 32](#_Toc158570563)

# Загальні положення

Фіскальний Сервер контролюючого органу, через який реалізується фіскальна функція програмних реєстраторів розрахункових операцій, забезпечує:

* Одержання документів (чеків, Z-звітів, повідомлень тощо) від програмних реєстраторів розрахункових операцій;
* Перевірку коректності структури і складу одержаних документів;
* Призначення документам фіскальних номерів;
* Надсилання відправнику документа відповіді, що містить результат обробки документа і призначений документу фіскальний номер;
* Зберігання документів для подальшого використання в інформаційно-аналітичних системах;
* Обробку запитів щодо надання документів (чеків покупцям, даних розрахункових операцій для формування ПРРО, Z-звітів тощо).
* Збереження сертифікатів електронних підписів та/або печаток, що використовуються ПРРО;
* Збереження даних про зареєстровані ПРРО та господарські одиниці, на яких ПРРО застосовується (реєстр ПРРО).

Реєстрація документів (чеків, Z-звітів, повідомлень тощо) здійснюється:

* в режимі оперативного («онлайн») надсилання документів на Фіскальний Сервер;
* в режимі відкладеного («офлайн») надсилання документів на Фіскальний Сервер.

Документи в системі засвідчуються кваліфікованим електронним підписом (КЕП) фізичної особи або електронною печаткою суб’єкта господарювання, на якого зареєстровано ПРРО. Час засвідчення документа фіксується накладанням на КЕП позначки часу (за виключенням документів, створених в режимі офлайн).

Джерелом позначки часу може виступати від будь-який кваліфікований надавач електронних довірчих послуг.

# Поняття і визначення

**ПРРО** – Програмний реєстратор розрахункових операцій. Система виписки електронних касових чеків, що використовується продавцем.

**Сервер** – Фіскальний Сервер реєстрації розрахункових операцій.

**Локальний номер документа** – Номер, призначений документу (чеку, Z-звіту) системою виписки електронних касових чеків, що використовується продавцем.

**Фіскальний номер документа** – Номер, призначений документу (чеку, Z-звіту) Фіскальним Сервером.

**Офлайн сесія** – Сукупність документів, створених в режимі офлайн, між припиненням і відновленням зв’язку ПРРО з Фіскальним Сервером.

**Локальний номер реєстратора** – Номер, призначений ПРРО організацією продавця і зареєстрований формою 1-ПРРО «Заява про реєстрацію програмного реєстратора розрахункових операцій».

**Фіскальний номер реєстратора** – Номер, призначений ПРРО Фіскальним Сервером в процесі реєстрації форми 1-ПРРО.

# Порядок локальної нумерації документів

Локальна нумерація документів ведеться окремо кожним ПРРО. Нумерація здійснюється в межах:

* юридичної/фізичної особи продавця;
* господарської одиниці;
* фіскального номера ПРРО.

Локальна нумерація документів ведеться наскрізь, незалежно від класу документа (чек, Z-звіт).

# Порядок фіскальної нумерації онлайн документів

Фіскальна нумерація документів ведеться Сервером.

Фіскальна нумерація документів ведеться наскрізь, незалежно від джерела і класу документа.

# Загальний порядок взаємодії ПРРО з Фіскальним Сервером

1. Відкриття зміни;
2. Реєстрація фіскальних касових чеків;
3. Реєстрація Z-звіту і закриття зміни.

## Відкриття зміни

1. ПРРО створює XML документ повідомлення із призначеним локальним номером і типом «Відкриття зміни», засвідчує його кваліфікованим електронним підписом продавця і надсилає на Сервер. Відкриття кожної наступної зміни відбувається за умови закриття попередньої зміни (формування Z-звіту).
2. Сервер здійснює необхідні перевірки.
3. У разі помилки, Сервер надсилає у відповідь XML квитанції.
4. У разі успішної обробки, Сервер реєструє документ шляхом присвоєння йому номера, який є номером зміни, зберігає документ в БД і надсилає у відповідь XML квитанції щодо успішної реєстрації документа.
5. Сервер встановлює для ПРРО стан «зміну відкрито».

## Реєстрація фіскального касового чека

1. ПРРО створює XML документ чека із призначеним локальним номером, засвідчує його КЕП продавця і надсилає на Сервер.
2. Сервер здійснює необхідні перевірки.
3. У разі помилки, Сервер надсилає у відповідь XML квитанції, засвідченої КЕП Сервера.
4. У разі успішної обробки, Сервер зберігає документ в БД, призначає документу фіскальний номер і надсилає у відповідь XML квитанції щодо успішної реєстрації документа, що містить фіскальний номер.

## Реєстрація Z-звіту і закриття зміни

1. Сервер формує проект Z-звіту за запитом ПРРО на дату та час отримання запиту та надсилає його до ПРРО. Отриманий проект Z-звіту має бути перевірений, може бути змінений. Кількість запитів для формування проекту Z-звіту необмежена.
2. ПРРО (самостійно або на основі проекту, отриманого від Фіскального Сервера) створює XML документ Z-звіту із призначеним локальним номером, засвідчує його КЕП продавця і надсилає на Сервер.
3. Сервер здійснює необхідні перевірки.
4. У разі помилки, Сервер надсилає у відповідь XML квитанції, засвідченої КЕП Сервера.
5. У разі успішної обробки, Сервер зберігає документ в БД, і надсилає у відповідь XML квитанції щодо успішної реєстрації документа.
6. ПРРО надсилає на Сервер повідомлення (технічний документ) із типом «Закриття зміни», на підставі якого відбувається закриття зміни.
7. Сервер встановлює для ПРРО стан «зміну закрито».

# Офлайн сесія

Для обробки чеків в режимі офлайн, використовується поняття **«Офлайн сесія»**.

Офлайн сесія обмежена документами «Початок офлайн сесії» і «Завершення офлайн сесії».

Тривалість однієї офлайн сесії не може перевищувати встановлене законодавством значення (36 годин за сесію, 168 годин за місяць). (В програмних РРО рекомендується реалізувати можливість інформування користувача про сплив одного з термінів, дозволених для роботи ПРРО в режимі офлайн).

Кількість документів у межах одної офлайн сесії не обмежена.

Офлайн сесія має унікальний ідентифікатор (число). Ідентифікатор офлайн сесії надається Сервером у відповідь на документ «Відкриття зміни».

Коли Сервер знову стає доступний, ПРРО надсилає пакет, що містить документи: «Початок офлайн сесії», чеки, Z-звіти, «Завершення офлайн сесії». У відповідь Сервер надсилає номер наступної офлайн сесії.

Офлайн сесія застосовується тільки з ПРРО, для якого виданий ідентифікатор офлайн сесії та фіскальні номери із Діапазону. З іншим ПРРО офлайн сесія не може бути застосована.

В межах офлайн сесії може бути відкрито і закрито необмежену кількість змін (якщо зміни не відкриті одночасно і кожному відкриттю нової зміни передує закриття попередньої). Для закриття зміни формується Z-звіт і повідомлення (технічний документ) із типом «Закриття зміни».

## Режими онлайн і офлайн

ПРРО знаходиться в режимі онлайн, якщо на ньому не створений документ «Початок офлайн сесії».

ПРРО знаходиться в режимі офлайн, якщо на ньому створений документ «Початок офлайн сесії», що не надісланий на Сервер, і для якого відсутній документ «Завершення офлайн сесії».

