



Специфікація використання

## **Cashalot API Bridge**

v. 5.9

## Зміст

Зміст	2
Поняття і визначення	4
Загальна інформація	4
Апаратні та програмні вимоги	5
<b>API ПРРО Cashälot з .NET інтерфейсом CashalotApiBridge</b>	<b>6</b>
Відповідь на виконання методів	6
Порядок роботи обліково-розрахункової системи	8
з API бібліотекою ПРРО Cashälot	8
1. Встановлення параметрів	8
2. Відкриття зміни	13
3. Службове внесення грошових коштів в касу	13
4. Службова видача грошових коштів з каси	14
5. Фіскалізація чека продажу товарів, послуг	14
6. Проведення операції повернення товару	25
7. Визначення фіскального номера чека	27
8. Визначення локального номера чека по фіскальному номеру	27
9. Отримання XML-структури чека	27
10. Відображення фіскалізованого чека	28
11. Створення X-звіту	28
12. Закриття зміни та створення Z-звіту	28
13. Ручний перевод ПРРО в режим офлайн	29
14. Ручний перевод ПРРО в режим онлайн	29
15. Синхронізація залишків по товарам	30
16. Синхронізація залишків та товарів	30
17. Сторнування чека продажу/повернення	30
18. Відображення QR-кода зареєстрованого чека	31
19. Відправка чека покупцю за допомогою сервісів доставки SMS, Viber або E-mail	31
20. Отримання переліку налаштованих пристроїв POS-терміналів в програмі	32
21. Встановлення платіжного POS-терміналу пристроєм за замовчуванням	33
22. Формування звіту про реалізовані товари	33
23. Проведення транзакції через POS-термінал	34
24. Повернення транзакції (оплати) через POS-термінал	35
25. Скасування транзакції (оплати) через POS-термінал	36
26. Проведення транзакції через функцію “Оплата частинами”	37
27. Повернення транзакції по операції “Оплата частинами”	38
28. Створення відкладеного чека продажу товарів/послуг	39
29. Фіскалізація відкладеного чеку продажу товарів/послуг	40
30. Отримати інформацію про відкладений чек за його ReceiptID	41
31. Отримати інформацію про відкладений чек за його номером EH (експрес-накладної)	41
32. Отримати інформацію про перший відкладений чек	42
33. Отримати інформацію наступний відкладений чек	43
34. Відображення візуальної форми чека за його ReceiptID	44
Приклади використання засобів оплат у чеку (спрощений та розширений варіант)	46
ЛОМБАРДНІ ОПЕРАЦІЇ	63
Особливості роботи з чеками типу «Інтернет-продаж» при доставці Новою поштою	63
Загальний опис	63
Створення відкладеного чека	63
Отримання інформації про відкладений чек	64
Реєстрація відкладеного чека	64
Особливості реєстрації відкладеного чека «Інтернет-продажу» в залежності від статусу EH(ТТН) та	

визначеного способу оплати по чеку  
Приклад реалізації функції для виклику API методів ПРРО CashÄlot

65

66

## Поняття і визначення

**КЕП** – кваліфікований електронний підпис.

**Локальний номер чека** – номер, призначений чеку системою виписки електронних касових чеків, що використовується продавцем.

**Офлайн сесія** – сукупність документів, створених в режимі офлайн, між припиненням і відновленням зв'язку ПРРО з ФСКО.

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

**Фіскальний номер чека(ФН)** – номер, призначений чеку сервером ФСКО.

**ФСКО** – фіскальний сервер контролюючого органу.

## Загальна інформація

**Cashälot API Bridge** використовується для інтеграції ПРРО з розрахунково-обліковими системами за допомогою підключення бібліотеки CashälotApiBridge.dll. В даному документі приведено специфікацію методів бібліотеки Cashälot.

Інтеграція надає користувачам можливість, безпосередньо з розрахунково-облікової системи, скористатися наступним функціоналом програмного забезпечення Cashälot:

- Відкриття та закриття касової зміни;
- Створення та реєстрація чеків продажу/повернення;
- Сторнуванн чеків продажу/повернення;
- Створення та реєстрація чеків з типами оплати передоплата, післяплата, інтернет-продажа, повна післяплата, оплата частинами;
- Створення видаткового чека на основі чека продажу та його реєстрація;
- Формування та друк Z-звіту за даними фіскального сервера контролюючого органу (скорочено ФСКО)/X-звіту за даними ПРРО;
- Створення та реєстрація службових чеків;
- Службове внесення;
- Службова видача;
- Робота в режимі офлайн;
- Можливість відображення на екрані зареєстрованого чека на пристрої продавця з QR-кодом після успішної реєстрації;
- Друк зареєстрованого чека;
- Формування періодичних звітів.

## Апаратні та програмні вимоги

Для оптимальної роботи ПРРО Cashälot рекомендовані наступні системні та програмні вимоги:

### **Операційна система**

Програмний комплекс коректно функціонує на комп'ютерах з сучасними операційними системами Microsoft та компонентами, що входять до її складу. Рекомендовано використовувати останню версію ОС, наприклад, Windows 10 або Windows Server 2019 відповідно до потреб, з встановленими актуальними оновленнями.

### **Апаратне забезпечення:**

- процесор з мінімальною тактовою частотою від 2 ГГц;
- оперативна пам'ять від 2 GB;
- вільне місце на жорсткому диску від 1,5 GB;
- кольоровий графічний дисплей;
- маніпулятор типу миша та клавіатура;
- встановлений український або російський мовний стандарт регіональних налаштувань операційної системи;
- підтримка кирилиці в операційній системі;
- доступ до поштового сервера;
- наявний зв'язок з фіскальним сервером контролюючого органу.

### **Мережа**

Для роботи користувача необхідний доступ до мережі інтернет. Також, необхідно мати доступ до кабінету Cashälot, фіскального сервера контролюючого органу та до ресурсу центру сертифікації ключів. Наприклад, якщо Ви використовуєте ключі від "АЦСК Україна", необхідно надати доступ до ресурсу: <https://uakey.com.ua/> - 443. Якщо ж ви використовуєте КЕП (ЕЦП) інших кваліфікованих надавачів (АЦСК), то необхідно надати доступ до серверів TSP, OCSP, CMP відповідних центрів.

### **Загальний список ресурсів:**

- <https://load.cashalot.org.ua/update/> - 443 - Сервіс завантаження інсталяцій програми (дистрибутиви, оновлення);
- <https://fs.tax.gov.ua:8643/fs/cmd> - Сервіс API ДПС;
- <https://my.cashalot.org.ua/> - 443 - Особистий кабінет Cashälot.

## API ПРРО Cashälot з .NET інтерфейсом CashalotApiBridge

Підключення ПРРО до розрахунково-облікової системи відбувається за допомогою розробленої API бібліотеки ПРРО Cashälot **CashalotApiBridge.dll**, шлях до якої необхідно вказати для коректного налаштування. Користувачу необхідно розробити функції, що будуть транслювати виклик методів із бібліотеки Cashälot.

**Зверніть увагу!** Виклик бібліотеки CashalotApiBridge.dll необхідно здійснювати виключно з кореневого каталогу встановленого екземпляру Cashälot.

Як приклад у цьому документі наведено **C#** реалізацію у вигляді методу:

**CallCashalotMethod** (“Метод”, “Параметри”).

Оскільки будь-яка дія в ПРРО (відкриття/закриття зміни, відправка чеків тощо) здійснюється шляхом підписання блоку даних КЕП, то під час роботи необхідно вказати шлях до сертифікатів, вибрати користувача і здійснити авторизацію.

За допомогою наявних методів API бібліотека реалізує виконання наступних операцій ПРРО Cashälot:

- Відкриття зміни;
- Службове внесення грошових коштів в касу;
- Службове вилучення грошових коштів з каси;
- Проведення розрахункових операцій продажу товарів/послуг(Фіскалізація чека);
- Проведення операції повернення товару;
- Визначення номера фіскального чека за локальним номером чека в касі;
- Створення X-Звіту;
- Визначення касового локального номера фіскального чека;
- Закриття зміни та створення Z-Звіту.

### Відповідь на виконання методів

Виконання методів повертає об'єкт **CashalotApiRsp**, що містить інформацію про результат роботи:

- **Ret** – ознака успішності виконання функції (**bool**);
- **ErrorString** – текст помилки (**string**);
- **Values** – масив параметрів (**Dictionary<string, string>**).

Приклад:

```
public class CashalotApiRsp
{
    //Ознака успішності виконання функції (true - успішно/false - не успішно)
    public bool Ret { get; set; } = false;
    //Рядок з текстом помилки
    public string ErrorString { get; set; } = String.Empty;
    //Масив параметрів, які можуть повернути функції АПІ
    public Dictionary<string, string> Values { get; set; }
}
```

**Values**, залежно від використаного методу, може містити такі параметри:

Назва параметра	Значення
ReceiptFiscalNumber	Фіскальний номер чека.
ReceiptLocalNumber	Локальний номер чека.
ShiftID	Ідентифікаційний номер поточної зміни.
OfflineMode	Ознака офлайн режиму.
FSKORceiptLink	Посилання на перегляд чека в ДПС
CashalotReceiptLink	Посилання на перегляд чека в кабінеті Cashalot
Type	<a href="#">Тип документа</a> .
Base64Str1251ReceiptXML	XML-структура фіскалізованого чека в кодуванні Base64.
ServiceInput	Загальна сума службових внесень по зміні.
ServiceOutput	Загальна сума службової видачі по зміні.
OrderCount	Кількість чеків продажу.
OrderExpCount	Кількість чеків повернення.
SumCash	Поточний залишок готівки в касі.
TotalSUM	Загальна сума продажу.
TotalSUMCash	Загальна сума продажу "Готівка".
TotalSUMCard	Загальна сума продажу "Картка".
TotalTAXSUM	Загальна сума ПДВ по продажу.
ReturnSUM	Загальна сума повернення.
ReturnSUMCash	Загальна сума повернення "Готівка".
ReturnSUMCard	Загальна сума повернення "Картка".
ReturnTAXSUM	Загальна сума ПДВ по поверненню.
TotalTAX_<1>_<2>%_<3>	Деталізація податків по фіскальним чекам.
ReturnTAX_<1>_<2>%_<3>	Деталізація податків по чекам повернення.
ReceiptPrePayOrderNum	Номер інтернет-замовлення (для чеків інтернет-продажу або передплати/післяплати)
Base64ReceiptJpeg	JPEG структура фіскалізованого чека у форматі base64
Base64ReceiptPdf	PDF структура фіскалізованого чека у форматі base64
LicEndDate	Дата завершення терміну дії ліцензії ПРРО у форматі (rrrr.мм.дд гг.хв.сс)
CertEndDate	Дата завершення терміну дії сертифікату касира у форматі (rrrr.мм.дд гг.хв.сс)

Назви об'єктів податків чеків оплати **TotalTAX** та повернення **ReturnTAX** побудовані з використанням складових, що визначаються на основі карток проданих товарів:

Складова	Зміст складової	Можливі значення	Приклад
<1>	Літера податку зазначена в картках товарів	A, B, V, ...	TotalTAX_A_20%_SUM ReturnTAX_B_5%_TURNOVER TotalTAX_Є_10%_REMOVEDTAXES
<2>	Відсоток податку	0, 5, 20, ...	
<3>	Сума	SUM	
	Обіг	TURNOVER	
	Знижка	DISCOUNTSUM	
	Виключені податки	REMOVEDTAXES	

Також повертаються службові повідомлення або графічні відображення чеків та звітів, які можна відправити на друк.

# Порядок роботи обліково-розрахункової системи з API бібліотекою ПРРО Cashälot

Перед початком роботи каси виконується операція відкриття зміни. При відкритій касовій зміні доступні проведення розрахункових операцій, повернення товарів, службові внесення та видача грошових коштів, а також, створення X-звітів.

Для завершення касової зміни необхідно провести операцію закриття зміни, під час якої відбувається службове вилучення грошових коштів з каси і формується Z-звіт.

Всі перелічені операції додатково дозволяють відправку документів на друк.

Деякі методи ПРРО Cashälot, вказані у розділі [загальна інформація](#), виконуються неявно під час обробки запитів від розрахунково-облікової системи і можуть повертати відповідні службові повідомлення (наприклад, перехід у офлайн-режим тощо).

## 1. Встановлення параметрів

Метод **SetParameter** призначений для встановлення значень параметрів налаштувань API бібліотеки. Рекомендовано встановити всі параметри на початку роботи. Дані параметри можна змінювати в процесі роботи з API. Наприклад, для того, щоб виконувати автоматичний друк тільки Z-звіту в процесі роботи програми.

```
public string SetParameter( string Name, string Value );
```

Кожен параметр повинен містити в собі:

Ім'я	Тип	Опис
Name	string	Ім'я параметра
Value	string	Значення параметра

### Приклад:

```
instparameter Param = new instparameter();  
  
Param.PathToCashalotDir = "C:\\ProgramData\\Cashalot\\Cashalot";  
Param.DeviceIDFnRRO = "4000000000";  
  
Object[] PathToCashalotDir = { "PathToCashalotDir", Param.PathToCashalotDir };  
var ret = CallCashaLotMethod("SetParameter", PathToCashalotDir);  
  
Object[] DeviceIDFnRRO = { "DeviceIDFnRRO", Param.DeviceIDFnRRO };  
var ret1 = CallCashaLotMethod("SetParameter", DeviceIDFnRRO);
```

**SetParameter** складається з обов'язкових параметрів (виділені червоним) та не обов'язкових параметрів. Параметри повинні мати послідовність, як в таблиці опису параметрів. Тип всіх параметрів **string**.

Опис параметрів вказано в таблиці:

Ім'я параметра(Name)	Значення параметра(Value)
PathToCashalotDir	Шлях до каталогу з встановленим ПРРО Cashälot.
DeviceIDFnRRO	Фіскальний номер ПРРО.
NOAUTOUPDATE	Перевірка на наявність нової версії на сервері <a href="https://load.cashalot.org.ua/update/">https://load.cashalot.org.ua/update/</a> та запит на оновлення програми Cashälot в момент початку роботи з API: "True" – заборонити оновлення; "False" – дозволити оновлення(за замовчуванням).
NOINTERFACEMODE	Ознака використання інтерфейсу РМК для авторизації та службових операцій видачі та внесення готівки в касу. "False" – при роботі з API викликається діалогове вікно. "True" – авторизація відбувається шляхом передачі параметрів авторизації PathToCertificate та PwdToCertificate. Вікна службових операцій видачі та внесення готівки не викликаються (проте, результат операції службового внесення/видачі готівки відображається).
PathToCertificate	Шлях до каталогу з файловим ключем користувача. Важливо: в папці повинні міститися ключ і сертифікат тільки одного користувача. Якщо касою користуються декілька касирів, необхідно помістити їх сертифікати і ключі в окремі каталоги.
PwdToCertificate	Пароль від ключа.
USETOKEN	Чи використовується токен для авторизації. "True" – використовується. "False" – не використовується (використовується файловий ключ). Важливо: параметр USETOKEN має бути заданий після параметрів PathToCashalotDir, DeviceIDFnRRO та NOINTERFACEMODE. Для пришвидшення завантаження програми рекомендовано не використовувати токени.
USECLOUDKEY	Чи використовується CloudKey для авторизації. "True" – використовується. "False" – не використовується. Параметр USECLOUDKEY має бути заданий після параметрів PathToCashalotDir, DeviceIDFnRRO та NOINTERFACEMODE.
NOAUTOOPENSIFT	Перевірка стану зміни (відкрита або закрита). Якщо зміна закрита, при виклику будь-якої СОМ-функції зміна автоматично відкривається. "False" – автоматично відкривати зміну, якщо вона закрита. "True" – не відкривати автоматично зміну. В цьому випадку робота СОМ-функції буде перервана і результат виконання буде повернуто у вигляді помилки.
DEFAULTPRINTERNAME	Визначає принтер для друку при натисканні кнопки "Друк" в діалогових вікнах: 1) "" – пустий рядок, виводиться діалог вибору принтера; 2) "DEFAULTWINDOWS" – використовувати принтер Windows за замовчуванням; 3) "ім'я принтера чеків" – в параметр передається ім'я принтера чеків, який буде використовуватись для друку чеків в ПРРО Cashälot.
AUTOPRINTMODE	Параметр визначає автоматичну відправку на друк чека або службового звіту без відображення інформаційного вікна: "True" – відправляти на друк; "False" – не відправляти на друк.

