

Специфікація рішень API ПРРО Cashalot

Актуально починаючи з версії ПРРО Cashalot 1.00.34

Зміст

Поняття і визначення.....	5
Апаратні та програмні вимоги	5
Додаткові пристрої	6
ВАРІАНТ 1 – 1С BAS ДРАЙВЕР ПРРО CASHALOT, CASHALOTAPIDRV32.DLL АБО CASHALOTAPIDRV64.DLL.....	6
Ініціалізація драйверу.....	7
Отримання налаштувань драйверу	10
Встановлення налаштувань драйверу	10
Підключення фіскального реєстратора ПРРО Cashalot.....	11
Відключення екземпляра фіскального реєстратора ПРРО Cashalot	12
Перевірка ПРРО Cashalot	12
Відкриття зміни.....	13
Реєстрація чеків.....	13
Відміна відкритого чеку.....	19
Отримання інформації про поточний стан ПРРО	19
PrintPReportDate - Надрукувати періодичний звіт по датам	20
Друк довільної текстової інформації	21
Повернення товару	22
Службове внесення	23
Службова видача	23
Формування X-звіту	24
Формування Z-звіту	24
Закриття зміни	24
Порядок виклику функцій при виконанні касових операцій	25
Приклад реалізації підключення драйверу до Розрахунково-облікової платформи (.epf)	25
ВАРІАНТ 2А – API ПРРО CASHALOT з COM ІНТЕРФЕЙСОМ CASHALOTAPI32.DLL, CASHALOTAPI64.DLL	27
Встановлення налаштувань API бібліотеки SetParameter.....	29
Встановлення налаштувань API бібліотеки SetParameterEx	32
GetVersion – Отримання номеру версії API бібліотеки	34
GetInterfaceRevision – Отримання номеру ревізії API бібліотеки	34
OpenShift – Відкриття зміни	35
FiscalizeCheck – Фіскалізація чека продажу товарів, послуг	36
FiscalizeReturnCheck – Фіскалізація чека повернення товарів, послуг	43
GetReceiptLocalNumberByFiscalNumber – Отримання локального номеру чеку за його фіскальним номером.....	46
GetReceiptXML – Отримання XML структури фіскального чеку.....	47
GetFirstReceiptByPeriod – Отримати перший чек з періоду	49
GetFirstReceiptByShift – Отримати перший чек з зміни.....	50
GetNextReceipt – Отримати наступний чек з періоду	51

GetCurrentStatus – Отримати інформацію по поточному стану каси.....	53
GetFirstZReportByPeriod – Отримати перший Z-звіт за період.....	54
GetNextZReport – Отримати наступний Z-звіт з періоду.....	56
GetZReportByShift– Отримати Z-звіт касової зміни	58
ShowReceipt – Вивести візуальне відображення чеку на екран	60
PrintPReportDate - Надрукувати періодичний звіт по датам	61
PrintTextDocument – Надрукувати довільний нефіскальний текст	63
ServiceInput – Службове внесення грошей в касу.....	64
ServiceInputEx – Службове внесення грошей в касу.....	65
ServiceOutput – Службова видача грошей з каси.....	65
ServiceOutputEx – Службова видача грошей з каси	66
GetLastError – Отримання опису останньої помилки.....	67
PrintXReport - Формування X-звіту.....	67
GetXReport - Формування X-звіту	67
GetLastZReport – Отримати останній Z-звіт	71
PrintZReport – Формування Z-звіту	74
CloseShift - Закриття зміни.....	74
Приклад роботи з кількома ПРРО одночасно в одній обліковій системі використовується новий метод SetParameterEx.....	78
Можна задавати параметри, які ПРРО використовувати для кожного методу окремо.....	80
Ломбардні операції	81
Рекомендований порядок роботи з CashalotApi	84
ВСТАНОВЛЕННЯ ПАРАМЕТРІВ	84
ВІДКРИТТЯ ЗМІНИ.....	85
ПРОДАЖ ТОВАРІВ, ПОСЛУГ	85
ПОВЕРНЕННЯ ТОВАРІВ, РЕКОМПЕНСАЦІЯ ПОСЛУГ	85
ПОШУК ФІСКАЛЬНОГО ТА ЛОКАЛЬНОГО НОМЕРУ ЧЕКУ	85
ОТРИМАННЯ XML-СТРУКТУРИ ЧЕКУ	85
ВІДОБРАЖЕННЯ ЗАРЕЄСТРОВАНОВОГО ЧЕКУ	85
НАЛАШТУВАННЯ ЕЛЕКТРОННОЇ ПОШТИ	85
СЛУЖБОВЕ ВНЕСЕННЯ ГРОШЕЙ В КАСУ	85
СЛУЖБОВА ВИДАЧА ГРОШЕЙ З КАСИ.....	85
ФОРМУВАННЯ X-ЗВІТУ	86
ОТРИМАННЯ ЧЕКІВ Для ОТРИМАННЯ ЧЕКІВ З ІСТОРІЇ ОПЕРАЦІЙ РЕАЛІЗОВАНО ФУНКЦІЇ, ЩО ПЕРЕДБАЧАЮТЬ ЗАВАНТАЖЕННЯ ЧЕКІВ ЗА ПЕВНИЙ ПЕРІОД ЧАСУ АБО ЗА ПЕВНУ КАСОВУ ЗМІНУ	86
ОТРИМАННЯ ДОКУМЕНТІВ Z-ЗВІТІВ ЗАКРИТИХ КАСОВИХ ЗМІН	86
ОТРИМАННЯ ОСТАННЬОГО Z-ЗВІТУ	86
ЗАКРИТТЯ ЗМІНИ, Z-ЗВІТ	86
Приклад реалізації підключення COM бібліотеки на C++	87
СТВОРЕННЯ ФАЙЛІВ ЗАГОЛОВКІВ *.h	87
ЗАУВАЖЕННЯ	87
ЛІСТИНГ	87
ВАРІАНТ 2Б – API ПРРО CASHALOT З .NET ІНТЕРФЕЙСОМ CASHALOTAPIBRIDGE.DLL	103
Специфікація методів API бібліотеки ПРРО Cashalot при підключенні до облікової системи	103
Порядок роботи обліково-розрахункової системи з API бібліотекою ПРРО Cashalot	103
Специфікація методів бібліотеки CashalotApiBridge.dll	104
ВІДКРИТТЯ ЗМІНИ.....	104

Відкриття зміни.....	104
Службове внесення грошових коштів в касу.....	105
Службове вилучення (видача) грошових коштів з каси.....	106
Проведення розрахункових операцій продажу товарів/послуг	107
Визначення фіскального номеру чеку	113
Проведення операції повернення товару	114
Створення Х-Звіту	116
Визначення локального номеру чека по фіскальному номеру	117
Отримання XML-структури чеку	118
Надрукувати періодичний звіт по датам	118
Відображення фіскалізованого чеку	119
Закриття зміни та створення Z-Звіту	119
Ломбардні операції	120
Приклад використання API для ломбардних операцій.....	121
Приклад реалізації функції для виклику API методів PPPO Cashalot.....	122

Загальна інформація

Розроблені API ПРРО Cashalot використовуються для інтеграції ПРРО з розрахунково-обліковими системами. В даному документі приведено специфікацію методів бібліотек для пропонуванних варіантів підключення:

Додано можливість одночасної роботи декількох ПРРО Cashalot в одній обліковій системі використовуючи бібліотеку (**CashaLotApiBridgeHlp.dll** та метод [SetParameterEx](#)), що сумісні з усіма нижче перелічуваними варіантами.

Варіант 1 - драйвер ПРРО Cashalot для BAS призначений для інтеграції ПРРО Cashalot з розрахунково-обліковою системою BAS (**API драйвер CashalotApiDrv**).

Варіант 2 - API ПРРО Cashalot призначений для підключення до сторонніх розрахунково-облікових систем:

- а) з використанням COM-методів (**COM API CashalotApi**);
- б) на основі .NET-технології (**API CashalotApiBridge**).

Варіант 3 - сервер API ПРРО Cashalot призначений для взаємодії з фіскальним сервером програмних реєстраторів розрахункових операцій.

Поставляється у вигляді окремого застосунку, що розгортається локально на стороні користувача як web-служба.

Інтеграція надає користувачам можливість безпосередньо з розрахунково-облікової системи скористатися функціоналом програмного забезпечення Cashalot:

- Відкриття та закриття касової зміни;
- Створення та реєстрація чеків продажу/повернення;
- Створення видаткового чека на основі чека продажу та його реєстрація;
- Формування та друк Z-Звіту за даними фіскального сервера контролюючого органу (скорочено ФСКО)/Х-Звіту за даними ПРРО;
- **Створення та реєстрація службових чеків:**
 - - Службове внесення;
 - - Службова видача;
- **Реєстрація чека:**
 - - Засвідчення документа кваліфікованим електронним підписом (КЕП);
 - - Відправка по API XML чека на фіскальний сервер контролюючого органу;
 - - Отримання та обробка позитивної відповіді від ФСКО (отримання фіскального номера);
 - - Отримання та обробка негативної відповіді;
- **Робота в режимі Офлайн:**
 - - Контроль зв'язку з фіскальним сервером контролюючого органу;
 - - Використання резервних номерів;
 - - Контроль таймера 36 (36 годин однієї офлайн-сесії);
 - - Контроль таймера 168 (168 годин офлайн-сесій на місяць);
 - - Збереження чеків у Тимчасовий ланцюжок;
 - - Реєстрація непереданих чеків (контрольна стрічка), створених в режимі Офлайн (після переходу в режим Онлайн);
- Авторизація по КЕП (вхід в програму);
- Відображення на екрані (візуалізація) зареєстрованого чека на пристрої продавця з QR-кодом після успішної оплати;
- Друк зареєстрованого чека покупцю за допомогою принтера чеків.
- **Робота з обладнанням:**
 - Інтеграція зі сканером штрих-кодів;
 - Інтеграція з принтером чеків.

1. 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.

Розрахунково-облікова платформа – платформа BAF (реалізована підтримка версій 7.7 і вище).

Розрахунково-облікова система – стороння система обліку розрахунків.

Фіскальний номер чека – номер, призначений чеку Сервером ФСКО.

ФСКО – фіскальний сервер контролюючого органу.

BAS – конфігурація Business Automation Software for Retail для розрахунково-облікової платформи BAF.

Обов'язкові Параметри* - відповідають за ідентифікацію та виклик необхідних компонентів, функцій, параметрів то що.

Далі розглянуто варіанти роботи розрахунково-облікових систем або платформ з API бібліотеками ПРРО Cashalot:

Варіант 1. CashalotApiDrv32.dll, CashalotApiDrv64.dll

Варіант 2а. CashalotApi32.dll, CashalotApi64.dll

Варіант 2б. CashalotApiBridge.dll

Варіант 3. Ознайомитися з умовами підключення і можливостями роботи з серверним рішенням API ПРРО Cashalot можна за телефоном (044) 344-07-44.

2. Апаратні та програмні вимоги

Для оптимальної роботи ПРРО Cashalot, що є клієнтським десктопним програмним забезпеченням, рекомендовані 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:

глобальні системні оновленнями, що презентує компанія Microsoft для систем безпеки та інших програмних компонентів в залежності від операційної системи яка використовується;

рекомендовано використовувати останню версію ОС, наприклад, Windows 10 або Windows Server 2019 відповідно до потреб.

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 (розподільна здатність від 1440 x 900);
- 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. Необхідна пропускна здатність каналу залежить від інтенсивності роботи.

3. Додаткові пристрої

Можливий набір пристроїв для обладнання робочого місця касира:

- системний пристрій з встановленими програмними модулями комп'ютер/планшет);
- монітор касира;
- принтер чеків;
- платіжний пристрій (POS-термінал);
- ваги;
- сканер штрих-кодів.

Варіант 1 – 1С BAS драйвер ПРРО Cashalot, CashaLotApiDrv32.dll або CashaLotApiDrv64.dll

В конфігураторі BAS для драйверу фіскального реєстратора вказуються функції, призначені для розробки драйверів обладнання, що підключається. Ці функції відправляють запити і отримують відповіді від драйверу CashaLotApiDrv32.dll / CashaLotApiDrv64.dll (далі - CashaLotApiDrv).

Інструкція по підключенню і налаштуванню драйверу Cashalot для інтеграції в BAS детально описує підготовку до роботи в обліково-розрахунковій системі.

1. Ініціалізація драйверу:

Для використання драйверу необхідно зареєструвати бібліотеки CashaLotApiDrv32.dll / CashaLotApiDrv64.dll.

Для цього в командному рядку Windows виконайте команду:

regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApiDrv32.dll

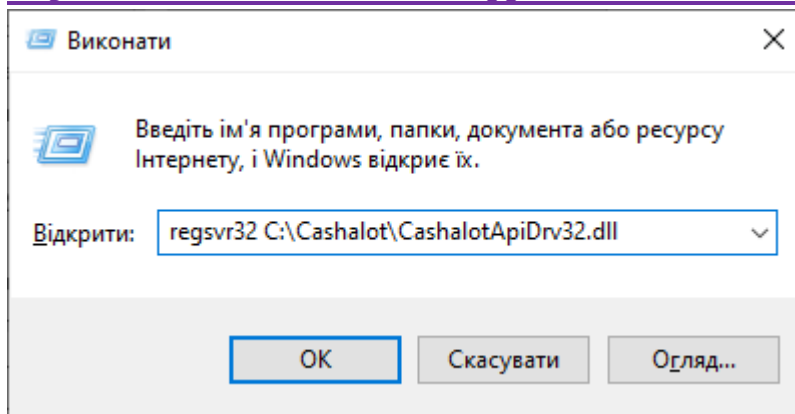
(для 32-бітної платформи)

regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApiDrv64.dll

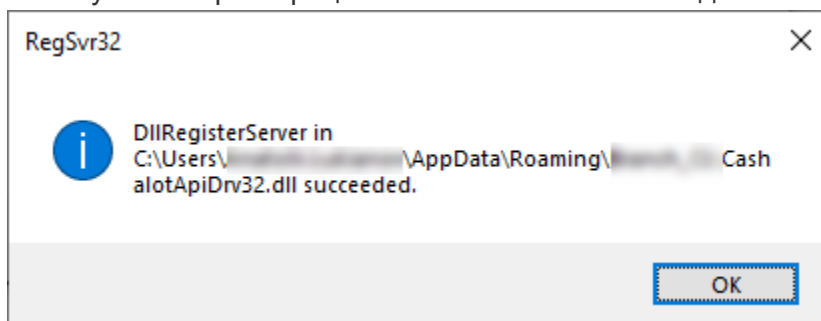
(для 64-бітної платформи)

Наприклад:

regsvr32 C:\Users\User1\AppData\Local\Cashalot\CashalotApiDrv32.dll



Після успішної реєстрації бібліотеки з'явиться повідомлення зразку:



Ініціалізація компонент відбувається шляхом створення об'єкту відповідно до синтаксису:

*ПодключитьВнешнююКомпоненту(<Местоположение>, <Имя>, ТипВнешнейКомпоненты.Native);
Компонента = Новый("AddIn.<Имя>.AddInNativeExtension");*

Драйвер ПРРО Cashalot ініціалізується наступним чином:

ПодключитьВнешнююКомпоненту("C:\Cashalot\CashaLotApiDrv32.dll", "CashaLotApiDrv ")

ОбъектДрайвера = Новый("AddIn.CashaLotApiDrv.CashaLotApiDrv");

В драйвері CashalotApiDrv реалізована обробка наступних функцій:

ПолучитьНомерВерсии

ПолучитьОшибку

ПолучитьПараметры

УстановитьПараметр

Подключить

Отключить
ТестУстройства
ОткрытьСмену
ОткрытьЧек
НапечататьФискальнуюСтроку
НапечататьФискСтроку
ЗакрытьЧек
ОтменитьЧек
НапечататьЧекВнесенияВыемки
НапечататьОтчетБезГашения
НапечататьОтчетСГашением
ЗакрытьСмену

Необхідно встановити налаштування в 1С КонфігураторіBAS таким чином щоб драйвер мав можливість розрізняти різні екземпляри ПРРО Cashalot, використовуючи їх одночасно для декількох екземплярів ПРРО Cashalot.

Користувач має можливість налаштувати власну конфігурацію 1С BAS для декількох екземплярів ПРРО одночасно, для цього необхідно вказати обов'язкові значень параметру.

Обов'язкові значення параметру*

Параметр: DeviceIDFnRRO*, "PathToCashalotDir".

Ім'я	Опис
<u>"DeviceIDFnRRO"</u>	Фіскальний номер ПРРО
<u>"PathToCashalotDir"</u>	Шлях до каталогу з встановленим ПРРО Cashalot

виконує завдання опису властивостей, унікальне значення параметру(«ИДУстройства») призначатися та відображається для кожного екземпляру окремо.

За умови, що було виконано процедуру підключення відповідних бібліотек, що відповідальні за ініціалізацію компонент шляхом створення об'єкту, відповідно до синтаксису, встановлених функцією драйверу, за їхніми іменами.

Рекомендовано кожну операцію в BAS супроводжувати встановлювати обов'язкове значення параметру*; Фіскальний номер ПРРО* та (шлях до ПРРО Cashalot)*.

1. УстановитьПараметр

2. Операция (например, регистрация чеку)

Отримання номеру версії драйверу

Для отримання номеру версії драйверу використовується функція

СТРОКА ПолучитьНомерВерсии()

Приклад:

Версия = *ОбъектДрайвера.ПолучитьНомерВерсии()*;

Отримання опису помилки

Для отримання опису помилки використовується функція

ЧИСЛО ПолучитьОшибку(ОписаниеОшибки)

Параметри функції зазначені в Таблиці 2.1.

Таблица 2.1 Параметри функції опису помилки.

Ім'я	Тип	Опис
ОписаниеОшибки	СТРОКА [OUT]	Опис помилки

Приклад:

ИмяПараметра = "DeviceIDFnRRO";

ЗначениеПараметра = "4000001234";

Ответ = *ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра)*;

ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ОбъектДрайвера.ПолучитьОшибку(*ВыходныеПараметры*[1]);

2. Отримання налаштувань драйверу

Для отримання списку параметрів налаштувань драйверу та їх типів використовують функцію **БУЛЕВО** `ПолучитьПараметры` (**ТаблицаПараметров**)

Параметри функції зазначені в Таблиці 3.1

Таблиця 3.1 Параметри функції отримання параметрів драйверу.

Ім'я	Тип	Опис
ТаблицаПараметров	СТРОКА [OUT]	Опис параметрів.

Таблиця списку параметрів представляє собою XML-рядок.

Зауваження. Функція не повертає поточні значення параметрів.

Приклад:

```
ИмяПараметра = "DeviceIDFnRRO";  
ЗначениеПараметра = "4000001234";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
ИмяПараметра = "PathToCashalotDir";  
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
ОбъектДрайвера.ПолучитьПараметры(ПараметрыДрайвера);
```

3. Встановлення налаштувань драйверу

Сценарії користувача, що встановлюють значень параметрів налаштувань драйверу декількох екземплярів ПРРО Cashalot (), кожен з яких має унікальне значенням параметрів, встановлених функцією драйверу , за їхніми іменами використовують функцію

БУЛЕВО `УстановитьПараметр`(**Имя**, **Значение**)

Параметри функції зазначені в Таблиці 4.1.

Таблиця 4.1 Параметри функції установки параметрів драйверу.

Ім'я	Тип	Опис
Имя	СТРОКА	Ім'я параметру
Значение	СТРОКА	Значення параметру

При роботі з **CashalotApi**, необхідно зазначити такі параметри:

Ім'я	Опис	Обов'язкові
"DeviceIDFnRRO"	Фіскальний номер ПРРО	Так, для Всіх
"PathToCshalotDir"	Шлях до каталогу з встановленим ПРРО Cshalot	Так, для Всіх
"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 Кашалот.	Так, для цієї функції
"AUTOPRINTMODE"	Параметр визначає автоматичну відправку на друк чеку або службового звіту без відображення інформаційного вікна "True" – відправляти на друк "False" – не відправляти на друк Якщо задано принтер за замовчуванням (DEFAULTPRINTERNAME), то чек одразу друкується, в іншому випадку відображається вікно вибору принтера для друку.	Так, для цієї функції

Приклад:

```

ІмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(ІмяПараметра, ЗначениеПараметра);
ІмяПараметра = "PathToCshalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cshalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(ІмяПараметра, ЗначениеПараметра);
Для Каждого Параметр Из Параметры Цикл
    Если Лев(Параметр.Ключ, 2) = "P_" Тогда
        ЗначениеПараметра = Параметр.Значение;
        ІмяПараметра = Сред(Параметр.Ключ, 3);
        Ответ = ОбъектДрайвера.УстановитьПараметр(ІмяПараметра, ЗначениеПараметра)
    КонецЕсли;
КонецЦикла;

```

4. Підключення фіскального реєстратора ПРРО Cshalot

Для підключення ПРРО Cshalot з поточними значенням параметрів, встановлених функцією «УстановитьПараметр», використовується функція

БУЛЕВО Підключить(ИДУстройства)

Параметри функції зазначені в Таблиці 5.1

Таблиця 5.1 Параметри функції підключення пристрою.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА [OUT]	Ідентифікатор пристрою

При успішному підключенні, функція повертає ідентифікатор підключеного ПРРО.

Приклад:

```
ІмяПараметра = "DeviceIDFnRRO";
```

ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Ответ = ОбъектДрайвера.Подключить(*ПараметрыПодключения*.ИДУстройства);

5. Відключення екземпляра фіскального реєстратора ПРРО Cashalot

Для відключення ПРРО Cashalot(), в залежності від налаштувань користувача з одним чи декількома встановленими екземплярами ПРРО(), використовується функція «УстановитьПараметр», яка використовується функцією

БУЛЕВО Отключить(**ИДУстройства**)

Параметри функції зазначені в Таблиці 7.1.

Таблица 7.1 Параметры функции відключення пристрою.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Функція повертає ідентифікатор відключеного ПРРО().

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "3000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Ответ = ОбъектДрайвера.Отключить(*ПараметрыПодключения*.ИДУстройства);

6. Перевірка ПРРО Cashalot

Сценарій в якому функція використовується користувачем для перевірки одного чи декількох ПРРО(), шляхом пробного підключення і опитування пристрою з поточними значеннями

параметрів, встановлених для кожного екземпляру окремо, функцію «УстановитьПараметр», використовується функція

БУЛЕВО ТестУстройства (Описание, ПризнакДемоРежима)

Параметри функції зазначені в Таблиці 8.1.

Таблиця 8.1 Параметри функції

Ім'я	Тип	Опис
Описание	СТРОКА [OUT]	Опис результату виконання тесту
ПризнакДемоРежима	СТРОКА [OUT]	Опис обмежень демонстраційного режиму при його наявності, і пустий результат при його відсутності.

При успішній перевірці повертається повідомлення "Тест пристрою виконано вдало" та результат перевірки приймає значення ИСТИНА.

Увага! На даному етапі розробки результат виконання функції не містить значень вихідних параметрів.

Приклад:

```
ИмяПараметра = "DeviceIDFnRRO";  
ЗначениеПараметра = "4000001234";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
ИмяПараметра = "PathToCashalotDir";  
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
Ответ = ОбъектДрайвера.ТестУстройства("", "");
```

7. Відкриття зміни

Сценарій в якому функція використовується користувачем при відкритті зміни, в залежності від налаштувань для одного ПРРО чи для декількох екземплярами ПРРО, для кожного екземпляру окремо необхідно вказати **Обов'язкові значення параметру*-'ИДУстройства'**, щоб мати змогу застосувати функцію

БУЛЕВО ОткрытьСмену(ИДУстройства)

Параметри функції зазначені в Таблиці 9.1. При виклику цієї функції драйвер ПРРО Cashalot викликає вікно авторизації, в якому користувач зазначає папку з файловим ключом касира або старшого касира, пароль, після чого відбувається відкриття касової зміни.

Таблиця 9.1 Параметри функції відкриття зміни.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою

Приклад:

```
ИмяПараметра = "DeviceIDFnRRO";  
ЗначениеПараметра = "4000001234";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
ИмяПараметра = "PathToCashalotDir";  
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";  
Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);  
Результат = ОткрытьСмену(ОбъектДрайвера, Параметры, ПараметрыПодключения,  
ВходныеПараметры, ВыходныеПараметры);
```

8. Реєстрація чеків

При реєстрації чеків дані передаються через ланцюжок функцій BAS, призначених для друку чеків на фіскальних реєстраторах:

БУЛЕВО ОткрытьЧек(ИДУстройства, ФискальныйЧек, ЧекВозврата, АннулироватьОткрытыйЧек, НомерЧека, НомерСмены)

БУЛЕВО НапечататьФискальнуюСтроку(ИДУстройства, Наименование*, КодУКТЗЕД, Артикул*, Штрихкод, ЕдиницаИзмерения*, НомерСекции, Количество*, Цена*, Сумма*, ПроцентСкидки, СуммаСкидки, НДС*,

**ПризнакСуммаВключаетНДС, Сумма НДС, ПризнакАкцизногоНалога
НомерСтроки, ДопРеквизиты*);**

або

**БУЛЕВО НапечататьФискСтроку (ИДУстройства, Наименование*, Количество*,
Цена*, Сумма*, НомерСекции, СтавкаНДС*, ДопРеквизиты*);**

БУЛЕВО PrintNoFiscString (ИДУстройства, Коментарий);

**БУЛЕВО ЗакрытьЧек (ИДУстройства, НаличнаяОплата, ОплатаКартой,
ОплатаКредитом, ОплатаСертификатом,
СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления,
ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат, НомерЧекаЧисловой,
ФискальныйНомерСозданногоЧека);**

Параметри, відмічені зірочкою (*), є обов'язковими при передачі даних товару.

Функція **ОткрытьЧек** формує заголовок чеку.

Функції **НапечататьФискальнуюСтроку** та **НапечататьФискСтроку** призначені для заповнення чеку рядками товарами, що купуються. Ці функції рівноцінні, проте, перша з них має повний перелік параметрів товару, а друга має скорочений перелік. Обидві функції мають параметр **ДопРеквизиты**, в якому передається сформований XML-рядок з переліком реквізитів товару. Реквізити, які зазначені в XML-рядку, мають пріоритет над явно заданими параметрами функції, тому рекомендується використовувати **ДопРеквизиты**, а також функцію **НапечататьФискСтроку** з використанням цього параметру. Крім того, у випадку використання

функції **НапечататьФискСтроку**, в **ДопРеквизиты** необхідно обов'язково вказати значення для артикулу та одиниці вимірювання товару.

Функція **ЗакретьЧек** виконує завершення формування чеку, передає дані в ПРРО Cashalot, де чек підписується і відправляється на сервер ФСКО для реєстрації, та повертає в якості кінцевих параметрів фіскальний номер зареєстрованого чеку та номер для наступного чеку. Порушення послідовності функцій призведе до помилки реєстрації чеків.

ТОВ "Тест-гідрометцентр"
Тест-гідромет
УКРАЇНА, м. Київ, Адреса
ІД 24234227
ФН ПРРО 4000007619
КАСІР Пірометр П. П.
Чек №297664

НазваТовару1	
1 X 20000,03	20000,03 A
Знижка	-1000,00
НазваТовару2	
1 X 20000,03	20000,03 A
Знижка	-1000,00

Сума, грн	38000,06
ПДВ А=20%	6333,34

Заокруглення	0,04
До сплати	38000,10
ГОТІВКА, грн	20000,00
Решта, грн	999,90
КАРТКА	19000,00

297664 3/4/2021 3:04:15 PM

ФІСКАЛЬНИЙ ЧЕК ОНЛАЙН Cashalot

Друкувати звіт

Готово

Заокруглення суми по чеку реалізовано наступним чином. В функцію **ЗакретьЧек** передається значення готівки, що надана клієнтом на касі, значення оплати картою, сертифікатом та кредитом, а також заокруглена сума, розрахована за певним алгоритмом користувача (наприклад, відповідно до Постанови НБУ¹, 1-4 коп заокруглюється до 0 коп, 5-9 коп заокруглюється до 10 коп). В чеку буде відображено заокруглення, надану готівку та інші форми оплати, заокруглену суму до сплати, решту:

Для друку нефіскальних рядків допускається один виклик функції **PrintNoFiscString()**. Якщо необхідно надрукувати декілька нефіскальних рядків, необхідно побудувати параметр **Коментарий** у вигляді:

*"<Перший рядок коментаря> +Символы.ПС + <Другий рядок коментаря>
>+Символы.ПС +...+Символы.ПС +< Останній рядок коментаря >*

де **Символы.ПС** – символ переходу на новий рядок. При побудові нефіскального рядку необхідно врахувати ширину чекової стрічки та допустиму кількість символів в одному рядку. У випадку, якщо параметр

СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления не визначений (null),

¹ <https://zakon.rada.gov.ua/laws/main/v0025500-18> п.4.

або дорівнює нулю, то він замінюється сумою по всім формам оплати. В такому випадку заокруглення буде дорівнювати нулю.

Параметри функцій зазначені в Таблиці 9.1.

Таблиця 9.1 Параметри функцій для формування чеку.