## Тривалість офлайн сесії

Розрахунок тривалості офлайн сесії здійснюється як різниця між часом реєстрації:

* Останнього фінансово значущого документа (чека, Z-звіту) в офлайн сесії, що передує документу типу «Завершення офлайн сесії»
* Документа типу «Початок офлайн сесії»

*Примітка*. Фінансово значущими є документи, які містять інформацію щодо руху або підсумках грошових коштів і товарів/послуг.

## Пакет офлайн документів

Коли Сервер стає доступний, ПРРО надсилає один (або декілька) пакетів, що містить документи: «Початок офлайн сесії», чеки, Z-звіти, «Завершення офлайн сесії».

Структура пакету:

<Розмір документа 1 (4 байта))><Документ 1><Розмір документа 2 (4 байта)><Документ 2>…<Розмір документа N (4 байта)><Документ N>

Кожен документ в пакеті повинен бути засвідчений КЕП.

Весь пакет також засвідчується КЕП відправника.

Максимальна кількість документів в пакеті обмежена (100).

Якщо кількість створених офлайн документів перевищує максимальну, надсилаються декілька пакетів. Пакети надсилаються по черзі, тобто кожний наступний пакет надсилається після одержання від Сервера позитивної відповіді щодо одержання попереднього пакета.

Сервер надсилає номер наступної офлайн сесії та нове «секретне число» у відповідь на пакет, який містить документ «Завершення офлайн сесії» та поповнює діапазон використаних номерів.

Формат JSON відповіді:

{

    “OfflineSessionId”**:** <Ідентифікатор офлайн сесії>,

    “OfflineSeed”**:** <"Секретне число" для обчислення фіскального номера офлайн документа>

}

Якщо пакет не містить документ «Завершення офлайн сесії», Сервер надсилає у відповідь “OK”.

### Відкликання останнього онлайн чека

Якщо перехід в режим офлайн було здійснено внаслідок розірвання зв’язку з Сервером під час реєстрації чека, але Сервер отримав документ і здійснив реєстрацію чека з призначенням фіскального номера, і чек було переформовано в режимі офлайн із присвоєнням номера із діапазону, виникає дублювання онлайн і офлайн чеків.

Для усунення такого дублювання:

* ПРРО зберігає чек, який надіслано до Сервера, але фіскальний номер не отриманий внаслідок розірвання зв’язку;
* Після відновлення зв’язку, ПРРО одержує з Сервера останній зареєстрований чек;
* Якщо останній зареєстрований чек дорівнює збереженому, в документі «Початок офлайн сесії» додається елемент <REVOKELASTONLINEDOC>true</REVOKELASTONLINEDOC>.

*Примітка*. Зазначений порядок стосується лише фінансово значущих документів.

### Сторнування останнього чека

Якщо покупець відмовляється від проведеної операції з торгівлі валютними цінностями і при цьому здійснено реєстрацію чека Фіскальним Сервером, або якщо чек з помилковою сумою зареєстрований Фіскальним Сервером:

* ПРРО надсилає на Фіскальний Сервер фіскальний чек типу «Чек сторнування попереднього чека» із присвоєним Сервером фіскальним номером, що підлягає сторнуванню, чек має відповідну ознаку «сторно» (елемент <ORDERSTORNUM > Фіскальний номер чека, для якого здійснюється сторнування </ORDERSTORNUM>).
* Фіскальний Сервер перевіряє фіскальний номер проведеної розрахункової операції, реєструє операцію «сторно» із нульовим значенням сторнованої суми, присвоює фіскальний номер чеку з ознакою «сторно» та направляє користувачу.

## Структура фіскального номера офлайн документа

Фіскальний номер документа, що виданий офлайн, має таку структуру:

<Ідентифікатор офлайн сесії>.<Локальний номер документа в офлайн сесії>.<Контрольне число>

Локальний номер документа в межах кожної офлайн сесії починається з 1 і збільшується послідовно.

*Зауваження:* Принцип безперервної локальної нумерації документів залишається незмінним. Тобто, поле «Локальний номер документа» продовжує безперервне збільшення значень, незалежно від режиму (онлайн чи офлайн). Значення поля «Локальний номер документа» не дорівнює значенню «Локальний номер документа в офлайн сесії».

Приклад фіскального номера офлайн документа: “82563.25.6127”

(при цьому «Локальний номер документа» може дорівнювати “9537”)

## Формування Діапазону та облік фіскальних номерів

Новий Діапазон визначається в межах номера офлайн сесії, який видається ПРРО Фіскальним сервером під час першого відкриття зміни на ПРРО.

Діапазон містить послідовний ряд натуральних чисел, починаючи з 1.

Номери із Діапазону використовуються послідовно (1, 2, 3, …), як зазначено в розділі «Структура фіскального номера офлайн документа».

## Розрахунок контрольного числа

Одночасно з передачею номера офлайн сесії, Сервер надає ПРРО «секретне число».

Для розрахунку контрольного числа, ПРРО створює текстовий рядок, що містить об’єднання значень, розділених символом «,»:

* «Секретне число»
* Дата документа (ДДММРРРР)
* Час документа (ГГХХСС)
* Локальний номер документа
* Фіскальний номер реєстратора
* Локальний номер реєстратора
* Загальна сума оплати по документу класу «Чек» (формат «0.00») (якщо елемент присутній)
* Геш попереднього документа (крім першого документа, створеного в межах офлайн сесії, тобто документа «Початок офлайн сесії») (якщо елемент присутній)

*Примітка*. Для документа «Початок офлайн сесії» геш попереднього документа не використовується в розрахунку тому, що попередній документ відсутній. Для наступного за ним документа геш попереднього документа не використовується в розрахунку тому, що попередній документ («Початок офлайн сесії») може бути змінений (може бути додано елемент «REVOKELASTONLINEDOC»).

Від текстового рядку розраховується геш за алгоритмом CRC32 (див. <https://github.com/damieng/DamienGKit/blob/master/CSharp/DamienG.Library/Security/Cryptography/Crc32.cs>), як десяткове беззнаковое число. Із розрахованого гешу беруться 4 молодші розряди. Ведучі нулі відкидаються (наприклад, “0123” -> “123”).

Контрольне число не може дорівнювати 0. Якщо у результаті розрахунку контрольного числа одержано 0, призначається значення 1.

### Приклад розрахунку контрольного числа

Документ:

*<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="check01.xsd">*

*<CHECKHEAD>*

*<DOCTYPE>103</DOCTYPE>*

*<UID>795E22D9-D6C8-4B0E-8FAC-F4C42CBA9ABE</UID>*

*<TIN>44082020</TIN>*

*<IPN>440820207777</IPN>*

*<ORGNM>ТОВ Тестова організація</ORGNM>*

*<POINTNM>Кафе "Сонечко"</POINTNM>*

*<POINTADDR>УКРАЇНА, м. Київ, вул. Сонячна, 27</POINTADDR>*

*<ORDERDATE>20082020</ORDERDATE>*

*<ORDERTIME>142338</ORDERTIME>*

*<ORDERNUM>10</ORDERNUM>*

*<CASHDESKNUM>10</CASHDESKNUM>*

*<CASHREGISTERNUM>4000002411</CASHREGISTERNUM>*

*<CASHIER>Восьмий Касир Тест</CASHIER>*

*<VER>1</VER>*

*<ORDERTAXNUM>5008.3.4758</ORDERTAXNUM>*

*<OFFLINE>true</OFFLINE>*

*<PREVDOCHASH>cdd68bb111f8993f3603f0179341571b35b73a07d5acee9b28fbfb714698e1b3</PREVDOCHASH>*