	Якщо задано принтер за замовчуванням (DEFAULTPRINTERNAME), чек друкується одразу, в іншому випадку відображається вікно вибору принтера для друку.
NOPRINTMODE	Дозвіл на друк: "True" – заборонити друк; "False" – дозволити друк.
SHOWREPORTADDITIONALINFO	Виводити додаткову інформацію в нижній частині Z-звіту. В разі встановленого значення "True", відображаються такі поля: - Залишок готівки в касі; - Сума знижки; - Сума націнки. "True" - відображає додаткову інформацію; "False" – не відображає додаткову інформацію.

## USEGOODSSTORAGEMODE

Параметр [USEGOODSSTORAGEMODE](#) визначає використання ведення складського обліку по товарам та автоматичну синхронізацію залишків по ним при використанні API Cashälot за умови активованого налаштування в кабінеті користувача в розділі «Склад - Налаштування - Облік товарів у ПРРО». Параметр може приймати наступні значення:

«0» - програма працює в стандартному режимі без ведення обліку товарів в API та синхронізації залишків по товарам з кабінетом користувача.

«1» - режим передбачає використання автоматичної синхронізації залишків по товарам та інформування користувача окремим повідомленням про перевищення їх ліміту під час спроби реєстрації фіскального чека. Після появи повідомлення користувачу доступна можливість відмовитись від подальшої обробки чека або погодитись та продовжити реєстрацію, але таким чином залишки по товарам будуть мати від'ємне значення кількості.

«2» - режим передбачає використання автоматичної синхронізації залишків по товарам та інформування користувача окремим повідомленням про перевищення їх ліміту без можливості продовжити обробку та реєстрацію чека.

Візуалізація повідомлення про перевищення ліміту залишків з можливістю продовжити/перервати процедуру фіскалізації чека, використовуючи перший режим параметра [USEGOODSSTORAGEMODE](#), буде виникати також при ознаці використання режиму роботи без інтерфейса ([NOINTERFACEMODE=true](#)).

\*Кроки по практичному застосуванню параметра [USEGOODSSTORAGEMODE](#)

1. Імпортувати номенклатуру, яка використовується в API Cashälot, до кабінету користувача. Після імпорту необхідно перенести номенклатуру до еталонної групи.
2. В кабінеті користувача в модулі «Склад» сформувати та провести документ «Надходження» по необхідним товарним позиціям.
3. Налаштувати та встановити використання параметра [USEGOODSSTORAGEMODE](#) у необхідному режимі.
4. Виконати синхронізацію даних по товарам (номенклатурі) та залишкам товарів методом [SyncGoodsWithBalance](#). Подальше оновлення інформації по залишкам відбувається в автоматичному режимі при відкриті зміни та реєстрації фіскального чека, а також в ручному режимі, з використанням методів синхронізації товарів та залишків по ним або окремо, виконавши синхронізацію залишків.
5. Сформувати та заповнити фіскальний чек даними по номенклатурі, для якої доступні залишки по товарам.

	<p><b>Зверніть увагу!</b> Коректна робота обраного параметра відбувається лише за умови, що в API Cashalot використовується одна й та сама номенклатура, що заведена в кабінеті користувача.</p>
<p>USEPRGPOSTERMINAL</p>	<p>Параметр відповідає за можливість проведення оплати платіжною картою через POS-термінал, який підключений в Cashalot в розділі «Налаштування - Додаткове обладнання - POS-термінал». Оплату на терміналі можливо здійснити після передачі чека на реєстрацію. В разі використання обраного параметра фіскальний чек буде містити всі необхідні обов'язкові реквізити банківського терміналу (ЕПЗ, RRN, еквайр та ін.).</p> <p>“True” - використовується POS-термінал, підключений в Cashalot для оплати під час передачі чека на фіскалізацію з обраною формою оплати Картка (<a href="#">SumPayByCard</a>) .</p> <p>“False” - під час фіскалізації чека з формою оплати Картка підключений POS-термінал в Cashalot не використовується.</p> <p>В разі неуспішної транзакції на банківському терміналі операція реєстрації чека не відбувається.</p>
<p>RATELETTERCREATEIFNOTEXIST</p>	<p>Параметр відповідає за автоматичне створення податкової ставки та літерного позначення при реєстрації чеків продажу. Рекомендовано до використання, якщо в кабінеті Cashalot зайняті доступні комбінації податкової ставки та літери.</p> <p>“True” - відбувається автоматичне створення податкової ставки та літерного позначення в разі її відсутності в Cashalot</p> <p>“False” - використовуються наявні завантажені в Cashalot податкові ставки та їх літерні позначення</p>
<p>USESELTERMINALINFOWND</p>	<p>Параметр призначений для можливості ручного підтвердження термінальної операції користувачем за умови, якщо операція з POS-терміналом пройшла неуспішно (обірвався зв'язок терміналу з Cashalot в момент проведення операції або ін.). В такому разі з'явиться додаткове вікно за допомогою якого можливо підтвердити операцію або виконати запит результату операції з POS-терміналу.</p> <p>Параметр працює за умови використання параметра <a href="#">USEPRGPOSTERMINAL</a>. Вікно підтвердження буде з'являтися навіть якщо використовується параметр <a href="#">NOINTERFACEMODE</a>.</p> <p>“True” - після отримання помилки від POS-терміналу при проведенні операції з'явиться додаткове вікно з можливістю ручного підтвердження операції.</p> <p>“False” - в разі отримання помилки від POS-терміналу необхідно буде повторити спробу виконання операції.</p>

<p>AUTOCLOSETERMINALSHIFT</p>	<p>Параметр призначений для закриття термінальної зміни під час закриття зміни ПРРО.</p> <p>“0” - закриття зміни на POS-терміналі за умови закриття зміни ПРРО не відбувається.</p> <p>“1” - під час закриття зміни на ПРРО буде перевірятись наявність проведених термінальних чеків та в разі їх наявності буде здійснено закриття зміни на POS-терміналі (за умови налаштування POS-терміналу в Cashalot)</p> <p>“2” - під час закриття зміни на ПРРО буде завжди закриватись зміна на POS-терміналі (за умови налаштування POS-терміналу в Cashalot)</p>
<p>ISNEEDAPIRETRECEIPTJPEG</p>	<p>Параметр використовується для отримання чека продажу або повернення у форматі JPEG, закодованого в Base64. Результат повертається у структурі відповіді в полі JsonVal, де значення доступне в параметрі Base64ReceiptJpeg.</p>
<p>ISNEEDAPIRETRECEIPTPDF</p>	<p>Параметр використовується для отримання чека продажу або повернення у форматі PDF, закодованого в Base64. Результат повертається у структурі відповіді в полі JsonVal, де значення доступне в параметрі Base64ReceiptPdf.</p>

## 2. Відкриття зміни

Для відкриття касової зміни необхідно викликати метод **OpenShift** з параметром, у якому вказано **Фіскальний номер ПРРО**. Cashalot виконує запит щодо стану каси з обраним фіскальним номером, після чого створює XML повідомлення із призначеним локальним номером і типом «Відкриття зміни», засвідчує його кваліфікованим електронним підписом продавця і надсилає на ФСКО.

```
public string OpenShift( string rroFiscalNumber );
```

*Приклад:*

```
Object[] parameters = { "4000000000"/*фіскальний номер ПРРО*/};
var ret = CallCashalotMethod("OpenShift", parameters);
```

При успішному виконанні метод повертає структуру **CashalotApiResponse** у вигляді JSON, де буде вказаний ID зміни, дата завершення терміну дії сертифікату касира та ліцензії ПРРО. *Приклад:*

```
{"Ret":true,"ErrorString":"","Values":{"ShiftID":"e8353c00-f9eb-4b4c-aebc-ad0e55fb9109","LicEndDate":"2027-10-28T23:59:59","CertEndDate":"2027-05-11T23:59:59"}}
```

## 3. Службове внесення грошових коштів в касу

Для проведення операції службового внесення грошових коштів необхідно викликати метод **ServiceInput** з параметрами, де буде вказаний **ФН ПРРО** та **сума службового внесення**.

```
public string ServiceInput( string rroFiscalNumber, string sumVal );
```

#### Приклад:

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "150.25" /* сума службового внесення*/
};
var ret = CallCashaLotMethod( "ServiceInput", parameters );
```

Cashälot створює службове діалогове вікно для перевірки отриманих параметрів запиту і після підтвердження, відправляє відповідне XML повідомлення на сервер ФСКО.

При виконанні відкривається чек службового внесення та повертається [CashalotApiRsp](#).

#### 4. Службова видача грошових коштів з каси

Для проведення операції службової видачі грошових коштів необхідно викликати метод **ServiceOutput** з параметрами, де буде вказаний **ФН ПРРО та сума службової видачі**.

```
public string ServiceOutput( string rroFiscalNumber, string sumVal )
```

#### Приклад:

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "150.25" /* сума службової видачі*/
};
var ret = CallCashaLotMethod( "ServiceOutput", parameters );
```

Cashälot створює службове діалогове вікно для перевірки отриманих параметрів запиту і після підтвердження, відправляє відповідне XML повідомлення на сервер ФСКО.

При виконанні відкривається чек службової видачі та повертається [CashalotApiRsp](#).

#### 5. Фіскалізація чека продажу товарів, послуг

Для виконання розрахункової операції продажу товарів або послуг необхідно викликати метод фіскалізації чека **FiscalizeCheck**, в якості параметрів якого передаються ФН каси та два JSON-рядки: **strJSONData** для списку товарів/послуг та **payDataJSON** для повної вартості по чеку.

```
public string FiscalizeCheck( string rroFiscalNumber, string strJSONData, string payDataJSON );
```

#### Опис **strJSONData**:

Структура містить масив **ReceiptLst**, кожен елемент якого містить параметри номенклатури та об'єкти **Comment**, **DocType**, **SaleOrderNumber**.

#### Опис **ReceiptLst**:

Поле	Значення
VendorCode*	Артикул.
Name	Найменування товару.
DocNumber**	Номер договору (при використанні ломбардних послуг).

CdUKDZED	Код УКТЗЕД.
CdDKPP	Код ДКПП.
GoodsType	Тип номенклатури. За замовчуванням товар, якщо параметр не задано. 1 – товар; 2 – послуга; 3 – ломбардна послуга **.
Barcode	Штрих-код.
UnitType*	Одиниця вимірювання.
Quantity*	Кількість товару.
Price*	Ціна одиниці товару (в форматі <Гривні, Копійки>).
Amount*	Кінцева сума по позиції чека (з урахуванням всіх націнок та знижок, у форматі <Гривні, Копійки>).
DiscountPrc	Відсоток знижки. <b>Увага!</b> При заповненні DiscountPrc, необхідно також обов'язково заповнювати та передавати DiscountSum
DiscountSum	Сума знижки на позицію.
VATRate	ПДВ у відсотках (від 0 до 99,99).
VATLetter	Літера ставки ПДВ.
IsPriceIncludeVAT	Для випадків, коли сума по позиції не містить суму ПДВ (true/false).
SumVAT	Сума ПДВ за одиницю товару (ігнорується, якщо вказати ознаку, що враховує ПДВ "true").
IsExcise	Ознака підакцизного товару(true/false).
ExciseLetter	Літера ставки акцизного податку.
ExciseStampBarcode	Штрих-код марки акцизного податку. У випадку коли товарів більше ніж 1, по одній позиції номенклатури, то всі штрих-коди заповнюються через кому. Поле ігнорується, якщо ознака підакцизного товару IsExcise = false.
OtherParametrs	Додаткові параметри у вигляді XML таблиці.

\* обов'язкові параметри при передачі даних товару.

\*\* для [Ломбардних операцій](#).

**Зауваження!** Для передачі кількості товару з дробовою частиною (зокрема, нульовою, наприклад, "2,00") допускається використовувати десятковий розділювач "." (крапка) або "," (кома). Якщо параметр не задано, або задано некоректно, його значення приймається рівним одиниці (1).

Параметр **OtherParametrs** передається в форматі XML з кодуванням UTF-8. Атрибути XML-рядка вказані в таблиці нижче.

Зазначимо, що параметри XML-рядка **OtherParametrs** мають пріоритет над параметрами, які передані в **ReceiptLst**.

Атрибути XML-рядка **OtherParametrs**:

Назва атрибута	Опис атрибута
Nomenclature *	Найменування товару.
UKTZED	Код УКТЗЕД.
VendorCode *	Артикул.

Barcode	Штрих-код.
Dimension *	Одиниця вимірювання.
Discount	Відсоток знижки. <b>Увага!</b> При заповненні Discount, необхідно також обов'язково заповнювати та передавати DiscountAmount
DiscountAmount	Сума знижки.
VAT *	Ставка ПДВ в відсотках.
VATLetter	Літера ставки ПДВ.
PriceIncludesVAT або PriceIncludeVAT	Для випадків, коли сума по позиції не містить суму ПДВ (true/false).
AmountVAT	Сума ПДВ за одиницю товару, у форматі <Гривні, Копійки>.
IsExcisable	Ознака підакцизного товару, True/False.
ExciseLetter	Літера ставки акцизного податку.
ExciseStampBarcode	Штрих-код марки акцизного податку. У випадку коли товарів більше ніж 1, по одній позиції номенклатури, то всі штрих-коди заповнюються через кому. Поле ігнорується, якщо ознака підакцизного товару <b>IsExcise</b> = false.

\* обов'язкові параметри в рядку [OtherParams](#), якщо не вказані відповідні явні параметри.

Приклад [OtherParams](#):

```
<?xml version="1.0" encoding="UTF-8"?> <Table> <Record VAT="5" IsExcisable="True" Nomenclature="Товар1" UKTZED="0202" VendorCode="1" Barcode="1452" Dimension="шт" Discount="0.54" DiscountAmount="0.25" PriceIncludeVAT="1" AmountVAT="0" ExciseStampBarcode="11111, 22222, 33333"/> </Table>
```

Об'єкт **Comment** використовується для додавання нефіскальної інформації:

Comment	Опціональний рядок нефіскальної інформації, коментар. Допускається використання одного коментаря на чек.
---------	---

Об'єкт **DocType** є обов'язковим при проведенні ломбардних операцій, призначений для визначення типу документа (чека). [Ломбардні операції](#).

DocType	Тип документа: 0 – чек продажу; 4 – ломбардний чек; За замовчуванням DocType = 0, якщо об'єкт не задано.
---------	---

Об'єкт **SaleOrderNumber** (Номер замовлення/договору) потрібен для заповнення номера договору при проведенні інтернет продажу, передоплати та післяплати. Для відображення номера замовлення в післяплаті обов'язково потрібно заповнювати в [payDataJSON](#) фіскальний номер чека передоплати [ParentReceiptFiscalNumber](#).