Функція	Параметри функцій		
	Ім'я	Тип	Опис
ОткрытьЧек	<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
	ФискальныйЧек	БУЛЕВО	Ознака фіскального чеку
	ЧекВозврата	БУЛЕВО	Ознака чеку повернення
	АннулироватьОткрытыйЧек	БУЛЕВО	Ознака автоматичного анулювання раніше відкритого чеку
	НомерЧека	ЧИСЛО [OUT]	Номер нового чеку
	НомерСмены	ЧИСЛО [OUT]	Номер відкритої зміни
НапечататьФискальнуюСтроку	<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
	Наименование	СТРОКА	Найменування товару, відповідно до специфікації ФСКО, допускається довжина до 127 символів
	КодУКТЗЕД	СТРОКА	Код УКТЗЕД
	Артикул	СТРОКА	Артикул, відповідно до специфікації ФСКО, допускається довжина до 63 символів.
	Штрихкод	СТРОКА	Штрихкод
	ЕдиницаИзмерения	СТРОКА	Одиниця вимірювання
	НомерСекции	СТРОКА	Номер секції (не використовується)
	Количество	ЧИСЛО	Кількість товару
	Цена	ЧИСЛО	Ціна одиниці товару (в форматі <Гривні, Копійки>)
	Сумма	ЧИСЛО	Кінцева сума по позиції чеку (з врахуванням всіх націнок та знижок, в форматі <Гривні, Копійки>)
	ПроцентСкидки	ЧИСЛО	Процент знижки (ігнорується, якщо заповнена сума знижки)
	СуммаСкидки	ЧИСЛО	Сума знижки на позицію

	НДС	ЧИСЛО	ПДВ в процентах (від 0 до 99,99)
	ПризнакСуммаВключаетНДС	БУЛЕВО	Ознака, що сума включає ПДВ
	СуммаНДС	ЧИСЛО	ПДВ в процентах (від 0 до 99,99)
	ПризнакАкцизногоНалога	БУЛЕВО	Ознака, що сума включає ПДВ
	НомерСтроки	БУЛЕВО	Номер рядку (не використовується)
	ДопРеквизиты	СТРОКА	Додаткові параметри (XML таблиця)
НапечататьФискСтроку	<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
	Наименование	СТРОКА	Найменування товару, відповідно до специфікації ФСКО, допускається довжина до 127 символів
	Количество	ЧИСЛО	Кількість товару
	Цена	ЧИСЛО	Ціна одиниці товару (в форматі <Гривні, Копійки>)
	Сумма	ЧИСЛО	Кінцева сума по позиції чеку (з врахуванням всіх націнок та знижок, в форматі <Гривні, Копійки>)
	НомерСекции	СТРОКА	Номер секції (не використовується)
	СтавкаНДС	ЧИСЛО	ПДВ в процентах (від 0 до 99,99)
	ДопРеквизиты	СТРОКА	Додаткові параметри
PrintNoFiscString	<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
	Коментарий	СТРОКА	Нефіскальний рядок, коментар. Для побудови декількох рядків використовуйте +Символы.ПС +
ЗакрытьЧек	<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
	НаличнаяОплата	ЧИСЛО	Сума оплати готівкою
	ОплатаКартой	ЧИСЛО	Сума оплати картою
	ОплатаКредитом	ЧИСЛО	Сума оплати кредитом
	ОплатаСертификатом	ЧИСЛО	Сума оплати сертифікатом

СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления	ЧИСЛО	Сума до оплати, що враховує всі форми оплати, знижки та заокруглення
ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат	СТРОКА	Фіскальний номер чеку, за яким виконується повернення
НомерЧекаЧисловой	ЧИСЛО [OUT]	Номер нового чеку
ФискальныйНомерСозданногоЧека	СТРОКА [OUT]	Фіскальний номер нового чеку

*** обов'язкові параметри.**

Зауваження! Для передачі параметрів функцій з дробовою частиною (в т.ч. нульовою, наприклад, "2,00") допускається використовувати десятковий розділювач "." (крапка) або "," (кома). Якщо параметр не задано, або задано некоректно, його значення приймається рівним одиниці (1).

ДопРеквизиты

Текст в форматі XML з кодуванням UTF-8, що передається за допомогою параметрів типу **СТРОКА**. Атрибути XML-рядку вказані в Таблиці 9.2.

Нагадаємо, що параметри рядку чеку, які вказані в XML-рядку мають пріоритет над параметрами, які явно передані в функції НапечататьФискальнуюСтроку або НапечататьФискСтроку, тому допускається передача цих параметрів рядку чеку через XML-рядок.

Таблица 9.2. Атрибути XML-рядку параметру ДопРеквизиты.

Атрибут XML-рядку	Опис атрибута
NOMENCLATURE	Найменування товару, відповідно до специфікації ФСКО, допускається довжина до 127 символів
UKTZED	Код УКТЗЕД
VENDORCODE	Артикул, відповідно до специфікації ФСКО, допускається довжина до 63 символів
BARCODE	Штрихкод
DIMENSION	Одиниця вимірювання
DISCOUNT	Процент знижки (ігнорується, якщо заповнена сума знижки)
DISCOUNTAMOUNT	Сума знижки
VAT	Ставка ПДВ в процентах
PRICEINCLUDESVAT або PRICEINCLUDEVAT	Ознака, що сума включає ПДВ
AMOUNTVAT	Сума ПДВ, в форматі <Гривні, Копійки>
ISEXCISABLE	Ознака підакцизного товару, True/False
EXCISESTAMPBARCODE	Штрих-код марки акцизного податку. Поле ігнорується, якщо ознака підакцизного товару ISEXCISABLE = false.

OtherParameters	Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML). Необов'язкове поле.
-----------------	---

9. Відміна відкритого чеку

Для відміни раніше відкритого чеку використовується функція

БУЛЕВО ОтменитьЧек(**ИДУстройства**)

Параметри функції зазначені в Таблиці 10.1.

Таблиця 10.1 Параметри функції відміни чеку.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою

Приклад:

Результат = ОтменитьЧек(*ОбъектДрайвера, Параметры, ПараметрыПодключения, ВыходныеПараметры*);

10. Отримання інформації про поточний стан ПРРО

Для отримання інформації про поточний стан використовується функція

БУЛЕВО ПолучитьТекущееСостояние (ИДУстройства, ВходныеПараметры, ВыходныеПараметры)

Параметри функції зазначені в Таблиці 11.1.

Таблиця 11.1 Параметри функції поточного стану ПРРО.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
<u>ВходныеПараметры</u>	СТРОКА	Таблиця вхідних параметрів, параметр ігнорується.
<u>ВыходныеПараметры</u>	СТРОКА	XML-структура параметрів з поточними значеннями

Таблиця вихідних параметрів має вигляд XML-структури з переліком параметрів, які наведено в Таблиці 11.2., та їх значень.

Таблиця 11.2 Параметри XML-структури вихідних параметрів.

Ім'я параметру	Значення
CheckNumber	Локальний номер останнього чеку
CashBalance	Залишок готівки в касі
IsOfflineMode	Стан офлайн-режиму каси: 0 – каса в онлайн-режимі; 1 – каса в офлайн-режимі
NextLocalNumber	Наступний локальний номер чеку
LastCheckFiscalNumber	Фіскальний номер останнього чеку
ShiftNumber	Номер поточної відкритої зміни/ останньої закритої зміни
ShiftID	Ідентифікатор зміни
ShiftFiscalNumber	Фіскальний номер зміни

ShiftState	Стан зміни: 1 – закрита, 2 – відкрита.
DateTime	Дата і час формування фіскального документу

Приклад:

ИДУстройства = "4000001234";

ВыходныеПараметры = "";

Результат = ОбъектДрайвера.ПолучитьТекущееСостояние (*ИДУстройства* , "" ,*ВыходныеПараметры*);

Приклад вихідних параметрів:

ВыходныеПараметры =

```
<?xml version = "1.0" encoding= "UTF-8" ?>< DocumentOutputParameters > <
Parameters CheckNumber = "0" CashBalance = "0" IsOfflineMode = "0"
NextLocalNumber = "2472" LastCheckFiscalNumber = "" ShiftNumber= "2471" ShiftID =
"96acfaa1-22bd-4d6c-9f42-b5dbe958f6d1" ShiftFiscalNumber = "510130" ShiftState=
"2" DateTime= "0001-01-01T00:00:00" /> </ DocumentOutputParameters >
```

11. Надрукувати періодичний звіт по датам

Функція **PrintPReportDate** надрукувати періодичний звіт по датам використовується для можливості друку періодичного звіту, використовуючи проміжки часу, що фіксується значеннями:

(Усі параметри рядкові, для сумісності).

//Надрукувати періодичний звіт по датам(ФіскальнийНомерРРО, датаПочаткуПеріоду-\'dd.MM.yyyy\'', датаЗакінченняПеріоду-\'dd.MM.yyyy\'', ОзнакаСкороченогоЗвіту-\'TRUE/FALSE\'')

БУЛЕВО НапечататьПериодическийОтчетПоДатам (*ИДУстройства*, *ВходныеПараметры*, *ВыходныеПараметры*, *Сокращенный*)

Параметри

Ім'я	Тип	Опис
<u><i>ИДУстройства</i></u>	String	Ідентифікатор пристрою
<u><i>ВходныеПараметры</i></u>	DATE	Початок періоду в форматі ДДММГГГГ
<u><i>ВыходныеПараметры</i></u>	DATE	Кінець періоду в форматі ДДММГГГГ
<u><i>Сокращенный</i></u>	BOOL	ОзнакаСкороченогоЗвіту

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення початку періоду та дату закінчення заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, ОзнакаСкороченогоЗвіту, Base64 рядок XML структури чеку	
DeviceID	String	Ідентифікатор устройства	ФіскальнийНомерРРО
DateBegin	DATE	Дата начала периода	dd.MM.yyyy
DateEnd	DATE	Дата конца периода	dd.MM.yyyy
IsShort	BOOL	Признак сокращенного отчета	TRUE/FALSE

Приклад:

// Функция осуществляет печать периодического отчёта по датам.

Функция НапечататьПериодическийОтчетПоДатам(ОбъектДрайвера, Параметры,
ПараметрыПодключения, ВходныеПараметры, ВыходныеПараметры)

Результат = Истина;

ОбщегоНазначенияКлиентСервер.СообщитьПользователю(НСтр("ru='Печать периодического отчета по датам, с даты ' ;uk='Друк періодичного звіту по датах, з дати '"') +

Формат(ВходныеПараметры[0], "ДФ=dd.MM.yyyy") +

НСтр("ru=' по дату ' ;uk=' за дату '"') +

Формат(ВходныеПараметры[1], "ДФ=dd.MM.yyyy"));

Если ПараметрыПодключения.ТипОборудования = "ФискальныйРегистратор" Тогда

// Ветка на стандарт 1.4. Поддержка команды драйвера "НапечататьПериодическийОтчетПоДатам".

Если ПараметрыПодключения.РевизияИнтерфейса > 1004 Тогда

Попытка

Результат =

ОбъектДрайвера.НапечататьПериодическийОтчетПоДатам(ПараметрыПодключения.ИДУстройства,
ВходныеПараметры[0],ВходныеПараметры[1],Результат);

Исключение

Результат = Ложь;

ВыходныеПараметры.Очистить();

ВыходныеПараметры.Добавить(999);

ВыходныеПараметры.Добавить(НСтр("ru='Ошибка вызова метода

<ОбъектДрайвера.НапечататьПериодическийОтчетПоДатам>.' ;uk='Помилка виклику методу

<ОбъектДрайвера.НапечататьПериодическийОтчетПоДатам>.'"') + Символы.ПС + ОписаниеОшибки());

Возврат Результат;

КонецПопытки;

КонецЕсли;

КонецЕсли;

// Заполнение выходных параметров

ВыходныеПараметры.Добавить(1);

ВыходныеПараметры.Добавить(1);

ВыходныеПараметры.Добавить(0);

ВыходныеПараметры.Добавить(ТекущаяДата());

Возврат Результат;

КонецФункции

12. Друк довільної текстової інформації

Для друку на чекову стрічку довільного тексту використовується функція

БУЛЕВО НапечататьТекстовыйДокумент(ИДУстройства, Текст)

Після виклику відображається вікно з зображенням тексту і можливістю роздрукувати.

У випадку, коли задано автоматичну відправку чеків на друк і принтер для друку за замовчуванням, друк заданого тексту відбувається без попереднього перегляду:

УстановитьПараметр("AUTOPRINTMODE", "true");

УстановитьПараметр("DEFAULTPRINTERNAME", "xprint80-name");

Функцію можливо викликати без відкриття касової зміни, але необхідно авторизуватися.

Параметри функції зазначені в Таблиці 11.1.

Таблиця 12.1 Параметри функції поточного стану ПРРО.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
Текст	СТРОКА	Довільний текст, який передається на друк.

Приклад:

ИДУстройства = "4000001234";

Текст = "Еней був парубок моторний"+Символы.ПС+"І хлопець хоть куди козак,"+Символы.ПС+"Удавься на всеє зле проворний,"+Символы.ПС+"Завзятіший од всіх бурлак."";

Результат = ОбъектДрайвера.НапечататьТекстовыйДокумент (*ИДУстройства* , *Текст*);

Еней був парубок моторний
І хлопець хоть куди козак,
Удавься на всеє зле проворний,
Завзятіший од всіх бурлак.

Друк

Готово

13. Повернення товару

Виконання операції повернення товару відбувається аналогічно операції формування чеку (див. Реєстрація чеків), але із зазначенням параметрів

Ім'я параметру	Значення
ЧекВозврата	True
ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат	Фіскальний номер чеку, за яким виконується повернення

Приклад:

ИмяПараметра = "DeviceIDFnRRO";

ЗначениеПараметра = "4000001234";

Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);

ИмяПараметра = "PathToCashalotDir";

ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";

Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);

ЧекВозврата = ВходныеПараметры[1];

ФискальныйЧек = ВходныеПараметры[0];

Результат = ОткрытьЧек(ОбъектДрайвера, ВходныеПараметры, ПараметрыПодключения, ЧекВозврата, ФискальныйЧек, ВыходныеПараметры);

ИДУстройства = "4000001234";

Наименование = "Товар1";

КодУКТЗЕД = "000001";

Артикул = "артикул1";

Штрихкод = "9999999999";

ЕдиницаИзмерения = "шт";

НомерСекции = 1;

Количество = 5;

Цена = 50,45;

Сумма = 239,64;

ПроцентСкидки = 5;
 СуммаСкидки = 12,61;
 НДС = 20;
 ПризнакСуммаВключаетНДС = 1;
 СуммаНДС = 50,45;
 ПризнакАкцизногоНалога = 0;
 Ответ = ОбъектДрайвера.НапечататьФискСтроку(ИДУстройства, Наименование, Количество, Цена, Сумма, НомерСекции, СтавкаНДС, ДопРеквизиты);
 СуммаНаличнойОплаты = 239,60;
 СуммаБезналичнойОплаты1 = 0;
 СуммаБезналичнойОплаты2 = 0;
 СуммаБезналичнойОплаты3 = 0;
 Ответ = ОбъектДрайвера.ЗакрытьЧек(ИДУстройства, СуммаНаличнойОплаты, СуммаБезналичнойОплаты1, СуммаБезналичнойОплаты2, СуммаБезналичнойОплаты3, СуммаЧека, ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат, НомерСозданногоЧека, ФискальныйНомерСозданногоЧека);

14. Службове внесення

Для виконання операції службового внесення використовується функція

БУЛЕВО НапечататьЧекВнесенияВыемки (ИДУстройства, Сумма)

Параметри функції зазначені в Таблиці 14.1.

Таблица 14.1 Параметри функції службового внесення.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
Сумма	ЧИСЛО	Сума внесення у форматі <Грн, Коп>

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
 ЗначениеПараметра = "4000001234";
 Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);
 ИмяПараметра = "PathToCashalotDir";
 ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
 Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);
 Результат = ОбъектДрайвера.НапечататьЧекВнесенияВыемки (ПараметрыПодключения.ИДУстройства, ?(ТипИнкассации = 1, Сумма, -Сумма));

15. Службова видача

Для виконання операції службової видачі використовується функція

БУЛЕВО НапечататьЧекВнесенияВыемки (ИДУстройства, Сумма)

з від'ємним значенням **Сумма** (див. розділ Службове внесення).

Параметри функції зазначені в Таблиці 15.1.

Таблица 15.1 Параметри функції службового внесення.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою
Сумма	ЧИСЛО	(від'ємна) сума видачі у форматі <Грн, Коп>

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
 ЗначениеПараметра = "4000001234";
 Ответ = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, ЗначениеПараметра);
 ИмяПараметра = "PathToCashalotDir";
 ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";

Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Результат = ОбъектДрайвера.НапечататьЧекВнесенияВыемки (*ПараметрыПодключения*.ИДУстройства,
 ?(ТипИнкассации = 0, *Сумма*, -*Сумма*));

16. Формування Х-звіту

Для виконання операції формування звіту без гасіння (Х-звіту) використовується функція
БУЛЕВО НапечататьОтчетБезГашения(ИДУстройства)

Параметри функції зазначені в Таблиці 16.1.

Таблиця 16.1 Параметри функції Х-звіту.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Результат = ОбъектДрайвера.НапечататьОтчетБезГашения(*ПараметрыПодключения*.ИДУстройства);

17. Формування Z-звіту

Для формування Z-звіту використовується функція

БУЛЕВО НапечататьОтчетСГашением(ИДУстройства)

Параметри функції зазначені в Таблиці 15.1.

Таблиця 17.1 Параметри функції Z-звіту.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
Результат = ОбъектДрайвера. НапечататьОтчетСГашением(*ПараметрыПодключения*.ИДУстройства);

18. Закриття зміни

Для виконання операції службового внесення використовується функція

БУЛЕВО ЗакрытьСмену(ИДУстройства)

Параметри функції зазначені в Таблиці 18.1.

Таблиця 18.1 Параметри функції закриття зміни.

Ім'я	Тип	Опис
<u>ИДУстройства</u>	СТРОКА	Ідентифікатор пристрою

Приклад:

ИмяПараметра = "DeviceIDFnRRO";
ЗначениеПараметра = "4000001234";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);
ИмяПараметра = "PathToCashalotDir";
ЗначениеПараметра = "C:\Users\User1\AppData\Local\Cashalot";
Ответ = ОбъектДрайвера.УстановитьПараметр(*ИмяПараметра*, *ЗначениеПараметра*);

Результат = ЗакретьСмену(ОбъектДрайвера, Параметры, ПараметрыПодключения, ВходныеПараметры, ВыходныеПараметры);

Порядок виклику функцій при виконанні касових операцій

При роботі з касою в системі BAS рекомендується послідовність виклику функцій ПРРО Cashalot, зазначена в Таблиці 19.

Таблиця 19. Порядок виклику функцій при роботі з касою.

Касова операція	Послідовність функцій драйверу
Відкриття зміни	УстановитьПараметр ОткрытьСмену
Фіскалізація чеків	УстановитьПараметр ОткрытьЧек НапечататьФискальнуюСтроку або НапечататьФискСтроку ЗакретьЧек або ОтменитьЧек
Повернення товару	УстановитьПараметр ОткрытьЧек НапечататьФискальнуюСтроку або НапечататьФискСтроку ЗакретьЧек або ОтменитьЧек
Х-звіт	УстановитьПараметр НапечататьОтчетБезГашения
З-звіт	УстановитьПараметр НапечататьОтчетСГашением
Закриття зміни	УстановитьПараметр ЗакретьСмену
Внесення грошей	УстановитьПараметр НапечататьЧекВнесенияВыемки
Видача грошей	УстановитьПараметр НапечататьЧекВнесенияВыемки
Службові операції	ПолучитьОшибку або ПолучитьПараметры або ТестУстройства

Приклад реалізації підключення драйверу до Розрахунково-облікової платформи (.erf)

&НаКлиенте

Процедура КоманднаяПанельТаблицаДокументовМодосЗапуск (Команда)

Если

ПодключитьВнешнююКомпоненту ("C:\Users\User1\Cashalot\CashaLotApiDrv32.dll", "CashaLotApiDrv") Тогда

ОбъектДрайвера = Новый("AddIn.CashaLotApiDrv.CashaLotApiDrv");

ИмяПараметра = "PathToCashalotDir";

Параметр = " C:\Users\User1\Cashalot\CashaLotApiDrv32.dll";

```

УстановленПутьCashalot = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра,
Параметр);
Сообщить(" Установлен Путь Cashalot : " + УстановленПутьCashalot);

ИмяПараметра = "DeviceIDFnRRO";
Параметр = "4000007619";

ФН = ОбъектДрайвера.УстановитьПараметр(ИмяПараметра, Параметр);

Сообщить(" Установлен Номер Фискальной Кассы: " + ФН);
таблицапарам = "";
ответ = ОбъектДрайвера.ПолучитьПараметры (таблицапарам);
Сообщить(" Параметры: " + таблицапарам);

ответ = ОбъектДрайвера.Подключить("4000007619");
Сообщить(" Подключить: " + ответ);

ответ = ОбъектДрайвера.ОткрытьСмену("4000007619");
Сообщить(" ОткрытьСмену: " + ответ);

//службове внесения
ответ = ОбъектДрайвера.НапечататьЧекВнесенияВыемки ("4000007619", 50);
Сообщить(" Службове внесения: 50 грн " + ответ);

ФискальныйНомерКассы ="4000007619";
НомерСмены = 0;
НомерЧека = 0;
НомерЧекаВозврата = 0;
ФискальныйЧек = Истина;
ТипЧека = 0;

ответ = ОбъектДрайвера.ОткрытьЧек(ФискальныйНомерКассы, ФискальныйЧек,
ТипЧека, Истина, НомерЧека, НомерСмены);
Сообщить("===== Відкрито чек ===== " + ответ );
Сообщить("ФискальныйНомерКассы " + ФискальныйНомерКассы );
Сообщить("НомерЧека " + НомерЧека );
Сообщить("НомерСмены " + НомерСмены );

ИДУстройства = "4000007619";
Наименование = "НазваТовару1";
КодУКТЗЕД = "";
Артикул = "АртикулТовару1";
Штрихкод = "123456789";
ЕдиницаИзмерения = "шт";
НомерСекции = "";
Количество = "1";
Цена = "183";
Сумма = "181,17";
ПроцентСкидки = "";
СуммаСкидки = "1.83";
НДС = "20";
ПризнакСуммаВключаетНДС = "true";
СуммаНДС = "30.2";
ПризнакАкцизногоНалога = "false";
НомерСтроки = "1";
ДопРеквизиты = "";

```

```

        ответ = ОбъектДрайвера.НапечататьФискальнуюСтроку
(ИДУстройства, Наименование, КодУКТЗЕД, Артикул, Штрихкод, ЕдиницаИзмерения, НомерСекци
и, Количество, Цена, Сумма, ПроцентСкидки, СуммаСкидки, НДС, ПризнакСуммаВключаетНДС, Сум
маНДС, ПризнакАкцизногоНалога, НомерСтроки, ДопРеквизиты);
        Сообщить ("===== Фискальный рядок 1 ===== " + ответ );
        ответ = ОбъектДрайвера.НапечататьФискальнуюСтроку
(ИДУстройства, Наименование, КодУКТЗЕД, Артикул, Штрихкод, ЕдиницаИзмерения, НомерСекци
и, Количество, Цена, Сумма, ПроцентСкидки, СуммаСкидки, НДС, ПризнакСуммаВключаетНДС, Сум
маНДС, ПризнакАкцизногоНалога, НомерСтроки, ДопРеквизиты);
        Сообщить ("===== Фискальный рядок 2 ===== " + ответ );

        НаличнаяОплата                                = "300";
        ОплатаКартой                                    = "10";
        ОплатаКредитом                                  = "20";
        ОплатаСертификатом                              = "37";
        СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления = "362,34";
        ФискальныйНомерЧекаПоКоторомуПроизводитсяВозврат = "";
        НомерЧекаЧисловой                               = "";
        ФискальныйНомерСозданногоЧека                   = "";

        ответ =
ОбъектДрайвера.PrintNoFiscString(ИДУстройства, "#####
##" + Символы.ПС+"#Тестовий коментар#"
+Символы.ПС+"#####");
        Сообщить ("===== Нефискальні рядки ===== " + ответ );

        ответ =
ОбъектДрайвера.ЗакрытьЧек(ИДУстройства, НаличнаяОплата, ОплатаКартой, ОплатаКредитом
, ОплатаСертификатом, СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления, Ф
искальныйНомерЧекаПоКоторомуПроизводитсяВозврат, НомерЧекаЧисловой, ФискальныйНомер
СозданногоЧека);
        Сообщить ("===== Закрито чек ===== " + ответ );
        Сообщить ("ФН чека " + ФискальныйНомерСозданногоЧека );
        Сообщить ("локальний номер чека " + НомерЧекаЧисловой );

        ответ = ОбъектДрайвера.НапечататьОтчетБезГашения(ИДУстройства);
        Сообщить ("===== X-звіт ===== " + ответ );

        //службова видача
        ответ = ОбъектДрайвера.НапечататьЧекВнесенияВыемки ("4000007619", -20);
        Сообщить (" Службова видача: 20 грн " + ответ);

        ответ = ОбъектДрайвера.ЗакрытьСмену("4000007619");
        Сообщить ("ЗакрытьСмену: " + ответ );

        КонецЕсли;

КонецПроцедуры

```

Варіант 2a – API PPPO Cashalot з COM
інтерфейсом CashalotApi32.dll, CashalotApi64.dll

В останньому оновленні заявила можливість використання декількох екземплярів ПРРО Cashalot, що значно розширив перелік варіантів використовуватися для вже існуючих параметрів та функцій.

Для зручності користувачів та запобігання дублюванню ключових назв в вже існуючих варіантів підключення та використання нових компонентів та бібліотек, додано поняття **Обов'язкові параметри***.

В склад яких можуть входити також додаткові залежності від призначення функції входять додаткові **Обов'язкові параметри***, що можуть бути не обов'язковими для інших функцій.

Обов'язкові параметри* не залежать від Інтерфейсу користувача та повинні бути використані окремо в випадку з Одним екземпляром ПРРО [SetParameter](#).

Для використання розрахунково-обліковою платформою або системою декількох екземплярів ПРРО, використовується параметр [SetParameterEx](#).

Використовують бібліотеки елементів інтерфейсу **CashalotApi**, що розміщені в корневих каталогах шлях до яких необхідно вказати при встановленні підключення до об'єктів COM, кожного екземпляру ПРРО Cashalot окремо. Шлях до корневих каталогів є **Обов'язковим параметри* PathToCashalotDir***. (див. Встановлення налаштувань бібліотеки).

```
regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApi32.dll
```

(для 32-бітної платформи)

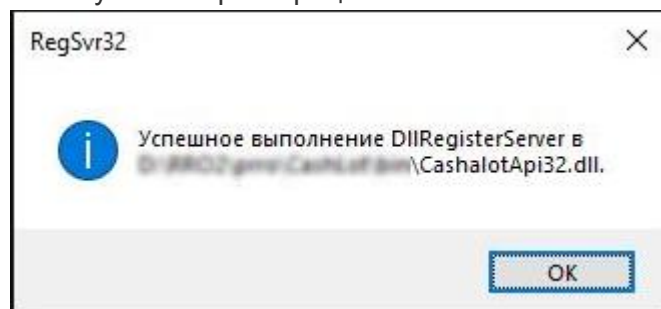
```
regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApi64.dll
```

(для 64-бітної платформи)

Наприклад:

```
regsvr32 C:\Users\User1\AppData\Local\Cashalot\CashalotApi32.dll
```

Після успішної реєстрації бібліотеки з'явиться повідомлення зразку:



В бібліотеках CashalotApi реалізована обробка наступних функцій:

NEW

1. [SetParameter](#)
2. [SetParameterEx](#)
3. [GetVersion](#)
4. [GetInterfaceRevision](#)
5. [ServiceInput](#)
6. [ServiceInputEx](#)
7. [ServiceOutput](#)
8. [ServiceOutputEx](#)
9. [OpenShift](#)
10. [FiscalizeCheck](#)
11. [FiscalizeReturnCheck](#)
12. [GetReceiptLocalNumberByFiscalNumber](#)
13. [GetReceiptFiscalNumberByLocalNumber](#)
14. [GetReceiptXML](#)
15. [GetFirstReceiptByPeriod](#)
16. [GetFirstReceiptByShift](#)
17. [GetNextReceipt](#)
18. [GetFirstZReportByPeriod](#)

- 19. **GetNextZReport**
- 20. **GetZReportByShift**
- 21. **ShowReceipt**
- 22. GetLastError
- 23. PrintXReport
- 24. **GetXReport**
- 25. **GetLastZReport**
- 26. PrintZReport
- 27. **CloseShift**

Функції, виділені жирним шрифтом, повертають відповідь у вигляді структури **CashalotApiRetVal**, що може містити такі властивості:

Властивість	Тип	Значення
Description	STRING	Опис
JsonVal	STRING	JSON-структура
ReceiptFiscalNum	STRING	Фіскальний номер чеку
ReceiptLocalNum	STRING	Локальний номер чеку
Return	BOOL	Успішність операції
ShiftID	STRING	Ідентифікатор касової зміни
XML	STRING	XML-структура документа
Type	INTEGER	Тип документа (див. FiscalizeCheck())

Приклади використання COM-функцій наведено у вигляді

```
Ret = App.GetVersion()
```

1. Встановлення налаштувань API бібліотеки [SetParameter](#)

[SetParameter](#) щодо встановлення налаштувань підключення API бібліотек Функція призначена для встановлення значень параметрів налаштувань API бібліотеки. Рекомендовано встановити всі параметри на початку роботи.

Параметри

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

Значення Name

Name	Тип	Опис
DeviceIDFnRRO	STRING	Фіскальний номер ПРРО * ²
PathToCashalotDir	STRING	Шлях до каталогу з встановленим ПРРО Cashalot * ³

²

³ Обов'язкові параметри

NOINTERFACEMODE	STRING	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).
NOAUTOOPENSHIFT	STRING	Перевірка стану зміни (відкрита або закрита). Якщо зміна закрита при виклику будь-якої COM-функції, зміна автоматично відкривається. "False" – автоматично відкривати зміну, якщо вона закрита. "True" – не відкривати автоматично зміну. В цьому випадку робота COM-функції буде перервана і результат виконання буде повернуто у вигляді помилки.
USETOKEN	STRING	Чи використовується токен для авторизації. "True" – використовується "False" – не використовується (використовується файловий ключ) Зауваження. Параметр USETOKEN має бути заданий після параметрів PathToCashlotDir, DeviceIDFnRRO та NOINTERFACEMODE. Для пришвидшення завантаження програми рекомендовано не використовувати токени.
PathToCertificate	STRING	Шлях до папки з файловим ключем користувача. Зауваження. В папці повинні міститися ключ і сертифікат тільки одного користувача. Якщо касою користуються декілька касирів, необхідно помістити їх сертифікати і ключі в окремі папки.
PwdToCertificate	STRING	Пароль користувача
DEFAULTPRINTERNAME	STRING	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 Кашалот
AUTOPRINTMODE	STRING	Параметр визначає автоматичну відправку на друк чеку або службового звіту без відображення інформаційного вікна "True" – відправляти на друк "False" – не відправляти на друк Якщо задано принтер за замовчуванням (DEFAULTPRINTERNAME), то чек одразу друкується, в іншому випадку відображається вікно вибору принтера для друку.
NOPRINTMODE	STRING	Дозвіл на друк чеку або службового звіту в режимі AUTOPRINTMODE = "True" "True" – заборонити друк "False" – дозволити друк
SHOWREPORTADDITIONALINFO	STRING	Відображає додаткову інформацію Z-звіту чеку печатної форми

NEW

NEW

		Структура чеку Z-звіту: 1) інформація по продажам, поверненням, службовим операціям та реквізітам ППРО 2) відокремити основні дані пунктирною лінією. 3) виводити напис Додаткова інформація, під написом виводити такі дані: - Залишок готівки в касі - Сума знижки - Сума націнки SHOWREPORTADDITIONALINFO" = "TRUE" "TRUE" - Відображає додаткову інформацію "False" – Не відображає додаткову інформацію.⁴
	<u>OtherParametrs</u>	STRING
		Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML). Необов'язкове поле.