*</CHECKHEAD>*

*</CHECK>*

Текстовий рядок для розрахунку CRC32:

*“179625192271939,20082020,142338,10,4000002411,10,cdd68bb111f8993f3603f0179341571b35b73a07d5acee9b28fbfb714698e1b3”*

Розраховане значення CRC32:

* в шітнадцятковому форматі: *0xbddbbda6*
* як беззнакове ціле: *3185294758*

Контрольне число: *4758*

## Розрахунок гешу офлайн документа

Геш офлайн документа розраховується за стандартним алгоритмом геш-функції SHA-256 (згідно з FIPS PUB 180-4 “Secure Hash Standard (SHS)”, <http://dx.doi.org/10.6028/NIST.FIPS.180-4>), від попереднього документа поточної офлайн сесії, і зберігається в елементі “PREVDOCHASH” як текстовий рядок в шітнадцятковому форматі. Наприклад, <PREVDOCHASH>68f2e680d84e900a4279fb3b614d86138f1a9f6c67df0510fe837c75ec7cf0ec</PREVDOCHASH>.

Вихідними даними для розрахунку геша є блок документа, підготовлений для надсилання на Фіскальний Сервер (тобто XML, засвідчений КЕП і, за наявності, позначкою часу).

У разі неможливості здійснення підписання документу, дозволяється використовувати для розрахунку геша безпосередньо блок XML.

Послідовність гешів розраховується, починаючи з першого розрахункового документа (чека на оплату або Z-звіту). Документи типу «Початок офлайн сесії» і «Завершення офлайн сесії» не приймають участь в розрахунку послідовності гешів.

Для першого розрахункового документа офлайн сесії елемент “PREVDOCHASH” не створюється.

## Блокування роботи ПРРО

Фіскальний Сервер блокує роботу ПРРО у разі:

* Перевищення допустимого терміну роботи в офлайн режимі «36 годин» або «168 годин» протягом календарного місяця та надсилає повідомлення користувачеві. Розблоковує роботу ПРРО після настання нового часового періоду з якого починається відлік дозволеного часу роботи в режимі офлайн.
* Наявності повідомлення за формою 2-ПРРО з позначкою «несправність»

# Сценарії роботи

## Новий екземпляр ПРРО

* 1. Користувач заходить в ПРРО з аутентифікацією по КЕП. ПРРО одержує з Сервера довідники (перелік господарських одиниць (ГО), перелік фіскальних номерів ПРРО, тощо).
  2. Користувач вибирає доступну ГО і фіскальний номер ПРРО, і дає команду на відкриття зміни. ПРРО надсилає на Сервер документ «Відкриття зміни», і одержує у відповідь фіскальний номер документа, ідентифікатор офлайн сесії та «секретне число».
  3. Користувач видає чеки.

## ПРРО працює з відкритою зміною (Сервер недоступний)

* 1. Користувач заходить в ПРРО з аутентифікацією по КЕП.
  2. Користувач заходить в ГО і фіскальний номер ПРРО, для якого відкрито офлайн сесію.
  3. ПРРО перевіряє, чи зміну відкрито тим самим користувачем. Якщо ні – робота Користувача неможлива до закриття зміни.
  4. Користувач видає чеки.

## ПРРО працює з закритою зміною (Сервер недоступний)

* 1. Користувач заходить в ПРРО з аутентифікацією по КЕП.
  2. Користувач заходить в доступну ГО і фіскальний номер ПРРО, для якого відкрито офлайн сесію, і дає команду на відкриття зміни.
  3. Користувач видає чеки.
  4. Користувач закриває зміну.

## Перевірка зв’язку з Сервером (ПРРО знаходиться в режимі онлайн)

* 1. Якщо при передачі документа Сервер недоступний, ПРРО запитує у користувача підтвердження переходу в режим офлайн.
  2. Якщо користувач згодний, ПРРО створює документ «Початок офлайн сесії»

## Перевірка зв’язку з Сервером (ПРРО знаходиться в режимі офлайн)

* 1. ПРРО періодично перевіряє зв’язок з Сервером (наприклад, за допомогою команди «Запит стану ПРРО», або команди «Запит стану сервера»).
  2. Якщо зв’язок присутній, ПРРО автоматично переходить в режим онлайн, про що інформує користувача.
  3. ПРРО надсилає на Сервер пакет, що містить документи, створені в режимі офлайн.
  4. У разі відмови Сервера на підставі некоректного сертифікату, пакет засвідчується підписом старшого користувача або керівника.
  5. Якщо пакет успішно прийнятий, ПРРО отримує підтвердження (квитанцію) від фіскального Сервера.

## Скасування реєстрації ПРРО

* 1. Користувач надсилає заяву з позначкою «Скасування реєстрації»
  2. Користувач надсилає повідомлення з позначкою «несправність» або «крадіжка пристрою»
  3. Автоматичне скасування (припинено державну реєстрацію, закрито господарську одиницю тощо).

# Зауваження щодо реалізації

1. Кодування текстових полів документів XML: Windows-1251.
2. Тип HTTP-запиту – POST.
3. Заголовок “Content-Type”: “application/octet-stream” для повідомлень, засвідчених КЕП. Або “Content-Type”: “application/json” чи “Content-Type”: “application/json; charset=UTF-8” для команд, не засвідчених КЕП.
4. Використання компресії Gnu Zip (GZip) або Zip (deflate). Вхідні та вихідні документи оброблюються алгоритмом GZip або Zip для зменшення розміру повідомлення.
5. У разі використання компресії, заголовок “Content-Encoding” вхідного повідомлення повинен мати значення “gzip” або “deflate”.
6. Повідомлення містить XML документ, засвідчений КЕП відправника.
7. Вхідне повідомлення може містити серйозні помилки, що не дозволяють його подальшу обробку. Наприклад, повідомлення не являє собою засвідчений блок або містить більше ніж одну КЕП. У випадку грубого порушення вимог до формату вхідного повідомлення, у відповідь надсилається код статусу обробки HTTP 4xx “Client Error”. У тілі відповіді надається інформація щодо причини помилки. Наприклад, “Очікується значення 'Content-Type' рівне 'application/octet-stream' або 'application/json; charset=UTF-8'”, “Повідомлення повинно бути засвідчене кваліфікованим електронним підписом” і таке інше.
8. Сервер оброблює Url:
   * “<адреса>/doc“ – одержання документів (чеків, Z-звітів тощо)
   * “<адреса>/pck“ – одержання пакетів документів (офлайн документи тощо)
   * “<адреса>/cmd“ – одержання команд
9. Команди надсилаються у форматі JSON. Див. розділ «Команди».