Опис [payDataJSON](#):

Поле	Значення
SumCash	Сума наданої готівки.
SumPayByCard	Сума оплати чека банківською картою.
SumPayByBank	Сума оплати безготівкою

SumPayByCredit	Сума оплати чека в кредит.
SumPayByCertificate	Сума оплати чека сертифікатом.
SumPayCheck	Сума чека до оплати з урахуванням всіх надбавок, знижок та округлень.
PaymentOrderType	Встановлення порядку сплати: 0 - звичайна; 1 - передплата; 2 - післяплата; 3 - інтернет продаж; 4 - повна післяплата; 5 - оплата частинами (передплата); 6 - оплата частинами (чергова сплата); 7 - оплата частинами (остаточний розрахунок).
SumPreparePayed	Сума попередньої оплати.
PartPaySum	Сума попередньої оплати або чергової сплати при використанні способу оплати «Оплата частинами». Сума остаточного розрахунку не потребує заповнення, її розрахунок відбувається автоматично на основі залишків до сплати по ланцюжку чеків зі способом оплати «Оплата частинами».
ParentReceiptFiscalNumber	Фіскальний номер чека, по якому здійснюється доплата/повернення.
ParentRROFiscalNumber	Фіскальний номер РРО, по якому здійснюється доплата/повернення.
ParentReceiptOrderDate	Дата продажу чека по якому здійснюється повернення (формат: дд.мм.рррр). Використовується тільки для чеків продажів.
CustomerEmail	Опціонально. Електронна адреса покупця, на яку можна відправити фіскалізований чек, передається в поле електронної адреси при натисканні кнопки "Надіслати чек на Email" екрану відображення чека.
CustomerSmsNumber	Номер телефону покупця на який можна відправити фіскалізований чек у вигляді SMS повідомлення. Передається в поле номера телефону при натисканні кнопки «Надіслати на SMS».
CustomerViberNumber	Номер телефону покупця на який можна відправити фіскалізований чек у вигляді повідомлення у Viber. Передається в поле номера телефону при натисканні кнопки «Надіслати на Viber».
AutosendCustomerEmail	Ознака автоматичної відправки чека покупцю на E-mail після його реєстрації. Параметр може приймати значення true або false. Відправка чека буде здійснена якщо передано заповнений параметр CustomerEmail, виконано налаштування електронної пошти в РМК в розділі «Налаштування - Загальні налаштування - Господарська одиниця - Налаштування пошти» та встановлене значення true для AutosendCustomerEmail

AutosendCustomerSms	Ознака автоматичної відправки чека покупцю на SMS після його реєстрації. Параметр може приймати значення true або false. Відправка чека буде здійснена якщо передано заповнений параметр <b>CustomerSmsNumber</b> , виконано налаштування сервісу доставки SMS в кабінеті користувача Cashalot ( <a href="https://my.cashalot.org.ua/">https://my.cashalot.org.ua/</a> ) та встановлено значення true для <b>AutosendCustomerSms</b>
AutosendCustomerViber	Ознака автоматичної відправки чека покупцю на SMS після його реєстрації. Параметр може приймати значення true або false. Відправка чека буде здійснена якщо передано заповнений параметр <b>CustomerViberNumber</b> , виконано налаштування сервісу доставки повідомлень у Viber в кабінеті користувача Cashalot ( <a href="https://my.cashalot.org.ua/">https://my.cashalot.org.ua/</a> ) та встановлено значення true для <b>AutosendCustomerViber</b>
OtherParametrs	Додаткові параметри у вигляді XML таблиці.
BankPayToolsNM	Найменування засобу оплати при використанні способу оплати 0,1,2,5, 6 та 7
PrePaymentPayToolsNM	Найменування засобу оплати при використанні способу оплати 3
AfterPaymentPayToolsNM	Найменування засобу оплати при використанні способу оплати 3,4
UseTransactionDetailsToSumPayByBank	Ознака (true/false) яка дозволяє застосувати реквізити транзакції POS-терміналу до безготівкового засобу оплати (Платіж Mono Pay, Платіж LIQPAY, Платіж WAYFORPAY) використаних в способах оплати 0,1,2,5, 6 та 7
PaymentFormsLst	Масив який містить інформацію про засоби оплати по чеку які передаються у вигляді окремих масивів. Масив з вказаними значеннями по засобу оплати повинен містити: код засобу оплати, суму безготівкової оплати, найменування засобу оплати та ознаку (true/false) застосування реквізитів транзакції POS-терміналу до використаного засобу оплати. Важливо зберігати саме таку послідовність заповнення (див. приклад застосування). Реквізити транзакції можливо застосувати до засобів оплати: Платіж LIQPAY, Платіж Mono pay, Платіж WAYFORPAY.
ReceiptID	Параметр призначений для можливості контролю дублюючої передачі чека на реєстрацію. Значення параметру містить GUID контроль за генерацією та використанням котрого необхідно реалізувати на рівні обліково-розрахункової системи користувача. В разі повторної передачі значення GUID чека який вже був зареєстрований, API поверне відповідь на раніше сформований чек та відобразить його візуальну форму.
ExpressWaybillNumber	Номер експрес-накладно (ТТН)
<b>Необов'язкові параметри інформації по транзакції терміналу при безготівкових розрахунках:</b>	
TerminalID	Код терміналу.
ApprovalCode	Код підтвердження.
RRN	РРН транзакції.
IssuerName	Платіжна система.
PAN	Номер картки.
TransactionDate	Дата транзакції (в форматі dd.mm.yyyy hh:mm:ss).

SignVerif	Електронний підпис.
AcquireName	Термінал.
InvoiceNumber	Номер чека.
ParentRRN	RRN транзакції чека, по якому здійснено повернення.
AcquireID	Ідентифікатор еквайра

**Увага!** Різниця між сумою оплати чека та [SumPayCheck](#) (сумою по всіх видах оплати) вважається значенням заокруглення. Тому розробник має самостійно реалізувати алгоритм округлення значення виданої готівки або суми оплати по чеку.

**Заокруглення суми** по чеку реалізовано наступним чином:

В [payDataJSON](#) передається значення готівки, що надана клієнтом на касі, значення оплати картою, сертифікатом та кредитом, а також заокруглена сума, розрахована за певним алгоритмом користувача [SumPayCheck](#) (відповідно до [Постанови НБУ №115](#)):

- якщо сума закінчується від 1 до 24 копійок – здійснювати заокруглення до 00 копійок, в бік зменшення;
- якщо закінчуються від 75 до 99 копійок – здійснювати заокруглення до 00 копійок, в бік збільшення;
- якщо сума закінчується від 51 до 74 копійок - здійснювати заокруглення до 50 копійок, в бік зменшення
- якщо сума закінчується від 25 до 49 копійок - здійснювати заокруглення до 50 копійок, в бік збільшення.

В чеку буде відображено заокруглення, надану готівку та інші форми оплати, заокруглену суму до сплати, решту.

Значення заокруглення, яке буде зареєстроване на сервері ФСКО та відображено в чеку, розраховується по формулі:

**Сума заокруглення = [SumPayCheck](#) - Загальна сума товарів**

Загальна сума товарів – це сума рядків [Amount](#), вказаних в [strJSONData](#). Тобто:

**Загальна сума товарів = Сума товару 1(Amount) + Сума товару 2(Amount) + ...**

**Решта** по чеку розраховується таким чином:

**Сума решти = [SumCash](#) + [SumPayByCard](#) + [SumPayByBank](#) + [SumPayByCredit](#) + [SumPayByCertificate](#) – [SumPayCheck](#)**

Решта розраховується програмою Cashlot автоматично, згідно вказаних користувачем даних.

### **Приклад використання засобів оплат в чеках**

В разі потреби відображення в чеку засобів оплат таких як «Платіж Monopay», «Переказ з картки», «Платіж EasyPay» або інших, наявно два варіанти їх відображення та використання - спрощений та розширений. Засоби оплати можливо використовувати для всіх способів оплат по чеку (продаж, передплата/післяплата, інтернет-продаж, тощо).

Стандартні засоби оплат (Готівка (Готівкові кошти), Картка, Сертифікат, Кредит) заповнюються стандартно відповідно до опису встановлення порядку оплати в [payDataJSON](#).

**Примітка!** Описаний функціонал є виключно доповненням до існуючого функціоналу налаштування та застосування оплат по чеку. Таким чином новий функціонал не змінює та не порушує

роботу раніше реалізованого механізму оплати, а працює як окреме, самостійне розширення. Його впровадження є повністю зворотно сумісним із поточними інтеграціями.

### Варіант використання засобів оплат №1 (спрощений)

Для налаштування спрощеного варіанту використання засобів оплати у чеку слід використовувати наступний перелік параметрів: [SumPayByBank](#), [BankPayToolsNM](#), [PrePaymentPayToolsNM](#), [AfterPaymentPayToolsNM](#) та [UseTransactionDetailsToSumPayByBank](#). Спрощений варіант надає змогу використовувати лише один засіб оплати у чеку.

За допомогою параметру [SumPayByBank](#) передається сума для безготівкового засобу оплати за умови, що використовується спосіб оплати 0, 1, 2, 5, 6 або 7. Залежно від обраного способу оплати по чеку, для передачі назви засобу оплати слід використовувати один з атрибутів: [BankPayToolsNM](#), [PrePaymentPayToolsNM](#) або [AfterPaymentPayToolsNM](#). Використання параметру [BankPayToolsNM](#) дозволяється у випадку застосування способів оплати 0, 1, 2, 5, 6 або 7. Параметр [PrePaymentPayToolsNM](#) можна використовувати, якщо застосовується спосіб оплати 3, а [AfterPaymentPayToolsNM](#) — якщо використовується спосіб оплати 3 або 4.

Як найменування засобів оплати радимо використовувати наступний [перелік](#) готових варіантів. Якщо в переліку відсутня потрібна назва, користувач може ввести власну. Створення власного засобу оплати відбувається під час його першого використання у чеку. Для успішного створення, на момент реєстрації такого чека необхідно мати доступ до кабінету користувача Cashalot.

За потреби додавання параметрів транзакції POS-терміналу до засобу оплати, вказаного у [BankPayToolsNM](#), необхідно використовувати параметр [UseTransactionDetailsToSumPayByBank](#). Опція доступна лише за умови, що використовується один із таких засобів оплати: Платіж Mono Pay, Платіж LIQPAY або Платіж WAYFORPAY. Якщо було використано інший засіб оплати (наприклад, Інтернет-еквайринг) разом із [UseTransactionDetailsToSumPayByBank](#), то параметри транзакції POS-терміналу не будуть відображені у чеку.

**Важливо!** Параметр [UseTransactionDetailsToSumPayByBank](#) надає пріоритет параметрам транзакції, які застосовуються до використаного засобу оплати. У такому випадку, якщо разом із засобом оплати (наприклад, Платіж LIQPAY) також використано засіб оплати Картка (SumPayByCard), то параметри транзакції буде відображено у чеку саме під засобом Платіж LIQPAY

### Варіант використання засобів оплат №2 (розширений)

Розширений варіант використання засобів оплат передбачає заповнення параметра PaymentFormsLst в структурі оплат payDataJSON. Параметр є масивом який містить необхідну кількість засобів оплат заповнених у вигляді окремих масивів. За допомогою розширеного варіанту можливо передати декілька засобів оплат одночасно.

**Приклад payDataJSON з заповненим PaymentFormsLst :**

```
payDataJSON = {"PaymentOrderType":0,"SumCash":"","SumPayCheck":"","100","PaymentFormsLst":[
  [
    "202",
    "50",
    "Платіж Mono Pay",
    "True"
  ],
  [
    "202",
    "25",
```

"Переказ з поточного рахунку"

]  
]  
}"

де **202** - це код засобу оплати;

**50** - сума засобу оплати;

**Платіж Portmone** - назва засобу оплати.

**True** - ознака застосування реквізитів POS-терміналу до використаного засобу оплати.

Перелік рекомендованих засобів оплати які доступні до використання для способів оплати 0,1,2,5,6 та 7:

Код засобу оплати	Найменування засобу оплати
202	Платіж Mono Pay
202	Платіж LIQPAY
202	Платіж WAYFORPAY
202	Інтернет еквайринг
202	Переказ з картки
202	Переказ через ННПП
202	Переказ через ПТКС ННПП
202	Переказ через ПТКС банку
202	Переказ через ННПП NovaPay
202	Інтернет банкінг
202	Переказ з поточного рахунку
202	Платіж mono checkout
202	Платіж Portmone
202	Платіж EasyPay
202	Переказ через ННФПП
202	Платіж через сервіс переказу ROZETKA Pay

В рамках одного фіскального чека допускається використання засобів оплат з дублюючим значенням коду але за умови, що найменування засобу оплати заповнено відмінне від значення попереднього засобу. В разі відсутності потрібного найменування засобу оплати користувач може використати необхідну власну назву. Створення власного засобу оплати відбувається під час його першого використання у чеку. Для успішного створення, на момент реєстрації такого чека необхідно мати доступ до кабінету користувача Cashalot.

За потреби додавання параметрів транзакції POS-терміналу до засобу оплати, вказаного в масиві, необхідно встановити ознаку True. Опція доступна лише за умови, що найменування засобу

оплати вказано як: Платіж Mono Pay, Платіж LIQPAY або Платіж WAYFORPAY. Якщо було використано інший засіб оплати (наприклад, Інтернет-еквайринг) разом із ознакою True, то параметри транзакції POS-терміналу не будуть відображені у чеку. Якщо ознака не встановлена (не використовується), то це означає, що використовується його ознака False.

**Важливо!** Ознака використання реквізитів транзакції POS-терміналу надає пріоритет параметрам транзакції, які застосовуються до використаного засобу оплати. У такому випадку, якщо разом із засобом оплати (наприклад, Платіж LIQPAY) також використано засіб оплати Картка (SumPayByCard), то параметри транзакції буде відображено у чеку саме під засобом Платіж LIQPAY. Встановлення ознаки доступно тільки для засобів оплат які використовуються в способах оплати 0,1,2,5,6 та 7.

Перелік рекомендованих засобів оплати які доступні до використання для способів оплати 3 та 4:

Код засобу оплати	Найменування засобу оплати
2 або 201*	Платіж Mono Pay
2 або 201*	Платіж LIQPAY
2 або 201*	Платіж WAYFORPAY
2 або 201*	Інтернет еквайринг
2 або 201*	Переказ з картки
2 або 201*	Переказ через ННПП
2 або 201*	Переказ через ПТКС ННПП
2 або 201*	Переказ через ПТКС банку
2 або 201*	Переказ через ННПП NovaPay
2 або 201*	Інтернет банкінг
2 або 201*	Переказ з поточного рахунку
2 або 201*	Платіж mono checkout
2 або 201*	Платіж Portmone
2 або 201*	Платіж EasyPay
2 або 201*	Переказ через ННФПП
2 або 201*	Платіж через сервіс переказу ROZETKA Pay