Відповідь

	BOOL	Успішність операції
--	------	---------------------

Приклад:

```

App = CreateObject("AddIn.CashaLotApi")

param = "PathToCashalotDir"
paramVal = "C:\Users\User\AppData\Roaming\Cashalot"
ret = App.SetParameter(param, paramVal)
param = "DeviceIDFnRRO"
paramVal = "4000007621"
ret = App.SetParameter(param, paramVal)
param = "NOINTERFACEMODE"
paramVal = "True"
ret = App.SetParameter(param, paramVal)
param = "NOAUTOOPENSHIFT"
paramVal = "True"
ret = App.SetParameter(param, paramVal)
param = "USETOKEN"
paramVal = "False"
ret = App.SetParameter(param, paramVal)
param = "PathToCertificate"
paramVal = "C:\Users\User\Certificate\Cashier1\"
ret = App.SetParameter(param, paramVal)
param = "PwdToCertificate"
paramVal = "Use0n!yStr0ngPa$swords"
ret = App.SetParameter(param, paramVal)
param = "DEFAULTPRINTERNAME"
paramVal = "Microsoft Print to PDF"

```

⁴ Фіскальний звітний чек (Z-звіт) та/або електронний фіскальний звітний чек – це документ, створений програмним РРО, що містить дані денного звіту, зареєстрований ФСКО, і інформація в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити фіскального звітного чека за встановленою формою.

Відповідно печатна форма у ППРО та електронна форма Z-звіту зареєстрована на ФСКО повинні містити та відображати ідентичні дані. у xsd-форматі Z-звіту (zger01.xsd) відсутній відповідний тег для відображення залишку готівки в касі у xml-форматі Z-звіту.

Також реквізит «Залишок готівки в касі» відсутній у затверджених Вимогах щодо забезпечення виконання фіскальних функцій програмними реєстраторами розрахункових операцій для різних сфер застосування, затверджених Постановою Кабінету міністрів України №199 від 18 лютого 2002 р. (із змінами, внесеними згідно з Постановою КМ №629 від 22.07.2020 р.).

Тому додавати на печатну форму Z-звіту відображення реквізиту «Залишок готівки в касі», який відсутній у xsd-форматі Z-звіту, категорично не можна.

```
ret = App.SetParameter(param, paramVal)
param = "AUTOPRINTMODE"
paramVal = "True"
ret = App.SetParameter(param, paramVal)
param = "NOPRINTMODE"
paramVal = "False"
ret = App.SetParameter(param, paramVal)
```

Приклад відповіді:

```
ret = True;
```

NEW

2. Встановлення налаштувань API бібліотеки [SetParameterEx](#)

Функція призначена для **використання** розрахунково-обліковою платформою або системою **декількох екземплярів ПРРО** одночасно. Перед початком роботи необхідно задати всі обов'язкові параметри. Якщо використовується робота одночасно з декількома ПРРО то встановлення параметрів здійснюється виключно через [SetParameterEx](#). У такому випадку [SetParameter](#) не використовується.

При встановленні параметрів за допомогою SetParameterEx, обов'язково потрібно зазначити шлях до кореневого каталогу кожного екземпляру ПРРО Cashalot та ФН номер каси окремо. Обов'язково необхідно створити для кожного екземпляру ПРРО окремо каталог та вказати шлях до кожного екземпляру індивідуально. За потреби вкласти в цей каталог сертифікати користувачів. Власне таким чином може бути реалізовано розподілення доступу між користувачами на основі вкладених індивідуально ключів для кожного каталогу окремо.

Приклад вказаний нижче:

Параметри

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

Значення Name

Name	Тип	Опис
DeviceIDFnRRO	STRING	Фіскальний номер ПРРО *
PathToCashalotDir	STRING	Шлях до каталогу з встановленим ПРРО Cashalot *
NOINTERFACEMODE	STRING	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).
NOAUTOOPENSIFT	STRING	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 COM-функції буде перервана і результат виконання буде повернуто у вигляді помилки.
USETOKEN	STRING	Чи використовується токен для авторизації. "True" – використовується "False" – не використовується (використовується файловий ключ) Зауваження. Параметр USETOKEN має бути заданий після параметрів PathToCashalotDir, DeviceIDFnRRO та NOINTERFACEMODE. Для пришвидшення завантаження програми рекомендовано не використовувати токени.
PathToCertificate	STRING	Шлях до папки з файловим ключем користувача. Зауваження. В папці повинні міститися ключ і сертифікат тільки одного користувача. Якщо касою користуються декілька касирів, необхідно помістити їх сертифікати і ключі в окремі папки.
PwdToCertificate	STRING	Пароль користувача
DEFAULTPRINTERNAME	STRING	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 Кашалот
AUTOPRINTMODE	STRING	Параметр визначає автоматичну відправку на друк чеку або службового звіту без відображення інформаційного вікна "True" – відправляти на друк "False" – не відправляти на друк Якщо задано принтер за замовчуванням (DEFAULTPRINTERNAME), то чек одразу друкується, в іншому випадку відображається вікно вибору принтера для друку.
NOPRINTMODE	STRING	Дозвіл на друк чеку або службового звіту в режимі AUTOPRINTMODE = "True" "True" – заборонити друк "False" – дозволити друк
SHOWREPORTADDITIONALINFO	STRING	Відображає додаткову інформацію Z-звіту чеку печатної форми Структура чеку Z-звіту: 1) інформація по продажам, поверненням, службовим операціям та реквізітам ППРО 2) відокремити основні дані пунктирною лінією. 3) виводити напис Додаткова інформація, під написом виводити такі дані: - Залишок готівки в касі - Сума знижки - Сума націнки

NEW

		SHOWREPORTADDITIONALINFO ="TRUE" "TRUE" - Відображає додаткову інформацію "False" – Не відображає додаткову інформацію.
OtherParametrs	STRING	Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML). Необов'язкове поле.

Відповідь

	BOOL	Успішність операції
--	------	---------------------

Приклад:

```
Set App = CreateObject("AddIn.CashaLotApi")

App.SetParameterEx " FiscalNumberRRO1", "PathToCashalotDir",
"C:\Users\User\AppData\Roaming\Cashalot1\"
App.SetParameterEx " xxxxxxxxxxxx ", "DeviceIDFnRRO", " xxxxxxxxxxxx "
set ret1 = App.OpenShift ("xxxxxxxxxxx")
MsgBox "resultOpenShift1.Json" & ret1.JsonVal

App.SetParameterEx " FiscalNumberRRO2", "PathToCashalotDir",
"C:\Users\User\AppData\Roaming\Cashalot2\"
App.SetParameterEx " yyyyyyyyyy ", "DeviceIDFnRRO", " yyyyyyyyyy "
set ret2 = App.OpenShift ("yyyyyyyyyyy ")
MsgBox "resultOpenShift2.Json" & ret2.JsonVal
Set ret3 = App.CloseShift ("xxxxxxxxxxx ")
MsgBox ret3.JsonVal
set ret2 = App.CloseShift ("yyyyyyyyyyy ")
MsgBox ret4.JsonVal
```

Приклад відповіді:

```
ret = True;
```

3. GetVersion – Отримання номеру версії API бібліотеки

Параметри

Функція не має вхідних параметрів.

Відповідь

	STRING	Номер версії API бібліотеки
--	--------	-----------------------------

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetVersion()
```

Приклад відповіді:

```
Ret = "1,1";
```

4. GetInterfaceRevision – Отримання номеру ревізії API бібліотеки

Параметри

Функція не має вхідних параметрів.

Відповідь

	STRING	Номер ревізії API бібліотеки
--	--------	------------------------------

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetInterfaceRevision ()
```

Приклад відповіді:

```
Ret = "3003";
```

5. OpenShift – Відкриття зміни

При виклику цієї функції API бібліотеки ПРРО Cashalot відкриває вікно авторизації, в якому потрібно зазначити папку з файловим ключем касира або старшого касира, ввести пароль, після чого відбудеться відкриття касової зміни.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumberRRO</u>	STRING	Ідентифікатор пристрою

Для відкриття зміни в "тихому" режимі без відображення вікна авторизації необхідно за допомогою функції SetParameter() встановити параметр

NOINTERFACEMODE = True

При виклику функцій, якщо зміна не відкрита, відбувається автоматичне відкриття зміни з викликом функції OpenShift(). Для запобігання автоматичного відкриття зміни за допомогою функції SetParameter() встановіть параметр

NOAUTOOPENSIFT= True

В цьому випадку, поки не буде явно відкрито зміну викликом **OpenShift()**, виклик інших функцій, які вимагають відкритої зміни, буде перериватись з поверненням помилки:

"ErrorString": "Операція не можлива, необхідно відкрити зміну"

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	STRING	JSON-структура, що містить значення успішності операції, помилки, ID зміни
Return	BOOL	Успішність операції
ShiftID	STRING	Ідентифікаційний номер поточної зміни

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
Ret = App.OpenShift("4000007621")
```

Приклад відповіді:

```
Ret.Description = ""
```

```
Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"1a9eb031-b1ce-4ee7-8a11-329824d895a7"}}
```

```
Ret.Return = True
```

```
Ret.ShiftID = 1a9eb031-b1ce-4ee7-8a11-329824d895a7
```

або:

```
Ret.Description = "Авторизація відмінена користувачем"
```

```
Ret.JsonVal = {"Ret":false,"ErrorString":"Авторизація відмінена користувачем", "Values":{}}
```

```
Ret.Return = False
```

Ret.ShiftID =""

6. FiscalizeCheck – Фіскалізація чека продажу товарів, послуг

Для створення та фіскалізації чеку продажу товарів та/або послуг необхідно сформувати JSON структуру JSONGoodsLstData, що міститиме дані номенклатури, та структуру JSON структуру JSONPayData з формами оплати

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	STRING	Ідентифікатор пристрою
strJSONData	STRING	JSON структура з описом проданих товарів або послуг
strJSONPayData	STRING	JSON структура з підсумками оплати по чеку

Структура **JSONGoodsLstData** містить масив **ReceiptLst**, кожен елемент якого містить параметри номенклатури, та об'єкти **Comment** і [DocType](#):

Поле	Значення
VendorCode*	Артикул
Name	Найменування товару
DocNumber**	Номер договору (при використанні ломбардних послуг)
CdUKDZED	Код УКТЗЕД
CdDKPP	Код ДКПП
GoodsType	Тип номенклатури: 1 – товар 2 – послуга 3 – ломбардна послуга ** За замовчуванням товар, якщо параметр не задано.
Barcode	Штрих-код
UnitType*	Одиниця вимірювання
Quantity*	Кількість товару
Price*	Ціна одиниці товару (в форматі <Гривні, Копійки>)
Amount*	Кінцева сума по позиції чека (з врахуванням всіх націнок та знижок, у форматі <Гривні, Копійки>)
DiscountPrc	Процент знижки (ігнорується, якщо заповнена сума знижки)

NEW

DiscountSum	Сума знижки на позицію
VATRate	ПДВ в процентах (від 0 до 99,99)
IsPriceIncludeVAT	Ознака, що ціна враховує ПДВ
SumVAT	ПДВ в процентах (від 0 до 99,99)
IsExcise	Ознака підакцизного товару
ExciseStampBarcode	Штрих-код марки акцизного податку. Поле ігнорується, якщо ознака підакцизного товару IsExcise = false.
OtherParametrs	Додаткові параметри у вигляді XML таблиці

NEW

NEW

*** обов'язкові параметри при передачі даних товару.**

** Ломбардні операції описані в розділі ["Ломбардні послуги"](#).

Зауваження! Для передачі параметрів функцій з дробовою частиною (в т.ч. 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 явно передані в ReceiptLst.

Атрибути XML-рядка [OtherParametrs](#)

XML атрибут	Опис атрибута
NOMENCLATURE *	Найменування товару
UKTZED	Код УКТЗЕД
VENDORCODE *	Артикул
BARCODE	Штрих-код
DIMENSION *	Одиниця вимірювання
DISCOUNT	Процент знижки (ігнорується, якщо заповнена сума знижки)
DISCOUNTAMOUNT	Сума знижки
VAT *	Ставка ПДВ в процентах
PRICEINCLUDESVAT або PRICEINCLUDEEVAT	Ознака, що ціна враховує ПДВ
AMOUNTVAT	Сума ПДВ, у форматі <Гривні, Копійки>
ISEXCISABLE	Ознака підакцизного товару, True/False

* обов'язкові параметри в рядку [OtherParametrs](#), якщо не вказані відповідні явні параметри. Опціональний об'єкт **Comment** використовується для додавання нефіскальної інформації:

Comment	Опціональний рядок нефіскальної інформації, коментар. Допускається використання одного коментарю на чек. За необхідності сформувати коментар з декількох рядків, для переходу на новий рядок використовується комбінація " <i>r</i> " або " <i>n</i> ".
----------------	---

NEW

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

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

Структура **JSONPayData** містить підсумки оплати по чеку, і має наступні поля:

Поле	Значення
SumCash	Сума оплати чеку готівкою
SumPayByCard	Сума оплати чеку банківською картою
SumPayByCredit	Сума оплати чеку в кредит
SumPayByCertificate	Сума оплати чеку сертифікатом
SumPayCheck	Сума чеку до оплати, з врахуванням всіх надбавок і знижок
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 чеку. Рекомендовано використовувати при встановленому параметрі відображення вікон інтерфейсу: <u>SetParameter("NOINTERFACEMODE", "False")</u>

Необов'язкові параметри інформації по транзакції терміналу при безготівкових розрахунках:

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 чеку
ParentRRN	PPH транзакції чека по якому здійснено повернення

Увага! Різниця між SumPayCheck та сумою по всіх видах оплати вважається значенням заокруглення. Тому розробник має самостійно реалізувати алгоритм заокруглення значення наданої готівки або суми оплати по чеку.

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер чеку
ReceiptLocalNum	String	Локальний номер чеку
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип документу (див. нижче)
XML	String	XML-структура фіскалізованого чеку

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

1<Клас документу><Тип документу><Додатковий тип документу>

Де

Число	Кількість знаків	Значення
<Клас документу>	1	0 - Чек 1 - Z-звіт 2 - X-звіт
<Тип документу>	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 документу>	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» 005 - Чек сторнування попереднього чека

Приклад:

NPg0KICAgICAgPFBBWVNZUz4NCiAgICAgICAgPFJPVYBSTld0VU09IjEiPg0KICAgICAgICAgIDxOQU1F
Pk1BU1RFUkNBukQ8L05BTUU+DQogICAgICAgICAgPEFDUVVJUkVOT7P80ji4PIgweDt6jwvQUNRVU1SR
U5NPg0KICAgICAgICAgIDxBQ1FVSVJFVFJBT1NJRD4wNdG4NjcxNzM2MjA8L0FDUVVJUkVUUKFOU01EPg
0KICAgICAgICAgIDxQT1NUUkFOU0RBVEU+MTkwODIwMjExMDQwNDA8L1BPu1RSQU5TREUFURT4NCiAgICA
gICAgICA8UE9TVFJBT1NOVU0+MjI0PC9QT1NUUkFOU05VTT4NCiAgICAgICAgICA8RVBAREVUQUlMUz5Y
WFhYWFhYWFhYWFg2ODgzPC9FUFpERVRBSUxTPg0KICAgICAgICAgIDxBVVRIQ0Q+NjgwMTRCPC9BVVRIQ
0Q+DQogICAgICAgICAgPFNVTT4zMtAuMzQ8L1NVTT4NCiAgICAgICAgPC9ST1c+DQogICAgICA8L1BBWV
NZUz4NCiAgICA8L1JPVz4NCiAgPC9DSEVDS1BBWT4NCiAgPENIRUNLVEFYpG0KICAgIDxST1cgUk9XT1V
NPSIxIj4NCiAgICAgIDxUWVBFPjA8L1RZUEU+DQogICAgICA8TkFNRT7PxMI8L05BTUU+DQogICAgICA8
TEVUVEVSPsA8L0xFVFRFUj4NCiAgICAgIDxQUkM+MjAuMDA8L1BSQz4NCiAgICAgIDxUVVJOT1ZFUj4zN
jYuMDA8L1RVUk5PVkVSPg0KICAgICAgPFNPVJDRVNVTT4zNjIuMzQ8L1NPVJDRVNVTT4NCiAgICAgID
xTVU0+NjAuMzk8L1NVTT4NCiAgICA8L1JPVz4NCiAgPC9DSEVDS1RBWD4NCiAgPENIRUNLQk9EWT4NCiA
gICA8Uk9XIFJPV05VTT0iMSI+DQogICAgICA8Q09ERT7A8PLo6vPr0u7i4PDzMTwvQ09ERT4NCiAgICAg
IDxCQVJDT0RFPjEyMzQ1Njc4OTwvQkFSQ09ERT4NCiAgICAgIDxOQU1FPs3g5+Lg0u7i4PDzMTwvTkFNR
T4NCiAgICAgIDxVTklUQ0Q+MjAwOTwvVU5JVENEPg0KICAgICAgPFVOSVROTT748jwvVU5JVE5NPg0KIC
AgICAgPEFNT1VOVD4xLjAwMDwvQU1PVU5UPg0KICAgICAgPFBSUNFPjE4My4wMDwvUFJJQ0U+DQogICA
gICA8TEVUVEVSUz7APC9MRVRURVJTPg0KICAgICAgPENPU1Q+MTgzLjAwPC9DT1NUPg0KICAgICAgPERJ
U0NPVU5UVFlQRT4wPC9ESVNDT1VOVFRZUEU+DQogICAgICA8RELtQ09VTlRTVU0+MS44MzwvRELtQ09VT
lRTVU0+DQogICAgPC9ST1c+DQogICAgPFJPVYBSTld0VU09IjEiPg0KICAgICAgPENPREU+wPDy6Orz69
Lu4uDw8zI8L0NPUEU+DQogICAgICA8QkFSQ09ERT45OTk5OTk5OTk5PC9CQVJDT0RFPg0KICAgICAgPE5
BTUU+zeDn4uDS7uLg8PMYPC9OQU1FPg0KICAgICAgPFVOSVRDRD4yMDA5PC9VTklUQ0Q+DQogICAgICA8
VU5JVE5NPvjyPC9VTklUTk0+DQogICAgICA8QU1PVU5UPjEuMDAwPC9BTU9VTlQ+DQogICAgICA8UFJJQ
0U+MTgzLjAwPC9QUklDRT4NCiAgICAgIDxMRVRURVJTPsA8L0xFVFRFUlM+DQogICAgICA8Q09TVD4xOD
MuMDA8L0NPu1Q+DQogICAgICA8RELtQ09VTlRUWVBFPjA8L0RJU0NPVU5UVFlQRT4NCiAgICAgIDxESVN
DT1VOVFNVT4xLjgzPC9ESVNDT1VOVFNVT4NCiAgICA8L1JPVz4NCiAgPC9DSEVDS0JPRFk+DQo8L0NI
RUNLPg=="} }

Ret.ReceiptFiscalNum = 474775

Ret.ReceiptLocalNum = 1162

Ret.Return = true

Ret.Type = 10000000

Ret.XML = " <?xml version="1.0" encoding="windows-1251" ?>

<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespacesSchemaLocation="check01.xsd">

<CHECKHEAD>

<DOCTYPE>0</DOCTYPE>

<DOCSUBTYPE>0</DOCSUBTYPE>

<UID>8C446439-6B01-4833-A326-AF8FA14CDCDF</UID>

<TIN>24234227</TIN>

<ORGNM>TOB "Тест-гідрометцентр"</ORGNM>

<POINTNM>Тест-гідромет</POINTNM>

<POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>

<ORDERDATE>06092021</ORDERDATE>

<ORDERTIME>180649</ORDERTIME>

<ORDERNUM>1189</ORDERNUM>

<CASHDESKNUM>3</CASHDESKNUM>

<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>

<CASHIER>Лабораторний П. П.</CASHIER>

<COMMENT>#####

Тестовий коментар

#####</COMMENT>

<VER>1</VER>

<ORDERTAXNUM>494296</ORDERTAXNUM>

<!--<CASHALOTCOMMENT>

<PAYMENTSTATUS>3</PAYMENTSTATUS>

</CASHALOTCOMMENT>-->

</CHECKHEAD>

<CHECKTOTAL>

```

<SUM>362.34</SUM>
<DISCOUNTSUM>3.66</DISCOUNTSUM>
</CHECKTOTAL>
<CHECKPAY>
  <ROW ROWNUM="1">
    <PAYFORMCD>0</PAYFORMCD>
    <PAYFORMNM>ГОТОВКА</PAYFORMNM>
    <SUM>52.00</SUM>
  </ROW>
  <ROW ROWNUM="2">
    <PAYFORMCD>1</PAYFORMCD>
    <PAYFORMNM>КАРТКА</PAYFORMNM>
    <SUM>310.34</SUM>
    <PAYSYS>
      <ROW ROWNUM="1">
        <NAME>MASTERCARD</NAME>
        <ACQUIRENM>Приват Банк</ACQUIRENM>
        <ACQUIRETRANSID>048867173620</ACQUIRETRANSID>
        <POSTRANSDATE>06092021180646</POSTRANSDATE>
        <POSTRANSNUM>224</POSTRANSNUM>
        <EPZDETAILS>XXXXXXXXXXXX6883</EPZDETAILS>
        <AUTHCD>68014B</AUTHCD>
        <SUM>310.34</SUM>
      </ROW>
    </PAYSYS>
  </ROW>
</CHECKPAY>
<CHECKTAX>
  <ROW ROWNUM="1">
    <TYPE>0</TYPE>
    <NAME>ПДВ</NAME>
    <LETTER>O</LETTER>
    <PRC>20.00</PRC>
    <TURNOVER>366.00</TURNOVER>
    <SOURCESUM>353.71</SOURCESUM>
    <SUM>58.95</SUM>
  </ROW>
  <ROW ROWNUM="2">
    <TYPE>1</TYPE>
    <NAME>Акциз</NAME>
    <LETTER>Д</LETTER>
    <PRC>5.00</PRC>
    <TURNOVER>183.00</TURNOVER>
    <SOURCESUM>181.17</SOURCESUM>
    <SUM>8.63</SUM>
  </ROW>
</CHECKTAX>
<CHECKBODY>
  <ROW ROWNUM="1">
    <CODE>АртикулТовару1</CODE>
    <BARCODE>123456789</BARCODE>
    <NAME>НазваТовару1</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>183.00</PRICE>
    <LETTERS>ОД</LETTERS>

```

```

<COST>183.00</COST>
<DISCOUNTTYPE>0</DISCOUNTTYPE>
<DISCOUNTSUM>1.83</DISCOUNTSUM>
<EXCISELABELS>
  <ROW ROWNUM="1">
    <EXCISELABEL>1111</EXCISELABEL>
  </ROW>
</EXCISELABELS>
</ROW>
<ROW ROWNUM="2">
  <CODE>АртикулТовару2</CODE>
  <BARCODE>999999999</BARCODE>
  <NAME>НазваТовару2</NAME>
  <UNITCD>2009</UNITCD>
  <UNITNM>шт</UNITNM>
  <AMOUNT>1.000</AMOUNT>
  <PRICE>183.00</PRICE>
  <LETTERS>O</LETTERS>
  <COST>183.00</COST>
  <DISCOUNTTYPE>0</DISCOUNTTYPE>
  <DISCOUNTSUM>1.83</DISCOUNTSUM>
</ROW>
</CHECKBODY>
</CHECK>" "

```

7. FiscalizeReturnCheck – Фіскалізація чека повернення товарів, послуг

Для створення та фіскалізації чеку повернення товарів та/або послуг необхідно сформувати JSON структуру JSONGoodsLstData, що міститиме дані номенклатури, що повертається, структуру JSONPayData з формами повернення грошей та номер чеку, за яким здійснюється повернення.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою *
strJSONData	String	JSON структура з описом проданих товарів або послуг
strJSONPayData	String	JSON структура з підсумками оплати по чеку
strFiscalNumberReturnCheck	String	Фіскальний номер чеку, за яким здійснюється повернення

Структури **JSONGoodsLstData** та **JSONPayData** визначені так само, як і в специфікації функції формування чеку продажу [FiscalizeCheck](#).

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер чеку
ReceiptLocalNum	String	Локальний номер чеку

1JERVJUQVhOVU0+DQogICAgPCEtLTxDQVNIQUxPVENPTU1FTlQ+DQogIDxQQVlNRU5UU1RBVFTVTPjE8L1
BBWU1FTlRTVEFUVVM+DQogIDxQQVJFTlRJRd4yNzZkZTFkYi00ZTAyLTRhYWQtOTQ3Ny1jODMwNzlkMGQ
1MmM8L1BBUkVOVElEPg0KPC9DQVNIQUxPVENPTU1FTlQ+LS0+DQogIDwvQ0hFQ0tIRUFEPg0KICA8Q0hF
Q0tUT1RBTd4NCiAgICA8U1VNPjM2Mi4zNDwvU1VNPg0KICAgIDxEsVNDT1VOVFNVTt4zLjY2PC9ESVNDT
1VOVFNVTt4NCiAgPC9DSEVDS1RPVEFMPg0KICA8Q0hFQ0tQQVtDQogICAgPFJFVjBSTldOVU09IjEiPg
0KICAgICAgPFBBWUZPUk1DRD4wPC9QQVlGTlJNQ0Q+DQogICAgICA8UEFZrK9STU5NPSP00rLCysA8L1B
BWUZPUk1OTT4NCiAgICAgIDxTVU0+MzYyLjM0PC9TVU0+DQogICAgPC9STlc+DQogIDwvQ0hFQ0tQQVtDQog
IDwvQ0hFQ0tQQVtDQogIDxSEVDS1RBWD4NCiAgICA8Uk9XIFJPV05VTt0iMSI+DQogICAgICA8VF1QRT4wPC9UWVBFPg0KI
CAGICAgPE5BTUU+z8TCPC9QU1FPg0KICAgICAgPExFVFRFUj7APC9MRVRURVI+DQogICAgICA8UFJDPj
IwLjAwPC9QUkM+DQogICAgICA8VfVStk9WRVI+MzY2LjAwPC9UVVJOTlZFUj4NCiAgICAgIDxTt1VSQ0V
TVU0+MzYyLjM0PC9Tt1VSQ0VTVU0+DQogICAgICA8U1VNPjYwLjM5PC9TVU0+DQogICAgPC9STlc+DQog
IDwvQ0hFQ0tQQVtDQogIDxSEVDS0JPRFk+DQogICAgPFJFVjBSTldOVU09IjEiPg0KICAgICAgPENPR
EU+wPDY6Orz69Lu4uDw8zE8L0NPREU+DQogICAgICA8QkFSQ09ERT4xMjM0NTY3ODk8L0JBuKNPREU+DQ
ogICAgICA8TkFNRT7N4Ofi4NLu4uDw8zE8L05BTUU+DQogICAgICA8VU5JVENEPjIwMDk8L1VOSVRDRD4
NCiAgICAgIDxVTklUTk0++PI8L1VOSVROTT4NCiAgICAgIDxBTU9VTlQ+MS4wMDA8L0FNT1VOVD4NCiAg
ICAgIDxQUk1DRt4xODMuMDA8L1BSSUNFPg0KICAgICAgPExFVFRFUlM+wDwvTEVUVEVSUz4NCiAgICAgI
DxDt1NUPjE4My4wMDwvQ09TVD4NCiAgICAgIDxEsVNDT1VOVFRZUEU+MDwvRELtQ09VTlRUWVBFPg0KIC
AgICAgPERJU0NPVU5UU1VNPjEuODM8L0RJu0NPVU5UU1VNPg0KICAgIDwvUk9XPg0KICAgIDxSt1cgUk9
XTlVNPSIyIj4NCiAgICAgIDxDt0RFPsDw8ujq8+vS7uLg8PMYPC9DT0RFPg0KICAgICAgPEJBuKNPREU+
OTk5OTk5OTk5OTwvQkFSQ09ERT4NCiAgICAgIDxOQU1FPs3g5+Lg0u7i4PDzMjwvTkFNRT4NCiAgICAgI
DxVTklUQ0Q+MjAwOTwvVU5JVENEPg0KICAgICAgPFVOSVROTT748jwvVU5JVE5NPg0KICAgICAgPEFNT1
VOVD4xLjAwMDwvQU1PVU5UPg0KICAgICAgPFBSUNFPjE4My4wMDwvUFJJQ0U+DQogICAgICA8TEVUVEV
SUz7APC9MRVRURVJTPg0KICAgICAgPENPU1Q+MTgzLjAwPC9DT1NUPg0KICAgICAgPERJU0NPVU5UVFlQ
RT4wPC9ESVNDT1VOVFRZUEU+DQogICAgICA8RELtQ09VTlRTVU0+MS44MzwvRELtQ09VTlRTVU0+DQogI
CAGPC9STlc+DQogIDwvQ0hFQ0tCT0RZPg0KPC9DSEVDSz4="}}}

Ret.ReceiptFiscalNum = 444799

Ret.ReceiptLocalNum = 860

Ret.Return = true

Ret.Type = 10000001

Ret.XML = "<?xml version='1.0' encoding='windows-1251'?"

<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="check01.xsd">

<CHECKHEAD>

<DOCTYPE>0</DOCTYPE>

<DOCSUBTYPE>1</DOCSUBTYPE>

<UID>985F658E-71F6-4D29-8551-023D9DEA2804</UID>

<TIN>24234227</TIN>

<ORGNM>TOB "Тест-гідрометцентр"</ORGNM>

<POINTNM>Тест-гідромет</POINTNM>

<POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>

<ORDERDATE>26072021</ORDERDATE>

<ORDERTIME>112136</ORDERTIME>

<ORDERNUM>860</ORDERNUM>

<CASHDESKNUM>3</CASHDESKNUM>

<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>

<ORDERRETNUM>444798</ORDERRETNUM>

<CASHIER>Лабораторний П. П.</CASHIER>

<COMMENT>#####

Тестовий коментар

#####</COMMENT>

<VER>1</VER>

<ORDERTAXNUM>444799</ORDERTAXNUM>

</CHECKHEAD>

<CHECKTOTAL>

<SUM>362.34</SUM>

<DISCOUNTSUM>3.66</DISCOUNTSUM>