## Класи документів

/// <summary>

///     Клас документа

/// </summary>

public enum DocumentClass {

    /// <summary>

    ///     Чек

    /// </summary>

**Check** = 0,

    /// <summary>

    ///     Z-звіт

    /// </summary>

**ZRep** = 1

}

## Чек. Типи документів

/// <summary>

///     Чек. Тип документа

/// </summary>

public enum CheckDocumentType {

    /// <summary>

    ///     Чек реалізації товарів/послуг

    /// </summary>

**SaleGoods** = 0,

    /// <summary>

    ///     Чек переказу коштів

    /// </summary>

**TransferFunds** = 1,

    /// <summary>

    ///     Чек операції обміну валюти

    /// </summary>

**CurrencyExchange** = 2,

    /// <summary>

    ///     Чек видачі готівки

    /// </summary>

**CashWithdrawal** = 3,

    /// <summary>

    ///     Відкриття зміни

    /// </summary>

**OpenShift** = 100,

    /// <summary>

    ///     Закриття зміни

    /// </summary>

**CloseShift** = 101,

    /// <summary>

    ///     Початок офлайн сесії

    /// </summary>

**OfflineBegin** = 102,

    /// <summary>

    ///     Завершення офлайн сесії

    /// </summary>

**OfflineEnd** = 103

}

## Чек. Розширені типи документів

/// <summary>

///     Чек. Розширений тип документа

/// </summary>

public enum CheckDocumentSubType {

    /// <summary>

    ///     Касовий чек (реалізація)

    /// </summary>

**CheckGoods** = 0,

    /// <summary>

    ///     Видатковий чек (повернення)

    /// </summary>

**CheckReturn** = 1,

    /// <summary>

    ///     Чек операції «службове внесення»/«отримання авансу»

    /// </summary>

**ServiceDeposit** = 2,

    /// <summary>

    ///     Чек операції «отримання підкріплення»

    /// </summary>

**AdditionalDeposit** = 3,

    /// <summary>

    ///     Чек операції «службова видача»/«інкасація»

    /// </summary>

**ServiceIssue** = 4,

    /// <summary>

    ///     Чек сторнування попереднього чека

    /// </summary>

**CheckStorno** = 5

}

## Типи даних запитів документів

/// <summary>

/// Тип даних запиту документа

/// </summary>

public enum DocumentRequestType {

/// <summary>

/// Перевірка наявності документа

/// </summary>

**Availability** = 0,

/// <summary>

/// Оригінальний XML

/// </summary>

**OriginalXml** = 1,

/// <summary>

/// XML засвідчений КЕП Фіскального Сервера

/// </summary>

**SignedByServerXml** = 2,

/// <summary>

/// Документ в текстовому форматі для відображення (UTF-8)

/// </summary>

**Visualization** = 3,

/// <summary>

/// XML засвідчений КЕП відправника

/// </summary>

**SignedBySenderXml** = 4,

/// <summary>

/// XML засвідчений КЕП відправника і КЕП Фіскального Сервера

/// </summary>

**SignedBySenderAndServerXml** = 5

}

## Коди результату обробки запитів документів

/// <summary>

/// Код результату обробки запиту документа

/// </summary>

public enum DocumentRequestResultCode {

/// <summary>

/// OK

/// </summary>

**Ok** = 0,

/// <summary>

/// Документ з фіскальним номером, що відповідає режиму онлайн, не зареєстрований на ПРРО

/// </summary>

**OnlineDocumentAbsent** = 1,

/// <summary>

/// Фіскальний номер зарезервований для використання в режимі офлайн на ПРРО, але наразі ще не переданий

/// з ПРРО до контролюючого органу

/// </summary>

**OfflineNumberReserved** = 2,

/// <summary>

/// Фіскальний номер не зарезервований для використання в режимі офлайн на ПРРО

/// </summary>

**OfflineNumberNotReserved** = 3,

/// <summary>

/// ПРРО не зареєстрований

/// </summary>

**TransactionsRegistrarNotRegistered** = 4,

/// <summary>

/// Документ не зареєстрований на ПРРО

/// </summary>

**DocumentAbsent** = 5

}

## Коди помилок

/// <summary>

///     Код помилки

/// </summary>

public enum ErrorCode {

    /// <summary>

    ///     OK

    /// </summary>

**Ok** = 0,

    /// <summary>

    ///     ПРРО не зареєстрований

    /// </summary>

**TransactionsRegistrarAbsent** = 1,

    /// <summary>

    ///     Відсутній доступ до ПРРО для користувача

    /// </summary>

**OperatorAccessToTransactionsRegistrarNotGranted** = 2,

    /// <summary>

    ///     В документі зазначено реєстраційний код платника, що не дорівнює реєстраційному коду господарської одиниці

    /// </summary>

**InvalidTin** = 3,

    /// <summary>

    ///     Зміну для ПРРО наразі відкрито

    /// </summary>

**ShiftAlreadyOpened** = 4,

    /// <summary>

    ///     Зміну для ПРРО наразі не відкрито

    /// </summary>

**ShiftNotOpened** = 5,

    /// <summary>

    ///     Останній документ, зареєстрований перед закриттям зміни, повинен бути Z-звітом

    /// </summary>

**LastDocumentMustBeZRep** = 6,

    /// <summary>

    ///     Некоректний локальний номер чека

    /// </summary>

**CheckLocalNumberInvalid** = 7,

    /// <summary>

    ///     Z-звіт наразі зареєстрований для поточної зміни

    /// </summary>

**ZRepAlreadyRegistered** = 8,

    /// <summary>

    ///     Помилка валідації документа

    /// </summary>

**DocumentValidationError** = 9,

    /// <summary>

    ///     Помилка валідації пакету офлайн документів

    /// </summary>

**PackageValidationError** = 10,

    /// <summary>

    ///     Некоректний параметр запиту

    /// </summary>

**InvalidQueryParameter** = 11,

    /// <summary>

    ///     Помилка криптографічних функцій

    /// </summary>

**CryptographyError** = 12,

    /// <summary>

    ///     Запитані дані відсутні

    /// </summary>

**NoContent** = 13

}

## Коди HTTP

200 “OK”

400 “Bad Request” – Некоректний зміст запиту

401 “Unauthorized” – Власнику сертифіката не дозволений доступ до зазначеного ПРРО;

403 “Forbidden” – Некоректний URL звернення до сервера;

404 “Not Found” – ПРРО відсутній в реєстрі;

405 “Method Not Allowed” – Метод звернення до сервера не дозволяється для зазначеної операції;

415 “Unsupported Media Type” – Недопустиме значення 'Content-Type';

416 “Requested Range Not Satisfiable” – Недопустимий розмір повідомлення;

500 “Internal Server Error” – Внутрішня помилка сервера;

503 “Service Unavailable” – Сервер недоступний.

Відповідь містить опис помилки у вигляді “Код помилки: <Числовий код помилки> <Символьний код помилки>\r\n<Опис помилки>”.

## Команди

### Загальні параметри

Загальні параметри запиту:

{

    “Command”**:** “<Команда>”,

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

“Timestamp”**:** <Час надсилання команди (не обов’язковий)>,

“Timeout”**:** <Таймаут дії команди (мсек) (не обов’язковий)>

}

При заданих параметрах “Timestamp” та “Timeout”, команда скасовується, якщо поточний час сервера перевищує значення “Timestamp + Timeout”.

Загальні параметри відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

“Timestamp”**:** <Час надсилання відповіді>,

“ErrorCode”**:** <Код помилки (у разі виникнення, в режимі “resultAsJson=true”)>,

“ErrorMessage”**:** <Повідомлення про помилку (у разі виникнення, в режимі “resultAsJson=true”)>

}

Дата і час представлені текстом у форматі ISO 8601 (наприклад, "2018-10-17T01:23:00+03:00" ).

#### Отримання відповіді у форматі JSON

У разі виникнення помилки в процесі обробки команди, Сервер надсилає код помилки і повідомлення про помилку як текст.