## Приклади використання та заповнення чеку

Приклад виконання методу *FiscalizeCheck*:

```
private void btnFiscalizeReceipt_Click( object sender, EventArgs e )
{
    //Майже всі поля задаються як строки(для сумісності з різними мовами програмування, щоб
уникнути проблем з конвертацією)
    LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
    lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();
    CheckInfo chInfo = new CheckInfo();
    chInfo.VendorCode = "АртикулТовару1";
    chInfo.Name = "НазваТовару1";
    chInfo.CdUKDZED = String.Empty;
    chInfo.Barcode = "123456789";
    chInfo.UnitType = "шт";
    chInfo.Quantity = "1";
    chInfo.Price = "20000.03";
    chInfo.Amount = "19000.03";
    chInfo.DiscountPrc = null;
    chInfo.DiscountSum = "-1";
    chInfo.IsPriceIncludeVAT = true;
    chInfo.VATRate = "20";
    chInfo.SumVAT = "3800.01";
    chInfo.IsExcise = true;
    chInfo.ExciseStampBarcode = "AFFD6723344";
    //chInfo.OtherParameters = "ХМЛ структура реквізитів, якщо реквізит не заповнено в
основній структурі, та заповнено в ХМЛ, значення беруться з ХМЛ"
    //chInfo.OtherParameters = "<?xmlversion=\\"1.0\\"encoding=\\"UTF-8\\"?><Table><Record
VAT=\\"20\\" IsExcisable=\\"False\\" Nomenclature=\\"Товар1\\" UKTZED=\\"000001\\" VendorCode=\\"79078\\"
Barcode=\\"1234567890\\" Dimension=\\"шт\\" Discountamount=\\"1.83\\" PriceIncludeVAT=\\"1\\"
AmountVAT=\\"30.2\\"/></Table>";

    //додати рядок чека з товаром №1
    lstGoodsInCheck.ReceiptLst.Add( chInfo );

    CheckInfo chInfo2 = new CheckInfo( chInfo );
    chInfo2.VendorCode = "АртикулТовару2";
    chInfo2.Name = "НазваТовару2";
    chInfo2.Barcode = "9999999999";
    chInfo2.VATRate = null;
    chInfo2.SumVAT = null;
    chInfo2.ExciseStampBarcode = "AFFD55223344,AAFF667788";
    chInfo2.DiscountSum = "2";
    //chInfo2.OtherParameters = "<?xmlversion=\\"1.0\\"encoding=\\"UTF-8\\"?><Table><Record
VAT=\\"20\\" IsExcisable=\\"False\\" Nomenclature=\\"Товар2\\" UKTZED=\\"000001\\" VendorCode=\\"79078\\"
Barcode=\\"1234567890\\" Dimension=\\"шт\\" Discountamount=\\"1.83\\" PriceIncludeVAT=\\"1\\"
AmountVAT=\\"30.2\\"/></Table>";
    //додати строку чеку з товаром №2
    lstGoodsInCheck.ReceiptLst.Add( chInfo2 );

    PayInfoRsp payInfo = new PayInfoRsp();
    payInfo.SumCash = "20000"; //Сума готівки наданої покупцем( може бути більшою за
сумму до сплати, з неї вираховується решта/здача)

    payInfo.SumPayByCard = "19000";//Сума оплати картою

    //-----{
    // Інформація по транзакції терміналу:
    //Код терміналу
```

```

payInfo.TerminalID = "S1K700J8";
//Код підтвердження
payInfo.ApprovalCode = "68014B";
//PPH транзакції
payInfo.RRN = "048867173620";
//Платіжна система
payInfo.IssuerName = "MASTERCARD";
//Номер картки
payInfo.PAN = "XXXXXXXXXXXX6883";
//Дата транзакції(в форматі dd.ММ.уууу HH:mm:ss)
payInfo.TransactionDate = DateTime.Now.ToString( "dd.ММ.уууу HH:mm:ss" );
//Електроний підпис
//payInfo.SignVerif="12334444";
//Термінал
payInfo.AcquireName = "Приват Банк";
//Номер чека
payInfo.InvoiceNumber = "224";
//-----}

payInfo.SumPayCheck = "38000.10";//Сума оплати загальна з урахуванням заокруглення
(+0,04 грн)
payInfo.CustomerEmail = edCustomerEmail.Text;
//Конвертація структур до формату джейсон строки
string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
string payDataJSON = JsonConvert.SerializeObject( payInfo );

//payDataJSON =
"{\"SumCash\": \"240\", \"SumPayByCard\": \"0\", \"SumPayByCredit\": \"0\", \"SumPayByCertificate\": \"0\", \"SumPayCheck\": \"239,64\"}";
//Параметри функції
Object[] parameters =
{
    edFiscalNumberRRO.Text.Trim(), // фіскальний номер PPO
    strJSONData, // структура інформації зі строками чека
    payDataJSON // структура інформації про оплату по чеку
};

var ret = CallCashLotMethod( "FiscalizeCheck", parameters );
if ( ret?.Ret != true )
{
    MessageBox.Show( String.IsNullOrEmpty( ret?.ErrorString ) ? "Не вдалося
фіскалізувати чек" : ret?.ErrorString );
}
else if ( ret?.Values != null )
{
    //отримання параметра в результаті виконання функції
    var vals = ret.Values.ToArray();
    //перебираємо параметри
    for ( int i = 0; i < vals.Length; i++ )
    {
        // отримуємо значення по ключу
        if ( vals[i].Key.ToLower() == "receiptlocalnumber" )
        {
            edLocalCheckNum.Text = vals[i].Value;
        }
        else
        if ( vals[i].Key.ToLower() == "receiptfiscalnumber" )
        {
            edFiscalCheckNum.Text = vals[i].Value;
        }
    }
}

```

```

    }
    MessageBox.Show( $"Чек було успішно фіскалізовано.\n" +
        $"Чеку присвоєно локальний номер:{edLocalCheckNum.Text},\n" +
        $"та фіскальний номер: {edFiscalCheckNum.Text} "
    );
}
}

```

У відповіді на **FiscalizeCheck** повертається **CashalotApiRsp**, в якому заповнений масив параметрів **Values**, що містить:

ReceiptFiscalNumber	Фіскальний номер чека.
ReceiptLocalNumber	Локальний номер чека.
ShiftID	Ідентифікаційний номер поточної зміни.
OfflineMode	Ознака офлайн режиму.
Type	<a href="#">Тип документа.</a>
FSKORceiptLink	Посилання на перегляд чека в ДПС
CashalotReceiptLink	Посилання на перегляд чека в кабінеті Cashalot
ReceiptID	ID (guid) чека
ReceiptSum	Сума до сплати по чеку
ReceiptDateTime	Дата та час створення у форматі (дд.мм.рррр гг:хв:сс)
Base64Str1251ReceiptXML	XML-структура фіскалізованого чека в кодуванні Base64.
ReceiptPrePayOrderNum	Номер інтернет-замовлення (для чеків інтернет-продажу або передплати/післяплати)
Base64ReceiptJpeg	JPEG структура фіскалізованого чека у форматі base64
Base64ReceiptPdf	PDF структура фіскалізованого чека у форматі base64

Після виконання запиту Cashalot створює діалогове вікно з графічним відображенням результатів реєстрації продажу у вигляді чека.

Тип документа **"Type"** визначається числом, яке формується наступним чином:

**1<Клас документа><Тип документа><Додатковий тип документа>**

Опис **"Type"**:

Число	Кількість знаків	Значення
<Клас документа>	1	0 - Чек 1 - Z-звіт 2 - X-звіт
<Тип документа>	3	000 - Чек реалізації товарів/послуг 001 - Чек переказу коштів 002 - Чек операції обміну валюти 003 - Чек видачі готівки 100 - Відкриття зміни 101 - Закриття зміни 102 - Початок офлайн сесії 103 - Завершення офлайн сесії
<Додатковий тип документа>	3	000 - Касовий чек (реалізація) 001 - Видатковий чек (повернення) 002 - Чек операції «службове внесення»/«отримання авансу» 003 - Чек операції «отримання підкріплення» 004 - Чек операції «службова видача»/«інкасація»

## 6. Проведення операції повернення товару

Для виконання повернення товару чи товарів з фіскального чека з відомим (визначеним) фіскальним номером використовується метод **FiscalizeReturnCheck**, для якого формується чек повернення аналогічно операції [фіскалізації чека продажу товарів, послуг](#) із зазначенням тих товарів, які реєструються для повернення. Тобто, створюються **strJSONData** для списку товарів повернення, **payDataJSON** для повної вартості покупок та додатково зазначається параметр з заповненим **ФН чека, по якому здійснюється повернення**:

Параметр	Значення	Приклад	Примітка
<b>strFiscalNumReturnCheck</b>	Фіскальний номер чека, по якому здійснюється повернення	"275301" "19255.3.7284"	онлайн-чек офлайн-чек

```
public string FiscalizeReturnCheck( string rroFiscalNumber, string strJSONData, string payDataJSON,
string strFiscalNumReturnCheck)
```

### Приклад:

```
// Відправка чека повернення
LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();
CheckInfo chInfo = new CheckInfo();
chInfo.VendorCode = "АртикулТовару1";
chInfo.Name = "НазваТовару1";
chInfo.CdUKDZED = String.Empty;
chInfo.Barcode = "123456789";
chInfo.UnitType = "шт";
chInfo.Quantity = "1";
chInfo.Price = "183";
chInfo.Amount = "181,17";
chInfo.DiscountPrc = String.Empty;
chInfo.DiscountSum = "1.83";
chInfo.IsPriceIncludeVAT = true;
chInfo.VATRate = "20";
chInfo.SumVAT = "30.2";
lstGoodsInCheck.ReceiptLst.Add( chInfo );
CheckInfo chInfo = new CheckInfo();
chInfo.VendorCode = "АртикулТовару2";
chInfo.Name = "НазваТовару2";
chInfo.Barcode = "9999999999";
lstGoodsInCheck.ReceiptLst.Add( chInfo );
PayInfoRsp payInfo = new PayInfoRsp();
payInfo.SumCash = "366";
payInfo.SumPayCheck = "362,34";
string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
string payDataJSON = JsonConvert.SerializeObject( payInfo );
Object[] parametersChRt =
{
    edFiscalNumberRRO.Text.Trim(),
    strJSONData,
    payDataJSON,
    edFiscalCheckNum.Text /*фіскальний номер чека, по якому здійснюється повернення*/
};
var retChRt = CallCashalotMethod( "FiscalizeReturnCheck", parametersChRt );
if ( !retChRt.Ret )
{
```

```
MessageBox.Show( String.IsNullOrEmpty( retChRt.ErrorString ) ? "Не вдалося здійснити повернення" : retChRt.ErrorString );  
}
```

Після виконання методу `CashÄlot` створює діалогове вікно з графічним відображенням результатів реєстрації повернення товару у вигляді чека та повертається [CashalotApiRsp](#).

## 7. Визначення фіскального номера чека

З метою визначення ФН чека, зареєстрованого раніше в ФСКО, за локальним номером, що зареєстрований в касі при продажу, виконується метод `GetReceiptFiscalNumberByLocalNumber` з параметрами, де буде вказаний **ФН ПРРО** та **локальний номер чека**.

```
public string GetReceiptFiscalNumberByLocalNumber(string rroFiscalNumber,string checkLocalNumber);
```

*Приклад:*

```
Object[] parameters = {  
    "4000000000", /* фіскальний номер ПРРО*/  
    "1554" /* локальний номер чека*/  
};  
var ret = CallCashaLotMethod( "GetReceiptFiscalNumberByLocalNumber", parameters );
```

Визначений фіскальний номер чека використовується, зокрема, в якості додаткового параметра при фіскалізації чека повернення товару.

---

## 8. Визначення локального номера чека по фіскальному номеру

В деяких випадках виникає необхідність визначення локального номера чека, зареєстрованого в касі, по відомому фіскальному номеру чека. Для визначення локального номера чека по ФН використовується метод `GetReceiptLocalNumberByFiscalNumber` з параметрами, де заповнені **Фіскальний номер ПРРО** та **ФН чека**.

```
public string GetReceiptFiscalNumberByLocalNumber(string rroFiscalNumber,string checkLocalNumber);
```

*Приклад:*

```
Object[] parameters = {  
    "4000000000", /* фіскальний номер ПРРО*/  
    "1234567" /* фіскальний номер чеку*/  
};  
var ret = CallCashaLotMethod( "GetReceiptLocalNumberByFiscalNumber", parameters);
```

## 9. Отримання XML-структури чека

Для отримання XML-структури фіскалізованого чека використовується метод `GetReceiptXML` з параметрами, де заповнені **Фіскальний номер ПРРО** та **ФН чека**.

```
public string GetReceiptXML(string rroFiscalNumber, string checkFiscalNumber );
```

*Приклад:*

```
Object[] parameters = {  
    "4000000000", /* фіскальний номер ПРРО*/  
    "1234567" /* фіскальний номер чеку*/  
};  
var ret = CallCashaLotMethod( "GetReceiptXML", parameters );
```

Після виконання методу Cashalot повертається **CashalotApiRsp**, в якому заповнений масив параметрів **Values**, що містить заповнений параметр **Base64Str1251ReceiptXML**. Дані в ньому закодовані в Base64.

---

## 10. Відображення фіскалізованого чека

У випадку коли після фіскалізації чека з'являється необхідність відобразити на екрані його повторно, наприклад для друку, використовується метод **ShowReceipt** з заповненими параметрами **Фіскальний номер ПРРО** та **ФН чека**.

```
public string ShowReceipt(string rroFiscalNumber, string checkFiscalNumber );
```

*Приклад виконання:*

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "1234567"     /* фіскальний номер чеку*/
};
var ret = CallCashaLotMethod( "ShowReceipt", parameters );
```

---

## 11. Створення X-звіту

Для створення X-звіту необхідно викликати метод **ShowXReport** з параметром, в якому вказано **Фіскальний номер ПРРО**.

```
public string ShowXReport( string rroFiscalNumber );
```

*Приклад:*

```
Object[] parameters = {"4000000000", /* фіскальний номер ПРРО*/};
var ret = CallCashaLotMethod( "ShowXReport", parameters );
```

Після виконання методу Cashalot повертається **CashalotApiRsp**, в якому заповнений масив параметрів **Values**. В результаті виконання операції створюється X-звіт та відображається на екрані, після чого його можна роздрукувати.

---

## 12. Закриття зміни та створення Z-звіту

Для закриття поточної зміни використовується метод **CloseShift**. В параметрах **CloseShift** має бути заповнено тільки **ФН ПРРО**.

**Увага!** Якщо встановлений параметр **NOINTERFACEMODE = True**, то службову видачу потрібно передбачити перед виконанням методу **CloseShift**.

```
public string CloseShift( string rroFiscalNumber );
```

#### Приклад виконання:

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
};
var ret = CallCashaLotMethod( "CloseShift", parameters );
```

Перед створенням Z-звіту та закриттям зміни, описаним в цьому розділі, автоматично виконується метод [службового вилучення наявних грошових коштів з каси](#). Також виконується графічне відображення Z-звіту для можливості подальшого друку.

---

### 13. Ручний перевод ПРРО в режим офлайн

Для ручного переводу ПРРО в режим офлайн використовується метод [SetOfflineMode](#). В якості параметрів використовується **Фіскальний номер ПРРО** та **Ознака автоматичного виходу в онлайн**(true - автоматично виходити з офлайну при наявності інтернету. false - виходити з офлайну лише при виклику методу [TryGoToOnlineMode](#) або після закриття зміни).

```
public string SetOfflineMode(string rroFiscalNumber, bool autoExitToOnline);
```

#### Приклад виконання:

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    false, /* true - автоматично виходити з офлайну при наявності інтернету. false -
виходити з офлайну лише при виклику методу TryGoToOnlineMode або після закриття зміни*/
};
var ret = CallCashaLotMethod( "SetOfflineMode", parameters );
```

Виклик [SetOfflineMode](#) не означає автоматичний перехід в офлайн одразу в момент виводу, перехід до офлайну буде здійснено при будь-якій фіскальній операції(відкриття/закриття зміни, службовій видачі/внесенню або реєстрації чека) після виводу [SetOfflineMode](#).

В якості відповіді повертається стандартна структура параметрів повернення [CashalotApiRsp](#) з заповненим параметром Return(Ознака успішності виконання).

---

### 14. Ручний перевод ПРРО в режим онлайн

Для ручного переводу ПРРО в режим онлайн використовується метод [TryGoToOnlineMode](#).

```
public string TryGoToOnlineMode( string rroFiscalNumber );
```

#### Приклад виконання:

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
};
var ret = CallCashaLotMethod( "TryGoToOnlineMode", parameters );
```

В якості відповіді повертається стандартна структура параметрів повернення [CashalotApiRsp](#) з заповненим параметром Return(Ознака успішності виконання).

---

## 15. Синхронізація залишків по товарам

Метод призначений для ручної синхронізації залишків по товарам з кабінету користувача в локальну базу даних Cashalot. Для можливості виконання методу необхідно встановити [параметр ведення обліку товарів в API](#) в режимі 1 або 2 та ознаку ведення обліку товарів в кабінеті користувача в розділі «Склад - Налаштування - Облік товарів у ПРРО».

```
public string SyncBalanceOnGoods( string rroFiscalNumber );
```

**Приклад виконання:**

```
Object[] parameters = {  
    "4000000000", /* фіскальний номер ПРРО*/  
};  
var ret = CallCashaLotMethod( "SyncBalanceOnGoods", parameters );
```

В якості відповіді повертається стандартна структура параметрів повернення CashalotApiRsp з заповненим параметром Return(Ознака успішності виконання).

---

## 16. Синхронізація залишків та товарів

Метод призначений для ручної синхронізації товарів та залишків по ним з кабінету користувача в локальну базу даних Cashalot. Для можливості виконання методу необхідно встановити [параметр ведення обліку товарів в API](#) в режимі 1 або 2 та ознаку ведення обліку товарів в кабінеті користувача в розділі «Склад - Налаштування - Облік товарів у ПРРО».

```
public string SyncGoodsWithBalance( string rroFiscalNumber );
```

**Приклад виконання:**

```
Object[] parameters = {  
    "4000000000", /* фіскальний номер ПРРО*/  
};  
var ret = CallCashaLotMethod( "SyncGoodsWithBalance", parameters );
```

В якості відповіді повертається стандартна структура параметрів повернення CashalotApiRsp з заповненим параметром Return(Ознака успішності виконання).