```

</CHECKTOTAL>
<CHECKPAY>
  <ROW ROWNUM="1">
    <PAYFORMCD>0</PAYFORMCD>
    <PAYFORMNM>ГОТІВКА</PAYFORMNM>
    <SUM>362.34</SUM>
  </ROW>
</CHECKPAY>
<CHECKTAX>
  <ROW ROWNUM="1">
    <TYPE>0</TYPE>
    <NAME>ПДВ</NAME>
    <LETTER>A</LETTER>
    <PRC>20.00</PRC>
    <TURNOVER>366.00</TURNOVER>
    <SOURCESUM>362.34</SOURCESUM>
    <SUM>60.39</SUM>
  </ROW>
</CHECKTAX>
<CHECKBODY>
  <ROW ROWNUM="1">
    <CODE>АртикулТовару1</CODE>
    <BARCODE>123456789</BARCODE>
    <NAME>НазваТовару1</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>183.00</PRICE>
    <LETTERS>A</LETTERS>
    <COST>183.00</COST>
    <DISCOUNTTYPE>0</DISCOUNTTYPE>
    <DISCOUNTSUM>1.83</DISCOUNTSUM>
  </ROW>
  <ROW ROWNUM="2">
    <CODE>АртикулТовару2</CODE>
    <BARCODE>999999999</BARCODE>
    <NAME>НазваТовару2</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>183.00</PRICE>
    <LETTERS>A</LETTERS>
    <COST>183.00</COST>
    <DISCOUNTTYPE>0</DISCOUNTTYPE>
    <DISCOUNTSUM>1.83</DISCOUNTSUM>
  </ROW>
</CHECKBODY>
</CHECK>"

```

8. GetReceiptLocalNumberByFiscalNumber – Отримання локального номеру чеку за його фіскальним номером

Функція дозволяє визначити локальний номеру чеку за відомим фіскальним номером чеку. Пошук чеку відбувається в локальній базі чеків, а якщо його не знайдено, то виконується пошук чеку на сервері ФСКО.

Параметри

Ім'я	Тип	Опис
------	-----	------

FiscalNumberRRO1	String	Ідентифікатор пристрою
ReceiptFiscalNumber	Number	Фіскальний номер чеку

Відповідь

	Number	Локальний номер чеку
--	--------	----------------------

Якщо номер чеку не знайдено, то повертається пустий рядок.

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається і відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Приклад:

```
FiscalNumberRRO1 = "4000014020"
ReceiptFiscalNumber = "391930"
App = CreateObject("AddIn.CashaLotApi")
ret = ComObjApp.FiscalizeCheck(FiscalNumberRRO1 , JSONGoodsLstData,
JSONPayData );
ReceiptLocalNumber = ret.ReceiptLocalNum ;
```

Приклад відповіді:

Ret = 503

9. GetReceiptXML – Отримання XML структури фіскального чеку

Функція дозволяє отримати XML-структуру зареєстрованого чеку за його відомим фіскальним номером.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою *
ReceiptFiscalNumber	Number	Фіскальний номер чеку

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається і відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Відповідь

	String	XML-структура чеку
--	--------	--------------------

Приклад:

```
FiscalNumberRRO = "4000014020"
ReceiptFiscalNumber = "391930"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetReceiptXML(FiscalNumberRRO, ReceiptFiscalNumber)
```

Приклад відповіді:

```
Ret = <?xml version="1.0" encoding="windows-1251"?>
<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="check01.xsd">
  <CHECKHEAD>
    <DOCTYPE>0</DOCTYPE>
    <DOCSUBTYPE>1</DOCSUBTYPE>
    <UID>3E7D7D79-A0E4-4EB9-A8CE-1EC78D04B038</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Киев, Адреса</POINTADDR>
    <ORDERDATE>22072021</ORDERDATE>
```

```
<ORDERTIME>122452</ORDERTIME>
<ORDERNUM>810</ORDERNUM>
<CASHDESKNUM>3</CASHDESKNUM>
<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
<ORDERRETNUM>442921</ORDERRETNUM>
<CASHIER>Лабораторний П. П.</CASHIER>
<VER>1</VER>
<ORDERTAXNUM>442922</ORDERTAXNUM>
</CHECKHEAD>
<CHECKTOTAL>
  <SUM>362.34</SUM>
  <DISCOUNTSUM>3.66</DISCOUNTSUM>
</CHECKTOTAL>
<CHECKPAY>
  <ROW ROWNUM="1">
    <PAYFORMCD>0</PAYFORMCD>
    <PAYFORMNM>ГОТІВКА</PAYFORMNM>
    <SUM>362.34</SUM>
  </ROW>
</CHECKPAY>
<CHECKTAX>
  <ROW ROWNUM="1">
    <TYPE>0</TYPE>
    <NAME>ПДВ</NAME>
    <LETTER>А</LETTER>
    <PRC>20.00</PRC>
    <TURNOVER>366.00</TURNOVER>
    <SOURCESUM>362.34</SOURCESUM>
    <SUM>60.39</SUM>
  </ROW>
</CHECKTAX>
<CHECKBODY>
  <ROW ROWNUM="1">
    <CODE>АртикулТовару1</CODE>
    <BARCODE>123456789</BARCODE>
    <NAME>НазваТовару1</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>183.00</PRICE>
    <LETTERS>А</LETTERS>
    <COST>183.00</COST>
    <DISCOUNTTYPE>0</DISCOUNTTYPE>
    <DISCOUNTSUM>1.83</DISCOUNTSUM>
  </ROW>
  <ROW ROWNUM="2">
    <CODE>АртикулТовару2</CODE>
    <BARCODE>999999999</BARCODE>
    <NAME>НазваТовару2</NAME>
    <UNITCD>2009</UNITCD>
    <UNITNM>шт</UNITNM>
    <AMOUNT>1.000</AMOUNT>
    <PRICE>183.00</PRICE>
    <LETTERS>А</LETTERS>
    <COST>183.00</COST>
    <DISCOUNTTYPE>0</DISCOUNTTYPE>
    <DISCOUNTSUM>1.83</DISCOUNTSUM>
```



```

</ROW>
</CHECKBODY>
</CHECK>

```

10. GetFirstReceiptByPeriod – Отримати перший чек з періоду

Функцію **GetFirstReceiptByPeriod()** рекомендовано використовувати в ланцюжку запитів для отримання чеків за певний період часу. Дана функція викликається першою для підготовки списку чеків, і для отримання першого чеку з вказаного періоду часу.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою
dateBeg	String	Початок періоду в форматі ДДММГГГГ
dateEnd	String	Кінець періоду в форматі ДДММГГГГ

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення першого чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип першого чеку, помилки, локальний та фіскальний номери першого чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер першого чеку
ReceiptLocalNum	String	Локальний номер першого чеку
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип першого чеку (див. нижче)
XML	String	XML-структура першого чеку

Тип документу **"Type"** визначається числом, опис якого наведено в специфікації **FiscalizeCheck**.

Приклад:

```

FiscalNumberRRO = "4000007621"
StartDate = "19072021"
EndDate = "22072021"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetFirstReceiptByPeriod(FiscalNumberRRO, StartDate, EndDate)

```

Приклад відповіді:

(перший чек – відкриття зміни)

```

Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"1d936740-7255-444e-a6e0-f717c0c50fae","ReceiptFiscalNumber":"438767","ReceiptLocalNumber":"780","OfflineMode":"False","Type":"10100000","Base64Str1251ReceiptXML":"PD94bWwgdmVyc2lrbj0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQo8Q0hFQ0sgeGlsbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYS1pbmN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZW1hTG9jYXRpb249ImNoZWNRmDEueHNkIj4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVFlQRT4xMDA8L0RPQ1RZUEU+DQogICA8PFVJRDR4RDkzNjc0MC03MjU1LTQ0NEUtQTZFMCI1GnZlE3QzBDNTBGQUU8L1VJRDR4NCiAgICA8T1JERVJ

```

```
EQVRFPjE5MDcyMDIxPC9PUkRFUkRBVEU+DQogICAgPE9SREVSVElNRT4wOTU1MzE8L09SREVSVElNRT4NCiAgICA8T1JERVJOVU0+NzgwPC9PUkRFUk5VTT4NCiAgICA8Q0FTSERFU0tOVU0+MzwwQ0FTSERFU0tOVU0+DQogICAgPENBU0hSRUdJU1RFUk5VTT40MDAwMDA3NjIxPC9DQVNIUkVHSVNUURVJOVU0+DQogICAgPENBU0hJRVI+Y+Dh7vDg8u7w7eipIM/w6Orr4OQgz/Do6uvvg5O7i6Pc8L0NBU0hJRVI+DQogICAgPFZFUj4xPC9WRVI+DQogICAgPE9SREVSVEFYtLVNPjQzODc2NzwwT1JERVJUQVhOVU0+DQogIDwvQ0hFQ0tIRUFE
Pg0KPC9DSEVDSz4="}}
Ret.ReceiptFiscalNum = 438767
Ret.ReceiptLocalNum = 780
Ret.Return = true
Ret.Type = 10100000
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<CHECK xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='check01.xsd'>
  <CHECKHEAD>
    <DOCTYPE>100</DOCTYPE>
    <UID>1D936740-7255-444E-A6E0-F717C0C50FAE</UID>
    <ORDERDATE>19072021</ORDERDATE>
    <ORDERTIME>095531</ORDERTIME>
    <ORDERNUM>780</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний Приклад Прикладович</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>438767</ORDERTAXNUM>
  </CHECKHEAD>
</CHECK>
"
```

11. GetFirstReceiptByShift – Отримати перший чек з зміни

Функцію **GetFirstReceiptByShift()** рекомендовано використовувати в ланцюжку запитів для отримання чеків певної касової зміни. Дана функція викликається першою для підготовки списку чеків, і для отримання першого чеку з вказаної касової зміни.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою
dateBeg	String	Початок періоду в форматі ДДММГГГГ
dateEnd	String	Кінець періоду в форматі ДДММГГГГ

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення першого чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип першого чеку, помилки, локальний та фіскальний номери першого чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер першого чеку
ReceiptLocalNum	String	Локальний номер першого чеку
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни

Type	Integer	Тип першого чеку (див. нижче)
XML	String	XML-структура першого чеку

Тип документу *"Type"* визначається числом, опис якого наведено в специфікації [FiscalizeCheck\(\)](#).

Приклад:

```
FiscalNumberRRO = "4000007621"
ShiftID = "ef956a75-3760-4880-a5fe-07c126f81930"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetFirstReceiptByShift(deviceID, ShiftID)
```

Приклад відповіді:

(перший чек – відкриття зміни)

```
Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"ef956a75-3760-4880-a5fe-07c126f81930","ReceiptFiscalNumber":"435498","ReceiptLocalNumber":"678","OfflineMode":"False","Type":"10100000","Base64Str1251ReceiptXML":"PD94bWwgdmVyc2lvdj0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQo8Q0hFQ0sgeG1sbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYS1pbnN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZWlhTG9jYXRpb249ImNoZWNRmDEueHNkIj4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVFlQRT4xMDA8L0RPQ1RZUEU+DQogICAgPFVJRDR5FRjk1NkE3NS0zNzYwLTQ4ODAtQTVGRS0wN0MxMjZGODE5MzA8L1VJRDR4NCiAgICA8T1JERVJEQVRFPjE2MDcyMDIxPC9PUkRFUkRBVEU+DQogICAgPE9SREVSVE1NRT4xNTE3MzQ8L09SREVSVE1NRT4NCiAgICA8T1JERVJOVU0+Njc4PC9PUkRFUk5VTT4NCiAgICA8Q0FTSERFU0tOVU0+MzwwQ0FTSERFU0tOVU0+DQogICAgPENBU0hSRUdJU1RFUk5VTT40MDAwMDA3NjIxPC9DQVNIUkVHSVNUURVJOVU0+DQogICAgPENBU0hJRVI+yDh7vDg8u7w7eipIM/w6Orr4OQgz/Do6uvvg5O7i6Pc8L0NBU0hJRVI+DQogICAgPFZFUj4xPC9WRVI+DQogICAgPE9SREVSVEFYt1VNPjQzNTQ5ODwvT1JERVJUQVhOVU0+DQogIDwvQ0hFQ0tIRUFEPg0KPC9DSEVDSz4="}}
Ret.ReceiptFiscalNum = 435498
Ret.ReceiptLocalNum = 678
Ret.Return = true
Ret.Type = 10100000
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<CHECK xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='check01.xsd'>
  <CHECKHEAD>
    <DOCTYPE>100</DOCTYPE>
    <UID>EF956A75-3760-4880-A5FE-07C126F81930</UID>
    <ORDERDATE>16072021</ORDERDATE>
    <ORDERTIME>151734</ORDERTIME>
    <ORDERNUM>678</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний Приклад Прикладович</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>435498</ORDERTAXNUM>
  </CHECKHEAD>
</CHECK>"
```

12. GetNextReceipt– Отримати наступний чек з періоду

Функція **GetNextReceipt()** використовується в ланцюжку запитів для отримання чеків за певний період часу. Дана функція викликається після виклику функції

GetFirstReceiptByPeriod() або **GetFirstReceiptByShift()**, які готують список чеків. Функцію можна викликати декілька разів, поки не буде отримано весь список чеків. Ознакою кінця списку вважається неуспішність чергового виклику функції.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumberRRO</u>	String	Ідентифікатор пристрою

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення наступного чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип першого чеку, помилки, локальний та фіскальний номери чеку, ID зміни та Base64 рядок XML структури чеку
ReceiptFiscalNum	String	Фіскальний номер чеку
ReceiptLocalNum	String	Локальний номер чеку
Return	Bool	Успішність операції. Коли в списку більше нема чеків повертається false .
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип поточного чеку (див. нижче)
XML	String	XML-структура поточного чеку

Тип документу **"Type"** визначається числом, опис якого наведено в специфікації [FiscalizeCheck\(\)](#).

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
FiscalNumberRRO = "4000007621"
StartDate = "19072021"
EndDate = "22072021"
Set Ret = App.GetFirstReceiptByPeriod(FiscalNumberRRO, StartDate,
EndDate)
Do while Ret.Return
'Обробити отримані чеки
Set Ret = App.GetNextReceipt(FiscalNumberRRO)
Loop
```

Приклад відповіді:

(успішна операція – чек повернення)

```
Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"9de7f9d4-046e-4f86-b686-6ad20306893a","ReceiptFiscalNumber":"442759","ReceiptLocalNumber":"808","OfflineMode":"False","Type":"10000001","Base64Str1251ReceiptXML":"PD94bWwgdmVyc2lrbj0iMS4wIiBlbmNvZGluz0id2luZG93cy0xMjUxIj8+DQo8Q0hFQ0sgeGlsbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYS1pbnN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZWlhTG9jYXRpb249ImNoZWNRmDEueHNkIj4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVF1QRT4wPC9ET0NUWVBFPg0KICAgIDxET0NTVUJUWVBFPjE8L0RPQ1NVQ1RZUEU+DQogICAgPFVJRj41REJGNjU5Ni1GRtU5LTQyRDUtQUM1RC00RDNDENTE0MjgzMTE8L1VJRj4NCiAgICA8T1JERVJEQVRFPjIyMDcyMDIxPC9PUkRFUkRBVEU+DQogICAgPE9SREVSVElnRT4xMDIwMzM8L09SREVSVElnRT4NCiAgICA8T1JERVJOUVU0+ODA4PC9PUkRFUk5VTT4NCiAgICA8Q0FTSERFU0tOVU0+MzwvQ0FTSERFU0tOVU0+DQogICAgPENBU0hSRUdJU1RFUk5VTT40MDAwMDA3NjIxPC9DQVNIUkVHSVNVURVJOUVU0+DQogICAgPENBU0hJRVI+y+Dh7vDg8u7w7eJpIM8uIM8uPC9DQVNIUWVSPg0KICAgIDxWRVI+MTwvVkVSPg0KICAgIDxPUkRFU1RBWE5VTT40NDI3NTk8L09SREVSVEFYTLVN
```

```

PiANCiAgPC9DSEVDS0hFQUQ+DQogIDxDSEVDS1RPVEFMPg0KICAgIDxTVU0+MzYyLjM0PC9TVU0+DQogIDwvQ0hFQ0tUT1RBTd4NCjwvQ0hFQ0s+"))}
Ret.ReceiptFiscalNum = 442759
Ret.ReceiptLocalNum = 808
Ret.Return = true
Ret.Type = 10100001
Ret.XML = " <?xml version="1.0" encoding="windows-1251"?>
<CHECK xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="check01.xsd">
  <CHECKHEAD>
    <DOCTYPE>0</DOCTYPE>
    <DOCSUBTYPE>1</DOCSUBTYPE>
    <UID>5DBF6596-FE59-42D5-AC5D-4D3D51428311</UID>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>102033</ORDERTIME>
    <ORDERNUM>808</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>442759</ORDERTAXNUM>
  </CHECKHEAD>
  <CHECKTOTAL>
    <SUM>362.34</SUM>
  </CHECKTOTAL>
</CHECK>"

```

(неуспішна операція – кінець ланцюжку отриманих чеків)

```

Ret.JsonVal = {"Ret":false,"ErrorString":"","Values":{}}
Ret.Return = false

```

13. GetCurrentStatus – Отримати інформацію по поточному стану каси

Функція **GetCurrentStatus()** використовується для отримання інформації по поточному стану каси. Перелік параметрів, доступних для перевірки, наведено в таблиці.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою

Відповідь

В якості відповіді повертається структура **CashlotApiRetVal**, що містить значення наступного чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення "Ret" – успішність операції, "ErrorString" – рядок з описом помилки (в разі невдачі), "Values" – масив параметрів каси (див. таблицю нижче).
Return	Bool	Успішність операції.
ShiftID	String	Ідентифікаційний номер зміни. Якщо зміна відкрита, то відповідає ідентифікатору поточної зміни, якщо зміна закрита, то відповідає ідентифікатору останньої зміни.

Параметри та значення, які передаються в масиві *Values* структури *JsonVal*

ShiftState	Стан зміни: 1 – закрита, 2 – відкрита
ShiftStateStr	Стан зміни: "Closed" – закрита, "Opened" – відкрита.
ShiftLocalNumber	Локальний номер зміни
ShiftFiscalNumber	Фіскальний номер зміни
ShiftID	Ідентифікатор зміни
LastCheckLocalNumber	Локальний номер останнього чеку
LastCheckFiscalNumber	Фіскальний номер останнього чеку
LastCheckDateTime	Мітка часу останнього чеку
IsOfflineMode	Стан офлайн-режиму каси: 0 – каса в онлайн-режимі; 1 – каса в офлайн-режимі
NextLocalNumber	Наступний локальний номер чеку
CashBalance	Значення залишків готівки в касі
Base64Str1251CurrentStatusXML	XML-рядок вказаних вище параметрів в форматі Base64.

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
FiscalNumberRRO = "4000007621"
Set Ret = App.GetCurrentStatus(FiscalNumberRRO)
```

Приклад відповіді:

```
Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ShiftState":"2","ShiftStateStr":"Opened",
"ShiftLocalNumber":"1306","ShiftFiscalNumber":"509939","ShiftID":"a6d67115-5a3e-
4e70-9cbc-de663f8cb256","LastCheckLocalNumber":"1307","LastCheckFiscalNumber":"509940","Las
tCheckDateTime":"2021-09-
23T13:02:32","IsOfflineMode":"0","NextLocalNumber":"1308","CashBalance":"52","Bas
e64Str1251CurrentStatusXML":"PD94bWwgdMvYc2lvdW11PSAiVVRGLTgi
ID8+PCBEb2N1bWVudE91dHB1dFBhcmFtZXRLcnMgPiA8IFBhcmFtZXRLcnMgQ2hlY2tOdWliZXIgaPSAiM
TMwNyIgQ2FzaEJhbGFuY2UgPSAiNTIiIElzM2ZmbGluZU1vZGUgPSAiMCIGTmV4dExvY2FsTnVtYmVpID
0gIjEzMDgiIExhc3RDaGVja0Zpc2NhbmE5bWJlciA9ICl1MDk5NDAlIFNoaWZ0TnVtYmVpPSAiMTMwNiI
gU2hpZnRJRCA9ICJhNmQ2NzExNS01YTl1LTRlNzAtOWNiYy1kZTY2M2Y4Y2IyNTYiIFNoaWZ0RmlzY2Fs
TnVtYmVpID0gIjUwOTkzOSIgU2hpZnRtdGF0ZT0gIjIiIERhdGVUaW11PSAiMjAyMS0wOS0yM1QxMzowM
jozMlIgIz4gPC8gRG9jdW11bnRpdXRwdXRQYXJhbWV0ZXJzID4="}}
Ret.Return = true
Ret.ShiftID = "a6d67115-5a3e-4e70-9cbc-de663f8cb256"
```

14. GetFirstZReportByPeriod – Отримати перший Z-звіт за період

Функцію **GetFirstZReportByPeriod()** рекомендовано використовувати в ланцюжку запитів для отримання Z-звітів за певний період часу. Дана функція викликається першою для підготовки списку Z-звітів, і для отримання першого Z-звіту з вказаного періоду часу.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою

dateBeg	String	Початок періоду в форматі ДДММГГГГ
dateEnd	String	Кінець періоду в форматі ДДММГГГГ

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення першого Z-звіту з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, помилки, тип документу Z-звіту (11000000), локальний та фіскальний номери Z-звіту, ID зміни та Base64 рядок XML структури Z-звіту
ReceiptFiscalNum	String	Фіскальний номер першого Z-звіту
ReceiptLocalNum	String	Локальний номер першого Z-звіту
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип документу Z-звіту - 11000000
XML	String	XML-структура першого Z-звіту

Тип документу Z-звіту **"Type"** визначається числом, опис якого наведено в специфікації [FiscalizeCheck](#).

Приклад:

```
FiscalNumberRRO = "4000007621"
StartDate = "19072021"
EndDate = "22072021"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetFirstZReportByPeriod(FiscalNumberRRO, StartDate, EndDate)
```

Приклад відповіді:

(перший чек – відкриття зміни)

```
Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ShiftID":"1d936740-7255-444e-a6e0-f717c0c50fae","ReceiptFiscalNumber":"438772","ReceiptLocalNumber":"781","OfflineMode":"False","Type":"11000000","Base64Str1251ReportXML":"PD94bWwgdmVyc2l1bWVjOj0iMS4wIiBlbmNvZGl1Zz0id2luZG93cy0xMjUxIj8+DQo8WlJFUCB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiB4c2k6bm90YW1lc3BhY2VtY2h1bWFMb2NhdGl1bWVjOienJlcDAXLnzhZCI+DQogIDxaUkVQSEVB4RD4NCiAgICA8VU1EPjUyRThDQ0MyLTRBQUQtNEVEMy04RkYxLTU1NjRBD0U4MTRFQzZwvVU1EPg0KICAgIDxPUkRFUkRBVEU+MTkwNzIwMjE8L09SREVSREFURT4NCiAgICA8T1JERVJUSU1FPjA5NTgyNjwvT1JERVJUSU1FPg0KICAgIDxPUkRFUk5VTT43ODE8L09SREVST1VNPg0KICAgIDxDQVNIREVTS05VTT4zPC9DQVNIREVTS05VTT4NCiAgICA8Q0FTSFJFR01TVEVST1VNPjQwMDAwMDc2MjE8L0NB0hSRUdJU1RFUk5VTT4NCiAgICA8Q0FTSElFUj7L4OHu8ODy7vDt6Okgz/Do6uv5CDP8Ojq6+Dk7uLo9zwvQ0FTSElFUj4NCiAgICA8VkvSPjE8L1ZFUj4NCiAgICA8T1JERVJUQVhOVU0+NDM4NzcyPC9PUkRFU1RBWE5VTT4NCiAgPC9aUkVQSEVB4RD4NCiAgPFpSRVBCT0RZPg0KICAgIDxTRVJWSUNFSU5QVVQ+MC4wMDwvU0VSVklDRU1OUFVUPg0KICAgIDxTRVJWSUNFT1VUUFVUPjAuMDA8L1NFU1ZJQ0VPVVRQVWQ+DQogIDwvWlJFUEJPRFk+DQo8L1pSRVA+", "ServiceInput":null, "ServiceOutput":null, "OrderCount":"0", "OrderExpCount":"0", "TotalSUM":null, "TotalSUMCash":null, "TotalSUMCard":null, "TotalTAXSUM":null, "ReturnSUM":null, "ReturnSUMCash":null, "ReturnSUMCard":null, "ReturnTAXSUM":null}}
```

```
Ret.ReceiptFiscalNum =
Ret.ReceiptLocalNum =
Ret.Return = true
Ret.Type = 11000000
```



```
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<ZREP xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='zrep01.xsd'>
  <ZREPHEAD>
    <UID>52E8CCC2-4AAD-4ED3-8FF1-6564A85814EC</UID>
    <ORDERDATE>19072021</ORDERDATE>
    <ORDERTIME>095826</ORDERTIME>
    <ORDERNUM>781</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний Приклад Прикладович</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>438772</ORDERTAXNUM>
  </ZREPHEAD>
  <ZREPBODY>
    <SERVICEINPUT>0.00</SERVICEINPUT>
    <SERVICEOUTPUT>0.00</SERVICEOUTPUT>
  </ZREPBODY>
</ZREP>"
```

15. GetNextZReport – Отримати наступний Z-звіт з періоду

Функція **GetNextZReport()** використовується в ланцюжку запитів для отримання Z-звітів за певний період часу. Дана функція викликається після виклику функції

GetFirstZReportByPeriod(), яка готує список Z-звітів. Функцію можна викликати декілька разів, поки не буде отримано весь список Z-звітів. Ознакою кінця списку вважається неуспішність чергового виклику функції.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumberRRO</u>	String	Ідентифікатор пристрою

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення наступного чеку з заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери Z-звіту, ID зміни та Base64 рядок XML структури Z-звіту
ReceiptFiscalNum	String	Фіскальний номер Z-звіту
ReceiptLocalNum	String	Локальний номер Z-звіту
Return	Bool	Успішність операції. Коли в списку більше нема Z-звіту повертається false .
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип Z-звіту - - 11000000

XML	String	XML-структура Z-звіту
-----	--------	-----------------------

Тип Z-звіту *"Type"* визначається числом, опис якого наведено в специфікації [FiscalizeCheck\(\)](#).

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
FiscalNumberRRO = "4000007621"
StartDate = "19072021"
EndDate = "22072021"
Set Ret = App.GetFirstZReportByPeriod(FiscalNumberRRO, StartDate,
EndDate)
Do while Ret.Return
'Обробити отримані звіти
Set Ret = App.GetNextZReport(FiscalNumberRRO)
Loop
```

Приклад відповіді:

```
Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"2c93c78e-f8db-4672-8748-0f21e97545da","ReceiptFiscalNumber":"442727","ReceiptLocalNumber":"804","OfflineMode":"False","Type":"11000000","Base64Str1251ReportXML":"PD94bWwgdmVyc2lvcj0iMS4wIiBlbmNvZGluzZ0id2luZG93cy0xMjUxIj8+DQo8WlJFUCB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiB4c2k6bm90YW1lc3BhY2VTY2h1bWFMb2NhdGlvbj0ienJlcDAxLnhzZCI+DQogIDxaUkVQSEVBRD4NCiAgICA8VU1EPjYyRjQwQUYyLUJDREMtNDIzNC05NjQzLUU4MzY1QUYxOTNGRTwvVU1EPg0KICAgIDxPUkRFUkRBVEU+MjIwNzIwMjE8L09SREVSREFUR01TVEVSTlVNPjQwMDAwMDc2MjE8L0NBUE0hSRUdJU1RFUk5VTT4zPC9DQVNIREVTS05VTT4zPC9DQVNIREVTS05VTT4NCiAgICA8Q0FTSFJFR01TVEVSTlVNPjQwMDAwMDc2MjE8L0NBUE0hSRUdJU1RFUk5VTT4NCiAgICA8Q0FTSElFUj7L40Hu8ODy7vDt60kgzy4gzy48L0NBUE0hJRVI+DQogICAgPFZFUj4xPC9WRVI+DQogICAgPE9SREVSVEFYTVVNPjQ0Mjcyc2VzVU1JERVJUQVhOVU0+DQogIDwvWlJFUEhFUUQ+DQogIDxaUkVQSEVBRD4NCiAgICA8U0VSUk1DRU1OUFVUPjAuMDA8L1NFU1ZJQ0VJTlBVVD4NCiAgICA8U0VSUk1DRU1OUFVUPjAuMDA8L1pSRVBCT0RZPg0KPC9aUkVQPg==","ServiceInput":null,"ServiceOutput":null,"OrderCount":"0","OrderExpCount":"0","TotalSUM":null,"TotalSUMCash":null,"TotalSUMCard":null,"TotalTAXSUM":null,"ReturnSUM":null,"ReturnSUMCash":null,"ReturnSUMCard":null,"ReturnTAXSUM":null}}
Ret.ReceiptFiscalNum = 442727
Ret.ReceiptLocalNum = 804
Ret.Return = true
Ret.Type = 11000000
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<ZREP xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='zrep01.xsd'>
  <ZREPHEAD>
    <UID>62F40AF2-BCDC-4234-9643-E8365AF193FE</UID>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>093405</ORDERTIME>
    <ORDERNUM>804</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>442727</ORDERTAXNUM>
  </ZREPHEAD>
  <ZREPBODY>
    <SERVICEINPUT>0.00</SERVICEINPUT>
    <SERVICEOUTPUT>0.00</SERVICEOUTPUT>
```

</ZREPBODY>
</ZREP>"

16. GetZReportByShift– Отримати Z-звіт касової зміни

Функція **GetZReportByShift()** призначена для отримання Z-звіту вказаної касової зміни.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumberRRO</u>	String	Ідентифікатор пристрою
ShiftID	String	Ідентифікатор касової зміни

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення Z-звіту вказаної касової зміни:

JsonVal	String	JSON-структура, що містить значення успішності операції, помилки, тип документу Z-звіту (11000000) , локальний та фіскальний номери Z-звіту, ID зміни та Base64 рядок XML структури Z-звіту
ReceiptFiscalNum	String	Фіскальний номер чеку Z-звіту
ReceiptLocalNum	String	Локальний номер чеку Z-звіту
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер поточної зміни
Type	Integer	Тип документу Z-звіту - 11000000
XML	String	XML-структура Z-звіту

Тип документу Z-звіту "**Type**" визначається числом, опис якого наведено в специфікації [FiscalizeCheck\(\)](#).