Якщо потрібно отримувати код помилки і повідомлення про помилку у форматі JSON, до Url одержання команд необхідно додати параметр “resultAsJson=true”:

“<адреса>/cmd?resultAsJson=true“

В режимі “resultAsJson=true”, у разі виникнення помилки Сервер повертає код HTTP 200 “OK” та JSON з полями “ErrorCode” та “ErrorMessage”.

### Запит стану сервера

Запит повинен містити JSON.

Формат JSON запиту:

{

    “Command”**:** “ServerState”,

    “UID”**:** “<Унікальний ідентифікатор запиту>”

}

Формат JSON відповіді:

{

“Timestamp”**:** “<Дата і час відповіді сервера>”,

    “UID”**:** “<Унікальний ідентифікатор запиту>”

}

Дата і час представлені текстом у форматі ISO 8601 (наприклад, "2018-10-17T01:23:00+03:00" ).

При заданих параметрах “Timestamp” та “Timeout”, команда скасовується, якщо поточний час сервера перевищує значення “Timestamp + Timeout”.

### Запит XSD схем

Запит повинен містити JSON.

Формат JSON запиту:

{

    “Command”**:** “Schemas”,

    “UID”**:** “<Унікальний ідентифікатор запиту>”

}

Формат JSON відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

    “XsdSchemaCheck”**:** “<Зміст XSD схеми валідації чека в кодуванні Base64>”,

    “XsdSchemaZRep”**:** “<Зміст XSD схеми валідації Z-звіта в кодуванні Base64>”

}

### Запит доступних об'єктів

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “**Objects**”,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

У відповідь повертається перелік доступних користувачу господарських одиниць і ПРРО.

Формат JSON відповіді:

public class ResponseObjects {

    public string UID { get; set; } // Унікальний ідентифікатор запиту

    public TaxObjectItem[] TaxObjects { get; set; } // Перелік ГО

}

public class TaxObjectItem {

    public long Entity { get; set; } // Ідентифікатор запису ГО

    public string TaxObjGuid { get; set; } // Ідентифікатор запису ОО

    public long TaxObjId { get; set; } // Код запису ОО

    public bool SingleTax { get; set; } // Ознака ФОП – платника єдиного податку

    public string Name { get; set; } // Найменування ГО

    public string Address { get; set; } // Адреса ГО

    public string Tin { get; set; } // Код ЄДРПОУ/ДРФО платника податків

    public string Ipn { get; set; } // Податковий номер платника ПДВ

    public string OrgName { get; set; } // Найменування суб’єкта господарювання

    public bool ChiefCashier { get; set; } // Старший касир

    public TransactionsRegistrarItem[] TransactionsRegistrars { get; set; } // Перелік ПРРО, зареєстрованих для ГО

}

public class TransactionsRegistrarItem {

    public long NumFiscal { get; set; } // Фіскальний номер ПРРО

    public long NumLocal { get; set; } // Локальний номер ПРРО

    public string Name { get; set; } // Найменування ПРРО

    public bool Closed { get; set; } // Ознака скасованої реєстрації ПРРО, на якому наразі не закрито зміну

}

У разі відсутності даних, повертається код HTTP 204 “No Content”.

### Запит переліку операторів (касирів) для суб’єкта господарювання

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “**Operators**”,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

У відповідь повертається перелік операторів, що зареєстровані для суб'єкта господарювання, реєстраційний номер якого (ЄДРПОУ, ДРФО, Картка платника податків) міститься у сертифікаті КЕП, яким засвідчений запит.

Формат JSON відповіді:

public class ResponseOperators {

    public string UID { get; set; } // Унікальний ідентифікатор запиту

    public string Tin { get; set; } // Реєстраційний номер суб'єкта господарювання (ЄДРПОУ,ДРФО,Картка платника податків)

    public OperatorItem[] Operators { get; set; } // Перелік операторів

}

public class OperatorItem {

    public string SubjectKeyId { get; set; } // Ідентифікатор ключа суб’єкта

    public string RegNum { get; set; } // Реєстраційний номер особи оператора (ЄДРПОУ,ДРФО,Картка платника податків)

    public bool ChiefCashier { get; set; } // Старший касир

}

### Запит стану ПРРО

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “TransactionsRegistrarState”**,**

“NumFiscal”**:** “<Фіскальний номер ПРРО>”**,**

“OfflineSessionId”**:** <Ідентифікатор офлайн сесії, для якої будуть надіслані пакети документів (не обов’язковий)>**,**

“OfflineSeed”**:** <"Секретне число" для обчислення фіскального номера офлайн документа офлайн сесії, для якої будуть надіслані пакети документів (не обов’язковий)>**,**

“IncludeTaxObject”**:** <Ознака запиту відомостей об’єкту оподаткування (false/true) (не обов’язковий)>,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

Значення параметрів “OfflineSessionId” і “OfflineSeed” необхідно передавати для коректної обробки повернення до режиму онлайн із режиму офлайн.

Формат JSON відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

“ShiftState”**:** <0-зміну не відкрито, 1-зміну відкрито>,

“ShiftId”**:** <Ідентифікатор зміни>,

“OpenShiftFiscalNum”**:** “<Фіскальний номер документа “Відкриття зміни”>”,

    “ZRepPresent”**:** <Ознака присутності Z-звіту (false/true)>,

“Testing”**:** <Ознака зміни, що містить тестові документи (false/true)>,

“Name”**:** “<П.І.Б. оператора, що відкрив зміну>”,

“SubjectKeyId”**:** “<Ідентифікатор ключа суб’єкта сертифікату оператора, що відкрив зміну (null – зміну не відкрито)>”,

“FirstLocalNum”**:** <Перший внутрішній номер документа у поточній відкритій зміні (0 – зміну не відкрито)>,

“NextLocalNum”**:** <Наступний внутрішній номер документа>,

“LastFiscalNum”**:** <Останній фіскальний номер документа у поточній відкритій зміні (null – зміну не відкрито)>,

“OfflineSupported”**:** <Для ПРРО може використовуватись режим офлайн (false/true)>,

“ChiefCashier”**:** <Користувач є старшим касиром (false/true)>,

“OfflineSessionId”**:** <Ідентифікатор офлайн сесії (null – режим офлайн заборонений для ПРРО)>,

“OfflineSeed”**:** <"Секретне число" для обчислення фіскального номера офлайн документа (null – режим офлайн заборонений для ПРРО)>,

“OfflineNextLocalNum”**:** <Наступний локальний номер документа в офлайн сесії (null – режим офлайн заборонений для ПРРО)>,

“OfflineSessionDuration”**:** <Тривалість офлайн сесії (хвилин) (null – режим офлайн заборонений для ПРРО)>,

“OfflineSessionsMonthlyDuration”**:** <Сумарна тривалість офлайн сесій за поточний місяць (хвилин) (null – режим офлайн заборонений для ПРРО)>,

“OfflineSessionRolledBack”**:** <Ознака здійснення відміни закриття офлайн сесії (false/true) (null – режим офлайн заборонений для ПРРО)>,

“OfflineSessionRollbackCmdUID”**:** <Унікальний ідентифікатор запиту, за яким було здійснено відміну закриття офлайн сесії (null – режим офлайн заборонений для ПРРО)>,

“Closed”**:** <Ознака скасованої реєстрації ПРРО, на якому наразі не закрито зміну (false/true)>,

“TaxObject”**:** <Відомості об’єкту оподаткування (TaxObjectItem)>

}

У разі відсутності даних, повертається код HTTP 204 “No Content”.