---

## 17. Сторнування чека продажу/повернення

Метод призначений для сторнування зареєстрованого чека продажу, чека повернення, службового внесення та службової видачі за умови, що чек є зареєстрованим останнім за датою та часом у відкритій зміні.

```
public string StornoCheck ( string rroFiscalNumber )
```

#### Приклад виконання:

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
};
var ret = CallCashaLotMethod( "StornoCheck", parameters );
```

У відповіді на **StornoCheck** повертається **CashalotApiRsp**, в якому заповнений масив параметрів

**Values**, що містить:

ReceiptFiscalNumber	Фіскальний номер чека.
ReceiptLocalNumber	Локальний номер чека.
ShiftID	Ідентифікаційний номер поточної зміни.
OfflineMode	Ознака офлайн режиму.
Type	<a href="#">Тип документа.</a>
Base64Str1251ReceiptXML	XML-структура фіскалізованого чека в кодуванні Base64.
ParentFiscalNumber	Фіскальний номер чека який сторнується

## 18. Відображення QR-кода зареєстрованого чека

Метод призначений для візуалізації QR-кода з посиланням для перегляду зареєстрованого чека на ресурсі ДПС.

```
public string ShowReceiptQRCode (string rroFiscalNumber, string checkFiscalNumber );
```

#### Приклад:

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "1234567" /* фіскальний номер чеку*/
};
var ret = CallCashaLotMethod( "ShowReceiptQRCode", parameters);
```

У відповіді на **ShowReceiptQRCode** відображається QR-код та повертається **CashalotApiRsp**, в якому заповнений масив параметрів **Values**, що містить:

ReceiptFiscalNumber	Фіскальний номер чека.
ReceiptLocalNumber	Локальний номер чека.
ShiftID	Ідентифікаційний номер поточної зміни.
OfflineMode	Ознака офлайн режиму.
Type	<a href="#">Тип документа.</a>
Base64Str1251ReceiptXML	XML-структура фіскалізованого чека в кодуванні Base64.

## 19. Відправка чека покупцю за допомогою сервісів доставки SMS, Viber або E-mail

Метод використовується для відправки покупцю зареєстрованого чека продажу/повернення за допомогою сервісів доставки SMS, Viber або e-mail. Відправка чека можлива за умови, що ПРРО перебуває в режимі онлайн. В якості параметрів передається інформація про фіскальний номер ПРРО, фіскальний номер чека який необхідно відправити, адреса отримувача (номер телефону або електронна пошта) та використовуваний сервіс для відправки.

**Приклад:**

```
public string SendCheckToCustomer (string rroFiscalNumber, string checkFiscalNumber, string SendingAddressType, string SendingAddress );
```

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "1234567" /* фіскальний номер чеку*/
    "1" /* Сервіс для відправки. 1 - Viber, 2 - SMS, 3 - E-mail*/
    "+380991112223" /* Адреса отримувача. E-mail або номер телефону одержувача в залежності від обраного типу відправки. Номер телефону необхідно зазначити з урахуванням маски +380.
Приклад: cashier@cashalot.org.ua +380991002030*/
};
var ret = CallCashaLotMethod( "SendCheckToCutomer", parameters);
```

Для коректного функціонування методу попередньо слід виконати налаштування сервісів для відправки SMS та Viber в кабінеті користувача Cashalot ( <https://my.cashalot.org.ua/> ). Для можливості використання електронної пошти необхідно виконати її налаштування в програмі Cashalot в розділі «Налаштування - Загальні налаштування - Господарські одиниці - Електронна пошта».

В якості відповіді повертається стандартна структура параметрів повернення CashalotApiRsp з заповненим параметром Return(Ознака успішності виконання).

## 20. Отримання переліку налаштованих пристроїв POS-терміналів в програмі

Метод призначений для отримання списку та ID налаштованих пристроїв платіжних терміналів в програмі в розділі «Налаштування - Додаткове обладнання - POS-термінали».

**Приклад:**

```
public string GetPOSTerminalList (string rroFiscalNumber)
```

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
};
var ret = CallCashaLotMethod( "GetPOSTerminalList", parameters);
```

Після виконання методу Cashalot повертається **CashalotApiRsp**, в якому заповнений масив параметрів **Values**, що містить заповнений параметр Base64POSTerminalList. Дані в ньому закодовані в Base64.

Опис параметрів платіжного терміналу отриманих в масиві **Values**:

Назва параметра	Призначення
ID	ID платіжного терміналу
IPAddress	IP адреса за яким підключений платіжний термінал
Port	Порт за яким підключений платіжний термінал

DriverPath	Шлях до драйвера (емулятора)
MerchantName	Назва банку еквайра
MerchantID	Номер банку-еквайра (мерчанта)
ComPort	СОМ порт платіжного терміналу
DefaultDevice	Ознака пристрою який використовується за замовчуванням (true/false)
NameTerminal	Назва платіжного терміналу

## 21. Встановлення платіжного POS-терміналу пристроєм за замовчуванням

Метод призначений для встановлення платіжного POS-терміналу пристроєм який буде використовуватись за замовчуванням при проведенні безготівкових оплат в API за умови використання параметра **USEPRGPOSTERMINAL** .

**Приклад:**

**public string** SetDefaultPOSTerminal (**string** rroFiscalNumber, **string** ID)

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "2" /* ID платіжного POS-терміналу отриманого в методі GetPOSTerminalList */
};
var ret = CallCashaLotMethod( "SetDefaultPOSTerminal", parameters);
```

В якості відповіді повертається стандартна структура параметрів повернення CashalotApiRsp з заповненим параметром Return(Ознака успішності виконання).

## 22. Формування звіту про реалізовані товари

Метод PrintReportGoods використовується для формування звіту про реалізовані товари.

**Приклад:**

**public string** PrintReportGoods (**string** rroFiscalNumber)

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
};
var ret = CallCashaLotMethod( "PrintReportGoods", parameters);
```

У відповідь на PrintReportGoods відкривається діалогове вікно з можливістю обрати товар за яким необхідно сформуванати звіт, за необхідності вказавши додаткові параметри.

## 23. Проведення транзакції через POS-термінал

Для проведення транзакції на POS-терміналі необхідно використовувати метод `PayByPaymentCard`.

**Приклад:**

`public string PayByPaymentCard (string rroFiscalNumber, string SumPayByCard, string OtherParams)`

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "1.0" /* Сума оплати карткою*/
    "" /* Зарезервований параметр*/
};
var ret = CallCashaLotMethod("PayByPaymentCard", parameters);
```

В результаті виконання методу буде здійснено запит на POS-термінал який використовується за замовчуванням в налаштуваннях додаткового обладнання Cashalot. Після проведення транзакції у відповідь API буде повернуто реквізити [транзакції](#). В подальшому такі реквізити можливо використати в методі реєстрації чека `FiscalizeCheck`

Після виконання методу Cashalot повертається **CashalotApiRsp**, в якому заповнена ознака успішності виконання методу та масив параметрів **Values**

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Values	string	JSON-структура, що містить значення успішності операції, опис помилки та перелік реквізитів транзакції проведеної на POS-терміналі
InvoiceNumber	string	Номер чека
RRN	string	PPH транзакції
AcquireID	string	Ідентифікатор еквайра
AcquireName	string	Термінал
ApprovalCode	string	Код підтвердження
IssuerName	string	Платіжна система
PAN	string	Номер картки
SignVerif	string	Електронний підпис
TerminalID	string	Код терміналу
TransactionDate	string	Дата та час проведення транзакції

## 24. Повернення транзакції (оплати) через POS-термінал

За необхідності здійснення повернення оплати по раніше проведеній транзакції слід використовувати метод **ReturnPaymentByPaymentCard**. Для виконання повернення обов'язково потрібно зазначити **RRN** транзакції.

**Приклад:**

```
public string ReturnPaymentByPaymentCard (string rroFiscalNumber, string SumPayByCard, string RRN, string OtherParams)
```

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "1.0"        /* Сума оплати карткою*/
    "11111111111" /* PPH чека*/
    ""          /* Зарезервований параметр*/
};
var ret = CallCashalotMethod( "ReturnPaymentByPaymentCard", parameters);
```

В результаті виконання методу буде здійснено запит на POS-термінал який використовується за замовчуванням в налаштуваннях додаткового обладнання Cashalot. Після проведення транзакції у відповідь API буде повернуто реквізити [транзакції](#). В подальшому такі реквізити можливо використати в методі реєстрації чека FiscalizeCheck з типом "Повернення".

Після виконання методу Cashalot повертається **CashalotApiRsp**, в якому заповнена ознака успішності виконання методу та масив параметрів **Values**

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Values	string	JSON-структура, що містить значення успішності операції, опис помилки та перелік реквізитів транзакції проведеної на POS-терміналі
InvoiceNumber	string	Номер чека
RRN	string	PPH транзакції
AcquireID	string	Ідентифікатор еквайра
AcquireName	string	Термінал
ApprovalCode	string	Код підтвердження
IssuerName	string	Платіжна система
PAN	string	Номер картки
SignVerif	string	Електронний підпис
TerminalID	string	Код терміналу
TransactionDate	string	Дата та час проведення транзакції

---

## 25. Скасування транзакції (оплати) через POS-термінал

За потреби скасування проведеної транзакції на POS-терміналі необхідно використовувати метод **CancelPaymentByPaymentCard**. Для виконання повернення обов'язково потрібно зазначити номер чеку (InvoiceNumber)

**Приклад:**

```
public string CancelPaymentByPaymentCard(string rroFiscalNumber, string SumPayByCard, string InvoiceNumber, string OtherParametrs)
```

```
Object[] parameters = {  
    "4000000000", /* фіскальний номер ПРРО*/  
    "1.0" /* Сума оплати карткою*/  
    "4123312" /* Номер чека*/  
    "" /* Зарезервований параметр*/  
};  
var ret = CallCashaLotMethod("CancelPaymentByPaymentCard", parameters);
```

Після виконання методу Cashälot повертається **CashalotApiRsp**, в якому заповнена ознака успішності виконання методу та масив параметрів **Values**

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Values	string	JSON-структура, що містить значення успішності операції, опис помилки та перелік реквізитів транзакції проведеної на POS-терміналі
InvoiceNumber	string	Номер чека
RRN	string	PPH транзакції
AcquireID	string	Ідентифікатор еквайра
AcquireName	string	Термінал
ApprovalCode	string	Код підтвердження
IssuerName	string	Платіжна система
PAN	string	Номер картки
SignVerif	string	Електронний підпис
TerminalID	string	Код терміналу
TransactionDate	string	Дата та час проведення транзакції

## 26. Проведення транзакції через функцію “Оплата частинами”

Для проведення транзакції на POS-терміналі з використанням сервісу “Оплата частинами” необхідно використовувати метод **PayPartPaymentByPaymentCard**. Максимальна кількість частин може бути зазначено 12.

**Важливо!** Опція доступна лише для конфігурації обладнання POS-терміналу ПриватБанк (LAN) або ПриватБанк (USB). В налаштуваннях додаткового обладнання для POS-терміналу необхідно активувати опцію “Сервіс оплати частинами”. Окрім цього така опція має бути увімкнена в налаштуваннях безпосередньо самого POS-терміналу.

**Приклад:**

```
public string PayPartPaymentByPaymentCard(string rroFiscalNumber, string SumPayByCard, string CountParts, string OtherParametrs)
```

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "1.0"        /* Сума оплати карткою*/
    "6"         /* Кількість частин*/
    ""          /* Зарезервований параметр*/
};
var ret = CallCashaLotMethod("PayPartPaymentByPaymentCard", parameters);
```

Після виконання методу Cashalot повертається **CashalotApiRsp**, в якому заповнена ознака успішності виконання методу та масив параметрів **Values**

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Values	string	JSON-структура, що містить значення успішності операції, опис помилки та перелік реквізитів транзакції проведеної на POS-терміналі
InvoiceNumber	string	Номер чека
RRN	string	PPH транзакції
AcquireID	string	Ідентифікатор еквайра
AcquireName	string	Термінал
ApprovalCode	string	Код підтвердження
IssuerName	string	Платіжна система
PAN	string	Номер картки
SignVerif	string	Електронний підпис
TerminalID	string	Код терміналу
TransactionDate	string	Дата та час проведення транзакції
ServiceDogNum	string	Номер кредитного договору оплати частинами

## 27. Повернення транзакції по операції “Оплата частинами”

За необхідності здійснення повернення оплати по раніше проведеної транзакції з використанням сервісу “Оплата частинами” слід використовувати метод **ReturnPartPaymentByPaymentCard**. Для виконання повернення обов’язково потрібно зазначити RRN транзакції чека оплати частинами та його номер кредитного договору.

**Важливо!** Опція доступна лише для конфігурації обладнання POS-терміналу ПриватБанк (LAN) або ПриватБанк (USB). В налаштуваннях додаткового обладнання для POS-терміналу необхідно активувати опцію “Сервіс оплати частинами”. Окрім цього така опція має бути увімкнена в налаштуваннях безпосередньо самого POS-терміналу.

**Приклад:**

```
public string ReturnPartPaymentByPaymentCard(string rroFiscalNumber, string SumPayByCard, string RRN, string IsPartialReturn, string ServiceDogNum, string OtherParams)
```

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "1.0" /* Сума оплати карткою*/
    "111111111111" /* PPH чека*/
    "false" /* Ознака часткового повернення по оплаті частинами (TRUE/FALSE)*/
    "34125" /* Номер кредитного договору*/
    "" /* Зарезервований параметр*/
};
var ret = CallCashaLotMethod( "ReturnPartPaymentByPaymentCard", parameters);
```

Після виконання методу `Cashlot` повертається `CashlotApiRsp`, в якому заповнена ознака успішності виконання методу та масив параметрів `Values`

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Values	string	JSON-структура, що містить значення успішності операції, опис помилки та перелік реквізитів транзакції проведеної на POS-терміналі
InvoiceNumber	string	Номер чека
RRN	string	РРН транзакції
AcquireID	string	Ідентифікатор еквайра
AcquireName	string	Термінал
ApprovalCode	string	Код підтвердження
IssuerName	string	Платіжна система
PAN	string	Номер картки
SignVerif	string	Електронний підпис
TerminalID	string	Код терміналу
TransactionDate	string	Дата та час проведення транзакції

## 28. Створення відкладеного чека продажу товарів/послуг

Для створення відкладеного чека продажу методом `CreatePostponedReceipt` потрібно підготувати дві JSON-структури: `strJSONData` - з даними про товари чи послуги, та `payDataJSON` - з інформацією про оплату. Ці структури мають такий самий формат, як у методі `FiscalizeCheck`. Створений відкладений чек зберігається на необмежений термін.

**Приклад:**

`public string CreatePostponedReceipt (string rroFiscalNumber, string strJSONData, string payDataJSON)`

**Примітка!** Приклад виконання методу аналогічний з методом `FiscalizeCheck`

У відповіді на `CreatePostponedReceipt` повертається `CashlotApiRsp`, в якому заповнена ознака успішності та масив параметрів `Values`, що містить:

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Type	string	Тип документа.
ReceiptID	string	ID (guid) чека
ReceiptSum	string	Сума до сплати по чеку
ReceiptDateTime	string	Дата та час створення у форматі (дд.мм.рррр гг:хв:сс)
ExpressWaybillNumber	string	Номер експрес-накладної (ТТН)
StatusTTN	string	Числовий статус експрес-накладної
StatusTTNDescription	string	Статус експрес-накладної
StatusTTNCanFiscalize	string	Ознака набуття чека фіналізованого статусу доставки
ReceiptPrePayOrderNum	string	Номер інтернет-замовлення (для чеків інтернет-продажу або передплати/післяплати)

Метод **CreatePostponedReceipt** підтримує створення відкладеного чека з типом «Інтернет-продаж», призначеного для реєстрації після фактичного отримання товару покупцем через Нову пошту. Вимоги до структури такого чека наведено у [розділі](#).