Приклад:

```
FiscalNumberRRO = "4000007621"  
ShiftID = "2c93c78e-f8db-4672-8748-0f21e97545da"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.GetZReportByShift(FiscalNumberRRO, ShiftID)
```

Приклад відповіді:

```
Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"2c93c78e-f8db-4672-8748-0f21e97545da","ReceiptFiscalNumber":"442727","ReceiptLocalNumber":"804","OfflineMode":"False","Type":"11000000","Base64Str1251ReportXML":"PD94bWwgdmVyc2lvdj0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQo8WlJFUCB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZWlhLWluc3RhbmNlIiB4c2k6bm90YWw1lc3BhY2VtY2hlcWFMb2NhdGlvbj0ienJlcDAXLnzhZCI+DQogIDxUkVQSEVBRD4NCiAgICA8VU1EPjYyRjQwQUYyLUJDREMTNDIzNC05NjQzLUU4MzY1QUYxOTNGRTwvVU1EPg0KICAgIDxUSU4+MjQyMzQyMjc8L1RJTj4NCiAgICA8T1JHTk0+0s7CICLS5fHyLeOz5PDu7OXy9uXt8vAiPC9PUkd0TT4NCiAgICA8UE9JTlROT7S5fHyLeOz5PDu7OXyPC9QT0lOV E5NPg0KICAgIDxQT0lOVEFERFI+08rQwK/NwCwg7C4gyuj14iwgwOTw5fHgPC9QT0lOVEFERFI+DQogICAgPE9SREVSREFURT4yMjA3MjAyMTwvT1JERVJJEQVRFPg0KICAgIDxPUkRFU1RJTUU+MDkzNDAlPC9PUkRFU1RJTUU+DQogICAgPE9SREVSTlVNPjgWNDwvT1JERVJJOVU0+DQogICAgPENBU0hERVNLTlVNPjM8L0NB U0hERVNLTlVNPg0KICAgIDxQVNIUkVHSVNUVJVOVU0+NDAwMDAwNzYyMTwvQ0FTSFJFR01TVEVSTlVNP g0KICAgIDxQVNISUVSPsvg4e7w4PLu8O3o6SDPLiDPLjwvQ0FTSElFUj4NCiAgICA8VkvSPjE8L1ZFUj 4NCiAgICA8T1JERVJUQVh0VU0+NDQyNzI3PC9PUkRFU1RBWE5VTT4NCiAgPC9aUkVQSEVBRD4NCiAgPFp SRVBSRUFSMSVo+DQogICAgPFNVTT4zNjIuMzQ8L1NVTT4NCiAgICA8T1JERVJTTQ05UPjE8L09SREVSU0NO VD4NCiAgICA8UEFZRk9STVM+DQogICAgICA8Uk9XIFJPV05VTT0iMSI+DQogICAgICA8IDxQQVlGTlJNQ
```

```

0Q+MDwvUEFZRk9STUNEPg0KICAgICAgICA8UEFZRk9STU5NPSP00rLCysA8L1BBWUZPUk10TT4NCiAgIC
AgICAgPFNVTT4yMTIuMzQ8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgICAgPFJPVYBST1dOVU09IjIiPg0
KICAgICAgICA8UEFZRk9STUNEPjI8L1BBWUZPUk1DRD4NCiAgICAgICAgPFBBWUZPUk10TT7RxdDSyNSy
ysDSPC9QQVlGT1JNTk0+DQogICAgICAgIDxTVU0+NTAuMDA8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgI
CAgPFJPVYBST1dOVU09IjMiPg0KICAgICAgICA8UEFZRk9STUNEPjM8L1BBWUZPUk1DRD4NCiAgICAgIC
AgPFBBWUZPUk10TT7K0MXEYNI8L1BBWUZPUk10TT4NCiAgICAgICAgPFNVTT4xMDAuMDA8L1NVTT4NCiA
gICAgIDwvUk9XPg0KICAgIDwvUEFZRk9STVM+DQogICAgPFRBWEVTPg0KICAgICAgPFJPVYBST1dOVU09
IjEiPg0KICAgICAgICA8VF1QRT4wPC9UWVBFPG0KICAgICAgICA8TkFNRT7PxiMI8L05BTUU+DQogICAgI
CAgIDxMRVRURVI+wDwvTEVUVFVSPg0KICAgICAgICA8UFJDPjIwLjAwPC9QUkM+DQogICAgICAgIDxUVV
JOT1ZFUj4zNjYuMDA8L1RVUk5PVkVSPg0KICAgICAgICA8U09VUkNFU1VNPjM2Mi4zNDwvU09VUkNFU1V
NPg0KICAgICAgICA8U1VNPjYwLjM5PC9TVU0+DQogICAgICA8L1JPVz4NCiAgICA8L1RBWEVTPg0KICA8
L1pSRVBBSRUFSVo+DQogIDxaUkVQk9EWT4NCiAgICA8U0VSVklDRU1OUFVUPjAuMDA8L1NFU1ZJQ0VJT
lBVVD4NCiAgICA8U0VSVklDRU1VVFVVD4wLjAwPC9TRVJSUNFT1VUUFVUPg0KICA8L1pSRVBCT0RZPg
0KPC9aUkVQPg==", "ServiceInput": "0", "ServiceOutput": "0", "OrderCount": "1", "OrderExp
Count": "0", "TotalSUM": "362,34", "TotalSUMCash": "212,34", "TotalSUMCard": null, "Total
TAXSUM": "60,39", "ReturnSUM": null, "ReturnSUMCash": null, "ReturnSUMCard": null, "Retur
nTAXSUM": null, "TotalTAX_A_20%_SUM": "60,39", "TotalTAX_A_20%_TURNOVER": "362,34", "To
talTAX_A_20%_DISCOUNTSUM": "3,66", "TotalTAX_A_20%_REMOVEDTAXES": null}}
Ret.ReceiptFiscalNum = 442727
Ret.ReceiptLocalNum = 804
Ret.Return = true
Ret.Type = 11000000
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<ZREP xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='zrep01.xsd'>
  <ZREPHEAD>
    <UID>62F40AF2-BCDC-4234-9643-E8365AF193FE</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>093405</ORDERTIME>
    <ORDERNUM>804</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>442727</ORDERTAXNUM>
  </ZREPHEAD>
  <ZREPREALIZ>
    <SUM>362.34</SUM>
    <ORDERSCNT>1</ORDERSCNT>
    <PAYFORMS>
      <ROW ROWNUM="1">
        <PAYFORMCD>0</PAYFORMCD>
        <PAYFORMNM>ГОТІВКА</PAYFORMNM>
        <SUM>212.34</SUM>
      </ROW>
      <ROW ROWNUM="2">
        <PAYFORMCD>2</PAYFORMCD>
        <PAYFORMNM>СЕРТИФІКАТ</PAYFORMNM>
        <SUM>50.00</SUM>
      </ROW>
      <ROW ROWNUM="3">
        <PAYFORMCD>3</PAYFORMCD>
        <PAYFORMNM>КРЕДИТ</PAYFORMNM>

```

```

        <SUM>100.00</SUM>
    </ROW>
</PAYFORMS>
<TAXES>
    <ROW ROWNUM="1">
        <TYPE>0</TYPE>
        <NAME>пдв</NAME>
        <LETTER>А</LETTER>
        <PRC>20.00</PRC>
        <TURNOVER>366.00</TURNOVER>
        <SOURCESUM>362.34</SOURCESUM>
        <SUM>60.39</SUM>
    </ROW>
</TAXES>
</ZPREALIZ>
<ZREPBODY>
    <SERVICEINPUT>0.00</SERVICEINPUT>
    <SERVICEOUTPUT>0.00</SERVICEOUTPUT>
</ZREPBODY>
</ZREP>"

```

17. ShowReceipt – Вивести візуальне відображення чеку на екран

Функція використовується для виклику вікна з візуальним відображенням зареєстрованого чеку. Вікно з відображенням чеку також дозволяє використати додаткові функції друку або відправки на електронну пошту чеку, кнопки яких присутні у вікні чеку.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumberRRO</u>	String	Ідентифікатор пристрою
ReceiptFiscalNumber	Number	Фіскальний номер чеку

Відповідь

	Bool	Успішність операції
--	------	---------------------

Приклад:

```

FiscalNumberRRO = "4000007621"
ReceiptFiscalNumber = "391930"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.ShowReceipt (FiscalNumberRRO, ReceiptFiscalNumber)

```

Приклад відповіді:

```
Ret = True
```

ТОВ "Тест-Діагностцентр"
Тест-Діагностцентр
УКРАЇНА, м. Київ, Адреса
ID 24234227
Фін ПРРО 4000007621
КАСІР Лаборагорний П. П.
Чек №391929

НазваТовару1	1 X 183.00	183.00 A
Знижка		-1.83
НазваТовару2	1 X 183.00	183.00 A
Знижка		-1.83

Сума, грн 362.34
ПДВ А+20% 60.39

ГОТІВКА, грн 366.00
Решта, грн 3.66
391929 6/1/2021 3:37:18 PM



ФІСКАЛЬНИЙ ЧЕК ОНЛАЙН Cashalot

Друк
Надіслати на Email
Готово

18. PrintPReportDate - Надрукувати періодичний звіт по датам

Функція надрукувати періодичний звіт по датам використовується для можливості друку періодичного звіту використовуючи проміжки часу, що фіксується значеннями (ДатаПочаткуПеріоду-у форматі `"dd.MM.yyyy"` та ДатаЗакінченняПеріоду- у форматі `"dd.MM.yyyy"`).

(Усі параметри рядкові, для сумісності):

//Надрукувати періодичний звіт по датам(ФіскальнийНомерРРО, датаПочаткуПеріоду-
"dd.MM.yyyy", датаЗакінченняПеріоду-"dd.MM.yyyy", ОзнакаСкороченогоЗвіту-"TRUE/FALSE")
BOOL PrintPReportDate(BSTR FiscalNumberRRO, BSTR dtBeg, BSTR dtEnd, BSTR isShort)

Приклад:

```
FiscalNumberRRO = "4000007621"
PrintPReportDate = "391930"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.ShowReceipt (FiscalNumberRRO, ReceiptFiscalNumber)
```

Приклад відповіді:

Період це проміжок часу , що встановлюється між DateBegin та DateEnd.

Параметри

Ім'я	Тип	Опис
<u>ИДУстройства</u>	String	Ідентифікатор пристрою
<u>ВходныеПараметры</u>	DATE	Початок періоду в форматі ДДММГГГГ
<u>ВыходныеПараметры</u>	DATE	Кінець періоду в форматі ДДММГГГГ
<u>Сокращенный</u>	BOOL	ОзнакаСкороченогоЗвіту

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення початку періоду та дату закінчення заданого періоду:

JsonVal	String	JSON-структура, що містить значення успішності операції, ОзнакаСкороченогоЗвіту, Base64 рядок XML структури чеку	
DeviceID	String	Идентификатор устройства	ФіскальнийНомерРРО
DateBegin	DATE	Дата начала периода	dd.MM.yyyy
DateEnd	DATE	Дата конца периода	dd.MM.yyyy
IsShort	BOOL	Признак сокращенного отчета	TRUE/FALSE

<u>FiscalNumberRRO</u>	String	Ідентифікатор пристрою
DocumentPackage	String	Текст для друку

Відповідь

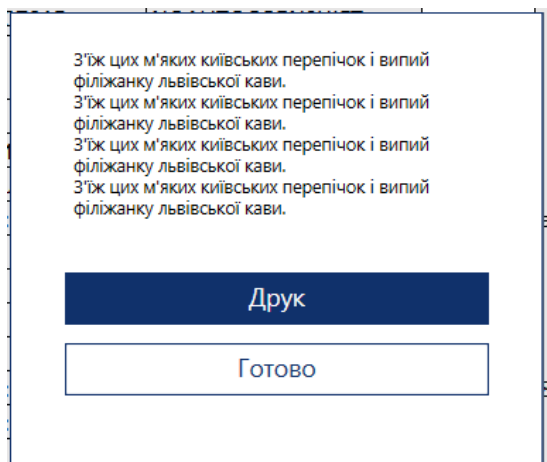
	Bool	Успішність операції
--	------	---------------------

Приклад:

```
FiscalNumberRRO = "4000007621"
DocumentPackage = "З'їж цих м'яких київських перепічок і випий філіжанку львівської кави.
З'їж цих м'яких київських перепічок і випий філіжанку львівської кави.
З'їж цих м'яких київських перепічок і випий філіжанку львівської кави.
З'їж цих м'яких київських перепічок і випий філіжанку львівської кави."
App = CreateObject("AddIn.CashaLotApi")
Ret = App.PrintTextDocument(FiscalNumberRRO, DocumentPackage)
```

Приклад відповіді:

Ret = True



20. ServiceInput – Службове внесення грошей в касу

Для виконання службового внесення в касу використовують функцію **ServiceInput**. Якщо задано параметр NOINTERFACEMODE = "False" (за допомогою функції [SetParameter](#)), то при цьому також виводиться службове вікно для підтвердження операції.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumRRO</u>	String	Ідентифікатор пристрою
Amount	Number	Додатне значення суми внесення у форматі <Грн, Коп>

Відповідь

	Bool	Успішність операції
--	------	---------------------

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритті зміні переривається, відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

і повертається **false**.

Приклад:

```
FiscalNumRRO = "4000007621"
Amount = "100"
```



```
App = CreateObject("AddIn.CashaLotApi")
Ret = App.ServiceInput (FiscalNumRRO, Amount)
```

NEW

21. ServiceInputEx – Службове внесення грошей в касу

Для виконання програмного службового внесення в касу використовують функцію **ServiceInputEx**.

Якщо задано параметр NOINTERFACEMODE = "False" (за допомогою функції [SetParameter](#)), то при цьому також виводиться службове вікно для підтвердження операції.

Якщо задано параметр NOINTERFACEMODE = "True", то на екран виводиться вікно з результатом проведеної операції.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою
Amount	Number	Додатне значення суми внесення у форматі <Грн, Коп>

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, помилки, локальний та фіскальний номери документу службового внесення
Return	Bool	Успішність операції
ShiftID	String	Ідентифікатор зміни

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається, відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Приклад:

```
FiscalNumRRO = "4000007621"
Amount = "100"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.ServiceInputEx(FiscalNumRRO, Amount)
```

Приклад відповіді:

```
Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"09992bfb-87cd-4f50-b685-1223a686091b"}}
Ret.ShiftID = "09992bfb-87cd-4f50-b685-1223a686091b"
Ret.Return = true
```

22. ServiceOutput – Службова видача грошей з каси

Для виконання службової видачі з каси використовують функцію **ServiceOutput**. Якщо задано параметр NOINTERFACEMODE = "False" (за допомогою функції [SetParameter](#)), то при цьому також виводиться службове вікно для підтвердження операції.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою
Amount	Number	Додатне значення суми видачі у форматі <Грн, Коп>

Відповідь

	Bool	Успішність операції
--	-------------	----------------------------

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається, відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Приклад:

```
FiscalNumRRO = "4000007621"
Amount = "100"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.ServiceOutput(FiscalNumRRO, Amount)
```

Приклад відповіді:

Ret = True

NEW 23. ServiceOutputEx – Службова видача грошей з каси

Для виконання програмного службового внесення в касу використовують функцію **ServiceOutputEx**.

Якщо задано параметр NOINTERFACEMODE = "False" (за допомогою функції [SetParameter\(\)](#)), то при цьому також виводиться службове вікно для підтвердження операції.

Якщо задано параметр NOINTERFACEMODE = "True", то на екран виводиться вікно з результатом проведеної операції.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою*
Amount	Number	Додатне значення суми видачі у форматі <Грн, Коп>

Відповідь

В якості відповіді повертається структура CashalotApiRetVal, що містить значення

JsonVal	String	JSON-структура, що містить значення успішності операції, помилки, локальний та фіскальний номери документу службової видачі
ReceiptFiscalNum	String	Фіскальний номер документу
ReceiptLocalNum	String	Локальний номер документу
Return	Bool	Успішність операції

Якщо задано параметр NOAUTOOPENSHIFT= True, виклик функції при закритій зміні переривається, відображається повідомлення

"Операція не можлива, необхідно відкрити зміну"

Приклад:

```
FiscalNumRRO = "4000007621"
Amount = "100"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.ServiceOutputEx(FiscalNumRRO, Amount)
```

Приклад відповіді:

```
Ret.JsonVal =
{"Ret":true,"ErrorString":"","Values":{"ReceiptFiscalNumber":"430098","ReceiptLocalNumber":"665"}}
Ret.ReceiptFiscalNum = 430098
Ret.ReceiptLocalNum = 665
Ret.Return = true
```

24. GetLastError – Отримання опису останньої помилки

У разі неуспішної операції виклик цієї функції дозволяє отримати опис помилки.

Параметри

Функція не має вхідних параметрів.

Відповідь

	String	Опис помилки
--	--------	--------------

Приклад:

```
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetLastError()
```

Приклади відповіді:

Ret = "Помилка закриття зміни"

Ret = "Помилка відправки чеку до ФСКО"

25. PrintXReport - Формування X-звіту

Функція дозволяє викликати на екран X-звіт для перегляду та відправки на друк.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumRRO</u>	String	Ідентифікатор пристрою

Відповідь

	Bool	Успішність операції
--	------	---------------------

Приклад:

```
FiscalNumRRO = "4000007621"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.PrintXReport(FiscalNumRRO)
```

Приклад відповіді:

Ret = True

26. GetXReport - Формування X-звіту

Виклик функції **GetXReport()** дозволяє сформувати і програмно отримати X-звіт у вигляді JSON-структури.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumberRRO</u>	String	Ідентифікатор пристрою *
ShowPrintReport	Boolean	Ознака необхідності вивести звіт на друк. "true" - на екран виводиться візуальне відображення X-звіту для перегляду та можливістю вручну роздрукувати звіт (аналогічно функції PrintXReport()).

Відповідь

В якості відповіді повертається структура CashalotApiRetVal, що містить значення

JsonVal	String	JSON-структура, яка містить інформацію X-звіту
Return	Bool	Успішність операції
ShiftId	String	Ідентифікатор зміни, в якій сформовано X-звіт
Type	Integer	Тип документу чеку - 12000000
XML	String	XML-структура X-звіту

JsonVal містить наступні об'єкти:

"ServiceInput"	Загальна сума службових внесень по зміні
"ServiceOutput"	Загальна сума службових видач 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 продажу "Готівка"
"ReturnSUMCard"	Загальна сума продажу "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
"Base64Str1251ReportXML"	Дані Z-звіту у вигляді XML-структури в кодуванні Windows1251 конвертованої в Base64-рядок

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	А, Б, В, ...	TotalTAX_A_20%_SUM ReturnTAX_Б_5%_TURNOVER

<2>	Відсоток податку	0, 5, 20, ...	TotalTAX_€_10%_REMOVEDTAXES
<3>	Сума	SUM	
	Обіг	TURNOVER	
	Знижка	DISCOUNTSUM	
	Виключені податки	REMOVEDTAXES	

Приклад:

```

FiscalNumRRO = "4000007621"
ShowPrintReport = false
App = CreateObject("AddIn.CashaLotApi")
Ret = App. GetXReport(FiscalNumRRO, ShowPrintReport)

```

Приклад відповіді:

```

Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"9de7f9d4-046e-4f86-b686-6ad20306893a","ReceiptFiscalNumber":null,"ReceiptLocalNumber":"0","OfflineMode":false,"Type":"12000000","Base64Str1251ReportXML":"PD94bWwgdmVyc2lvdj0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQo8WlJFUCB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZWlhLWluc3RhbmNlIiB4c2k6bm90YWllc3BhY2V2YTZhlbWFMb2NhdGlvbj0ienJlcDAXLnhzZCI+DQogIDxaUkVQSEVBRD4NCiAgICA8VU1EPjJDOEMzMEVELTcyMjEtNEY5QS1CRTU0LThCNTJGM0U4RTI3OTwvVU1EPg0KICAgIDxUSU4+MjQyMzQyMjc8L1RJTj4NCiAgICA8T1JHTk0+0s7CICLS5fHyLeOz5PDu7OXy9uXt8vAiPC9PUkd0TT4NCiAgICA8UE9JTlROT7S5fHyLeOz5PDu7OXyPC9QT0LOVE5NPg0KICAgIDxQT0LOVEFERFI+08rQwK/NwCwg7C4gyuj14iwgwOTw5fHgPC9QT0LOVEFERFI+DQogICAgPE9SREVSREFURT4yMjA3MjAyMTwvT1JERVJEQVRFPg0KICAgIDxPUkRFU1RJTUU+MTIyNTE2PC9PUkRFU1RJTUU+DQogICAgPE9SREVST1VNPjA8L09SREVST1VNPg0KICAgIDxQVNIREVTS05VTT4zPC9DQVNIREVTS05VTT4NCiAgICA8Q0FTSFJFR01TVEVST1VNPjQwMDAwMDc2MjE8L0NBUE0hSRUdJU1RFUk5VTT4NCiAgICA8Q0FTSElFUj7L4OHu8ODy7vDt6Okgy4gzY48L0NBUE0hJRVI+DQogICAgPFZFUj4xPC9WRVI+DQogIDwvWlJFUEhFQUQ+DQogIDxaUkVQUkVBTElaPg0KICAgIDxTVU0+NzI0LjY4PC9TVU0+DQogICAgPE9SREVVSU0NOVD4yPC9PUkRFU1NDTlQ+DQogICAgPFBBWUZPuk1TPg0KICAgICAgPFJPVjBSTldOVU09IjEiPg0KICAgICAgICA8UEFZRk9STUNEPjA8L1BBWUZPuk1DRD4NCiAgICAgICAgPFBBWUZPuk1OTT7DztKywsrAPC9QQVlGT1JNTk0+DQogICAgICAgIDxTVU0+NzI0LjY4PC9TVU0+DQogICAgICA8L1JPVz4NCiAgICA8L1BBWUZPuk1TPg0KICAgIDxUQVhFUz4NCiAgICAgIDxST1cgUk9XTlVNPSIxIj4NCiAgICAgICAgPFRZUEU+MDwvVFlQRT4NCiAgICAgICAgPE5BTUU+z8TCPC9QU1FPg0KICAgICAgICA8TEVUVEVSPsA8L0xVFRFUj4NCiAgICAgICAgPFBSQz4yMC4wMDwvUFJDPg0KICAgICAgICA8VFVSTk9WRVI+NzMyLjAwPC9UVVJOT1ZFUj4NCiAgICAgICAgPFNPVJDRVNVTT43MjQuNjg8L1NPVJDRVNVTT4NCiAgICAgICAgPFNVTT4xMjAuNzg8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgIDwvVEFYRVM+DQogIDwvWlJFUFJFQUxJWj4NCiAgPFpSRVBSRVRVUk4+DQogICAgPFNVTT43MjQuNjg8L1NVTT4NCiAgICA8T1JERVJTTQ05UPjI8L09SREVVSU0NOVD4NCiAgICA8UEFZRk9STVM+DQogICAgICA8Uk9XIFJFPV05VTT0iMSI+DQogICAgICAgIDxQQVlGT1JNQ0Q+MDwvUEFZRk9STUNEPg0KICAgICAgICA8UEFZRk9STU5NPsPO0rLCysA8L1BBWUZPuk1OTT4NCiAgICAgICAgPFNVTT43MjQuNjg8L1NVTT4NCiAgICAgIDwvUk9XPg0KICAgIDwvUEFZRk9STVM+DQogICAgPFRBWVTPg0KICAgICAgPFJPVjBSTldOVU09IjEiPg0KICAgICAgICA8VF1QRT4wPC9UWVBFPg0KICAgICAgICA8TktFNRT7PxMI8L05BTUU+DQogICAgICAgIDxMRVRURVI+wDwvTEVUVEVSPg0KICAgICAgICA8UFJDPjIwLjAwPC9QUkM+DQogICAgICAgIDxUVVJOT1ZFUj43MzIuMDA8L1RVUk5PVkVSPg0KICAgICAgICA8U09VUkNFU1VNPjcyNC42ODwvU09VUkNFU1VNPg0KICAgICAgICA8U1VNPjEyMC43ODwvU1VNPg0KICAgICAgPC9ST1c+DQogICAgPC9UQVhFUz4NCiAgPC9aUkVQUkVUVVJOPg0KICA8WlJFUEJPRFk+DQogICAgPFNFU1ZJQ0VJTTlBVVD4wLjAwPC9TRVJWSUNFSU5QVWQ+DQogICAgPFNFU1ZJQ0VPVVRQVWQ+MC4wMDwvU0VSVklDRU9VVFVBVD4NCiAgPC9aUkVQZk9EWT4NCjwwLjFUD4=","ServiceInput":null,"ServiceOutput":null,"OrderCount":"2","OrderExpCount":"2","TotalSUM":"724,68","TotalSUMCash":"724,68","TotalSUMCard":null,"TotalTAXSUM":"120,78","ReturnSUM":"724,68","ReturnSUMCash":"724,68","ReturnSUMCard":null,"ReturnTAXSUM":"120,78","TotalTAX_A_20%_SUM":

```

```

"120,78", "TotalTAX_A_20%_TURNOVER": "724,68", "TotalTAX_A_20%_DISCOUNTSUM": "7,32", "
TotalTAX_A_20%_REMOVEDTAXES": null, "ReturnTAX_A_20%_SUM": "120,78", "ReturnTAX_A_20%
_TURNOVER": "724,68", "ReturnTAX_A_20%_DISCOUNTSUM": "7,32", "ReturnTAX_A_20%_REMOVED
TAXES": null}}
Ret.Return = true
Ret.ShiftID = 9de7f9d4-046e-4f86-b686-6ad20306893a
Ret.Type = 12000000
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<ZREP xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='zrep01.xsd'>
  <ZREPHEAD>
    <UID>2C8C30ED-7221-4F9A-BE54-8B52F3E8E279</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОБ  "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>122516</ORDERTIME>
    <ORDERNUM>0</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
  </ZREPHEAD>
  <ZREPREALIZ>
    <SUM>724.68</SUM>
    <ORDERSCNT>2</ORDERSCNT>
    <PAYFORMS>
      <ROW ROWNUM="1">
        <PAYFORMCD>0</PAYFORMCD>
        <PAYFORMNM>ГОТІВКА</PAYFORMNM>
        <SUM>724.68</SUM>
      </ROW>
    </PAYFORMS>
    <TAXES>
      <ROW ROWNUM="1">
        <TYPE>0</TYPE>
        <NAME>ПДВ</NAME>
        <LETTER>A</LETTER>
        <PRC>20.00</PRC>
        <TURNOVER>732.00</TURNOVER>
        <SOURCESUM>724.68</SOURCESUM>
        <SUM>120.78</SUM>
      </ROW>
    </TAXES>
  </ZREPREALIZ>
  <ZREPRETURN>
    <SUM>724.68</SUM>
    <ORDERSCNT>2</ORDERSCNT>
    <PAYFORMS>
      <ROW ROWNUM="1">
        <PAYFORMCD>0</PAYFORMCD>
        <PAYFORMNM>ГОТІВКА</PAYFORMNM>
        <SUM>724.68</SUM>
      </ROW>
    </PAYFORMS>
    <TAXES>

```

```

<ROW ROWNUM="1">
  <TYPE>0</TYPE>
  <NAME>ПДВ</NAME>
  <LETTER>A</LETTER>
  <PRC>20.00</PRC>
  <TURNOVER>732.00</TURNOVER>
  <SOURCESUM>724.68</SOURCESUM>
  <SUM>120.78</SUM>
</ROW>
</TAXES>
</ZPREPRETURN>
<ZREPBODY>
  <SERVICEINPUT>0.00</SERVICEINPUT>
  <SERVICEOUTPUT>0.00</SERVICEOUTPUT>
</ZREPBODY>
</ZREP>

```

27. GetLastZReport – Отримати останній Z-звіт

Виклик функції **GetLastZReport ()** дозволяє отримати інформацію по останньому зареєстрованому Z-звіту у вигляді структури **CashalotApiRetVal**.

Параметри

Ім'я	Тип	Опис
FiscalNumberRRO	String	Ідентифікатор пристрою *
ShowPrintReport	Boolean	Ознака необхідності вивести звіт на друк. "true" - на екран виводиться візуальне відображення Z-звіту для перегляду та можливість вручну роздрукувати звіт (аналогічно функції PrintXReport()).

