

Опис API Фіскального Сервера (Єдине вікно подання електронної звітності)

Зміст

Загальні положення.....	2
Поняття і визначення.....	3
Порядок локальної нумерації документів.....	3
Порядок фіскальної нумерації онлайн документів	3
Загальний порядок взаємодії ПРРО з Фіскальним Сервером	3
Відкриття зміни.....	3
Реєстрація фіскального касового чека	4
Реєстрація Z-звіту і закриття зміни	4
Офлайн сесія	4
Режими онлайн і офлайн	5
Пакет офлайн документів	5
Відкликання останнього онлайн чека.....	5
Сторнування останнього чека.....	6
Структура фіскального номера офлайн документа	6
Розрахунок контрольного числа	6
Приклад розрахунку контрольного числа	7
Розрахунок гешу офлайн документа	7
Блокування роботи ПРРО.....	8
Формування Діапазону та облік фіскальних номерів.....	8
Сценарії роботи	8
1. Новий екземпляр ПРРО.....	8
2. ПРРО працює з відкритою зміною (Сервер недоступний)	8
3. ПРРО працює з закритою зміною (Сервер недоступний)	8
4. Перевірка зв'язку з Сервером (ПРРО знаходиться в режимі онлайн)	9
5. Перевірка зв'язку з Сервером (ПРРО знаходиться в режимі офлайн)	9
6. Скасування реєстрації ПРРО	9
Зауваження щодо реалізації.....	9
Класи документів.....	9
Чек. Типи документів	10
Чек. Розширені типи документів.....	11
Типи даних запитів документів	11
Коди результату обробки запитів документів	12

Коди помилок	13
Команди	14
Запит стану сервера.....	14
Запит доступних об'єктів.....	15
Запит стану ПРРО	15
Запит чека.....	16
Запит чека розширений	17
Запит Z-звіту	17
Запит переліку змін за період	17
Запит переліку документів зміни	18
Запит підсумків останньої зміни	18
Запит відомостей про документ за локальним номером.....	24
Видаткові чеки (повернення)	25
Сторнування попереднього чека	25
Порядок засвідчення повідомлень кваліфікованим електронним підписом.....	25
Позначка часу.....	25
Обмеження	25
Адреса Фіскального Сервера	26

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

6. ПРРО створює XML документ чека із призначеним локальним номером, засвідчує його КЕП продавця і надсилає на Сервер.
7. Сервер здійснює необхідні перевірки.
8. У разі помилки, Сервер надсилає у відповідь XML квитанції.
9. 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.
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 режиму (сесії)» і «Завершення офлайн режиму (сесії)».

Проміжок часу між першим і останнім документом (тобто між документами «Початок офлайн режиму (сесії)» і «Завершення офлайн режиму (сесії)») у межах 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 однієї офлайн сесії не може перевищувати надану кількість номерів (стандартно 2000, за окремим запитом суб'єкта кількість може бути збільшена).

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

Коли Сервер знову стає доступний, ПРРО надсилає пакет, що містить документи: «Початок офлайн режиму (сесії)», чеки, Z-звіти, «Завершення офлайн режиму (сесії)». У відповідь Сервер надсилає номер наступної офлайн сесії. (Поповнює фіскальні номери із Діапазону до 2000, або до більшої кількості, що використана ПРРО із збільшеним діапазоном).

Офлайн сесія застосовується тільки з ПРРО, для якого виданий номер 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 передуює 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 режиму (сесії)», що не надісланий на Сервер, і для якого відсутній документ «Завершення офлайн режиму (сесії)».

Пакет офлайн документів

Коли Сервер стає доступний, ПРРО надсилає один (або декілька) пакетів, що містить документи: «Початок офлайн режиму (сесії)», чеки, Z-звіти, «Завершення офлайн режиму (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 у відповідь на пакет, який містить документ «Завершення офлайн режиму (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 режиму (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 режиму (сесії)» додається елемент `<REVOKELASTONLINEDOC>true</REVOKELASTONLINEDOC>`.

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:

`<Номер офлайн сесії>.<Локальний номер документа в офлайн сесії>.<Контрольне число>`

Локальний номер документа в 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")

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 режиму (сесії)») (якщо елемент присутній)

Примітка. Для документа «Початок офлайн режиму (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 режиму (сесії)») може бути змінений (може бути додано елемент `<REVOKELASTONLINEDOC>`).

Від текстового рядку розраховується геш за алгоритмом CRC32, як десяткове беззнакове число. Із розрахованого гешу беруться 4 молодші розряди.

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 шістнадцятковому форматі: 0xbddbba6
- 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 шістнадцятковому 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).

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

Для першого розрахункового документа офлайн сесії елемент "PREVDOCHASH" не створюється.

Блокування роботи ПРРО

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»

Формування Діапазону та облік фіскальних номерів

Сервер формує діапазон фіскальних номерів для кожного ПРРО, який видається ПРРО під час присвоєння ПРРО фіскального номера (2000 або більше).