## 29. Фіскалізація відкладеного чеку продажу товарів/послуг

Метод **FiscalizePostponedReceipt** використовується для фіскалізації раніше створеного відкладеного чека. Вхідним параметром методу є ReceiptID, що повертається при виклику **CreatePostponedReceipt**.

**Приклад:**

```
public string FiscalizePostponedReceipt(string rroFiscalNumber, string ReceiptID)
```

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "b5ded7c1-75ab-4b58-8fae-ac671ba9c6b6" /* ReceiptID*/
};
var ret = CallCashaLotMethod("FiscalizePostponedReceipt", parameters);
```

У відповіді на **FiscalizePostponedReceipt** повертається **CashalotApiRsp**, в якому заповнена ознака успішності та масив параметрів **Values**, що містить:

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
ShiftID	string	Ідентифікаційний номер поточної зміни.
ReceiptFiscalNumber	string	Фіскальний номер чека.
ReceiptLocalNumber	string	Локальний номер чека.
OfflineMode	string	Ознака перебування в режимі офлайн
Type	string	Тип документа.
ReceiptID	string	ID (guid) чека
ReceiptSum	string	Сума до сплати по чеку
ReceiptDateTime	string	Дата та час створення у форматі (дд.мм.рррр гг:хв:сс)
FSKORceiptLink	string	Посилання на перегляд чека в ДПС
CashalotReceiptLink	string	Посилання на перегляд чека в кабінеті Cashalot
ExpressWaybillNumber	string	Номер експрес-накладної (ТТН)
StatusTTN	string	Числовий статус експрес-накладної
StatusTTNDescription	string	Статус експрес-накладної
StatusTTNCanFiscalize	string	Ознака набуття чека фіналізованого статусу доставки
ReceiptPrePayOrderNum	string	Номер інтернет-замовлення (для чеків інтернет-продажу або передплати/післяплати)
Base64Str1251ReceiptXML	string	XML-структура фіскалізованого чека у форматі base64

Для реєстрації чеків типу «Інтернет-продаж», створених методом **CreatePostponedReceipt** і призначених для фіскалізації після отримання товару покупцем через Нову пошту, слід враховувати вимоги, описані в окремому [розділі](#).

## 30. Отримати інформацію про відкладений чек за його ReceiptID

Метод **GetPostponedReceiptByID** призначений для отримання інформації про відкладений чек за його ReceiptID. Якщо чек має тип «Інтернет-продаж» і містить заповнений параметр ExpressWaybillNumber, програма додатково виконує перевірку статусу експрес-накладної через API Нової пошти.

Залежно від отриманого статусу доставки, система автоматично визначає, чи набув чек фіналізованого статусу для можливості реєстрації. Відповідно до отриманого статусу доставки на API Нової Пошти, програма встановить відповідну ознаку.

**Примітка!** Для можливості перевірки статусу експрес-накладної необхідно попередньо виконати налаштування в кабінеті користувача Cashalot у розділі «Всі каси → Налаштування → API Нова Пошта».

**Приклад:**

```
public string GetPostponedReceiptByID (string rroFiscalNumber, string ReceiptID)
```

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "b5ded7c1-75ab-4b58-8fae-ac671ba9c6b6" /* ReceiptID*/
};
var ret = CallCashalotMethod("FiscalizePostponedReceiptGetPostponedReceiptByID", parameters);
```

У відповіді на **GetPostponedReceiptByID** повертається **CashalotApiRsp**, в якому заповнена ознака успішності та масив параметрів **Values**, що містить:

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Type	string	Тип документа.
ReceiptID	string	ID (guid) чека
ReceiptSum	string	Сума до сплати по чеку
ReceiptDateTime	string	Дата та час створення у форматі (дд.мм.рррр :хв:сс)
ExpressWaybillNumber	string	Номер експрес-накладної (ТТН)
StatusTTN	string	Числовий статус експрес-накладної
StatusTTNDescription	string	Статус експрес-накладної
StatusTTNCanFiscalize	string	Ознака набуття чека фіналізованого статусу доставки
ReceiptPrePayOrderNum	string	Номер інтернет-замовлення (для чеків інтернет-продажу або передплати/післяплати)

## 31. Отримати інформацію про відкладений чек за його номером ЕН (експрес-накладної)

Метод призначений для отримання інформації про відкладений чек вказаним номером експрес-накладної під час створення. Додатково програма виконує перевірку статусу експрес-накладної на API Нової Пошти. В залежності від отриманого статусу доставки на API Нової Пошти, програма встановить відповідну ознаку доступності реєстрації такого чека.

**Примітка!** Для можливості перевірки статусу експрес-накладної обов'язково необхідно виконати налаштування в кабінеті користувача Cashalot в розділі «Всі каси -> Налаштування -> API Нова Пошта»

**Приклад:**

```
public string GetPostponedReceiptByWaybillNumber (string rroFiscalNumber, string ExpressWaybillNumber)
```

```
Object[] parameters = {
    "4000000000", /* фіскальний номер ПРРО*/
    "59001034050511" /* Номер ЕН (ТТН)*/
};
var ret = CallCashaLotMethod("GetPostponedReceiptByWaybillNumber", parameters);
```

У відповіді на `GetPostponedReceiptByWaybillNumber` повертається `CashalotApiRsp`, в якому заповнена ознака успішності та масив параметрів `Values`, що містить:

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Type	string	Тип документа.
ReceiptID	string	ID (guid) чека
ReceiptSum	string	Сума до сплати по чеку
ReceiptDateTime	string	Дата та час створення у форматі (дд.мм.рррр :хв:сс)
ExpressWaybillNumber	string	Номер експрес-накладної (ТТН)
StatusTTN	string	Числовий статус експрес-накладної
StatusTTNDescription	string	Статус експрес-накладної
StatusTTNCanFiscalize	string	Ознака набуття чека фіналізованого статусу доставки
ReceiptPrePayOrderNum	string	Номер інтернет-замовлення (для чеків інтернет-продажу або передплати/післяплати)

## 32. Отримати інформацію про перший відкладений чек

Метод `GetFirstPostponedReceipt` рекомендовано використовувати як перший крок у ланцюжку запитів для отримання відкладених чеків за певний період часу. Метод приймає два параметри: `DateBeg` та `DateEnd`, які визначають часовий інтервал для пошуку відкладених чеків. Обидва параметри є необов'язковими та можуть містити значення `null`. Якщо обидва параметри не задані (`DateBeg = null` і `DateEnd = null`), метод повертає усі доступні відкладені чеки без обмеження по даті їх створення. Якщо задано лише один параметр: `DateBeg` — метод повертає чеки від вказаної дати і далі. `DateEnd` — метод повертає чеки до вказаної дати включно. Якщо задано обидва параметри, метод повертає чеки в межах зазначеного періоду.

**Приклад:**

```
public string GetFirstPostponedReceipt (string rroFiscalNumber, string DateBeg, string DateEnd)
```

У відповіді на `GetFirstPostponedReceipt` повертається `CashalotApiRsp`, в якому заповнена ознака успішності та масив параметрів `Values`, що містить:

Ret	bool	Ознака успішності виконання.
ErrorString	string	Опис помилки за наявності
Type	string	Тип документа.
ReceiptID	string	ID (guid) чека
ReceiptSum	string	Сума до сплати по чеку

ReceiptDateTime	string	Дата та час створення у форматі (дд.мм.рррр :хв:сс)
ExpressWaybillNumber	string	Номер експрес-накладної (ТТН)
StatusTTN	string	Числовий статус експрес-накладної
StatusTTNDescription	string	Статус експрес-накладної
StatusTTNCanFiscalize	string	Ознака набуття чека фіналізованого статусу доставки
ReceiptPrePayOrderNum	string	Номер інтернет-замовлення (для чеків інтернет-продажу або передплати/післяплати)

### 33. Отримати інформацію наступний відкладений чек

Метод `GetNextPostponedReceipt` використовується після виклику `GetFirstPostponedReceipt` у ланцюжку запитів для отримання списку відкладених чеків за певний період часу. Після того як `GetFirstPostponedReceipt` підготує список чеків, метод `GetNextPostponedReceipt` дозволяє поетапно отримувати кожен наступний чек зі списку. Метод може викликатися багаторазово, доки не буде отримано всі чеки зі сформованого списку. Коли всі чеки вже отримано, метод повертає `false`, що сигналізує про завершення вибірки. Метод приймає єдиний обов'язковий параметр `FiscalNumberRRO`

**Примітка!** перед використанням `GetNextPostponedReceipt` необхідно обов'язково викликати `GetFirstPostponedReceipt`, інакше список чеків не буде сформовано.

*Приклад:*

`public string GetNextPostponedReceipt (string rroFiscalNumber)`

```

        var listPostponedReceipt = new List<ResponsePostponedReceipt>();
        Object[] parametersGetFirstPostponedReceipt = { fiscalNumberRro, startDate, endDate };
        var response = CallCashaLotMethod("GetFirstPostponedReceipt",
parametersGetFirstPostponedReceipt);

        if (response != null)
        {
            var resultGetFirstPostponedReceipt =
MapDictionaryToObject<ResponsePostponedReceipt>(response.Values);
            listPostponedReceipt.Add(resultGetFirstPostponedReceipt);
            Object[] parametersGetNextPostponedReceipt = { fiscalNumberRro };

            while (response.Ret)
            {
                response = CallCashaLotMethod("GetNextPostponedReceipt",
parametersGetNextPostponedReceipt);
                if (response.Ret)
                {
                    var resultGetNextPostponedReceipt =
MapDictionaryToObject<ResponsePostponedReceipt>(response.Values);
                    listPostponedReceipt.Add(resultGetNextPostponedReceipt);
                }
            }
        }
    }

```

У відповіді на `GetNextPostponedReceipt` повертається `CashalotApiRsp`, в якому заповнена ознака успішності та масив параметрів `Values`, що містить:

<code>Ret</code>	<code>bool</code>	Ознака успішності виконання.
<code>ErrorString</code>	<code>string</code>	Опис помилки за наявності
<code>Type</code>	<code>string</code>	<a href="#">Тип документа</a> .
<code>ReceiptID</code>	<code>string</code>	ID (guid) чека
<code>ReceiptSum</code>	<code>string</code>	Сума до сплати по чеку
<code>ReceiptDateTime</code>	<code>string</code>	Дата та час створення у форматі (дд.мм.рррр :хв:сс)
<code>ExpressWaybillNumber</code>	<code>string</code>	Номер експрес-накладної (ТТН)
<code>StatusTTN</code>	<code>string</code>	Числовий статус експрес-накладної
<code>StatusTTNDescription</code>	<code>string</code>	Статус експрес-накладної
<code>StatusTTNCanFiscalize</code>	<code>string</code>	Ознака набуття чека фіналізованого статусу доставки
<code>ReceiptPrePayOrderNum</code>	<code>string</code>	Номер інтернет-замовлення (для чеків інтернет-продажу або передплати/післяплати)

## 34. Відображення візуальної форми чека за його `ReceiptID`

Метод `ShowReceiptByID` призначений для отримання візуальної форми чека за його `ReceiptID`.

Для виклику методу необхідно передати два параметри: `FiscalNumberRRO` та `ReceiptID`.

За допомогою цього методу можна отримати візуальне представлення як зареєстрованого, так і відкладеного чека продажу, а також видаткового чека (у випадку повернення).

```
public string ShowReceiptByID (string rroFiscalNumber, string ReceiptID)
```

У відповідь на виклик методу відкривається візуальна форма з чеком, а також повертається значення `True` або `False`, що вказує на успішність виконання.



ТОВ "Фортеця"

"Фортифікація"

місто Київ, вул. Сталева, буд.1

ПН 444444555555

ФН ПРРО 4000024425

КАСІР ТОВ "Фортеця"(Печатка)

Чек №1566232

Повернення по чеку ФН 1566231

УКТ ЗЕД 170112

Штрихкод товару: 103

Цукор, ваговий

1,000 X 10,00 10,00 Ю

СУМА 10,00 ГРН

ПДВ Ю=20% 1,67

**ГОТІВКА 10,00 ГРН**

Дякуємо за покупку!

ЧЕК №1566232

17.09.2025 17:25:47



ВИДАТКОВИЙ ЧЕК

ОНЛАЙН  
Cashalot

Друк

Надіслати на Email

Зберегти чек

Готово

## Приклади використання засобів оплат у чеку (спрощений та розширений варіант)

### Чек зі способом оплати «Продаж» (PaymentOrderType=0)

Умова: Чек на суму 100грн з засобом оплати "Платіж LIQPAY" до якого застосовуються реквізити транзакції.

Пояснення: в структурі оплат для спрощеного варіанту застосування засобів оплат заповнюється SumPayByBank значенням суми безготівкової оплати. В параметрі SumPayCheck заповнюється загальна сума по чеку. За допомогою параметра BankPayToolsNm вказується необхідна назва засобу оплат, а ознака True встановлена в параметрі UseTransactionDetailsToSumPayByBank визначає те, що до обраного засобу оплати будуть додані реквізити транзакції POS-терміналу. У розширеному варіанті застосування суму безготівкової оплати та інші характеристики засобу необхідно заповнювати в складі параметра PaymentFormsLst у вигляді масиву з чіткою послідовністю заповнення параметрів.

Приклад заповнення структури оплат (спрощений):

```
{
  "PaymentOrderType": 0,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "100",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "100",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "BankPayToolsNm": "Платіж LIQPAY",
  "UseTransactionDetailsToSumPayByBank": "true"
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 0,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "100",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "PaymentFormsLst": [
    [
      "202",
      "100",
      "Платіж LIQPAY",
      "True"
    ]
  ]
}
```

ТОВ "Фортеця"  
"Фортифікація"  
місто Київ, вул. Сталева, буд.1  
ПН 444444555555  
ФН ПРРО 4000024021  
КАСІР Сталевий І. І.  
**Чек №1519122**

Штрихкод товару: 2562374154874  
Напій Соса-Cola сильногазований  
0.5л  
1,000 X 100,00 100,00 А

-----  
СУМА 100.00 ГРН  
ПДВ А=20% 16,67  
-----

**БЕЗГОТІВКОВА 100.00 ГРН**  
**Платіж LIQPAY**

Еквайр: ПриватБанк  
Термінал: t123456  
Вид операції: ОПЛАТА  
Сума: 100.00 грн  
Платіжна система: Visa Credit  
ЕПЗ: XXXXXXXXXXXX4455  
Код авторизації: 32456  
RRN: 987654321  
Дата та час: 10.04.2025 10:57:16  
Чек №: 555777

### Чек зі способом оплати «Передплата» (PaymentOrderType=1)

Умова: Чек на загальну суму 100грн по якому відбувається передплата 50грн з засобом оплати "Платіж LIQPAY". До засобу оплати застосовуються реквізити транзакції.

Пояснення: в структурі оплат для спрощеного варіанту застосування засобів оплат заповнюється SumPayByBank значенням суми безготівкової оплати яка має дорівнювати сумі передплати. В параметрі SumPayCheck заповнюється загальна сума по чеку передплати. В параметрі SumPreparePayed зазначається сума передплати по чеку. За допомогою параметра BankPayToolsNm вказується необхідна назва засобу оплати, а ознака True встановлена в параметрі UseTransactionDetailsToSumPayByBank визначає те, що до обраного засобу оплати будуть додані реквізити транзакції POS-терміналу. У розширеному варіанті застосування суму безготівкової оплати та інші характеристики засобу необхідно заповнювати в складі параметра PaymentFormsLst у вигляді масиву з чіткою послідовністю заповнення..

Приклад заповнення структури оплат (спрощений):

```
{
  "PaymentOrderType": 1,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "50",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "50",
  "SumPreparePayed": "50",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "BankPayToolsNm": "Платіж LIQPAY",
  "UseTransactionDetailsToSumPayByBank": "true"
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 1,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "50",
  "SumPreparePayed": "50",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "PaymentFormsLst": [
    [
      "202",
      "50",
      "Платіж LIQPAY",
      "True"
    ]
  ]
}
```

ТОВ "Фортеця"  
"Фортифікація"  
місто Київ, вул. Сталева, буд.1  
ПН 444444555555  
ФН ПРРО 4000024021  
КАСІР Сталевий І. І.