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення

JsonVal	String	JSON-структура, яка містить інформацію останнього зареєстрованого Z-звіту
Return	Bool	Успішність операції
ShiftId	String	Ідентифікатор зміни, в якій сформовано Z-звіт
Type	Integer	Тип документу чеку - 11000000
XML	String	XML-структура Z-звіту
ReceiptFiscalNum	String	Фіскальний номер документу
ReceiptLocalNum	String	Локальний номер документу

JsonVal містить наступні об'єкти:

"ServiceInput"	Загальна сума службових внесень по зміні
"ServiceOutput"	Загальна сума службових видач по зміні

"OrderCount"	Кількість чеків продажу
"OrderExpCount"	Кількість чеків повернення
"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 продажу "Готівка"
"ReturnSUMCard"	Загальна сума продажу "Картка"
"ReturnTAXSUM"	Загальна сума ПДВ по продажу
"TotalTAX_<1>_<2>%_<3>"	Деталізація податків по фіскальним чекам
"ReturnTAX_<1>_<2>%_<3>"	Деталізація податків по чекам повернення
"Base64Str1251ReportXML"	Дані Z-звіту у вигляді XML-структури в кодуванні Windows1251 конвертованої в Base64-рядок

Назви об'єктів податків чеків оплати TotalTAX та повернення ReturnTAX побудовані з використанням складових:

Складова	Зміст складової	Можливі значення	Приклад
<1>	Літера податку зазначена в картках товарів	А, Б, В, ...	TotalTAX_A_20%_SUM ReturnTAX_Б_5%_TURNOVER TotalTAX_Є_10%_REMOVEDTAXES
<2>	Відсоток податку	0, 5, 20, ...	
<3>	Сума	SUM	
	Обіг	TURNOVER	
	Знижка	DISCOUNTSUM	
	Виключені податки	REMOVEDTAXES	

Приклад:

```

FiscalNumRRO = "4000007621"
ShowPrintReport = false
App = CreateObject("AddIn.CashaLotApi")
Ret = App.GetLastZReport(FiscalNumRRO, ShowPrintReport)

```

Приклад відповіді:


```

Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"2c93c78e-f8db-4672-8748-0f21e97545da","ReceiptFiscalNumber":"442727","ReceiptLocalNumber":"804","OfflineMode":"False","Type":"11000000","Base64Str1251ReportXML":"PD94bWwgdmVyc2lvdj0iMS4wIiBlbmNvZGluZz0id2luZG93cy0xMjUxIj8+DQog8WlJFUCB4bWxuczp4c2k9Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hLWluc3RhbmNlIiB4c2k6bm90YW1lc3BhY2VTY2h1bWFMb2Nh dGlvbj0ienJlcDAXLnzhZCI+DQogIDxaUkVQSEVBRD4NCiAgICA8VU1EPjYyRjQwQUYyLUJDREMtNDIzNC05NjQzLUU4MzY1QUYxOTNGRTwvVU1EPg0KICAgIDxUSU4+MjQyMzQyMjc8L1RJTj4NCiAgICA8T1JHTk0+0s7CICLS5fHyLeOz5PDu7OXy9uXt8vAiPC9PUkdOTT4NCiAgICA8UE9JTlROTT7S5fHyLeOz5PDu7OXyPC9QT0lOVE5NPg0KICAgIDxQT0lOVEFERFI+08rQwK/NwCwg7C4gyujl4iwwOTw5fHgPC9QT0lOVEFERFI+DQogICAgPE9SREVSREFURT4yMjA3MjAyMTwvT1JERVJEQVRFPg0KICAgIDxPUkRFU1RJTUU+MDkzNDA1PC9PUkRFU1RJTUU+DQogICAgPE9SREVSTlVNPjgWdWvT1JERVJOUVU0+DQogICAgPENBU0hERVNLTlVNPjM8L0NB U0hERVNLTlVNPg0KICAgIDxDQVNIUkVHSVNUVJOUVU0+NDAAwMDAwNzYyMTwvQ0FTSFJFR01TVEVSTlVNPg0KICAgIDxDQVNISUVSPsvg4e7w4PLu8O3o6SDPLiDPLjwvQ0FTSElFUj4NCiAgICA8VkVSPjE8L1ZFUj4NCiAgICA8T1JERVJUQVhOVU0+NDQyNzI3PC9PUkRFU1RBWE5VTT4NCiAgPC9aUkVQSEVBRD4NCiAgPFpSRVBSRUFSMSVo+DQogICAgPFNVTT4zNjIuMzQ8L1NVTT4NCiAgICA8T1JERVJTQ05UPjE8L09SREVSU0NOVD4NCiAgICA8UEFZRk9STVM+DQogICAgICA8Uk9XIFJPV05VTT0iMSI+DQogICAgICA8IDxQQVlGT1JNQ0Q+MDwvUEFZRk9STUNEPg0KICAgICA8ICA8UEFZRk9STU5NPSP00rLCysA8L1BBWUZPUk1OTT4NCiAgICAgICA8PFNVTT4yMTIuMzQ8L1NVTT4NCiAgICA8IDwvUk9XPg0KICAgICA8PFJFVjBST1dOVU09IjIiPg0KICAgICA8ICA8UEFZRk9STUNEPjI8L1BBWUZPUk1DRD4NCiAgICA8ICA8PFBBWUZPUk1OTT7RxdDSyNSyysDSPC9QQVlGT1JNTk0+DQogICAgICA8IDxTVU0+NTAuMDA8L1NVTT4NCiAgICA8IDwvUk9XPg0KICAgICAgPFJFVjBST1dOVU09IjMiPg0KICAgICA8ICA8UEFZRk9STUNEPjM8L1BBWUZPUk1DRD4NCiAgICA8ICA8PFBBWUZPUk1OTT7K0MXEyNI8L1BBWUZPUk1OTT4NCiAgICA8ICA8PFNVTT4xMDAuMDA8L1NVTT4NCiAgICA8IDwvUk9XPg0KICAgIDwvUEFZRk9STVM+DQogICAgPFRBWEVTPg0KICAgICA8PFJFVjBST1dOVU09IjEiPg0KICAgICA8ICA8VF1QRT4wPC9UWVBFPG0KICAgICA8ICA8TkFNRT7PxMI8L05BTUU+DQogICAgICAgIDxMRVRURVI+wDwvTEVUVEVSPg0KICAgICA8ICA8UFJDPjIwLjAwPC9QUkM+DQogICAgICA8IDxUVVJOT1ZFUj4zNjYuMDA8L1RVUk5PVkVSPg0KICAgICA8ICA8U09VUkNFU1VNPjM2Mi4zNDwvU09VUkNFU1VNPg0KICAgICA8ICA8U1VNPjYwLjM5PC9TVU0+DQogICAgICA8L1JPVz4NCiAgICA8L1RBWEVTPg0KICAgL1pSRVBSRUFSMSVo+DQogIDxaUkVQKk9EWT4NCiAgICA8U0VSVklDRU1OUFVUPjAuMDA8L1NFU1ZJQ0VJTlBVVD4NCiAgICA8U0VSVklDRU1VVFVVD4wLjAwPC9TRVJWSUNFTlVUUFVUPg0KICAgL1pSRVBCT0RZPg0KPC9aUkVQPg==","ServiceInput":"0","ServiceOutput":"0","OrderCount":"1","OrderExpCount":"0","TotalSUM":"362,34","TotalSUMCash":"212,34","TotalSUMCard":null,"TotalTAXSUM":"60,39","ReturnSUM":null,"ReturnSUMCash":null,"ReturnSUMCard":null,"ReturnTAXSUM":null,"TotalTAX_A_20%_SUM":"60,39","TotalTAX_A_20%_TURNOVER":"362,34","TotalTAX_A_20%_DISCOUNTSUM":"3,66","TotalTAX_A_20%_REMOVEDTAXES":null}}

Ret.Return = true
Ret.ShiftID = 2c93c78e-f8db-4672-8748-0f21e97545da
Ret.ReceiptFiscalNum = 442727
Ret.ReceiptLocalNum = 804
Ret.XML = "<?xml version='\"1.0\"' encoding='\"windows-1251\"' ?>
<ZREP xmlns:xsi='\"http://www.w3.org/2001/XMLSchema-instance\"'
xsi:noNamespaceSchemaLocation='\"zrep01.xsd\"'>
  <ZREPHEAD>
    <UID>62F40AF2-BCDC-4234-9643-E8365AF193FE</UID>
    <TIN>24234227</TIN>
    <ORGNM>TOB  \"\"Тест-гідрометцентр\"\"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>093405</ORDERTIME>
    <ORDERNUM>804</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>442727</ORDERTAXNUM>
  </ZREPHEAD>

```

```

<ZPREPREALIZ>
  <SUM>362.34</SUM>
  <ORDERSCNT>1</ORDERSCNT>
  <PAYFORMS>
    <ROW ROWNUM="1">
      <PAYFORMCD>0</PAYFORMCD>
      <PAYFORMNM>ГОТІВКА</PAYFORMNM>
      <SUM>212.34</SUM>
    </ROW>
    <ROW ROWNUM="2">
      <PAYFORMCD>2</PAYFORMCD>
      <PAYFORMNM>СЕРТИФІКАТ</PAYFORMNM>
      <SUM>50.00</SUM>
    </ROW>
    <ROW ROWNUM="3">
      <PAYFORMCD>3</PAYFORMCD>
      <PAYFORMNM>КРЕДИТ</PAYFORMNM>
      <SUM>100.00</SUM>
    </ROW>
  </PAYFORMS>
  <TAXES>
    <ROW ROWNUM="1">
      <TYPE>0</TYPE>
      <NAME>ПДВ</NAME>
      <LETTER>A</LETTER>
      <PRC>20.00</PRC>
      <TURNOVER>366.00</TURNOVER>
      <SOURCESUM>362.34</SOURCESUM>
      <SUM>60.39</SUM>
    </ROW>
  </TAXES>
</ZPREPREALIZ>
<ZREPBODY>
  <SERVICEINPUT>0.00</SERVICEINPUT>
  <SERVICEOUTPUT>0.00</SERVICEOUTPUT>
</ZREPBODY>
</ZREP>"

```

28. PrintZReport – Формування Z-звіту

Виклик функції **PrintZReport()** призводить до виконання послідовності операцій аналогічних до виклику функції **CloseShift()**: службова видача з каси наявної готівки, формування Z-звіту та закриття зміни. Проте, виконання даної функції повертає ознаку успішності операції, в той час як виконання **CloseShift()** повертає структуру **CashalotApiRetVal**.

Параметри

Ім'я	Тип	Опис
<u>FiscalNumRRO</u>	String	Ідентифікатор пристрою

Відповідь

	Bool	Успішність операції
--	------	---------------------

Приклад:

```

FiscalNumRRO = "4000007621"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.PrintZReport(FiscalNumRRO)

```

29. CloseShift - Закриття зміни

Функція використовується для закриття поточної зміни.

Параметри

Ім'я	Тип	Опис
FiscalNumRRO	String	Ідентифікатор пристрою

Відповідь

В якості відповіді повертається структура **CashalotApiRetVal**, що містить значення Z-звіту закритої касової зміни:

JsonVal	String	JSON-структура, що містить значення успішності операції, тип документу, помилки, локальний та фіскальний номери документу Z-звіту, ID зміни та Base64 рядок XML структури звіту
ReceiptFiscalNum	String	Фіскальний номер Z-звіту
ReceiptLocalNum	String	Локальний номер Z-звіту
Return	Bool	Успішність операції
ShiftID	String	Ідентифікаційний номер закритої зміни
Type	Integer	Тип документу Z-звіту -
XML	String	XML-структура Z-звіту

JsonVal містить наступні об'єкти:

"ServiceInput"	Загальна сума службових внесень по зміні
"ServiceOutput"	Загальна сума службових видач по зміні
"OrderCount"	Кількість чеків продажу
"OrderExpCount"	Кількість чеків повернення
"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 продажу "Готівка"
"ReturnSUMCard"	Загальна сума продажу "Картка"
"ReturnTAXSUM"	Загальна сума ПДВ по продажу
"TotalTAX_<1>_<2>%_<3>"	Деталізація податків по фіскальним чекам


```
ICA8TkFNRT7PxMI8L05BTUU+DQogICAgICAgIDxMRVRURVI+wDwvTEVUVEVSPg0KICAgICAgICA8UFJDP
jIwLjAwPC9QUkM+DQogICAgICAgIDxUVVJOT1ZFUj43MzIuMDA8L1RVUk5PVkVSPg0KICAgICAgICA8U0
9VUkNFU1VNPjcyNC42ODwvU09VUkNFU1VNPg0KICAgICAgICA8U1VNPjEyMC43ODwvU1VNPg0KICAgICA
gPC9ST1c+DQogICAgPC9UQVhFUz4NCiAgPC9aUkVQUkVBTElaPg0KICA8WlJFUfJFVfVSTj4NCiAgICA8
U1VNPjcyNC42ODwvU1VNPg0KICAgIDxPUkRFU1NDTlQ+MjwvT1JERVJTQ05UPg0KICAgIDxQQVlGT1JNU
z4NCiAgICAgIDxST1cgUk9XTlVNPSIxIj4NCiAgICAgICAgPFBBWUZPUklDRD4wPC9QQVlGT1JNQ0Q+DQ
ogICAgICAgIDxQQVlGT1JNTk0+w87SssLKwDwvUEFZrk9STU5NPg0KICAgICAgICA8U1VNPjcyNC42ODw
vU1VNPg0KICAgICAgPC9ST1c+DQogICAgPC9QQVlGT1JNUz4NCiAgICA8VEFYRVM+DQogICAgICA8Uk9X
IFJpV05VTT0iMSI+DQogICAgICAgIDxUWVBFPjA8L1RZUEU+DQogICAgICAgIDxOQU1FPs/EwjwvTkFNR
T4NCiAgICAgICAgPExFVFRFUj7APC9MRVRURVI+DQogICAgICAgIDxQUkM+MjAuMDA8L1BSQz4NCiAgIC
AgICAgPFRVUk5PVkVSPjczMi4wMDwvVFVSTk9WRVI+DQogICAgICAgIDxTT1VSQ0VTVU0+NzI0LjY4PC9
TT1VSQ0VTVU0+DQogICAgICAgIDxTVU0+MTIwLjcz4PC9TVU0+DQogICAgICA8L1JPVz4NCiAgICA8L1RB
WEVTPg0KICA8L1pSRVBSRVRVUk4+DQogIDxaUkVQQk9EWT4NCiAgICA8U0VSVklDRU1OUFVUPjUwLjAwP
C9TRVJWSUNFSU5QVQ+DQogICAgPFNFU1ZJQ0VPVVRQVQ+MzAuMDA8L1NFU1ZJQ0VPVVRQVQ+DQogID
wvWlJFUEJPRFk+DQo8L1pSRVA+", "ServiceInput": "50", "ServiceOutput": "30", "OrderCount"
: "2", "OrderExpCount": "2", "TotalSUM": "724,68", "TotalSUMCash": "724,68", "TotalSUMCar
d": null, "TotalTAXSUM": "120,78", "ReturnSUM": "724,68", "ReturnSUMCash": "724,68", "Ret
urnSUMCard": null, "ReturnTAXSUM": "120,78", "TotalTAX_A_20%_SUM": "120,78", "TotalTAX_
A_20%_TURNOVER": "724,68", "TotalTAX_A_20%_DISCOUNTSUM": "7,32", "TotalTAX_A_20%_REMO
VEDTAXES": null, "ReturnTAX_A_20%_SUM": "120,78", "ReturnTAX_A_20%_TURNOVER": "724,68"
, "ReturnTAX_A_20%_DISCOUNTSUM": "7,32", "ReturnTAX_A_20%_REMOVEDTAXES": null}}
Ret.Return = True
Ret.ShiftID = 9de7f9d4-046e-4f86-b686-6ad20306893a
Ret.ReceiptFiscalNum = 443043
Ret.ReceiptLocalNum = 813
Ret.XML = "<?xml version='1.0' encoding='windows-1251'?>
<ZREP xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:noNamespaceSchemaLocation='\"zrep01.xsd\">
  <ZREPHEAD>
    <UID>90FDEE47-F299-4DAD-B482-89C737F5CF3D</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОБ \"Тест-гідрометцентр\"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>22072021</ORDERDATE>
    <ORDERTIME>143143</ORDERTIME>
    <ORDERNUM>813</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>443043</ORDERTAXNUM>
  </ZREPHEAD>
  <ZPREALIZ>
    <SUM>724.68</SUM>
    <ORDERSCNT>2</ORDERSCNT>
    <PAYFORMS>
      <ROW ROWNUM='1'>
        <PAYFORMCD>0</PAYFORMCD>
        <PAYFORMNM>ГОТІВКА</PAYFORMNM>
        <SUM>724.68</SUM>
      </ROW>
    </PAYFORMS>
    <TAXES>
      <ROW ROWNUM='1'>
        <TYPE>0</TYPE>
        <NAME>ПДВ</NAME>
```

```

        <LETTER>A</LETTER>
        <PRC>20.00</PRC>
        <TURNOVER>732.00</TURNOVER>
        <SOURCESUM>724.68</SOURCESUM>
        <SUM>120.78</SUM>
    </ROW>
</TAXES>
</ZREPREALIZ>
<ZREPRETURN>
    <SUM>724.68</SUM>
    <ORDERSCNT>2</ORDERSCNT>
    <PAYFORMS>
        <ROW ROWNUM="1">
            <PAYFORMCD>0</PAYFORMCD>
            <PAYFORMNM>ГОТОВКА</PAYFORMNM>
            <SUM>724.68</SUM>
        </ROW>
    </PAYFORMS>
    <TAXES>
        <ROW ROWNUM="1">
            <TYPE>0</TYPE>
            <NAME>ПДВ</NAME>
            <LETTER>A</LETTER>
            <PRC>20.00</PRC>
            <TURNOVER>732.00</TURNOVER>
            <SOURCESUM>724.68</SOURCESUM>
            <SUM>120.78</SUM>
        </ROW>
    </TAXES>
</ZREPRETURN>
<ZREPBODY>
    <SERVICEINPUT>50.00</SERVICEINPUT>
    <SERVICEOUTPUT>30.00</SERVICEOUTPUT>
</ZREPBODY>
</ZREP>"

```

NEW

30. Приклад роботи з кількома ПРРО одночасно в одній обліковій системі використовується новий метод [SetParameterEx](#)

Приклад відкриття зміни:

Процедура ПодключениеСОМОбъект()

Если ComObjApp = Неопределено Тогда

ИмяПлагина = "C:\Users\User\AppData\Roaming\Cashalot\CashaLotApi32.dll";

ПодключитьВнешнююКомпоненту(ИмяПлагина, "CashaLotApi");

Попытка

Компонент = Новый("AddIn.CashaLotApi.CashaLotApi");

Исключение

ПодключитьВнешнююКомпоненту("AddIn.CashaLotApi");

Компонент = Новый("AddIn.CashaLotApi");

КонецПопытки;

ComObjApp = Новый СОМОбъект("AddIn.CashaLotApi");

FiscalNumberRRO1 = "XXXXXXXXXX";

```
ComObjApp.SetParameterEx (FiscalNumberRRO1, "PathToCashalotDir",  
"C:\Users\User\AppData\Roaming\Cashalot\Cashalot1");  
ComObjApp.SetParameterEx (FiscalNumberRRO1, "DeviceIDFnRRO", FiscalNumberRRO1 );  
ComObjApp.SetParameterEx (FiscalNumberRRO1, "NOINTERFACEMODE", "False");  
ComObjApp.SetParameterEx (FiscalNumberRRO1, "PathToCertificate",  
"C:\Cashalot\Cashalot1\Ключи\Касир");  
ComObjApp.SetParameterEx (FiscalNumberRRO1, "PwdToCertificate", "111");  
ComObjApp.SetParameterEx (FiscalNumberRRO1, "USETOKEN", "False");  
ComObjApp.SetParameterEx (FiscalNumberRRO1, "NOAUTOOPENSHIFT", "False");  
ComObjApp.SetParameterEx (FiscalNumberRRO1, "DEFAULTPRINTERNAME", "");  
ComObjApp.SetParameterEx (FiscalNumberRRO1, "AUTOPRINTMODE", "True");
```

FiscalNumberRRO2 = "YYYYYYYYYY";

```
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "PathToCashalotDir" ),  
"C:\Users\User\AppData\Roaming\Cashalot\Cashalot2");  
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "DeviceIDFnRRO", FiscalNumberRRO2 );  
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "NOINTERFACEMODE", "False");  
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "PathToCertificate",  
"C:\Cashalot\Cashalot2\Ключи\Старший касир");  
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "PwdToCertificate", "111");  
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "USETOKEN", "False");  
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "NOAUTOOPENSHIFT", "False");  
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "DEFAULTPRINTERNAME", "");  
ComObjApp.SetParameterEx ( FiscalNumberRRO2, "AUTOPRINTMODE", "True");
```

КонецЕсли;

КонецПроцедуры

-

\\\\\\ Далі виконуємо відкриття змін для кас FiscalNumberRRO1 = " XXXXXXXXXXXX" та FiscalNumberRRO2 = "YYYYYYYYYY"

Процедура OpenShift (Команда)

ПодключениеСОМОбъект();

```
gh = ComObjApp.OpenShift(FiscalNumberRRO1);  
Сообщить("Зміна відкрита ");
```

КонецПроцедуры

&НаКлиенте

Процедура OpenShift_2 (Команда)

ПодключениеСОМОбъект();

```
gh = ComObjApp.OpenShift(FiscalNumberRRO2);
```

Сообщить ("Зміна відкрита ");

КонецПроцедуры

NEW 31. Можна задавати параметри, які ПРРО використовувати для кожного методу окремо.

Приклад для відкриття зміни

Процедура OpenShift (Команда)

ИмяПлагина = "C:\ProgramData\Cashalot\Cashalot2\CashaLotApi64.dll";

ПодключитьВнешнююКомпоненту (ИмяПлагина, "CashaLotApi");

FiscalNumberRRO = "XXXXXXXXXX";

Попытка

Компонент = Новый ("AddIn.CashaLotApi.CashaLotApi");

Исключение

ПодключитьВнешнююКомпоненту ("AddIn.CashaLotApi");

Компонент = Новый ("AddIn.CashaLotApi");

КонецПопытки;

ComObjApp = Новый COMОбъект ("AddIn.CashaLotApi");

ComObjApp.SetParameterEx (FiscalNumberRRO, "PathToCashalotDir",
"C:\ProgramData\Cashalot\Cashalot_2\");

ComObjApp.SetParameterEx FiscalNumberRRO, ("DeviceIDFnRRO", FiscalNumberRRO);

ComObjApp.SetParameterEx (FiscalNumberRRO, "AUTOPRINTMODE", "True");

gh = ComObjApp.OpenShift (FiscalNumberRRO);

Сообщить ("Зміна відкрита ");

КонецПроцедуры

- Реєстрація COM об'єкта

- Завдання налаштувань для кількох ПРРО за допомогою нового методу

SetParameterEx

- робота одночасно з двома примірниками програм SetParameterEx:

<u>OpenShift</u>	(Открытие смены)
<u>FiscalizeReceipt</u>	(Регистрация чека)
<u>FiscalizeReturnCheck</u>	(Возврат чека)
<u>ServiceInput</u>	(Служебное внесение)
<u>ServiceOutput</u>	(Служебная выдача)
<u>PrintXReport</u>	(X отчет)
<u>PrintZReport</u>	(Z отчет)
<u>CloseShift</u>	(Закрытие смены)
<u>GetReceiptLocalNumber</u>	(Получение локального номера чека)
<u>GetReceiptFiscalNumber</u>	(Получение фискального номера чека)
<u>GetReceiptXML</u>	(Получение XML чека)
<u>GetCurrentStatus</u>	(Состояние кассы)

32. Ломбардні операції

NEW

Для фіскалізації ломбардних послуг введено системні номенклатури, які необхідно використовувати при виклику методу фіскалізації чеку `Fiscalizecheck()`. Повернення товару при ломбардних операціях не підтримується.

Також необхідно вказати тип документу `DocType = 4` в JSON-структурі параметрів чеку.

За необхідності можна вказати номер договору `DocNumber`, за яким надається ломбардна послуга.

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"	+

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

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

Приклад

```
Set App = CreateObject("AddIn.CashaLotApi")
DeviceID = "4000007621"
Itmdata = [{"ReceiptLst": [{"VendorCode": "ЛО-00009", "Name": "Повернення кредиту", "DocNumber": "01/a12 від 15.08.21", "GoodsType": "2", "CdUKDZED": null, "CdDKPP": null, "Barcode": null, "UnitType": "шт", "Quantity": "1", "Price": "100", "Amount": "100", "DiscountPrc": null, "DiscountSum": null, "IsPriceIncludeVAT": false, "VATRate": null, "SumVAT": null, "IsExcise": false, "OtherParams": null}, {"VendorCode": "ЛО-00002", "Name": "Надання кредиту", "DocNumber": "31/б12 від 15.08.21", "GoodsType": "3", "CdUKDZED": null, "CdDKPP": null, "Barcode": null, "UnitType": "шт", "Quantity": "1", "Price": "150", "Amount": "150", "DiscountPrc": null, "DiscountSum": null, "IsPriceIncludeVAT": false, "VATRate": null, "SumVAT": null, "IsExcise": false, "OtherParams": null}]}]
```

Приклад відповіді

```
{ "Ret": true, "ErrorString": "", "Values": { "ShiftID": "17b433a0-1021-4d8c-b954-1bc3bfbf2950", "ReceiptFiscalNumber": "527988", "ReceiptLocalNumber": "1375", "OfflineMode": "False", "Type": "10004000", "Base64Str1251ReceiptXML": "PD94bWwgdmVyc2lvdj0iMS4wIiBlbmNvZGluc2lud2luZG93cy0xMjUxIj8DOo8O0hFO0sqeG1spnM6eHNpPSJodHRwOi8vd3d3Lnk=
```

zLm9yZy8yMDAxL1hNTFNjaGVtYS1pbmN0YW5jZSIgeHNpOm5vTmFtZXNwYWNlU2NoZWl1hTG9jYXRpb249
ImNoZWNRmDEueHNkIj4NCiAgPENIRUNLSEVBRD4NCiAgICA8RE9DVF1QRT40PC9ET0NUWVBFPg0KICAgI
DxET0NTVUJUWVBFPjA8L0RPQ1NVQ1RZUEU+DQogICAgPFVJRd5CNzFCMjRGQi010TEzLTQxQTktOENERS
04ODYxOTRCNkEzOTU8L1VJRd4NCiAgICA8VElOPjI0MjM0MjI3PC9USU4+DQogICAgPE9SR05NPtLOWiA
i0uXx8i3js+Tw7uzl8vbl7fLwIjwvTlJHTk0+DQogICAgPFbPSU5UTk0+0uXx8i3js+Tw7uzl8jwvUE9J
TlROTT4NCiAgICA8UE9JTlRBRERSPtPK0MCvzcAsIOwIMro5eIsIMdk8OXx4DwvUE9JTlRBRERSPg0KI
CAGIDxPukRFukRBVEU+MDcxMDIwMjE8L09SREVSREFURT4NCiAgICA8TlJERVJUSU1FPjE1MjM0MDwvTl
JERVJUSU1FPg0KICAgIDxPukRFuk5VTT4xMzc1PC9PukRFuk5VTT4NCiAgICA8Q0FTSERFU0tOVU0+Mzw
vQ0FTSERFU0tOVU0+DQogICAgPENBU0hSRUdJU1RFuk5VTT40MDAwMDA3NjIxPC9DQVNIUkVHsvNURVJO
VU0+DQogICAgPENBU0hJRVI+y+Dh7vDg8u7w7eJpIM8uIM8uPC9DQVNISUVSPg0KICAgIDxDT01NRU5UP
tLl8fLu4uAg6+7s4eDw5O3gIO7v5fDg9rP/PC9DT01NRU5UPg0KICAgIDxWRVI+MTwvVkvSPg0KICAgID
xPukRFU1RBWE5VTT41MjE8L09SREVSVEFYtLVNPg0KICAgIDwhLS08Q0FTSEFMT1RDT01NRU5UPg0
KICA8UEFZTUVOVFNUQVRVUz40PC9QVlNRU5UU1RBVfVTPg0KPC9DQVNIUkxPVENPTU1FTlQ+LS0+DQog
IDwvQ0hFQ0tIRUFEPg0KICA8Q0hFQ0tUT1RBTd4NCiAgICA8U1VNPi01MC4wMDwvU1VNPg0KICAgIDxQV
05TVU1JU1NVRUQ+MTUwLjAwPC9QV05TVU1JU1NVRUQ+DQogICAgPFbXTlNVTVJFQ0VJVkVEPjEwMC4wMD
wvUfDOU1VNukVDRULWRUQ+DQogIDwvQ0hFQ0tUT1RBTd4NCiAgPENIRUNLUEFZPg0KICAgIDxST1cgUk9
XTlVNPSIxIj4NCiAgICAgIDxQQVlGT1JNQ0Q+MDwvUEFZrk9STUNEPg0KICAgICAgPFBBWUZPUk1OTT7D
ztKywsrAPC9QQVlGT1JNTk0+DQogICAgICA8U1VNPi01MC4wMDwvU1VNPg0KICAgIDwvUk9XPg0KICA8L
0NIRUNLUEFZPg0KICA8Q0hFQ0tCT0RZPg0KICAgIDxST1cgUk9XTlVNPSIxIj4NCiAgICAgIDxDT0RFPs
vOLTAwMDA5PC9DT0RFPg0KICAgICAgPE5BTUU+z+7i5fDt5e3t/yDq8OXk6PLzPC9OQU1FPg0KICAgICA
gPFVOSVRDRD4yMDA5PC9VTklUQ0Q+DQogICAgICA8VU5JVE5NPvjyPC9VTklUTk0+DQogICAgICA8QU1P
VU5UPjEuMDAwPC9BTU9VTlQ+DQogICAgICA8UFJJQ0U+MTAwLjAwPC9QUklDRT4NCiAgICAgIDxDT1NUP
jEwMC4wMDwvQ09TVD4NCiAgICAgIDxQQVlFUlBBQ1ROVU0+MDEvYTEyIokZ5CAxNS4wOC4yMTwvUEFZRV
JQQUNUTlVNPg0KICAgICAgPFbXTkRjUj50cnVlPC9QV05ESVI+DQogICAgPC9ST1c+DQogICAgPFJPVYB
STldOVU09IjIiPg0KICAgICAgPENPREU+y84tMDAwMDI8L0NPRED+DQogICAgICA8TkFNRT7N4OTg7e3/
IORw5eTo8vM8L05BTUU+DQogICAgICA8VU5JVENEPjIwMDk8L1VOSVRDRD4NCiAgICAgIDxVTklUTk0++
PI8L1VOSVROTT4NCiAgICAgIDxBTU9VTlQ+MS4wMDA8L0FNT1VOVD4NCiAgICAgIDxQUklDRT4xNTAuMD
A8L1BSSUNFPg0KICAgICAgPENPU1Q+MTUwLjAwPC9DT1NUPg0KICAgICAgPFBBWUVSUEFDVE5VTT4zMS/
hMTIg4rPkIDE1LjA4LjIxPC9QVlFUlBBQ1ROVU0+DQogICAgICA8UFdORElSPmZhbHNlPC9QV05ESVI+
DQogICAgPC9ST1c+DQogIDwvQ0hFQ0tCT0RZPg0KPC9DSEVDSz4="}} }

Ret.ReceiptFiscalNum = 527988

Ret.ReceiptLocalNum = 1375

Ret.Return = true

Ret.Type = 10004000

ret.ShiftID = 17b433a0-1021-4dfc-b954-1bc3bfbf2950

Ret.XML = "<?xml version='1.0' encoding='windows-1251'>

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

<CHECKHEAD>

<DOCTYPE>4</DOCTYPE>

<DOCSUBTYPE>0</DOCSUBTYPE>

<UID>B71B24FB-5913-41A9-8CDE-886194B6A395</UID>

<TIN>24234227</TIN>

<ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>

<POINTNM>Тест-гідромет</POINTNM>

<POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>

<ORDERDATE>07102021</ORDERDATE>

<ORDERTIME>152340</ORDERTIME>

<ORDERNUM>1375</ORDERNUM>

<CASHDESKNUM>3</CASHDESKNUM>

<CASHREGISTERNUM>4000007621</CASHREGISTERNUM>

<CASHIER>Лабораторний П. П.</CASHIER>

<COMMENT>Тестова ломбардна операція</COMMENT>

<VER>1</VER>

<ORDERTAXNUM>527988</ORDERTAXNUM>

<!--<CASHALOTCOMMENT>

```

    <PAYMENTSTATUS>4</PAYMENTSTATUS>
</CASHALOTCOMMENT>-->
</CHECKHEAD>
<CHECKTOTAL>
    <SUM>-50.00</SUM>
    <PWNSUMISSUED>150.00</PWNSUMISSUED>
    <PWNSUMRECEIVED>100.00</PWNSUMRECEIVED>
</CHECKTOTAL>
<CHECKPAY>
    <ROW ROWNUM="1">
        <PAYFORMCD>0</PAYFORMCD>
        <PAYFORMNM>ГОТІВКА</PAYFORMNM>
        <SUM>-50.00</SUM>
    </ROW>
</CHECKPAY>
<CHECKBODY>
    <ROW ROWNUM="1">
        <CODE>ЛО-00009</CODE>
        <NAME>Повернення кредиту</NAME>
        <UNITCD>2009</UNITCD>
        <UNITNM>шт</UNITNM>
        <AMOUNT>1.000</AMOUNT>
        <PRICE>100.00</PRICE>
        <COST>100.00</COST>
        <PAYERPACTNUM>01/a12 від 15.08.21</PAYERPACTNUM>
        <PWNDIR>true</PWNDIR>
    </ROW>
    <ROW ROWNUM="2">
        <CODE>ЛО-00002</CODE>
        <NAME>Надання кредиту</NAME>
        <UNITCD>2009</UNITCD>
        <UNITNM>шт</UNITNM>
        <AMOUNT>1.000</AMOUNT>
        <PRICE>150.00</PRICE>
        <COST>150.00</COST>
        <PAYERPACTNUM>31/612 від 15.08.21</PAYERPACTNUM>
        <PWNDIR>false</PWNDIR>
    </ROW>
</CHECKBODY>
</CHECK>"

```

33. Рекомендований порядок роботи з CashalotApi

Для коректної роботи розрахунково-облікової системи рекомендується описана послідовність виклику функцій при виконанні касових операцій.