Сервер поповнює діапазон номерів до 2000 для ПРРО після отримання чеків, Z-звітів, повідомлень із присвоєними номерами із Діапазону.

Сервер веде облік виданих Діапазонів, присвоює статус кожному номеру:

- Зарезервований – після видачі номера користувачеві до моменту прийняття Сервером документів від ПРРО;
- Використаний – після прийняття документів, яким присвоєні номери із Діапазону.

Сценарії роботи

1. Новий екземпляр ПРРО

- 1.1. Користувач заходить в ПРРО з аутентифікацією по КЕП. ПРРО одержує з Сервера довідники (перелік господарських одиниць (ГО), перелік фіскальних номерів ПРРО, перші 2000 фіскальних номерів для присвоєння розрахунковим документам ПРРО, що працює в режимі офлайн тощо).
- 1.2. 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, номер офлайн сесії і «секретне число».
- 1.3. Користувач видає чеки.

2. ПРРО працює з відкритою зміною (Сервер недоступний)

- 2.1. Користувач заходить в ПРРО з аутентифікацією по КЕП.
- 2.2. Користувач заходить в ГО і фіскальний номер ПРРО, для якого відкрито офлайн сесію.
- 2.3. ПРРО перевіряє, чи зміну відкрито тим самим користувачем. Якщо ні – робота Користувача неможлива до закриття зміни.
- 2.4. Користувач видає чеки.

3. ПРРО працює з закритою зміною (Сервер недоступний)

- 3.1. Користувач заходить в ПРРО з аутентифікацією по КЕП.
- 3.2. Користувач заходить в доступну ГО і фіскальний номер ПРРО, для якого відкрито офлайн сесію, і дає команду на відкриття зміни.
- 3.3. Користувач видає чеки.

3.4. Користувач закриває зміну.

4. Перевірка зв'язку з Сервером (ПРРО знаходиться в режимі онлайн)

4.1. Якщо при передачі документа Сервер недоступний, ПРРО запитує у користувача підтвердження переходу в режим офлайн.

4.2. Якщо користувач згодний, ПРРО створює документ «Початок офлайн режиму (сесії)»

5. Перевірка зв'язку з Сервером (ПРРО знаходиться в режимі офлайн)

5.1. ПРРО періодично перевіряє зв'язок з Сервером (наприклад, за допомогою команди «Запит стану ПРРО», або команди «Запит стану сервера»).

5.2. Якщо зв'язок присутній, ПРРО автоматично переходить в режим онлайн, про що інформує користувача.

5.3. ПРРО надсилає на Сервер пакет, що містить документи, створені в режимі офлайн.

5.4. У разі відмови Сервера на підставі некоректного сертифікату, пакет засвідчується підписом старшого користувача або керівника.

5.5. Якщо пакет успішно прийнятий, ПРРО отримує підтвердження (квитанцію) від фіскального Сервера.

6. Скасування реєстрації ПРРО

6.1. Користувач надсилає заяву з позначкою «Скасування реєстрації»

6.2. Користувач надсилає повідомлення з позначкою «несправність» або «крадіжка пристрою»

6.3. 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>/psk" – 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 запита 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 запита документа
/// </summary>
public enum DocumentRequestResultCode {
    /// <summary>
    ///     ОК
    /// </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
}

Коди помилок
/// <summary>
///     Код помилки
/// </summary>
public enum ErrorCode {
    /// <summary>
    ///     ОК
    /// </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 до
рівнює 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, п
    овинен 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
}

```

Команди

Запит стану сервера

Запит повинен містити JSON.

Формат JSON запиту:

```

{
    "Command": "ServerState"
}

```

Формат JSON відповіді:

```

{
    Timestamp = <Дата і час відповіді сервера>
}

```

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

Запит доступних об'єктів

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

```
{
  "Command": "Objects"
}
```

У відповідь повертається перелік доступних користувачу господарських одиниць і ПРРО.

Формат JSON відповіді:

```
internal class ResponseObjects {
    public TaxObjectItem[] TaxObjects { get; set; } // Перелік ГО
}

public class TaxObjectItem {
    public string Address { get; set; } // Адреса ГО
    public long Entity { get; set; } // Ідентифікатор запису ГО
    public string Name { get; set; } // Найменування ГО
    public string OrgName { get; set; } // Найменування суб'єкта
    господарювання
    public string Tin { get; set; } // Код ЄДРПОУ/ДРФО платника
    податків
    public string Ipn { get; set; } // Податковий номер платника ПДВ

    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 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)>"
}
```


}

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

Формат JSON відповіді:

```
{
  ShiftState = <0-зміну не відкрито, 1-зміну відкрито>,
  ShiftId = <Ідентифікатор зміни>,
  OpenShiftFiscalNum = <Фіскальний номер документа "Відкриття
зміни">,
  ZRepPresent = <Ознака присутності Z-звіту (false/true)>,
  Name = <П.І.Б. оператора, що відкрив зміну>,
  SubjectKeyId = <Ідентифікатор ключа суб'єкта сертифікату
оператора, що відкрив зміну>,
  FirstLocalNum = <Перший внутрішній номер документа у зміні>,
  NextLocalNum = <Наступний внутрішній номер документа>,
  LastFiscalNum = <Останній фіскальний номер документа>,