**Чек №1519127**

**ПЕРЕДПЛАТА**

Штрихкод товару: 2562374154874  
Напій Соса-Сола сильногазований  
0.5л  
1,000 X 100,00 100,00 А  
-----  
**Сума по чеку** 100.00 грн  
Передплата 50.00 грн  
-----  
СУМА 50.00 ГРН  
ПДВ А=20% 8,33  
-----  
**БЕЗГОТІВКОВА Платіж LIQPAY 50.00 ГРН**  
Еквайр: ПриватБанк  
Термінал: t123456  
Вид операції: ОПЛАТА  
Сума: 50.00 грн  
Платіжна система: Visa Credit  
ЕПЗ: XXXXXXXXXXXX4455  
Код авторизації: 32456  
RRN: 987654321  
Дата та час: 10.04.2025 10:57:16  
Чек №: 555777  
-----  
Залишок до сплати 50.00 ГРН

Чек зі способом оплати «Післяплата» (PaymentOrderType=2)

Умова: Чек на загальну суму 100грн по якому відбувається доплата 50грн з засобом оплати "Платіж WAYFORPAY". До засобу оплати застосовуються реквізити транзакції.

Пояснення: в структурі оплат для спрощеного варіанту застосування засобів оплат заповнюється SumPayByBank значенням суми безготівкової оплати яка має дорівнювати сумі післяплати. За допомогою параметра BankPayToolsNm вказується необхідна назва засобу оплати, а ознака True встановлена в параметрі UseTransactionDetailsToSumPayByBank визначає те, що до обраного засобу оплати будуть додані реквізити транзакції POS-терміналу. У розширеному варіанті застосування суму безготівкової оплати та інші характеристики засобу необхідно заповнювати в складі параметра PaymentFormsLst у вигляді масиву з чіткою послідовністю заповнення. Обов'язково потрібно вказати за допомогою параметра ParentReceiptFiscalNumber ФН чека по якому була здійснена передплата.

Приклад заповнення структури оплат (спрощений):

```
{
  "PaymentOrderType": 2,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "50",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "ParentReceiptFiscalNumber": "1519246",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "BankPayToolsNm": "Платіж LIQPAY",
  "UseTransactionDetailsToSumPayByBank": "true"
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 2,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "ParentReceiptFiscalNumber": "1519246",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "PaymentFormsLst": [
    [
      "202",
      "50",
      "Платіж LIQPAY",
      "True"
    ]
  ]
}
```

ТОВ "Фортеця"  
"Фортифікація"  
місто Київ, вул. Сталева, буд.1  
ПН 444444555555  
ФН ПРРО 4000024021  
КАСІР Сталевий І. І.

**Чек №1519134**

ФН чека передплати 1519127

Штрихкод товару: 2562374154874

Напій Соса-Сола сильногазований  
0.5л

1,000 X 100,00 100,00 A

-----  
**Сума по чеку** 100.00 грн

Передплата 50.00 грн

Остаточний розрахунок 50.00 грн

-----  
СУМА 50.00 ГРН

ПДВ А=20% 8,33

-----  
**БЕЗГОТІВКОВА 50.00 ГРН**

**Платіж WAYFORPAY**

Еквайр: ПриватБанк

Термінал: t123456

Вид операції: ОПЛАТА

Сума: 50.00 грн

Платіжна система: Visa Credit

ЕПЗ: XXXXXXXXXXXX4455

Код авторизації: 32456

RRN: 987654321

Дата та час: 10.04.2025 10:57:16

Чек №: 555777

Чек зі способом оплати «Інтернет-продаж» (PaymentOrderType=3)

Умова: Чек на загальну суму 100грн по якому відбувається 100% передплата з власним засобом оплати "Власна назва".

Пояснення: для спрощеного варіанту застосування засобів оплат достатньо вказати найменування засобу який буде відображено в передплаті. В параметрі SumPreparePayed зазначається 100% сума передплати. Для розширеного варіанту необхідно в якості коду засобу оплати вказати значення 2, а для суми вказати значення яке відповідає сумі оплати по чеку. Параметр SumPreparePayed структури payDataJSON має містити однакову суму зі значенням в реквізиті суми. Ознака використання реквізитів транзакції POS-терміналу для способу оплати "Інтернет-продаж" не допускається.

Приклад заповнення структури оплат (спрощений):

```
{
  "PaymentOrderType": 3,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "SumPreparePayed": "100",
  "PrePaymentPayToolsNM": "Власна назва"
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 3,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "SumPreparePayed": "100",
  "PaymentFormsLst": [
    [
      "2",
      "100",
      "Власна назва"
    ]
  ]
}
```

ТОВ "Фортеця"

"Фортифікація"

місто Київ, вул. Сталева, буд.1

ПН 444444555555

ФН ПРРО 4000024021

КАСІР Сталевий І. І.

**Чек №1519149**

Номер інтернет-замовлення 31123

Штрихкод товару: 2562374154874

Напій Соса-Cola сильногазований

0.5л

1,000 X 100,00

100,00 A

СУМА

100.00 ГРН

ПДВ А=20%

16,67

Передплата

**БЕЗГОТІВКОВА**

**100.00 ГРН**

**Власна назва**

### Чек зі способом оплати «Інтернет-продаж» (ТипОплати=3)

Умова: Чек на загальну суму 100грн по якому відбувається комбінована оплата з заповненням засобів оплат для "Передплати" та "Післяплати" по чеку.

Пояснення: для спрощеного варіанту застосування засобів оплат необхідно вказати найменування засобу який буде відображено в передплаті та післяплаті за допомогою параметрів PrePaymentPayToolsNM та AfterPaymentPayToolsNM відповідно. В параметрі SumPreparePayed зазначається потрібна сума передплати. Різниця суми буде відображена в післяплаті. Для розширеного варіанту необхідно в якості коду засобу оплати вказати значення 2 та 201 заповнених в окремих масивах. В масиві засобу для передплати для суми вказати значення яке відповідає сумі передплати по чеку. В масиві засобу післяплати необхідно вказати суму післяплати. Ознака використання реквізитів транзакції POS-терміналу для способу оплати "Інтернет-продаж" не допускається.

Приклад заповнення структури оплат (спрощений):

```
{
  "PaymentOrderType": 3,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "SumPreparePayed": "50",
  "PrePaymentPayToolsNM": "Власна назва №2",
  "AfterPaymentPayToolsNM": "Власна назва №1"
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 3,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "SumPreparePayed": "50",
  "PaymentFormsLst": [
    [
      "2",
      "50",
      "Власна назва №2",
    ],
    [
      "201",
      "50",
      "Власна назва №1",
    ]
  ]
}
```

ТОВ "Фортеця"  
"Фортифікація"  
місто Київ, вул. Сталева, буд.1  
ПН 444444555555  
ФН ПРРО 400024021  
КАСІР Сталевий І. І.

**Чек №1519152**

Номер інтернет-замовлення 31123

Штрихкод товару: 2562374154874

Напій Соса-Сола сильногазований  
0.5л

1,000 X 100,00 100,00 А

СУМА 100.00 ГРН

ПДВ А=20% 16,67

-----  
Передплата  
**БЕЗГОТІВКОВА 50.00 ГРН**

**Власна назва №2**

Післяплата  
**БЕЗГОТІВКОВА 50.00 ГРН**

**Власна назва №1**

Чек зі способом оплати «Інтернет-продаж» (PaymentOrderType=3)

Умова: Чек на загальну суму 100грн по якому відбувається 100% післяплата з заповненням засобів оплат для "Післяплати" по чеку.

Пояснення: для спрощеного варіанту використання засобів оплати достатньо вказати найменування для післяплати по чеку в параметрі AfterPaymentPayToolsNM. Для розширеного варіанту в масиві засобу потрібно вказати код 201, суму яка дорівнює 100% післяплаті та потрібну назву оплати. Ознака використання реквізитів транзакції POS-терміналу для способу оплати "Інтернет-продаж" не допускається.

Приклад заповнення структури оплат (спрощений):

```
{  
  "PaymentOrderType": 3,  
  "SumCash": "",  
  "SumPayByCard": "",  
  "SumPayByBank": "",  
  "SumPayByCredit": "",  
  "SumPayByCertificate": "",  
  "SumPayCheck": "",  
  "AfterPaymentPayToolsNM": "Власна назва №1"  
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 3,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "PaymentFormsLst": [
    [
      "201",
      "100",
      "Власна назва №1"
    ]
  ]
}
```

ТОВ "Фортеця"  
"Фортифікація"  
місто Київ, вул. Сталева, буд.1  
ПН 444444555555  
ФН ПРРО 4000024021  
КАСІР Сталевий І. І.

**Чек №1519156**

Номер інтернет-замовлення 31123

Штрихкод товару: 2562374154874

Напій Соса-Cola сильногазований  
0.5л

1,000 X 100,00 100,00 А

СУМА 100.00 ГРН

ПДВ А=20% 16,67

Післяплата  
**БЕЗГОТІВКОВА 100.00 ГРН**

**Власна назва №1**

Чек зі способом оплати «Повна післяплата» (PaymentOrderType=4)

Умова: Чек на загальну суму 100грн по якому відбувається 100% післяплата з заповненням засобів оплат для "Передплати" та "Післяплати" по чеку.

Пояснення: для спрощеного варіанту використання засобів оплати достатньо вказати найменування для післяплати по чеку в параметрі AfterPaymentPayToolsNM. Для розширеного варіанту в масиві засобу потрібно вказати код 201, суму яка дорівнює 100% післяплаті та потрібну назву оплати. Ознака використання реквізитів транзакції POS-терміналу для способу оплати "Повна післяплата" не допускається.

Приклад заповнення структури оплат (спрощений):

```
{
  "PaymentOrderType": 4,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "AfterPaymentPayToolsNM": "Власна назва №3"
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 4,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "PaymentFormsLst": [
    [
      "201",
      "100",
      "Власна назва №3"
    ]
  ]
}
```

ТОВ "Фортеця"  
"Фортифікація"  
місто Київ, вул. Сталева, буд.1  
ПН 444444555555  
ФН ПРРО 4000024021  
КАСІР Сталевий І. І.  
**Чек №1519159**  
Номер замовлення 31123

Штрихкод товару: 2562374154874  
Напій Соса-Сола сильногазований  
0.5л  
1,000 X 100,00 100,00 А  
-----  
СУМА 100.00 ГРН  
ПДВ А=20% 16,67  
-----  
**БЕЗГОТІВКОВА** Післяплата  
**Власна назва №3** **100.00 ГРН**

Чек зі способом оплати «Передплата (оплата частинами)» (PaymentOrderType=5)

Умова: Чек на загальну суму 100грн по якому відбувається частковий розрахунок розділений на 3 платежі. Перший платіж складає 50грн.

Пояснення: в спрощеному варіанті використання засобів оплат необхідно встановити суму безготівкової оплати яка дорівнює першому платежу за допомогою параметра SumPayByBank. В параметрі PartPaySum зазначається сума цього першого платежу, а в SumPayCheck сума чека першої сплати. Параметром BankPayToolsNm визначається назва безготівкового засобу оплати до чека. В розширеному варіанті використання інформація про безготівкову оплату, найменування та код засобу зазначається в масиві.

Приклад заповнення структури оплат (спрощений):

```
{
  "PaymentOrderType": 5,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "50",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "50",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "PartPaySum": "50",
  "BankPayToolsNm": "Інтернет-еквайринг",
  "UseTransactionDetailsToSumPayByBank": "False"
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 5,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "50",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "PartPaySum": "50",
  "PaymentFormsLst": [
    [
      "202",
      "50",
      "Інтернет-еквайринг"
    ]
  ]
}
```

ТОВ "Фортеця"  
"Фортифікація"  
місто Київ, вул. Сталева, буд.1  
ПН 444444555555  
ФН ПРРО 4000024021  
КАСІР Сталевий І. І.

**Чек №1519162**  
**ПЕРЕДПЛАТА**

Штрихкод товару: 2562374154874  
Напій Соса-Сола сильногазований  
0,5л  
1,000 X 100,00 100,00 А

-----

<b>Сума по чеку</b>	100.00 грн
Передплата	50.00 грн
-----	
СУМА	50.00 ГРН
ПДВ А=20%	8,33
-----	
<b>БЕЗГОТІВКОВА</b>	<b>50.00 ГРН</b>
<b>Інтернет-еквайринг</b>	
-----	
Залишок до сплати	50.00 ГРН

Чек зі способом оплати «Чергова сплата» (PaymentOrderType=6)

Умова: Чек на загальну суму 100грн по якому відбувається черговий розрахунок в 25 грн. Перший платіж складає 50грн.

Пояснення: спрощений та розширений варіант застосування безготівкового засобу не відрізняється від прикладу формування чека першої сплати. За виключенням необхідно додати параметр ParentReceiptFiscalNumber з вказаним ФН чека першої сплати.

Приклад заповнення структури оплат (спрощений):

```
{
  "PaymentOrderType": 6,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "25",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "25",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "PartPaySum": "25",
  "BankPayToolsNm": "Інтернет-еквайринг",
  "ParentReceiptFiscalNumber": "1515457",
  "UseTransactionDetailsToSumPayByBank": "False"
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 6,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "25",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "PartPaySum": "25",
  "ParentReceiptFiscalNumber": "1519254",
  "PaymentFormsLst": [
    [
      "202",
      "25",
      "Інтернет-еквайринг"
    ]
  ]
}
```

ТОВ "Фортеця"  
"Фортифікація"  
місто Київ, вул. Сталева, буд.1  
ПН 444444555555  
ФН ПРРО 4000024021  
КАСИР Сталевий І. І.  
**Чек №1519163**  
ЧЕРГОВА СПЛАТА  
ФН чека передплати 1519162

Штрихкод товару: 2562374154874  
Напій Соса-Сола сильногазований  
0.5л  
1,000 X 100,00 100,00 А  
-----  
**Сума по чеку** 100.00 грн  
Чергова сплата 25.00 грн  
Сума передплат 50.00 грн  
-----  
СУМА 25.00 ГРН  
ПДВ А=20% 4,17  
-----  
**БЕЗГОТІВКОВА 25.00 ГРН**  
**Інтернет-еквайринг**  
-----  
Залишок до сплати 25.00 ГРН

Чек зі способом оплати «Остаточного розрахунку» (PaymentOrderType=7)

Умова: Чек на загальну суму 100грн по якому відбувається остаточний розрахунок в 25 грн. Загальна передплата по чеку складала 75 грн.

Пояснення: в спрощеному та розширеному варіанті застосування засобів оплати для чека остаточного розрахунку не потрібно зазначати суму передплат та загальну суму по чеку в параметрі SumPayCheck, сума буде розрахована автоматично. Обов'язково необхідно зазначати в параметрі ParentReceiptFiscalNumber ФН чека першої сплати.

Приклад заповнення структури оплат (спрощений):

```
{  
  "PaymentOrderType": 7,  
  "SumCash": "",  
  "SumPayByCard": "",  
  "SumPayByBank": "25",  
  "SumPayByCredit": "",  
  "SumPayByCertificate": "",  
  "SumPayCheck": "",  
  "TerminalID": "S1K700J8",  
  "ApprovalCode": "68014B",  
  "RRN": "048867173620",  
  "IssuerName": "MASTERCARD",  
  "PAN": "XXXXXXXXXXXX6883",  
  "TransactionDate": "03.04.2025 10:49:27",  
  "SignVerif": "12334444",  
  "AcquireName": "ПриватБанк",  
  "InvoiceNumber": "224",  
  "BankPayToolsNm": "Інтернет-еквайринг",  
  "ParentReceiptFiscalNumber": "1515370",  
  "UseTransactionDetailsToSumPayByBank": "False"  
}
```

Приклад заповнення структури оплат (розширений):

```
{
  "PaymentOrderType": 7,
  "SumCash": "",
  "SumPayByCard": "",
  "SumPayByBank": "",
  "SumPayByCredit": "",
  "SumPayByCertificate": "",
  "SumPayCheck": "",
  "TerminalID": "S1K700J8",
  "ApprovalCode": "68014B",
  "RRN": "048867173620",
  "IssuerName": "MASTERCARD",
  "PAN": "XXXXXXXXXXXX6883",
  "TransactionDate": "03.04.2025 10:49:27",
  "SignVerif": "12334444",
  "AcquireName": "ПриватБанк",
  "InvoiceNumber": "224",
  "ParentReceiptFiscalNumber": "1519254",
  "UseTransactionDetailsToSumPayByBank": "False",
  "PaymentFormsLst": [
    [
      "202",
      "25",
      "Інтернет-еквайринг"
    ]
  ]
}
```

ТОВ "Фортеця"

"Фортифікація"

місто Київ, вул. Сталева, буд.1

ПН 444444555555

ФН ПРРО 4000024021

КАСІР Сталевий І. І.