1. Встановлення параметрів

Рекомендовано за допомогою виклику функції

[SetParameter](#) або [SetParameterEx](#) (декілька PRPO)

встановити параметри в рекомендованій **послідовності**:

[FiscalNumRRO](#)

PathToCashalotDir

NOINTERFACEMODE

NOAUTOOPENSIFT

USETOKEN

PathToCertificate

PwdToCertificate

перед початком роботи.

2. Відкриття зміни

Для відкриття касової зміни необхідно скористатися функцією

OpenShift

3. Продаж товарів, послуг

Для формування чека та його фіскалізації створюються JSON-структури з масивом товарів та підсумками оплати по чеку, які передаються в якості параметрів в функцію

FiscalizeCheck

У чеку необхідно сформувати щонайменше 1 рядок товарів.

4. Повернення товарів, рекомпенсація послуг

Для повернення товару або рекомпенсації послуги необхідно скористатися функцією

FiscalizeReturnCheck

в якій потрібно вказати фіскальний номер чеку, за яким було продано товар або послугу.

5. Пошук фіскального та локального номеру чеку

Для визначення фіскального номеру чеку за його локальним номером використовується функція

GetReceiptFiscaNumberByLocalNumber

Для визначення локального номеру чеку за його фіскальним номером використовується функція

GetReceiptLocalNumberByFiscalNumber

6. Отримання XML-структури чеку

Функція

GetReceiptXML

повертає XML-структуру чеку за його вказаним фіскальним номером.

7. Відображення зареєстрованого чеку

Функція

ShowReceipt

відкриває вікно з візуальним відображенням чеку, яке аналогічне вікну з відображенням чеку після закриття чеку. У вікні є кнопки "Друк" та "Надіслати на email", які дозволяють роздрукувати або відправити на електронну пошту чек.

8. Налаштування електронної пошти

Електронну пошту можна налаштувати після фіскалізації чеку або виклику функції відображення чеку (**ShowReceipt**). Для цього, у вікні візуального відображення зареєстрованого чеку необхідно натиснути кнопку "Надіслати на email", після чого відкриється вікно налаштування електронної пошти ПРРО Кашалот.

Для налаштування необхідно зазначити електронну адресу, з якої будуть відправлятися повідомлення, а також налаштування з'єднання з поштовою скринькою: вказати SMTP-сервер, порт, ім'я для входу в поштову скриню, пароль поштової скрині.

Після успішного налаштування електронної пошти повторне натискання на кнопку "Надіслати на email" викликає діалогове вікно, в якому потрібно вказати електронну адресу клієнта, на яку відправляється чек, за необхідності додати вказану адресу в базу даних постійних клієнтів, та відправити чек.

9. Службове внесення грошей в касу

Для виконання службового внесення в касу грошей використовується функція

ServiceInput

або

ServiceInputEx

10. Службова видача грошей з каси

Для виконання службової видачі з каси використовується функція

ServiceOutput

або

ServiceOutputEx

11. Формування X-звіту

X-звіт формується в результаті виконання функції

PrintXReport

або

GetXReport

12. Отримання чеків

Для отримання чеків з історії операцій реалізовано функції, що передбачають завантаження чеків за певний період часу або за певну касову зміну.

Для формування списку чеків за заданий період часу скористайтеся функцією

GetFirstReceiptByPeriod

Для формування списку чеків за задану касову зміну скористайтеся функцією

GetFirstReceiptByShift

Обидві функції формують список чеків і повертають перший чек з сформованого списку.

Для отримання наступних чеків скористайтеся функцією

GetNextReceipt

необхідну кількість разів, або поки не буде вичерпано список чеків

13. Отримання документів Z-звітів закритих касових змін

Для отримання Z-звітів з історії операцій реалізовано функції, що передбачають завантаження документів за певний період часу або для певної касової зміни.

Для формування списку документів за заданий період часу скористайтеся функцією

GetFirstZReportByPeriod

Функція формує список чеків і повертає перший документ з сформованого списку.

Для отримання наступних документів скористайтеся функцією

GetNextZReport

необхідну кількість разів, або поки не буде вичерпано список документів.

Для отримання Z-звіту заданої касової зміни скористайтеся функцією

GetZReportByShift

Якщо звіт для вказаної касової зміни відсутній (наприклад, поточна касова зміна), функція поверне ознаку неуспішності операції.

14. Отримання останнього Z-звіту

Для отримання і перегляду останнього зареєстрованого Z-звіту використовується функція

GetLastZReport

15. Закриття зміни, Z-звіт

Формування Z-звіту відбувається при закритті зміни, тому, виконання операції закриття зміни та формування Z-звіту рівнозначні за результатом. Для цього виконується одна з функцій

PrintZReport

або

CloseShift

Виклик вказаних функцій призводить до виконання послідовності операцій: службова видача з каси наявної готівки, формування Z-звіту та закриття зміни. Проте, виконання функції

PrintZReport() повертає ознаку успішності операції, в той час як виконання **CloseShift()**

повертає структуру **CashlotApiRetVal**.

Приклад міститься у вкладенні вище.

34. Приклад реалізації підключення COM бібліотеки на C++

1. Створення файлів заголовків *.h

Якщо відсутні файли заголовків, або потрібно їх оновити:

1. генеруємо через імпорт, вказавши шлях до COM бібліотеки, наприклад:
`#import "C:\\Users\\User1\\AppData\\Local\\CashaLot\\CashaLotApi64.dll"`
2. Компілюємо solution з рядком import (після компіляції цей рядок прибираємо).
3. Далі в папці debug\ вашого solution з'являються файли CashaLotApi64.tlh та CashaLotApi64.tli
4. Переносимо їх в каталог solution, перейменовуємо в CashaLotApi.tlh, CashaLotApi.tli (за бажанням, можна зібрати два x32 та x64, та підключати директивами)
5. Всередині CashaLotApi.tlh прибираємо повний шлях до CashaLotApi64.tli і також міняємо ім'я на CashaLotApi.tli
6. Заголовочні файли готові до використання, підключаємо їх в solution:

```
#include "CashaLotApi.tlh"
using namespace CashaLotApi;
```

2. Зауваження

1. При старті програми додаємо ініціалізацію роботи з COM об'єктами: CoInitialize(NULL);
2. При закритті програми припиняємо роботу з COM об'єктами: CoUninitialize();
3. Перед початком роботи з COM-об'єктом ініціалізуємо його викликом InitCashaLotApiComInterface(), перед виходом з програми звільняємо ресурси викликом функції TermCashaLotApiComInterface();
4. Функції повертають структуру ICashalotApiRetVal або BOOL.

3. Лістинг

```
#include "stdafx.h"
#include "SampleUseCashaLotCOMAPI_CPP.h"
#include "SampleUseCashaLotCOMAPI_CPPDlg.h"
#include "afxdialogex.h"
#include <string>          // std::string
#include <sstream>         // std::ostringstream
#include <algorithm>
#include <list>

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

#include "CashaLotApi.tlh"
using namespace CashaLotApi;

extern LPCTSTR pAppName;

CSampleUseCashaLotCOMAPICPPDlg::CSampleUseCashaLotCOMAPICPPDlg(CWnd* pParent /*=nullptr*/)
: CDialogEx(IDD_SAMPLEUSECASHALOTCOMAPI_CPP_DIALOG, pParent)
, m_FiscalNumberRRO(_T(""))
, m_PathToCashaLot(_T(""))
, m_IsNoInterfaceMode(FALSE)
, m_PathCert(_T(""))
, m_pwdCert(_T(""))
, m_PrinterName(_T(""))
, m_useToken(FALSE)
, m_UseDefCashaLotPrinter(FALSE)
, m_PrintUseDefWindowsPrinter(FALSE)
, m_PrintUseNamedPrinter(FALSE)
, m_DateBeg(COleDateTime::GetCurrentTime())
, m_DateEnd(COleDateTime::GetCurrentTime())
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CSampleUseCashaLotCOMAPICPPDlg::DoDataExchange(CDataExchange* pDX)
{

```

```

CDialogEx::DoDataExchange(pDX);
DDX_Control(pDX, IDC_EDPATH, m_edPathToCashaLot);
DDX_Control(pDX, IDC_EDFISKNUMRCP, m_edFiscNumRcp);
DDX_Text(pDX, IDC_EDPATH2, m_FiscalNumberRRO);
DDX_Text(pDX, IDC_EDPATH, m_PathToCashaLot);
DDX_Control(pDX, IDC_EDPATH2, m_edFiscalNumberRRO);
DDX_Control(pDX, IDC_BTGETPATH, m_btnPath);
DDX_Check(pDX, IDC_CHECKNOINTERFACEMODE, m_IsNoInterfaceMode);
DDX_Text(pDX, IDC_EDPATHTOCERT, m_PathCert);
DDX_Text(pDX, IDC_EDPWD, m_pwdCert);
DDX_Control(pDX, IDC_RADIO1, m_rdbtPrintUseNamedPrinter);
DDX_Control(pDX, IDC_RADIO2, m_rdbtPrintUseDefPrinter);

DDX_Text(pDX, IDC_EDPATH5, m_PrinterName);
DDX_Control(pDX, IDC_EDPATHTOCERT, m_edPathCert);
DDX_Control(pDX, IDC_EDPWD, m_edPwdCert);
DDX_Control(pDX, IDC_CHECKNOINTERFACEMODE, m_chNoInterfaceMode);
DDX_Check(pDX, IDC_CHECKUSETOKEN, m_useToken);
DDX_Control(pDX, IDC_CHECKUSETOKEN, m_chUseToken);
DDX_Check(pDX, IDC_CHECKUSECASHALOTPRINTERSETTINGS, m_UseDefCashalotPrinter);
DDX_Control(pDX, IDC_CHECKUSECASHALOTPRINTERSETTINGS, m_chUseDefCashalotPrinter);
DDX_Control(pDX, IDC_EDPATH5, m_edPrinterName);
DDX_Radio(pDX, IDC_RADIO2, m_PrintUseDefWindowsPrinter);
DDX_Radio(pDX, IDC_RADIO1, m_PrintUseNamedPrinter);
DDX_DateTimeCtrl(pDX, IDC_DATETIMEPICKER1, m_DateBeg);
DDX_DateTimeCtrl(pDX, IDC_DATETIMEPICKER2, m_DateEnd);
}

BEGIN_MESSAGE_MAP(CSampleUseCashaLotCOMAPICPPDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BTOPENSHIFT, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtopenshift)
    ON_BN_CLICKED(IDC_BTcloseShift, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtcloseshift)
    ON_BN_CLICKED(IDC_BTfiscalizeReceipt,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtfiscalizereceipt)
    ON_BN_CLICKED(IDC_BTReturnReceipt, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtreturnreceipt)
    ON_BN_CLICKED(IDC_BTServiceInput, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtserviceinput)
    ON_BN_CLICKED(IDC_BTServiceOutput, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtserviceoutput)
    ON_BN_CLICKED(IDC_BTServiceOutput2, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedPrintXRep)
    ON_WM_DESTROY()
    ON_BN_CLICKED(IDC_BTGETPATH, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetpath)
    ON_BN_CLICKED(IDC_BTGETPATHTOCERT, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetpathtocert)
    ON_BN_CLICKED(IDC_BTPrintXRepEx, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtprintxrepex)
    ON_BN_CLICKED(IDC_CHECKUSECASHALOTPRINTERSETTINGS,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedCheckusecshalotprintersettings)
    ON_BN_CLICKED(IDC_RADIO2, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedRadio2)
    ON_BN_CLICKED(IDC_RADIO1, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedRadio1)
    ON_BN_CLICKED(IDC_BTGETPRINTERNAME, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetprintername)
    ON_BN_CLICKED(IDC_BTPrintLastZRep, &CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtprintlastzrep)
    ON_BN_CLICKED(IDC_BTGETRECEIPTANDREPPFROMPERIOD,
&CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedGetRcpByPeriod)
END_MESSAGE_MAP()

// CSampleUseCashaLotCOMAPICPPDlg message handlers

BOOL CSampleUseCashaLotCOMAPICPPDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE); // Set big icon
    SetIcon(m_hIcon, FALSE); // Set small icon

    // TODO: Add extra initialization here
    CWinApp* pApp = AfxGetApp();
    m_PathToCashaLot = pApp->GetProfileString(pAppName, _T("PathToCashaLot"));
    m_FiscalNumberRRO = pApp->GetProfileString(pAppName, _T("FiscalNumberRRO"));
    m_PathCert = pApp->GetProfileString(pAppName, _T("PathToCert"));
    m_pwdCert = pApp->GetProfileString(pAppName, _T("PwdToCert"));
    m_IsNoInterfaceMode = pApp->GetProfileString(pAppName, _T("NoInterfaceMode")) == L"1";
}

```



```

        m_useToken = pApp->GetProfileString(pAppName, _T("UseToken")) == L"1";

        m_PrintUseDefWindowsPrinter = (pApp->GetProfileString(pAppName, _T("PrintUseDefWindowsPrinter"))
== L"0")? 0 : -1;
        m_PrintUseNamedPrinter = (pApp->GetProfileString(pAppName, _T("PrintUseNamedPrinter")) == L"0")?
0 : -1;;
        m_PrinterName = pApp->GetProfileString(pAppName, _T("PrinterName"));

        m_edPrinterName.SetWindowText(m_PrinterName);

        CString defPrndPrm = pApp->GetProfileString(pAppName, _T("UseDefCashalotPrinter"));
        if (defPrndPrm == L"") defPrndPrm = L"1";
        m_UseDefCashalotPrinter = defPrndPrm == L"1";

        m_chUseDefCashalotPrinter.SetCheck(m_UseDefCashalotPrinter);

        m_chNoInterfaceMode.SetCheck(m_IsNoInterfaceMode);
        m_chUseToken.SetCheck(m_useToken);

        m_edFiscalNumberRRO.SetWindowText(m_FiscalNumberRRO);
        m_edPathToCashaLot.SetWindowText(m_PathToCashaLot);
        m_edPathCert.SetWindowText(m_PathCert);
        m_edPwdCert.SetWindowText(m_pwdCert);
        UpdateData(0);

        OnBnClickedCheckusecashalotprintersettings();
        OnBnClickedRadio2();

        return TRUE; // return TRUE unless you set the focus to a control
    }
    void CSampleUseCashaLotCOMAPICPPDlg::OnDestroy()
    {
        UpdateData();
        CDialogEx::OnDestroy();

        CWinApp* pApp = AfxGetApp();
        pApp->WriteProfileString(pAppName, _T("PathToCashaLot"), m_PathToCashaLot);
        pApp->WriteProfileString(pAppName, _T("FiscalNumberRRO"), m_FiscalNumberRRO);
        pApp->WriteProfileString(pAppName, _T("PathToCert"), m_PathCert);
        pApp->WriteProfileString(pAppName, _T("PwdToCert"), m_pwdCert);
        pApp->WriteProfileString(pAppName, _T("NoInterfaceMode"), ((m_IsNoInterfaceMode == TRUE)? L"1" :
L"0"));
        pApp->WriteProfileString(pAppName, _T("UseToken"), ((m_useToken == TRUE)? L"1" : L"0"));
        pApp->WriteProfileString(pAppName, _T("UseDefCashalotPrinter"), ((m_UseDefCashalotPrinter ==
TRUE)? L"1" : L"0"));

        pApp->WriteProfileString(pAppName, _T("PrintUseDefWindowsPrinter"), (m_PrintUseDefWindowsPrinter
== 0 ? L"0" : L"-1"));
        pApp->WriteProfileString(pAppName, _T("PrintUseNamedPrinter"), (m_PrintUseNamedPrinter == 0 ?
L"0" : L"-1"));
        pApp->WriteProfileString(pAppName, _T("PrinterName"), m_PrinterName );
    }

    void CSampleUseCashaLotCOMAPICPPDlg::OnSysCommand(UINT nID, LPARAM lParam)
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }

    // If you add a minimize button to your dialog, you will need the code below
    // to draw the icon. For MFC applications using the document/view model,
    // this is automatically done for you by the framework.

    void CSampleUseCashaLotCOMAPICPPDlg::OnPaint()
    {
        if (IsIconic())
        {
            CPaintDC dc(this); // device context for painting

            SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

            // Center icon in client rectangle

```

```

        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
}

// The system calls this function to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CSampleUseCashaLotCOMAPICPPDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

static int CALLBACK BrowseCallbackProc(HWND hwnd, UINT uMsg, LPARAM lParam, LPARAM lpData)
{
    if (uMsg == BFFM_INITIALIZED)
    {
        std::string tmp = (const char *)lpData;

        SendMessage(hwnd, BFFM_SETSELECTION, TRUE, lpData);
    }

    return 0;
}

CString BrowseFolder()
{
    TCHAR path[MAX_PATH] = { 0 };

    const char * path_param = "";

    BROWSEINFO bi = { 0 };
    bi.lpszTitle = _T("Вибір каталогу з програмою CashaLot");
    bi.ulFlags = BIF_RETURNONLYFSDIRS | BIF_NEWDIALOGSTYLE;
    bi.lpfncb = BrowseCallbackProc;
    bi.lParam = (LPARAM)path_param;

    LPITEMIDLIST pidl = SHBrowseForFolder(&bi);

    if (pidl != 0)
    {
        //get the name of the folder and put it in path
        SHGetPathFromIDList(pidl, path);

        //free memory used
        IMalloc * imalloc = 0;
        if (SUCCEEDED(SHGetMalloc(&imalloc)))
        {
            imalloc->Free(pidl);
            imalloc->Release();
        }

        return CString(path);
    }

    return CString("");
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedCheckusecashalotprintersettings()
{
    UpdateData();
}

```

```

        m_rdbtPrintUseDefPrinter.EnableWindow(!m_UseDefCashalotPrinter);
        m_rdbtPrintUseNamedPrinter.EnableWindow(!m_UseDefCashalotPrinter);
        m_edPrinterName.EnableWindow(!m_UseDefCashalotPrinter);
    }

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedRadio2()
{
    UpdateData();
    m_rdbtPrintUseDefPrinter.SetCheck(m_PrintUseDefWindowsPrinter == 0);
    m_rdbtPrintUseNamedPrinter.SetCheck(m_PrintUseDefWindowsPrinter == -1);
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedRadio1()
{
    UpdateData();
    m_rdbtPrintUseDefPrinter.SetCheck(m_PrintUseDefWindowsPrinter == -1);
    m_rdbtPrintUseNamedPrinter.SetCheck(m_PrintUseDefWindowsPrinter == 0);
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetprintername()
{
    CPrintDialog dlg(true, PD_NOPAGENUMS | PD_HIDEPRINTTOFILE | PD_NOCURRENTPAGE |
PD_NONETWORKBUTTON);
    if (dlg.DoModal() == IDOK)
    {
        m_PrinterName = dlg.GetDeviceName();
        UpdateData(0);
    }
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetpath()
{
    CString ret = BrowseFolder();
    if (ret.GetLength())
    {
        m_PathToCashaLot = ret;
        m_edPathToCashaLot.SetWindowText(m_PathToCashaLot);
    }
}

//////////ПРИКЛАД ВИКОРИСТАННЯ//////////
struct CashaLotApiRsp
{
    void Clear()
    {
        Ret = false;
        ErrorString = "";
        Values.clear();
    }

    CashaLotApiRsp()
    {
        Clear();
    }

    bool Ret;
    std::string ErrorString;
    std::list< std::pair<std::string, std::string> > Values;
};

CashaLotApiRsp rspEx;
CashaLotApiRsp DeserializeResp(std::string response)
{
    rspEx.Clear();
    return rspEx;
}

//Структура інформації по позиції товару в чеці(артикул, назва сумма, кількість, податки і тд.)
struct ReceiptItemInfo

```

```

{
    ReceiptItemInfo()
    {
        VendorCode = "";
        Name = "";
        CdUKDZED = "";
        CdDKPP = "";
        Barcode = "";
        GoodsType = 1; //1- товар, 2- услуга
        UnitType = "шт";
        Quantity = 0;
        Price = 0;
        Amount = 0;
        DiscountPrc = 0;
        DiscountSum = 0;
        VATRate = 0;
        IsPriceIncludeVAT = true;
        SumVAT = 0;
        IsExcise = false;
        OtherParameters = "";
    }

    std::string VendorCode;
    std::string Name;
    std::string CdUKDZED;
    std::string CdDKPP;
    std::string Barcode;
    int GoodsType; //1- товар, 2- услуга
    std::string UnitType;
    double Quantity;
    double Price;
    double Amount;
    double DiscountPrc;
    double DiscountSum;
    double VATRate;
    bool IsPriceIncludeVAT;
    double SumVAT;
    bool IsExcise;
    std::string OtherParameters;

    //Повертає JSON рядок з інформацією по позиції товару
    std::string ToJSON()
    {
        std::ostringstream ret;

        ret << "{" <<
            "\"VendorCode\": \"" << (VendorCode.c_str()) <<
            "\", \"Name\": \"" << (Name.c_str()) <<
            "\", \"CdUKDZED\": \"" << (CdUKDZED.c_str()) <<
            "\", \"CdDKPP\": \"" << (CdDKPP.c_str()) <<
            "\", \"Barcode\": \"" << (Barcode.c_str()) <<
            "\", \"GoodsType\": \"" << GoodsType <<
            "\", \"UnitType\": \"" << (UnitType.c_str()) <<
            "\", \"Quantity\": \"" << Quantity <<
            "\", \"Price\": \"" << Price <<
            "\", \"Amount\": \"" << Amount <<
            "\", \"DiscountPrc\": \"" << DiscountPrc <<
            "\", \"DiscountSum\": \"" << DiscountSum <<
            "\", \"VATRate\": \"" << VATRate <<
            "\", \"IsPriceIncludeVAT\": \"" << ((!IsPriceIncludeVAT) ? "false" : "true") <<
            "\", \"SumVAT\": \"" << SumVAT <<
            "\", \"IsExcise\": \"" << ((!IsExcise) ? "false": "true") <<
            "\", \"OtherParameters\": \"" << (OtherParameters.c_str()) << "\""
            << "}"
            ;

        return ret.str();
    }
}
};

struct ReceiptInfoLst

```

```

{
private:
    std::list<ReceiptItemInfo*> lstReceiptItemsInfo;
    std::string Comment;
public:
    ReceiptInfoLst()
    {
        Clear();
    };
    ~ReceiptInfoLst()
    {
        Clear();
    };

    std::string ToJSON()
    {
        std::ostringstream ret;
        std::ostringstream comment;
        long cnt=0;

        if(Comment.length() > 0)
            comment << "\",\"Comment\\\":\\\"\" << Comment.c_str() <<\"\"";

        ret << \"{\\\"ReceiptLst\\\":[\";

        for(std::list<ReceiptItemInfo*>::iterator x= lstReceiptItemsInfo.begin(); x!=
lstReceiptItemsInfo.end(); ++x)
        {
            if(cnt++!=0)
                ret << \", \";
            ret << ((*x)->ToJSON());
        }
        ret << \"]\" << (comment.str().c_str()) <<\"}\";

        return ret.str();
    }

    _bstr_t ToBSTRJSON()
    {
        return _bstr_t(ToJSON().c_str());
    }
    void Clear()
    {
        Comment.clear();
        for (std::list<ReceiptItemInfo*>::iterator x = lstReceiptItemsInfo.begin(); x !=
lstReceiptItemsInfo.end(); ++x)
            delete (*x);
        lstReceiptItemsInfo.clear();
    }

    void InsertItem( ReceiptItemInfo* check )
    {
        lstReceiptItemsInfo.push_back(check);
    }

    void SetComment(std::string strComment)
    {
        Comment = strComment;
    }
};

//Структура оплати по чеку
struct PayInfoRsp
{
    //Сума готівки наданої покупцем(може бути більшою за сумму до сплати, з неї вираховується
    решта/здача)
    double SumCash;
    //Сума оплати картою
    double SumPayByCard;
    //Сума оплати за рахунок кредитних коштів
    double SumPayByCredit;
    //Сума оплати сертифікатом

```

```

double SumPayByCertificate;
// Сума фактичної оплати всіма видами оплати, з урахуванням заокруглення
double SumPayCheck;
// Email платника
std::string CustomerEmail;

PayInfoRsp()
{
    //Сума готівки наданої покупцем(може бути більшою за сумму до сплати, з неї вираховується
    решта/здача)
    SumCash =
    SumPayByCard =
    SumPayByCredit =
    SumPayByCertificate =
    SumPayCheck = 0;
    CustomerEmail = "";
}

//Отримати JSON рядок з інформацією про оплату по чеку(готівка, карта, знижки та заокруглення)
std::string ToJSON()
{
    std::ostringstream ret;
    long cnt = 0;
    ret << "{\"SumCash\":\"" << SumCash <<
        "\",\"SumPayByCard\":\"" << SumPayByCard <<
        "\",\"SumPayByCredit\":\"" << SumPayByCredit <<
        "\",\"SumPayByCertificate\":\"" << SumPayByCertificate <<
        "\",\"SumPayCheck\":\"" << SumPayCheck <<
        "\",\"CustomerEmail\":\"" << CustomerEmail << "\"}";

    return ret.str();
}

_bstr_t ToBSTRJSON()
{
    return _bstr_t(ToJSON().c_str());
}
};

std::ostringstream strRet;
ReceiptInfoLst gsLstCheckItems;
ICashaLotApiRetValPtr retVal = NULL;
ICashaLotApiAddinPtr gsCashaLotApi = NULL;
BSTR fiscalNumRRO = NULL;
BSTR pathToCashaLot = NULL;
BSTR pathToCert = NULL;
BSTR pwdToCert = NULL;
bool IsNoInterfaceMode = false;
bool useToken = false;
int setupPrinterMode = 0;
BSTR printerName = NULL;

void SetPrinterParametr()
{
    if (gsCashaLotApi != NULL)
    {
        gsCashaLotApi->SetParameter( L"DEFAULTPRINTERNAME", (setupPrinterMode == 0)? BSTR(L"") :
        ((setupPrinterMode == 1)? BSTR(L"DEFAULTWINDOWS") : printerName) );
    }
}

//Ініціалізація COM об'єкта CashaLot
bool InitCashaLotApiComInterface()
{
    if ( gsCashaLotApi == NULL )
    {
        // Вказуємо ІД COM об'єкта AddIn.CashaLotApi, який має бути попередньо зареєстрований в
        реєстрі(regsvr32 D:\prro\CashLot\CashalotApi64.dll)
        char* szProgID = "AddIn.CashaLotApi";
        WCHAR szWideProgID[128];
        CLSID clsid;
        long lLen = MultiByteToWideChar(CP_ACP,

```

```

        0,
        szProgID,
        strlen(szProgID),
        szWideProgID,
        sizeof(szWideProgID));

szWideProgID[lLen] = '\\0';
HRESULT hr = ::CLSIDFromProgID(szWideProgID, &clsid);
if (FAILED(hr))
{
    MessageBox(0, L"Unable to get CLSID from ProgID. AddIn.CashaLotApi", 0, 0);
    return false;
}

IClassFactory* pCF;
// Отримати фабрику класів
hr = CoGetClassObject(clsid,
    CLSCTX_INPROC,
    NULL,
    IID_IClassFactory,
    (void**)&pCF);
if (FAILED(hr))
{
    MessageBox(0, L"Failed to GetClassObject server instance.", 0, 0);
    return false;
}

// за допомогою фабрики класів створити екземпляр
// компонента та отримати інтерфейс IUnknown.
IUnknown* pUnk;
hr = pCF->CreateInstance(NULL, IID_IUnknown, (void**)&pUnk);

// Release the class factory
pCF->Release();

if (FAILED(hr))
{
    MessageBox(0, L"Failed to create server instance.", 0, 0);
    return false;
}

hr = pUnk->QueryInterface(__uuidof(ICashaLotApiAddin), (LPVOID*)&gsCashaLotApi);
pUnk->Release();

if (FAILED(hr))
{
    MessageBox(0, L"Failed to create ICashaLotApiAddin", 0, 0);
    return false;
}

//-----Задаємо обов'язкові параметри(шлях до каталогу з програмою Кашалот, та фіскальний
номер РРО з якою будемо працювати) -----

gsCashaLotApi->SetParameter(L"PathToCshalotDir", pathToCashaLot);
gsCashaLotApi->SetParameter(L"DeviceIDFnRRO", fiscalNumRRO);
gsCashaLotApi->SetParameter(L"NOINTERFACEMODE", IsNoInterfaceMode ? L"true" : L"false");
gsCashaLotApi->SetParameter(L"NOAUTOOPENSHIFT", L"true");
//Ознака використанні захищеного носія(має бути після PathToCshalotDir та DeviceIDFnRRO та
NOINTERFACEMODE, якщо необхідно пришивдити завантаження, та немає необхідності використовувати
токени. Інакше токени будуть ініціалізуватися)
gsCashaLotApi->SetParameter(L"USETOKEN", useToken ? L"true" : L"false");

SetPrinterParametr();

gsCashaLotApi->SetParameter(L"PathToCertificate", pathToCert);
gsCashaLotApi->SetParameter(L"PwdToCertificate", pwdToCert);

}

return true;
}

```

```

bool CSampleUseCashaLotCOMAPICPPDlg::InitCashaLotApiComInterfaceEx()
{
    UpdateData();

    if (useToken != m_useToken && gsCashaLotApi != NULL)
        gsCashaLotApi->SetParameter(L"USETOKEN", m_useToken ? L"true" : L"false");

    useToken = m_useToken;

    if ( pathToCashaLot == NULL )
    {
        pathToCashaLot = ::SysAllocString(m_PathToCashaLot);
    }

    if (fiscalNumRRO == NULL )
    {
        fiscalNumRRO = ::SysAllocString(m_FiscalNumberRRO);
    }

    if (pathToCert != NULL && pathToCert != m_PathCert)
    {
        ::SysFreeString(pathToCert);
        pathToCert = ::SysAllocString(m_PathCert);
        if (gsCashaLotApi != NULL)
            gsCashaLotApi->SetParameter(L"PathToCertificate", pathToCert);
    }
    else
    if (pathToCert == NULL )
    {
        pathToCert = ::SysAllocString(m_PathCert);
    }

    if (pwdToCert != NULL && pwdToCert != m_pwdCert)
    {
        ::SysFreeString(pwdToCert);
        pwdToCert = ::SysAllocString(m_pwdCert);
        if (gsCashaLotApi != NULL)
            gsCashaLotApi->SetParameter(L"PwdToCertificate", pwdToCert);
    }
    else
    if (pwdToCert == NULL)
    {
        pwdToCert = ::SysAllocString(m_pwdCert);
    }

    int oldSetupPrinterMode = setupPrinterMode;
    setupPrinterMode = (m_UseDefCashalotPrinter)? 0 : m_PrintUseDefWindowsPrinter!=1? 1 : 2 ;

    if (printerName != NULL && (printerName != m_PrinterName || oldSetupPrinterMode !=
setupPrinterMode) )
    {
        ::SysFreeString(printerName);
        printerName = ::SysAllocString(m_PrinterName);
        SetPrinterParametr();
    }
    else
    if(printerName == NULL)
        printerName = ::SysAllocString(m_PrinterName);

    if(IsNoInterfaceMode != m_IsNoInterfaceMode && gsCashaLotApi != NULL )
        gsCashaLotApi->SetParameter(L"NOINTERFACEMODE", m_IsNoInterfaceMode ? L"true" : L"false");

    IsNoInterfaceMode = m_IsNoInterfaceMode;

    return InitCashaLotApiComInterface();
}

//Очистка всіх ресурсів по роботі з COM об'єктом CashaLot
bool TermCashaLotApiComInterface()
{
    try