Поля “Offline…” заповнюються, якщо запит засвідчено КЕП особи, яка відкрила зміну, або КЕП старшого користувача або керівника.

### Запит чека

Запит повинен містити JSON.

Формат JSON запиту:

{

    “Command”**:** “**Check**”**,**

“RegistrarNumFiscal”**:** “<Фіскальний номер ПРРО>”**,**

“NumFiscal”**:** “<Фіскальний номер чека>”**,**

“NumLocal”**:** <Локальний номер чека>**,**

“Original”**:** <Ознака запиту оригінального документа, надісланого продавцем (false/true)>,

“SetTimestamp”**:** <Ознака встановлення позначки часу на КЕП Фіскального Сервера (false/true)>

}

У запиті зазначається або фіскальний, або локальний номер документа.

У відповідь надається XML документ чека, засвідчений КЕП сервера.

У разі відсутності даних, повертається код HTTP 204 “No Content”.

### Запит чека розширений

Запит повинен містити JSON.

Формат JSON запиту:

{

    “Command”**:** “**CheckExt**”**,**

“RegistrarNumFiscal”**:** “<Фіскальний номер ПРРО>”**,**

“NumFiscal”**:** “<Фіскальний номер чека>”**,**

“NumLocal”**:** <Локальний номер чека>**,**

“Type”**:** <Тип даних запиту документа>,

“SetTimestamp”: <Ознака встановлення позначки часу на КЕП Фіскального Сервера (false/true)>,

“AcquireCabinetUrl”**:** <Ознака запиту посилання на сторінку візуалізації чека в Електронному кабінеті платника податків (false/true)>,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

У запиті зазначається або фіскальний, або локальний номер документа.

Формат JSON відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

    “Data”**:** “<Результат запиту в кодуванні Base64>”**,**

“ResultCode”**:** “<Код результату обробки запиту документа>”**,**

“ResultText”**:** “<Опис результату обробки запиту документа>”,

“CabinetUrl”**:** “<Посилання на сторінку візуалізації чека в Електронному кабінеті платника податків>”

}

*Примітка*. Елемент “CabinetUrl” заповнюється лише для документів, що присутні на Фіскальному Сервері.

### Запит Z-звіту

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “**ZRep**”**,**

“RegistrarNumFiscal”**:** “<Фіскальний номер ПРРО>”**,**

“NumFiscal”**:** “<Фіскальний номер Z-звіту>”**,**

“NumLocal”**:** <Локальний номер Z-звіту>**,**

“Original”**:** <Ознака запиту оригінального документа, надісланого продавцем (false/true)>,

“SetTimestamp”: <Ознака встановлення позначки часу на КЕП Фіскального Сервера (false/true)>

}

У запиті зазначається або фіскальний, або локальний номер документа.

У відповідь надається XML документ Z-звіту, засвідчений КЕП сервера.

У разі відсутності даних, повертається код HTTP 204 “No Content”.

### Запит Z-звіту розширений

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “**ZRepExt**”**,**

“RegistrarNumFiscal”**:** “<Фіскальний номер ПРРО>”**,**

“NumFiscal”**:** “<Фіскальний номер Z-звіту>”**,**

“NumLocal”**:** <Локальний номер Z-звіту>**,**

“Type”**:** <Тип даних запиту документа>,

“SetTimestamp”: <Ознака встановлення позначки часу на КЕП Фіскального Сервера (false/true)>,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

У запиті зазначається або фіскальний, або локальний номер документа.

Формат JSON відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

    “Data”**:** “<Результат запиту в кодуванні Base64>”**,**

“ResultCode”**:** “<Код результату обробки запиту документа>”**,**

“ResultText”**:** “<Опис результату обробки запиту документа>”

}

### Запит переліку змін за період

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “**Shifts**”**,**

“NumFiscal”**:** “<Фіскальний номер ПРРО>”,

“From”**:** “<Дата і час початку періоду>”,

“To”**:** “<Дата і час завершення періоду>”,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

Дата і час представлені текстом у форматі ISO 8601 (наприклад, "2018-10-17T01:23:00+03:00" ) або JavaScript (наприклад, "/Date(1539723599000)/").

Формат JSON відповіді:

{

“UID”**:** “<Унікальний ідентифікатор запиту>”,

“Shifts”:

[{

    “ShiftId”: <Ідентифікатор зміни>,

“OpenShiftFiscalNum”: <Фіскальний номер документа “Відкриття зміни”>,

“CloseShiftFiscalNum”: <Фіскальний номер документа “Закриття зміни”>,

“Testing”: <Ознака зміни, що містить тестові документи (false/true)>,

“Opened”: <Дата і час відкриття зміни>,

“OpenName”: <П.І.Б. оператора, що відкрив зміну>,

“OpenSubjectKeyId”: <Ідентифікатор ключа суб’єкта сертифікату оператора>,

“Closed”: <Дата і час закриття зміни>,

“CloseName”: <П.І.Б. оператора, що закрив зміну>,

“CloseSubjectKeyId”: <Ідентифікатор ключа суб’єкта сертифікату оператора>,

“ZRepFiscalNum”: <Фіскальний номер Z-звіту>

}]

}

### Запит переліку документів зміни

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “**Documents**”**,**

“NumFiscal”**:** “<Фіскальний номер ПРРО>”,

    “ShiftId”**:** “<Ідентифікатор зміни>”,

“OpenShiftFiscalNum”**:** “<Фіскальний номер документа “Відкриття зміни”>”,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

Повинно бути заповненим або поле “ShiftId”, або поле “OpenShiftFiscalNum”.

Формат JSON відповіді:

{

“UID”**:** “<Унікальний ідентифікатор запиту>”,

“Documents”:

[{

“NumFiscal”**:** “<Фіскальний номер документа>”,

“NumLocal”**:** “<Локальний номер документа>”,

“DocDateTime”**:** “<Дата і час операції, зафіксованої документом>”,

“DocClass”**:** “<Клас документа (“**Check**”, “**ZRep**”)>”,

“CheckDocType”**:** “<Тип чека (“**SaleGoods**”, …)>”,

“CheckDocSubType”**:** “<Розширений тип чека (“**CheckGoods**”, …)>”,

“Revoked”**:** “<Ознака відкликаного документа>”,

“Storned”**:** “<Ознака сторнованого документа>”

}]

}

### Запит підсумків останньої зміни

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “LastShiftTotals”**,**

“NumFiscal”**:** “<Фіскальний номер ПРРО>”,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

Формат JSON відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

    “ShiftState”**:** “<0-зміну не відкрито, 1-зміну відкрито>”,

    “ZRepPresent”**:** “<Ознака присутності Z-звіту (false/true)>”,

    “Testing”**:** “<Ознака зміни, що містить тестові документи (false/true)>”,

“Opened”**:** “<Дата і час відкриття зміни>”,

“OpenName”**:** “<П.І.Б. оператора, що відкрив зміну>”,

“OpenSubjectKeyId”**:** “<Ідентифікатор ключа суб’єкта сертифікату оператора>”,

“Closed”**:** “<Дата і час закриття зміни>”,

“CloseName”**:** “<П.І.Б. оператора, що закрив зміну>”,

“CloseSubjectKeyId”**:** “<Ідентифікатор ключа суб’єкта сертифікату оператора>”,

“ZRepFiscalNum”**:** “<Фіскальний номер Z-звіту>”,

“Totals”**:** “<Підсумки зміни (якщо зміну відкрито)>”

}

У разі відсутності даних, повертається код HTTP 204 “No Content”.