**Чек №1519164**

ФН чека передплати 1519162

Штрихкод товару: 2562374154874

Напій Соса-Сола сильногазований

0.5л

1,000 X 100,00 100,00 А

-----  
**Сума по чеку** 100.00 грн

Остаточний розрахунок 25.00 грн

Сума передплат 75.00 грн

-----  
СУМА 25.00 ГРН

ПДВ А=20% 4,17

-----  
**БЕЗГОТІВКОВА 25.00 ГРН**

**Інтернет-еквайринг**

## ЛОМБАРДНІ ОПЕРАЦІЇ

Для фіскалізації ломбардних послуг введено системні номенклатури, які необхідно використовувати при виклику методу фіскалізації чека [FiscalizeCheck](#). Повернення товару при ломбардних операціях не підтримується.

Також необхідно вказати тип документа [DocType = 4](#) в JSON-структурі параметрів чека. У разі потреби можна вказати номер договору [DocNumber](#), за яким надається ломбардна послуга.

Напрямок руху необхідно враховувати при обчисленні кінцевого підсумку в чеку: від позицій типу 2 віднімаються позиції типу 3.

Системні номенклатури для ломбардних операцій:

Артикул	Назва	Тип послуги	Напрямок руху*
"ЛО-00001"	"Повернення страхування"	"3"	-
"ЛО-00002"	"Надання кредиту"	"3"	-
"ЛО-00003"	"Добір кредиту"	"3"	-
"ЛО-00004"	"Надання додаткового кредиту"	"3"	-
"ЛО-00005"	"Зберігання"	"2"	+
"ЛО-00006"	"Страховий платіж"	"2"	+
"ЛО-00007"	"Часткове повернення кредиту (Частковий викуп)"	"2"	+
"ЛО-00008"	"Повне повернення кредиту (Повний викуп)"	"2"	+
"ЛО-00009"	"Повернення кредиту"	"2"	+
"ЛО-00010"	"Користування кредитом"	"2"	+
"ЛО-00011"	"Сплата відсотків по кредиту"	"2"	+
"ЛО-00012"	"Сплата пені"	"2"	+

\* Напрямок руху "-" видача грошей з каси; "+" повернення грошей в касу.

## Особливості роботи з чеками типу «Інтернет-продаж» при доставці Новою поштою

### Загальний опис

Чеки типу «Інтернет-продаж» застосовуються для фіскалізації операцій, у яких товар доставляється покупцю через Нову пошту. Реєстрація таких чеків має виконуватись після фактичного отримання товару покупцем, що підтверджується статусом експрес-накладної (ТТН), отриманим через API Нової пошти ([api.novaposhta.ua](#)). У деяких специфічних випадках дозволяється реєстрація чека за умови, якщо він не набув фінального статусу ЕН (ТТН)

API Cashalot надає інструмент у вигляді набору методів, які дозволяють реалізувати повний цикл взаємодії між продавцем і покупцем у рамках доставки:

- Створення відкладеного чека з номером ЕН (ТТН);
- Перевірка статусу доставки (отримання інформації про чек);
- Фіскалізація чека після підтвердження отримання товару.

Цей механізм забезпечує відповідність фіскальним вимогам та автоматизує процес обробки онлайн-продажів із доставкою. Використання механізму можливо за умови виконаних налаштувань в кабінеті користувача Cashalot в розділі «Всі каси → Налаштування → API Нова Пошта».

### Створення відкладеного чека

Для створення відкладеного чека використовується метод **CreatePostponedReceipt**. Під час виклику методу необхідно передати дві JSON-структури **strJSONData** та **payDataJSON**.

У випадку створення чека типу «Інтернет-продаж», в структурі **payDataJSON** обов'язково потрібно:

- Встановити параметр **PaymentOrderType** = 3, що вказує на тип операції «Інтернет-продаж».
- Заповнити параметр **ExpressWaybillNumber**, який відповідає номеру експрес-накладної (ТТН) Нової пошти.

Ці параметри дозволяють системі автоматично виконати запит до API Нової пошти після виконання методу для отримання актуального статусу доставки, що буде використано на наступних етапах обробки чека.

Можливі статуси доставки (ТТН) для відкладених чеків типу «Інтернет-продаж»

StatusTTNDescription	Пояснення до статусу	StatusTTNCanFiscalize
Потребує повторної перевірки стану доставки	Статус не вдалося визначити через відсутність зв'язку з API сервісом Нової Пошти.	False
Номер ЕН не знайдено	ЕН (ТТН) з таким номером не знайдено. Перевірте, чи правильно введено номер.	False
Відмова одержувача	Доставка не виконана - одержувач відмовився від посилки.	False
Очікується доставка одержувачу	Відправлення ще не доставлено - воно перебуває в дорозі.	False
Відправлення отримано	Посилка успішно доставлена отримувачу.	True

**Примітка!** У разі неналаштованої інтеграції в кабінеті користувача Cashalot, перевірка статусу експрес-накладної (ТТН) буде недоступна, і створення відкладеного чека типу «Інтернет-продаж» не відбудеться.

### Отримання інформації про відкладений чек

Для перевірки статусу доставки товару за номером експрес-накладної (ТТН) використовуються методи **GetPostponedReceiptByID** та **GetPostponedReceiptByWaybillNumber**, які приймають відповідно параметри **ReceiptID** або **ExpressWaybillNumber**. Якщо ці параметри відповідають чеку, створеному з типом «Інтернет-продаж» (**PaymentOrderType** = 3) та заповненим номером ТТН, система виконує запит до API Нової пошти для отримання актуального статусу доставки. Перелік можливих статусів доставки для відкладеного чека наведений в таблиці вище.

### Реєстрація відкладеного чека

У результаті виклику методів створення або отримання інформації про відкладений чек, які передбачають звернення до API Нової пошти, у відповіді повертається параметр **StatusTTNCanFiscalize**. Це логічне значення (true або false), що вказує на можливість фіскалізації чека залежно від статусу доставки. Якщо значення true, це означає, що посилка успішно доставлена, і чек може бути зареєстрований. Якщо false, доставка ще не завершена, але в окремих випадках реєстрація може бути дозволена згідно з внутрішньою логікою системи.

Для реєстрації відкладеного чека необхідно скористатися методом **FiscalizePostponedReceipt**, який приймає параметр **ReceiptID** - ідентифікатор створеного чека. Під час виклику цього методу здійснюється додаткове звернення до сервісів API Нової пошти з метою отримання актуального статусу ЕН (ТТН). Якщо статус вказує, що відправлення отримано, чек реєструється, і у відповіді

повертається оновлене значення **StatusTTNCanFiscalize = true**, що свідчить про те, що ТТН набула фіналізованого статусу.

**Примітка!** Реєстрація відкладеного чека неможлива, якщо статус ЕН (ТТН) - "Номер ЕН не знайдено" або "Відмова одержувача".

### Особливості реєстрації відкладеного чека «Інтернет-продажу» в залежності від статусу ЕН(ТТН) та визначеного способу оплати по чеку

Особливості реєстрації відкладеного чека типу «Інтернет-продаж» (**PaymentOrderType = 3**) із заповненим номером експрес-накладної (**ExpressWaybillNumber**) залежать від статусу доставки ЕН (ТТН) та способу оплати, визначеного по чеку. Основним орієнтиром є параметр **StatusTTNCanFiscalize**, який вказує, що ТТН набула фіналізованого статусу «Відправлення отримано» (**StatusTTNCanFiscalize = true**). У випадках, коли по чеку передбачена комбінована модель оплати - тобто частина суми вже сплачена наперед (визначається заповненістю параметра **SumPreparePayed**) або виконана 100% передплата засобами інтернет-еквайрингу, допускається можливість реєстрації чека навіть за умови, що статус доставки ще не фіналізований (**StatusTTNCanFiscalize = false**).

Спосіб оплати по чеку	Додаткові пояснення	Заповнення SumPrepare Payed	Можливість реєстрації якщо чек має StatusTTNCanFiscalize = false	Можливість реєстрації якщо чек має StatusTTNCanFiscalize = true
100% післяплата (накладений платіж)	Вид розрахунку, за яким покупець буде здійснювати оплату за товар службі доставки (експедитору) лише після огляду та отримання	Параметр не заповнено	Ні	Так
100% передплата	Вид розрахунку, за яким покупець сплачує повну суму (або її частину) ще до отримання товарів.	Значення параметру відповідає 100% сумі оплати по чеку	Так	Так
Оплата "50/50"	Часткова передплата та післяплата	Параметр заповнено на суму яка не відповідає повній сумі по чеку.	Так	Так

## Приклад реалізації функції для виклику API методів ПРРО Cashälot

```
//Структура інформації рядка чека (містить всі дані для створення товару в Cashälot)
public class CheckInfo
{
    public CheckInfo()
    { }
    public CheckInfo( CheckInfo ckInfo )
    {
        this.VendorCode = ckInfo.VendorCode;
        this.Name = ckInfo.Name;
        this.CdUKDZED = ckInfo.CdUKDZED;
        this.CdDKPP = ckInfo.CdDKPP;
        this.GoodsType = ckInfo.GoodsType;
        this.Barcode = ckInfo.Barcode;
        this.UnitType = ckInfo.UnitType;
        this.Quantity = ckInfo.Quantity;
        this.Price = ckInfo.Price;
        this.Amount = ckInfo.Amount;
        this.DiscountPrc = ckInfo.DiscountPrc;
        this.DiscountSum = ckInfo.DiscountSum;
        this.VATRate = ckInfo.VATRate;
        this.IsPriceIncludeVAT = ckInfo.IsPriceIncludeVAT;
        this.SumVAT = ckInfo.SumVAT;
        this.IsExcise = ckInfo.IsExcise;
        this.OtherParametrs = ckInfo.OtherParametrs;
        this.VATLetter = ckInfo.VATLetter;
        this.ExciseLetter = ckInfo.ExciseLetter;
        this.ExciseStampBarcode = ckInfo.ExciseStampBarcode;
    }
    //Артикул товару
    public String VendorCode { get; set; } = String.Empty;
    public String Article { get => VendorCode; set { VendorCode = value; } }

    //Назва товару
    public String Name { get; set; } = String.Empty;
    //Код УКДЗЕД
    public String CdUKDZED { get; set; } = String.Empty;
    //Код ДКПП(для послуг)
    public String CdDKPP { get; set; } = String.Empty;
    //Тип (1-товар/2-послуга(+)|3-послуга ломбардна(-)). За замовчуваннял товар, якщо
нічого не задано.
    public String GoodsType { get; set; } = String.Empty;
    //Штрихкод
    public String Barcode { get; set; } = String.Empty;
    //Код одиниці виміру
    public String UnitType { get; set; } = String.Empty;
    //Кількість товару
    public String Quantity { get; set; } = String.Empty;
    //Ціна товару
    public String Price { get; set; } = String.Empty;
    //Сума
    public String Amount { get; set; } = String.Empty;
    //Процент знижки
    public String DiscountPrc { get; set; } = String.Empty;
    //Сума знижки
    public String DiscountSum { get; set; } = String.Empty;
    //Процент податку ПДВ
    public String VATRate { get; set; } = String.Empty;
    //Ознака - чи включає сума ПДВ
    public bool? IsPriceIncludeVAT { get; set; } = true;
}
```

```

    //Сума ПДВ
    public String SumVAT { get; set; } = String.Empty;
    //Ознака - підакцизного товару
    public bool? IsExcise { get; set; } = false;
    //Інші параметри у вигляді строки XML
    //Приклад: <?xml version="1.0" encoding="UTF-8"?><Table><Record VAT ="20"
IsExcisable="False" Nomenclature="СТЕР Пазп 104 Винни Пух" UKTZED=""
VendorCode="82106" Barcode="" Dimension="шт" Discount="1"
DiscountAmount="0.56" PriceIncludeVAT="1" AmountVAT="9.24"/></Table>
    public String OtherParametrs { get; set; }
    //Літера ставки податку ПДВ
    public String VATLetter { get; set; } = String.Empty;
    //Літера ставки податку Акцизу
    public String ExciseLetter { get; set; } = String.Empty;
    //Штрихкод акцизної марки
    public String ExciseStampBarcode { get; set; } = String.Empty;
};

//Структура з масивом рядків чека
public class LstCheckInfoRsp
{
    public List<CheckInfo> ReceiptLst { get; set; }
    //Коментар
    public String Comment { get; set; }
    //Тип чека(0 - продаж, 4 - ломбард)
    public long DocType { get; set; } = 0;
};

//шлях до каталогу ПРРО Cashalot
public string edCashaLotPath = "C:\\ProgramData\\Cashalot\\Cashalot";

//Фіскальний номер РРО
public string edFiscalNumberRRO = "4000000000";

//завантажена бібліотека
private Assembly asmCashaLotDll;

//тип класу Апі
private Type asmCashaLotClass;

//екземпляр класу Апі
private object instCashaLotApi;

//клас відповіді апі для десеріалізації
public class CashaLotApiRsp
{
    //Ознака успішності виконання функції (true - успішно/false - не успішно)
    public bool Ret { get; set; } = false;
    //Рядок з текстом помилки
    public string ErrorString { get; set; } = String.Empty;
    //Масив параметрів, які можуть повернути функції АПІ
    public Dictionary<string, string> Values { get; set; }
}

/// <summary>
/// Функція для виклику методу з АПІ Cashalot
/// </summary>
/// <param name="methodName"> Назва методу </param>
/// <param name="parameters"> Список параметрів методу</param>
/// <returns></returns>
public CashaLotApiRsp CallCashaLotMethod(string methodName, Object[] parameters )

```

```

{
    CashaLotApiRsp ret = new CashaLotApiRsp();

    if ( edCashaLotPath.Trim().Length == 0 )
        MessageBox.Show( "Не вказано шлях до каталогу Cashalot" );

    if ( edFiscalNumberRRO.Trim().Length == 0 )
        MessageBox.Show( "Не вказано фіскальний номер РРО" );

    string strPath = $"{edCashaLotPath}\\CashaLotApiBridge.dll";
    if ( !File.Exists( strPath ) )
    {
        MessageBox.Show( "За вказаним каталогом не знайдено CashaLotApiBridge.dll" );
        return ret;
    }

    if ( asmCashaLotDll == null )
        asmCashaLotDll = System.Reflection.Assembly.LoadFrom( strPath
); //Завантаження бібліотеки

    if ( asmCashaLotDll != null )
    {
        if( asmCashaLotClass == null)
            asmCashaLotClass = asmCashaLotDll.GetType( "CashaLotApiBridge.BridgeApi"
);

        if ( asmCashaLotClass != null )
        {
            if( instCashaLotApi == null)
                instCashaLotApi = Activator.CreateInstance( asmCashaLotClass ); //Створення
об'єкту класу

            System.Reflection.MethodInfo mi = asmCashaLotClass.GetMethod(methodName
); //отримання інформації по вказанному методу
            if ( mi != null )
            {
                var retVal = mi.Invoke( instCashaLotApi, parameters ); // виклик методу
з передачею параметрів
                if( retVal !=null )
                {
                    try
                    {
                        //десеріалізація отриманої відповіді
                        ret = JsonConvert.DeserializeObject<CashaLotApiRsp>( retVal as
String );
                    }
                    catch
                    {
                        MessageBox.Show( "Не вдалося десеріалізувати відповідь від
Cashalot" );
                    }
                }
            }
        }
    }
    return ret;
}

```