  OfflineSupported = <Для ПРРО може використовуватись режим офлайн>,
  ChiefCashier = <Користувач є старшим касиром>,

  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)>,
  Closed = <Ознака скасованої реєстрації ПРРО, на якому наразі не
закрито зміну>
}
```

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>",
  "Original": <Ознака запиту оригінального документа, надісланого
продавцем (false/true)>
}
```

```
}
```

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>",
  "Type": "<Тип даних запита документа>"
}
```

Формат JSON відповіді:

```
{
  "Data": "<Результат запиту в кодуванні Base64>",
  "ResultCode": "<Код результату обробки запита документа>",
  "ResultText": "<Опис результату обробки запита документа>"
}
```

Запит Z-звіту

Запит повинен містити JSON, засвідчений КЕП користувача.

Формат JSON запиту:

```
{
  "Command": "ZRep",
  "RegistrarNumFiscal": "<Фіскальний номер ПРРО>",
  "NumFiscal": "<Фіскальний номер Z-звіту>",
  "Original": "<Ознака запиту оригінального документа, надісланого продавцем (false/true)>"
}
```

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

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

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>"
}
```

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:

```
{
  Shifts:
  [{
    ShiftId = <Ідентифікатор зміни>,
    OpenShiftFiscalNum = <Фіскальний номер документа "Відкриття
зміни">,

    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>
  }]
}
```

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">"
}
```

Повинно бути заповненим або поле "ShiftId", або поле "OpenShiftFiscalNum".

Формат JSON відповіді:

```
{
  Documents:
  [{
    NumFiscal = <Фіскальний номер документа>,
    NumLocal = <Локальний номер документа>,
    DocClass = <Клас документа ("Check", "ZRep")>,
    CheckDocType = <Тип чека ("SaleGoods", ...)>,
    Revoked = <Ознака відкликаноного 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>"
}
```

Формат JSON відповіді:

```
{
  ShiftState = <0-зміну не відкрито, 1-зміну відкрито>,
  ZRepPresent = <Ознака присутності Z-звіту (false/true)>,
  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
```

Опис API Фіскального Сервера (Єдине вікно подання електронної звітності)

```

/// </summary>
public class ShiftTotalsOrderType {
    /// <summary>
    ///     Загальна сума
    /// </summary>
    public decimal Sum { get; set; }

    /// <summary>
    ///     Загальна сума комісії від переказу коштів
    /// </summary>
    public decimal TotalCurrencyCommission { get; set; }

    /// <summary>
    ///     Кількість чеків
    /// </summary>
    public int OrdersCount { get; set; }

    /// <summary>
    ///     Кількість операції переказу коштів
    /// </summary>
    public int TotalCurrencyCost { get; set; }

    /// <summary>
    ///     Підсумки по формам оплати
    /// </summary>
    public List<ShiftTotalsPayForm> PayForm { get; set; }

    /// <summary>
    ///     Податки/збори
    /// </summary>
    public List<ShiftTotalsTax> Tax { get; set; }
}

/// <summary>
///     Підсумки по формам 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 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 пере
казів
    /// </summary>
    public decimal CommissionN { get; set; }

    /// <summary>
    ///     Кількість операцій (документів) переказів або виплат перек
азів
    /// </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 опе
    раціями «сторно»
    /// </summary>
    public decimal StorBuyValI { get; set; }

    /// <summary>
    ///     Загальна сума виданої клієнтам національної валюти за опер
    аціями «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 опер
    аціями «сторно»
    /// </summary>
    public decimal StorSellValN { get; set; }

    /// <summary>
    ///     Загальна сума прийнятої іноземної валюти за операціями кон
    вертації
    /// </summary>
    public decimal CInValI { get; set; }

```

```

    /// <summary>
    ///     Загальна сума виданої іноземної валюти за операціями конве
птації
    /// </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>"
}

```

}

Формат JSON відповіді:

{

```
NumFiscal = <Фіскальний номер документа>,
DocClass = <Клас документа ("Check", "ZRep")>,
CheckDocType = <Тип чека ("SaleGoods", ...)>,
Revoked = <Ознака відкликаного документа>,
Storned = <Ознака сторнованого документа>
```

}

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

Видаткові чеки (повернення)

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

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.

Порядок засвідчення повідомлень кваліфікованим електронним підписом

Повідомлення повинне бути засвідчене кваліфікованим електронним підписом у відповідності до вимог Закону України «Про електронні довірчі послуги» (див. <https://zakon.rada.gov.ua/laws/show/2155-19>).

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).

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 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.

Адреса Фіскального Сервера

<http://fs.tax.gov.ua:8609/fs>