#### Структура об’єкту підсумків зміни

/// <summary>

///     Підсумки по зміні

/// </summary>

public class ShiftTotals {

    /// <summary>

    ///     Підсумки реалізації

    /// </summary>

    public ShiftTotalsOrderType Real { get; set; }

    /// <summary>

    ///     Підсумки повернення

    /// </summary>

    public ShiftTotalsOrderType Ret { get; set; }

    /// <summary>

    ///     Підсумки видачі готівки

    /// </summary>

    public ShiftTotalsCash Cash { get; set; }

    /// <summary>

    ///     Підсумки операцій з іноземною валютою

    /// </summary>

    public ShiftTotalsCurrency Currency { get; set; }

    /// <summary>

    ///     Службове внесення/Отримання авансу/Отримання підкріплення

    /// </summary>

    public decimal ServiceInput { get; set; }

    /// <summary>

    ///     Службова видача/Інкасація

    /// </summary>

    public decimal ServiceOutput { get; set; }

}

/// <summary>

///     Підсумки по типу чека

/// </summary>

public class ShiftTotalsOrderType {

    /// <summary>

    ///     Загальна сума

    /// </summary>

    public decimal Sum { get; set; }

    /// <summary>

    ///     Загальна сума коштів, виданих клієнту ломбарду

    /// </summary>

    public decimal PwnSumIssued { get; set; }

    /// <summary>

    ///     Загальна сума коштів, одержаних від клієнта ломбарду

    /// </summary>

    public decimal PwnSumReceived { get; set; }

    /// <summary>

    ///     Заокруглення (наприклад, 0.71)

    /// </summary>

    public decimal RndSum { get; set; }

    /// <summary>

    ///     Загальна сума без заокруглення (наприклад, 1000.71)

    /// </summary>

    public decimal NoRndSum { get; set; }

    /// <summary>

    ///     Загальна сума переказів коштів

    /// </summary>

    public decimal TotalCurrencySum { get; set; }

    /// <summary>

    ///     Загальна сума комісії від переказів коштів

    /// </summary>

    public decimal TotalCurrencyCommission { get; set; }

    /// <summary>

    ///     Кількість чеків

    /// </summary>

    public int OrdersCount { get; set; }

    /// <summary>

    ///     Кількість чеків видачі коштів клієнтам ломбарду

    /// </summary>

    public int PwnOrdersCountIssued { get; set; }

    /// <summary>

    ///     Кількість чеків одержання коштів від клієнтів ломбарду

    /// </summary>

    public int PwnOrdersCountReceived { get; set; }

    /// <summary>

    ///     Кількість операції переказу коштів

    /// </summary>

    public int TotalCurrencyCost { get; set; }

    /// <summary>

    ///     Підсумки по формах оплати

    /// </summary>

    public List<ShiftTotalsPayForm> PayForm { get; set; }

    /// <summary>

    ///     Податки/збори

    /// </summary>

    public List<ShiftTotalsTax> Tax { get; set; }

}

/// <summary>

///     Підсумки по формах оплати

/// </summary>

public class ShiftTotalsPayForm {

    public int PayFormCode { get; set; } // Код форми оплати

    public string PayFormName { get; set; } // Найменування форми оплати

    public decimal Sum { get; set; } // Сума оплати

}

/// <summary>

///     Податки/збори

/// </summary>

public class ShiftTotalsTax {

    public int Type { get; set; } // Код виду податку/збору

    public string Name { get; set; } // Найменування виду податку/збору

    public string Letter { get; set; } // Літерне позначення виду і ставки податку/збору

    public decimal Prc { get; set; } // Відсоток податку/збору

    public bool Sign { get; set; } // Ознака податку/збору, не включеного у вартість

    public decimal Turnover { get; set; } // Сума обсягів для розрахування податку/збору

    public decimal TurnoverDiscount { get; set; } // Сума обсягів для розрахування податку/збору з урахуванням знижки

    public decimal SourceSum { get; set; } // Вихідна сума для розрахування податку/збору

    public decimal Sum { get; set; } // Сума податку/збору

}

/// <summary>

///     Підсумки видачі готівки

/// </summary>

public class ShiftTotalsCash {

    /// <summary>

    ///     Загальна сума

    /// </summary>

    public decimal Sum { get; set; }

    /// <summary>

    ///     Загальна сума комісії

    /// </summary>

    public decimal Commission { get; set; }

    /// <summary>

    ///     Кількість чеків

    /// </summary>

    public int OrdersCount { get; set; }

}

/// <summary>

///     Підсумки операцій з іноземною валютою

/// </summary>

public class ShiftTotalsCurrency {

    /// <summary>

    ///     Отримано авансів національною валютою

    /// </summary>

    public decimal TotalInAdvance { get; set; }

    /// <summary>

    ///     Отримано підкріплень національною валютою

    /// </summary>

    public decimal TotalInAttach { get; set; }

    /// <summary>

    ///     Здано по інкасації національною валютою

    /// </summary>

    public decimal TotalSurrCollection { get; set; }

    /// <summary>

    ///     Отримано комісії конвертації

    /// </summary>

    public decimal Commission { get; set; }

    /// <summary>

    ///     Кількість розрахункових документів за зміну

    /// </summary>

    public int CalcDocsCnt { get; set; }

    /// <summary>

    ///     Прийнято національної валюти для переказу

    /// </summary>

    public decimal AcceptedN { get; set; }

    /// <summary>

    ///     Видано національної валюти при виплаті переказу

    /// </summary>

    public decimal IssuedN { get; set; }

    /// <summary>

    ///     Отримано комісії в національній валюті при здійсненні переказів

    /// </summary>

    public decimal CommissionN { get; set; }

    /// <summary>

    ///     Кількість операцій (документів) переказів або виплат переказів

    /// </summary>

    public int TransfersCnt { get; set; }

    /// <summary>

    ///     Підсумки по видам іноземної валюти

    /// </summary>

    public List<TotalsCurrencyDetails> Details { get; set; }

}

/// <summary>

///     Підсумки по виду іноземної валюти

/// </summary>

public class TotalsCurrencyDetails {

    /// <summary>

    ///     Код валюти

    /// </summary>

    public int ValCd { get; set; }

    /// <summary>

    ///     Символьний код валюти

    /// </summary>

    public string ValSymCd { get; set; }

    /// <summary>

    ///     Загальна сума придбаної іноземної валюти

    /// </summary>

    public decimal BuyValI { get; set; }

    /// <summary>

    ///     Загальна сума проданої іноземної валюти

    /// </summary>

    public decimal SellValI { get; set; }

    /// <summary>

    ///     Загальна сума придбаної національної валюти

    /// </summary>

    public decimal BuyValN { get; set; }

    /// <summary>

    ///     Загальна сума проданої національної валюти

    /// </summary>

    public decimal SellValN { get; set; }

    /// <summary>

    ///     Загальна сума поверненої клієнтами іноземної валюти за операціями «сторно»

    /// </summary>

    public decimal StorBuyValI { get; set; }

    /// <summary>

    ///     Загальна сума виданої клієнтам національної валюти за операціями «сторно»

    /// </summary>

    public decimal StorSellValI { get; set; }

    /// <summary>

    ///     Загальна сума поверненої клієнтами національної валюти за операціями «сторно»

    /// </summary>

    public decimal StorBuyValN { get; set; }

    /// <summary>

    ///     Загальна сума виданої клієнтам національної валюти за операціями «сторно»

    /// </summary>

    public decimal StorSellValN { get; set; }

    /// <summary>

    ///     Загальна сума прийнятої іноземної валюти за операціями конвертації

    /// </summary>

    public decimal CInValI { get; set; }

    /// <summary>

    ///     Загальна сума виданої іноземної валюти за операціями конвертації

    /// </summary>

    public decimal COutValI { get; set; }

    /// <summary>

    ///     Загальна сума комісії за операціями конвертації

    /// </summary>

    public decimal Commission { get; set; }

    /// <summary>

    ///     Отримано авансів

    /// </summary>

    public decimal InAdvance { get; set; }

    /// <summary>

    ///     Отримано підкріплень

    /// </summary>

    public decimal InAttach { get; set; }

    /// <summary>

    ///     Здано по інкасації

    /// </summary>

    public decimal SurrCollection { get; set; }

    /// <summary>

    ///     Видано іноземної валюти по сторно конвертації

    /// </summary>

    public decimal StorCInValI { get; set; }

    /// <summary>

    ///     Повернуто іноземної валюти по сторно конвертації

    /// </summary>

    public decimal StorCOutValI { get; set; }

    /// <summary>

    ///     Повернуто суму комісії з сторно конвертації

    /// </summary>

    public decimal StorCommission { get; set; }

}

### Запит відомостей про документ за локальним номером

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “DocumentInfoByLocalNum”**,**

“NumFiscal”**:** “<Фіскальний номер ПРРО>”,

“NumLocal”**:** “<Локальний номер документа>”,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

Формат JSON відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

“NumFiscal”**:** “<Фіскальний номер документа>”,

“DocClass”**:** “<Клас документа (“**Check**”, “**ZRep**”)>”,

“CheckDocType”**:** “<Тип чека (“**SaleGoods**”, …)>”,

“CheckDocSubType”**:** “<Розширений тип чека (“**CheckGoods**”, …)>”,

“Revoked”**:** “<Ознака відкликаного документа>”,

“Storned”**:** “<Ознака сторнованого документа>”

}

У разі відсутності даних, повертається код HTTP 204 “No Content”.

### Запит надсилання документа

Запит повинен містити JSON.

Формат JSON запиту:

{

    “Command”**:** “SendDocument”**,**

“Document”**:** “<Документ в кодуванні Base64>”,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

Елемент “Document” містить документ, створений згідно з розділом “Загальний порядок взаємодії ПРРО з Фіскальним Сервером”.

Формат JSON відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

“Ticket”**:** “<Квитанція в кодуванні Base64>”

}

### Запит надсилання пакета офлайн документів

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

{

    “Command”**:** “SendOflPackage”**,**

“Documents”**:** <Масив офлайн документів в кодуванні Base64>,

“UID”**:** “<Унікальний ідентифікатор запиту>”

}

Масив “Documents” містить документи пакета в послідовності локальних номерів. Див. розділ “Пакет офлайн документів”.

Формат JSON відповіді:

{

    “UID”**:** “<Унікальний ідентифікатор запиту>”,

    “OfflineSessionId”**:** <Ідентифікатор офлайн сесії (надсилається у відповідь на пакет, який містить документ «Завершення офлайн сесії»)>,

    “OfflineSeed”**:** <"Секретне число" для обчислення фіскального номера офлайн документа (надсилається у відповідь на пакет, який містить документ «Завершення офлайн сесії»)>

}

Сервер надсилає номер наступної офлайн сесії та нове «секретне число» у відповідь на пакет, який містить документ «Завершення офлайн сесії» та поповнює діапазон використаних номерів.

## Видаткові чеки (повернення)

Видаткові чеки (<DOCTYPE>0</DOCTYPE>, <DOCSUBTYPE>1</DOCSUBTYPE>, <ORDERRETNUM>Фіскальний номер чека, для якого здійснюється повернення</ORDERRETNUM>) можуть реєструватись в режимі онлайн або офлайн.

## Сторнування попереднього чека

Чеки сторнування (<DOCTYPE>0</DOCTYPE>, <DOCSUBTYPE>5</DOCSUBTYPE>, <ORDERSTORNUM>Фіскальний номер чека, для якого здійснюється сторнування</ORDERSTORNUM>) можуть реєструватись в режимі онлайн або офлайн.

## Тестові документи

Необхідність тестових документів обумовлена задачами тестування програмного забезпечення під час розробки ПРРО.

Документ (чек, Z-звіт) є тестовим, якщо містить елемент <TESTING>true</TESTING>.

*Примітка*. Елемент «TESTING» не дозволяється для документів «Початок офлайн сесії» і «Завершення офлайн сесії».

Тестові документи не є фіскальними, попри те, що їм призначаються фіскальні номери. Для тестових документів у візуальній формі виводиться текст «ТЕСТОВИЙ НЕФІСКАЛЬНИЙ ДОКУМЕНТ».

В межах однієї зміни можуть знаходитись лише документи одного типу – або фіскальні, або тестові.

## Порядок засвідчення повідомлень кваліфікованим електронним підписом

Повідомлення повинне бути засвідчене кваліфікованим електронним підписом відповідно до вимог Закону України «Про електронні довірчі послуги» (див. [https://zakon.rada.gov.ua/laws/show/2155-19](https://zakon.rada.gov.ua/laws/show/2155-19#Text)).

Алгоритм підпису – криптографічний алгоритм, визначений ДСТУ 4145-2002, затвердженим наказом Державного комітету України з питань технічного регулювання та споживчої політики від 28 грудня 2002 № 31 (див. <http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=68769>).

Формат підписаних даних - CAdES-E-T відповідно до вимог ДСТУ ETSI EN 319 122-2:2016 (ETSI EN 319 122-2:2016, IDT) "Електронні підписи й інфраструктури (ESI). Цифрові підписи CAdES. Частина 2. Розширені підписи CAdES", затвердженого наказом державного підприємства "Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості" від 21 червня 2016 № 183 (див. <http://www.leonorm.com.ua/p/NL_DOC/UA/2016/Nak_183.htm>).

Підписаний документ повинен містити сертифікат підписувача (поле "certificates" підписаного повідомлення повинно бути присутнім та містити сертифікат підписувача, необхідний для перевірки підпису).

Підписаний документ не може містити:

* сертифікати видавця
* списки відкликаних сертифікатів
* результати онлайн перевірки сертифікатів

Дані, що підписані, зберігаються всередині підписаного повідомлення (поле "eContent" підписаного повідомлення повинно бути присутнім та містити дані, що були підписані).

### Позначка часу

Час засвідчення повідомлення кваліфікованим електронним підписом фіксується накладанням на КЕП кваліфікованої позначки часу (атрибут signature-time-stamp відповідно до вимог ДСТУ ETSI EN 319 122-2:2016, IDT).

Позначка часу контенту (атрибут content-time-stamp відповідно до вимог ДСТУ ETSI EN 319 122-2:2016, IDT) не дозволяється.

Позначка часу не обов’язкова для КЕП документів, створених в режимі офлайн.

## Обмеження

Розмір пакету даних, що надсилається, не може перевищувати 200 Kb. Якщо пакет даних передано з використанням компресії, перевіряється розмір оригінальних (не компресованих) даних.

# Адреси Фіскального Сервера

<https://fs.tax.gov.ua:8643/fs>

[http://fs.tax.gov.ua:8609/fs](http://fs.tax.gov.ua/fs)