```

```

{
    if (gsCashaLotApi != NULL)
    {
        gsCashaLotApi->Release();
        gsCashaLotApi = NULL;

        if (pathToCashaLot != NULL)
        {
            ::SysFreeString(pathToCashaLot);
            pathToCashaLot = NULL;
        }
        if (fiscalNumRRO != NULL)
        {
            ::SysFreeString(fiscalNumRRO);
            fiscalNumRRO = NULL;
        }
        if (pathToCert != NULL)
        {
            ::SysFreeString(pathToCert);
            pathToCert = NULL;
        }
        if (pwdToCert != NULL)
        {
            ::SysFreeString(pwdToCert);
            pwdToCert = NULL;
        }
        return true;
    }
}
catch (const char* msg)
{
    MessageBox(0, BSTR(msg), L"Exception", MB_OK);
}
catch (...)
{
    MessageBox(0, L"", L"Exception", MB_OK);
}

return false;
}

//Приклад відкриття зміни
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtopenshift()
{
    if(!InitCashaLotApiComInterfaceEx())
        return;

    gsCashaLotApi->SetParameter(L"NOINTERFACEMODE", IsNoInterfaceMode ? L"true" : L"false");

    if (pathToCert != NULL)
    {
        ::SysFreeString(pathToCert);
        pathToCert = NULL;
    }
    if (pwdToCert != NULL)
    {
        ::SysFreeString(pwdToCert);
        pwdToCert = NULL;
    }

    pathToCert = ::SysAllocString(m_PathCert);
    pwdToCert = ::SysAllocString(m_pwdCert);

    gsCashaLotApi->SetParameter(L"PathToCertificate", pathToCert);
    gsCashaLotApi->SetParameter(L"PwdToCertificate", pwdToCert);

    _variant_t ret = gsCashaLotApi->OpenShift(fiscalNumRRO);

    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)

```

```

        {
            m_edPathToCashaLot.EnableWindow(false);
            m_btnPath.EnableWindow(false);
            m_edFiscalNumberRRO.EnableWindow(false);

            strRet.str("");
            strRet << "Номер відкритої зміни: " << retVal->ShiftID;
            MessageBox(CString(strRet.str().c_str()));
        }
    }
}

//Приклад закриття зміни
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtccloseshift()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    _variant_t ret = gsCashaLotApi->CloseShift(fiscalNumRRO);

    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)
        {
            strRet.str("");
            strRet << "Номер закритої зміни: " << retVal->ShiftID<<"\n"
                << "Результат у вигляді JSON\n" << retVal->JsonVal;
            MessageBox(CString(strRet.str().c_str()));
        }
    }
}

//Приклад фіскалізації чека на дві позиції
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtfiscalizereceipt()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    gsLstCheckItems.Clear();

    ReceiptItemInfo* inf = new ReceiptItemInfo();
    inf->VendorCode = "АртикулТовару1";
    inf->Name = "НазваТовару1";
    inf->Barcode = "123456789";
    inf->UnitType = "шт";
    inf->Quantity = 1;
    inf->GoodsType = 1; /*товар*/
    inf->Price = 183;
    inf->Amount = 181.17;
    inf->DiscountSum = 1.83;
    inf->VATRate = 20;
    inf->SumVAT = 30.2;
    inf->IsExcise = false;
    gsLstCheckItems.InsertItem(inf);

    inf = new ReceiptItemInfo();
    inf->VendorCode = "АртикулПослуги1";
    inf->Name = "НазваПослуги1";
    inf->Barcode = "923456780";
    inf->UnitType = "шт";
    inf->Quantity = 1;
    inf->GoodsType = 2; /*послуга*/
    inf->Price = 183;
    inf->Amount = 181.17;
    inf->DiscountSum = 1.83;
    inf->VATRate = 20;
    inf->SumVAT = 30.2;
    inf->IsExcise = false;
    gsLstCheckItems.InsertItem(inf);
    gsLstCheckItems.SetComment("Тестовий рядок1\r\nТестовий рядок2\r\nТестовий рядок3");
}

```

```

_bstr_t jsonChkData = gsLstCheckItems.ToBSTRJSON();

PayInfoRsp payInfo;
payInfo.SumCash = 366;
payInfo.SumPayCheck = 362.34;
payInfo.SumPayByCredit = 100;
payInfo.SumPayByCertificate = 50;
_bstr_t jsonPayData = payInfo.ToBSTRJSON();

_variant_t ret = gsCashaLotApi->FiscalizeCheck(fiscalNumRRO, jsonChkData, jsonPayData);

if (V_VT(&ret) == VT_DISPATCH)
{
    retVal = (ICashalotApiRetVal*)ret.pdispVal;
    if (retVal != NULL && retVal->Return != FALSE)
    {
        m_edFiscNumRcp.SetWindowTextW(retVal->ReceiptFiscalNum);

        strRet.str("");
        strRet << "Фіскальний номер створеного чека: " << retVal->ReceiptFiscalNum;
        strRet << "\nXML створеного чека: \n" << retVal->XML;
        MessageBox(CString(strRet.str()).c_str());
    }
}

gsLstCheckItems.Clear();
}

//Приклад фіскалізації чека повернення на дві позиції, з вказанням чека по якому здійснюється повернення
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtreturnreceipt()
{
    CString strfiscNumRcpToRet;
    m_edFiscNumRcp.GetWindowText(strfiscNumRcpToRet);

    if (strfiscNumRcpToRet.GetLength() == 0)
    {
        MessageBox(L"Не заданий фіскальний номер чека по якому здійснюється повернення!");
        return;
    }

    if (!InitCashaLotApiComInterfaceEx())
        return;

    gsLstCheckItems.Clear();
    ReceiptItemInfo* inf = new ReceiptItemInfo();
    inf->VendorCode = "АртикулТовару1";
    inf->Name = "НазваТовару1";
    inf->Barcode = "123456789";
    inf->UnitType = "шт";
    inf->Quantity = 1;
    inf->GoodsType = 1; /*товар*/
    inf->Price = 183;
    inf->Amount = 181.17;
    inf->DiscountSum = 1.83;
    inf->VATRate = 20;
    inf->SumVAT = 30.2;
    inf->IsExcise = false;
    gsLstCheckItems.InsertItem(inf);

    inf = new ReceiptItemInfo();
    inf->VendorCode = "АртикулПослуги1";
    inf->Name = "НазваПослуги1";
    inf->Barcode = "923456780";
    inf->UnitType = "шт";
    inf->Quantity = 1;
    inf->GoodsType = 2; /*послуга*/
    inf->Price = 183;
    inf->Amount = 181.17;
    inf->DiscountSum = 1.83;
    inf->VATRate = 20;
    inf->SumVAT = 30.2;

```

```

inf->IsExcise = false;
gsLstCheckItems.InsertItem(inf);
_bstr_t jsonChkData = gsLstCheckItems.ToBSTRJSON();

PayInfoRsp payInfo;
payInfo.SumCash = 366;
payInfo.SumPayCheck = 362.34;
_bstr_t jsonPayData = payInfo.ToBSTRJSON();

_bstr_t retFiscNum = ::SysAllocString(strfiscNumRcpToRet);

_variant_t ret = gsCashaLotApi->FiscalizeReturnCheck(fiscalNumRRO, jsonChkData, jsonPayData,
retFiscNum );

if ( V_VT(&ret) == VT_DISPATCH )
{
    retVal = (ICashalotApiRetVal*)ret.pdispVal;
    if (retVal != NULL && retVal->Return != FALSE)
    {
        strRet.str("");
        strRet << "Фіскальний номер створеного видаткового чека: " << retVal->ReceiptFiscalNum;
        //strRet << "\nXML створеного видаткового чека: \n" << retVal->XML;
        strRet << "\nТип чека: \n" << retVal->Type;
        MessageBox(CString(strRet.str().c_str()));
    }
}

::SysFreeString(retFiscNum);
}

//Приклад чека службового внесення
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtserviceinput()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;
    BSTR strSm = ::SysAllocString(L"100");

    VARIANT vSm;
    ::VariantInit(&vSm);
    V_VT(&vSm) = VT_BSTR;
    V_BSTR(&vSm) = strSm;

    _variant_t ret = gsCashaLotApi->ServiceInput(fiscalNumRRO, vSm);

    if (V_VT(&ret) == VT_BOOL && V_BOOL(&ret) != FALSE)
        MessageBox(L"Службове внесення виконано успішно");
    else
        MessageBox(L"Не вдалося виконати службове внесення");

    ::SysFreeString(strSm);
}

//Приклад чека службової видачі
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtserviceoutput()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;
    BSTR strSm = ::SysAllocString(L"100");

    VARIANT vSm;
    ::VariantInit(&vSm);
    V_VT(&vSm) = VT_BSTR;
    V_BSTR(&vSm) = strSm;

    _variant_t ret = gsCashaLotApi->ServiceOutput(fiscalNumRRO, vSm);

    if (V_VT(&ret) == VT_BOOL && V_BOOL(&ret) != FALSE)
        MessageBox(L"Службова видача виконана успішно");
    else
        MessageBox(L"Не вдалося виконати службову видачу");

    ::SysFreeString(strSm);
}

```

```

//Приклад чека друку X-Звіту(без гасіння)
void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedPrintXRep()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    _variant_t ret = gsCashaLotApi->PrintXReport(fiscalNumRRO);

    if (V_VT(&ret) == VT_BOOL && V_BOOL(&ret) != FALSE)
        MessageBox(L"X-звіт сформований успішно");
    else
        MessageBox(L"Не вдалося сформувати X-звіт");
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtgetpathtocert()
{
    CString ret = BrowseFolder();
    if (ret.GetLength())
    {
        m_PathCert = ret;
        m_edPathCert.SetWindowText(m_PathCert);
    }
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtprintxrepex()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    _variant_t ret = gsCashaLotApi->GetXReport(fiscalNumRRO, VARIANT_FALSE);

    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)
        {
            strRet.str("");
            strRet << "Результат у вигляді JSON:\n" << retVal->JsonVal;
            MessageBox(CString(strRet.str().c_str()));
        }
    }
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedBtprintlastzrep()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    _variant_t ret = gsCashaLotApi->GetLastZReport(fiscalNumRRO, VARIANT_TRUE);

    if (V_VT(&ret) == VT_DISPATCH)
    {
        retVal = (ICashalotApiRetVal*)ret.pdispVal;
        if (retVal != NULL && retVal->Return != FALSE)
        {
            strRet.str("");
            strRet << "Результат у вигляді JSON:\n" << retVal->JsonVal;
            MessageBox(CString(strRet.str().c_str()));
        }
    }
}

void CSampleUseCashaLotCOMAPICPPDlg::OnBnClickedGetRcpByPeriod()
{
    if (!InitCashaLotApiComInterfaceEx())
        return;

    strRet.str("");
    BSTR dtBeg = ::SysAllocString(m_DateBeg.Format(CString("%d.%m.%Y")));

```

```

BSTR dtEnd = ::SysAllocString(m_DateEnd.Format(CString("%d.%m.%Y")));
//зачитуємо перший чек за період
_variant_t ret = gsCashaLotApi->GetFirstReceiptByPeriod(fiscalNumRRO, dtBeg, dtEnd);

if (V_VT(&ret) == VT_DISPATCH)
{
    retVal = (ICashalotApiRetVal*)ret.pdispVal;
    if (retVal != NULL && retVal->Return != FALSE)
    {
        strRet << "Фіскальні номери чеків за період:\n";
        strRet << retVal->ReceiptFiscalNum;

        bool retN = false;
        do
        {
            retN = false;
            //зачитуємо наступний чек за період
            ret = gsCashaLotApi->GetNextReceipt(fiscalNumRRO);
            if (V_VT(&ret) == VT_DISPATCH)
            {
                retVal = (ICashalotApiRetVal*)ret.pdispVal;
                if (retVal != NULL && retVal->Return != FALSE)
                {
                    retN = true;
                    strRet << ", ";
                    strRet << retVal->ReceiptFiscalNum;
                }
            }
            //читаємо поки будуть наступні
        } while (retN);
    }
}

ret = gsCashaLotApi->GetFirstZReportByPeriod(fiscalNumRRO, dtBeg, dtEnd);

if (V_VT(&ret) == VT_DISPATCH)
{
    retVal = (ICashalotApiRetVal*)ret.pdispVal;
    if (retVal != NULL && retVal->Return != FALSE)
    {
        strRet << "\nФіскальні номери Z-звітів за період:\n";
        strRet << retVal->ReceiptFiscalNum;

        bool retN = false;
        do
        {
            retN = false;
            //зачитуємо наступний чек за період
            ret = gsCashaLotApi->GetNextZReport(fiscalNumRRO);
            if (V_VT(&ret) == VT_DISPATCH)
            {
                retVal = (ICashalotApiRetVal*)ret.pdispVal;
                if (retVal != NULL && retVal->Return != FALSE)
                {
                    retN = true;
                    strRet << ", ";
                    strRet << retVal->ReceiptFiscalNum;
                }
            }
            //читаємо поки будуть наступні
        } while (retN);
    }
}

MessageBox(CString(strRet.str().c_str()));

::SysFreeString(dtBeg);
::SysFreeString(dtEnd);
}

```

Варіант 26 – API ПРРО Cashalot з .NET інтерфейсом CashalotApiBridge.dll

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

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

В якості прикладу в документі наведено **C#** реалізацію у вигляді методу **CallCashalotMethod** (**“Метод”, Параметру**).

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.

1. Специфікація методів API бібліотеки ПРРО Cashalot при підключенні до облікової системи

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

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

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

- Ознака успішності виконання функції (**bool**);
- Текст помилки (**string**);
- Масив параметрів (**Dictionary<string, string>**) (опціонально).

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 API
    public Dictionary<string, string> Values { get; set; }
}
```

Для користувачів також повертаються службові повідомлення або графічні відображення чеків, які можна відправити на друк.

2. Порядок роботи обліково-розрахункової системи з API бібліотекою ПРРО Cashalot

При відкритій касовій зміні доступні операції проведення розрахункових операцій, повернення товарів, службові внесення та видача грошових коштів, а також, створення Х-Звітів. Всі перелічені операції додатково дозволяють відправку документів на друк.

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

1. Відкриття зміни

Фіскальний номер РРО

```

/// <summary>
/// Відкрити зміну
/// </summary>

/// <param name="rroFiscalNumber"> Фіскальний номер РРО</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":""," "Values":{"ShiftID":"e8353c00-
f9eb-4b4c-aebc-ad0e55fb9109"}}
/// </returns>

public string OpenShift( string rroFiscalNumber );

```

[illegible]

Cashalot виконує запит щодо стану обраного фіскального номера каси, після чого створює XML повідомлення із призначенням локальним номером і типом «Відкриття зміни», засвідчує його кваліфікованим електронним підписом продавця і надсилає на ФСКО.

2. Відкриття зміни

Фіскальний номер РРО

```

/// <summary>
/// Відкрити зміну
/// </summary>
/// <param name="rroFiscalNumber"> Фіскальний номер РРО</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":""," "Values":{"ShiftID":"e8353c00-
f9eb-4b4c-aebc-ad0e55fb9109"}}
/// </returns>
public string OpenShift( string rroFiscalNumber );

```


Приклад:

```
Object[] parameters = {"1111111111111111"/>*фискальний номер ПРО*/};  
var ret = CallCashalotMethod( "OpenShift", parameters );
```

Cashalot виконує запит щодо стану обраного фіскального номера каси, після чого створює XML повідомлення із призначенням локальним номером і типом «Відкриття зміни», засвідчує його кваліфікованим електронним підписом продавця і надсилає на ФСКО.

Cashalot

v. 01.00.026

Файловий ключ

Апаратний ключ

Особистий ключ

C:\docs\ТОВ тігра\директор

...

Сертифікати

Лабораторний Приклад Прикладович(директо

▼

Пароль

👁

☒ Зберігати пароль протягом сеансу ?

Увійти

Відміна

⚙ Налаштувати прокси

❓ Відкрити довідку

Всі права захищені 2021
[my.cshalot.org.ua](#)

Cashalot

Відкриття зміни

✕

Каса

ПРРО №45

Касир

Сидоренко В. Л.

Службове внесення готівкових коштів на суму, грн

0,00

Відкрити зміну

Відмінити



Cashalot

Зміна успішно відкрита!

Ok

3. Службове внесення грошових коштів в касу

Для проведення операції службового внесення грошових коштів необхідно викликати метод **“ServiceInput”** з такими параметрами:

Фіскальний номер РРО ,
Сума службового внесення

Специфікація методу:

```

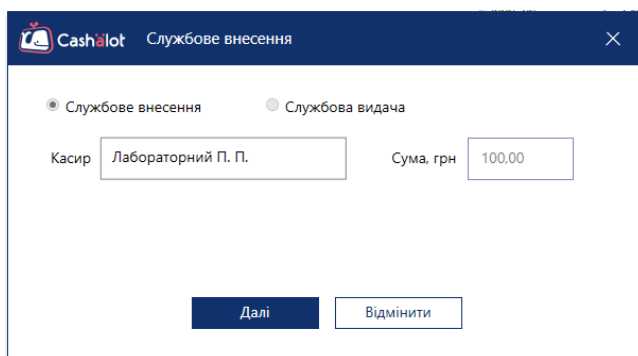
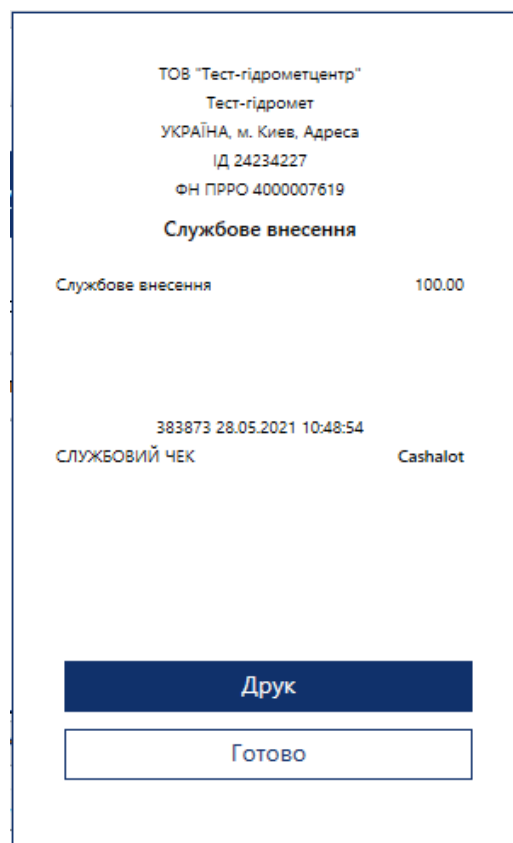
/// <summary>
/// Службове внесення готівки в касу
/// </summary>
/// <param name="roFiscalNumber">Фіскальний номер РРО</param>
/// <param name="sumVal"> Сума готівки </param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{}}
/// </returns>
public string ServiceInput( string roFiscalNumber, string sumVal
);

```

Приклад:

```
Object[] parameters = {"111111111111", /* фiскальний номер РРО*/
                       "5000.00"      /* сума службового внесення*/
                       };
var ret = CallCashalotMethod( "ServiceInput", parameters );
```

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

4. Службове вилучення (видача) грошових коштів з каси

Для проведення операції службового вилучення грошових коштів необхідно викликати метод **"ServiceOutput"** з такими параметрами:

```
{
    Фіскальний номер РРО ,
    Сума службової видачі
}
```

Специфікація методу:

```
/// <summary>
/// Службове виймання готівки з каси
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="sumVal"> Сума готівки </param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{}}
/// </returns>
public string ServiceOutput( string rroFiscalNumber, string sumVal);
```

Приклад:

```
Object[] parameters = {"111111111111", /* фіскальний номер РРО */
                       "5000.00"      /* сума службового вилучення*/
};
var ret = CallCashalotMethod( "ServiceOutput", parameters );
```

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


Cashalot
Службова видача
×

☐ Службове внесення
 ☒ Службова видача

Касир
 Сума, грн

ТОВ "Тест-гідрометцентр"
 Тест-гідромет
 УКРАЇНА, м. Київ, Адреса
 ІД 24234227
 ФН ПРРО 4000007619
Службова видача
 Службова видача 100.00
 383874 28.05.2021 10:49:38
 СЛУЖБОВИЙ ЧЕК Cashalot

5. Проведення розрахункових операцій продажу товарів/послуг

Для виконання розрахункової операції продажу товарів або послуг необхідно викликати метод фіскалізації чеку **"FiscalizeCheck"**, в якості параметрів якого передаються 2 JSON рядки: **strJSONData** для списку покупок та **payDataJSON** для повної вартості покупок. Вказані JSON рядки мають містити наступну інформацію:

strJSONData

Структура містить масив **ReceiptLst**, кожен елемент якого містить параметри номенклатури, та об'єкти **Comment** і [DocType](#).

Для кожного товару в чеку вказуються параметри у вигляді: *параметр = значення*:

| Параметр | Значення | Приклад |
|-------------------|---|-----------------------------------|
| VendorCode | Артикул товару | "24563" |
| Name | Найменування товару | "Бунчук гетьманський, сувенірний" |
| DocNumber | Номер договору (при використанні ломбардних послуг) | "01/a12 від 15.02.21" |
| CdUKDZED | Код УКДЗЕД | "1806903910" |
| CdDKPP | Код ДКПП | "47.00.69" |

NEW

NEW

| | | |
|-------------------------------|--|--|
| GoodsType | Тип товару: 1 – товар; 2 – послуга, 3 – послуга ломбардна *.
За замовчуванням: “1” | “1” |
| Barcode | Штрих-код | “123456789” |
| UnitType | Одиниця виміру. За замовчуванням: “шт” | “шт” |
| Quantity | Кількість | “1” |
| Price | Ціна (за одиницю товару) | “2000.00” |
| Amount | Вартість (сумарна, за всі одиниці товару, з врахуванням знижок) | “2000.00” |
| DiscountPrc | Сума знижки у відсотках (%) | “5.00” |
| DiscountSum | Сума знижки у грошових одиницях | “100.00” |
| IsPriceIncludeVAT | Чи включено ПДВ у ціну товару (true / false) | true |
| VATRate | Значення ПДВ у % | “20.00” |
| SumVAT | Значення ПДВ у грошових одиницях | “316.67” |
| IsExcise | Ознака підакцизного товару (true / false) | false |
| <u>OtherParameters</u> | Параметри у вигляді XML-структури реквізитів (якщо реквізит не заповнено в основній структурі, але заповнено в XML, то значення параметрів беруться з XML).
Необов'язкове поле. | <?xml version="1.0" encoding="UTF-8"?><Table><Record VAT ="20" IsExcisable="False" Nomenclature="STEP Пазл 104 Винни Пух" UKTZED="" VendorCode="82106" Barcode="" Dimension="шт" Discount="1" DiscountAmount="0.56" PriceIncludeVAT="1" AmountVAT="9.24"/></Table> |

* Використання ломбардних послуг описано в розділі "[Ломбардні послуги](#)".

Параметри повинні бути представлені у типі **string** (крім булевих параметрів **bool** **IsPriceIncludeVAT**, **IsExcise**), для уникнення помилок перетворення типів.

Опціональний об'єкт **Comment** використовується для додавання нефіскальної інформації:

| | |
|----------------|---|
| Comment | Опціональний рядок нефіскальної інформації, коментар. Допускається використання одного коментарю на чек. За необхідності сформулювати коментар з декількох рядків, для переходу на новий рядок використовується комбінація " \r " або " \n ". |
|----------------|---|

Об'єкт **DocType** є обов'язковим при проведенні ломбардних операцій, призначений для визначення типу документа (чека). Ломбардні операції описані в розділі "**Ломбардні послуги**".

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

payDataJSON

Для цього JSON рядка формується структура: *параметр = значення*:

| Параметр | Значення | Приклад |
|----------------------------|---|-----------|
| SumCash | Сума готівки, надана покупцем | "1960.00" |
| SumPayByCard | Сума безготівкової оплати | "0.00" |
| SumPayByCredit | Сума оплати в кредит | "0.00" |
| SumPayByCertificate | Сума оплати сертифікатом, карткою знижок та ін. | "0.00" |
| SumPayCheck | Сума по всім видам оплати, з урахуванням заокруглення | "1950.20" |

для якої значення також мають бути представлені у типі (**string**).

Зауваження! Для передачі параметрів функцій з дробовою частиною (в т.ч. нульовою, наприклад, "2,00") допускається використовувати десятковий розділювач "." (крапка) або "," (кома). Якщо параметр не задано, або задано некоректно, його значення приймається рівним одиниці (1).

Заокруглення суми по чеку та решта реалізовані наступним чином.

Сума до заокруглення обчислюється як сума параметрів рядків товарів в чеку, представлені в **strJSONData**:

$$SumInit = Amount_1 + Amount_2 + \dots$$

Сума після заокруглення обчислюється за певним алгоритмом користувача або відповідно до Постанови НБУ⁵ і передається в **payDataJSON** в якості параметру **SumPayCheck**.

Решта по чеку розраховується як:

$$SumRest = SumCash + SumPayByCard + SumPayByCredit + SumPayByCertificate - SumPayCheck$$

Значення заокруглення, яке буде зареєстроване на сервері ФСКО, є різницею:

$$SumDelta = SumPayCheck - SumInit$$

Додатні значення заокруглення відповідають заокругленню "вгору", від'ємні - заокругленню "вниз". Наприклад:

⁵ <https://zakon.rada.gov.ua/laws/main/v0025500-18> п.4.

| SumInit | SumPayCheck | SumDelta |
|---------|-------------|----------|
| 10,67 | 10,70 | 0,03 |
| 10,67 | 10,60 | - 0,07 |

У випадку, якщо параметр **SumPayCheck** не визначений (null) або дорівнює нулю, то він замінюється сумою по всім формам оплати. В такому випадку заокруглення буде дорівнювати нулю.

| | | |
|----------------|--|-----|
| <u>DocType</u> | Тип документа:
0 – чек продажу
4 – ломбардний чек
За замовчуванням <u>DocType</u> = 0, якщо параметр не задано. | "0" |
|----------------|--|-----|

Специфікація методу:

```

/// <summary>
/// Фіскалізувати чек
/// </summary>
/// <param name="roFiscalNumber">Фіскальний номер РРО</param>
/// <param name="strJSONData">
/// Параметри з інформацією про строки чеку (товар з характеристиками,
/// кількість, сума і тд.), у вигляді JSON строки
/// Приклад:
/// {"ReceiptLst":[{"VendorCode":"АртикулТовару1","Name":"НазваТовару1","CdUKDZED":"","CdDKPP":"","GoodsType":"","Barcode":"123456789","UnitType":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":false,"OtherParameters":null},{"VendorCode":"АртикулТовару2","Name":"НазваТовару2","CdUKDZED":"","CdDKPP":"","GoodsType":"","Barcode":"9999999999","UnitType":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":false,"OtherParameters":null}]}
/// </param>
/// <param name="payDataJSON">
/// Інформація про оплату у вигляді JSON строки
/// Приклад:
/// {"SumCash":"366","SumPayByCard":null,"SumPayByCredit":null,"SumPayByCertificate":null}
/// </param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{"ReceiptFiscalNumber":"203350","ReceiptLocalNumber":"1141"}}
/// </returns>
public string FiscalizeCheck( string roFiscalNumber, string strJSONData,
string payDataJSON );

```

Приклад:

```

LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();
lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();
CheckInfo chInfo = new CheckInfo();
VendorCode = "АртикулТовару1";
Name = "НазваТовару1";
CdUKDZED = String.Empty;
Barcode = "123456789";
UnitType = "шт";
Quantity = "1";

```

```

Price = "183";
Amount = "181,17";
DiscountPrc= String.Empty;
DiscountSum = "1.83";
IsPriceIncludeVAT = true;
VATRate = "20";
SumVAT = "30.2";
lstGoodsInCheck.ReceiptLst.Add( chInfo );//додати рядок чека з товаром №1

chInfo.VendorCode = "АртикулТовару2";
chInfo.Name = "НазваТовару2";
chInfo.Barcode = "9999999999";
lstGoodsInCheck.ReceiptLst.Add( chInfo );//додати рядок чека з товаром №2

PayInfoRsp payInfo = new PayInfoRsp();
payInfo.SumCash = "366"; //Сума готівки до сплати після заокруглення
//Конвертація структур до формату JSON рядка
string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
string payDataJSON = JsonConvert.SerializeObject( payInfo );
//Параметри функції
Object[] parameters =
{
    edFiscalNumberRRO.Text.Trim(), // фіскальний номер РРО
    strJSONData, // структура інформації з рядками чека
    payDataJSON // структура інформації про оплату по чеку
};
var ret = CallCashalotMethod( "FiscalizeCheck", parameters );
if ( !ret.Ret )
{
    MessageBox.Show( String.IsNullOrEmpty( ret.ErrorString ) ? "Не вдалося
        фіскалізувати чек" : ret.ErrorString );
}
else if ( ret.Values != null )
{
    string newCheckLocalNum = String.Empty;
    string newCheckFiscalNum = String.Empty;
    //отримання параметру 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" )
        {
            newCheckLocalNum = vals[i].Value;
        }
        else
        if ( vals[i].Key.ToLower() == "receiptfiscalnumber" )
        {
            newCheckFiscalNum = vals[i].Value;
        }
    }
    MessageBox.Show( $"Чек було успішно фіскалізовано.\n" +
        $"Чеку присвоєно локальний номер: { newCheckLocalNum
    },\n" +
        $"та фіскальний номер: { newCheckFiscalNum } "
    );
}
}

```

Приклад JSON рядків:

```

strJSONData=
{"ReceiptLst":
[
{
  "VendorCode":"АртикулТовару1",
  "Name":"НазваТовару1",
  "CdUKDZED":"",
  "CdDKPP":"",
  "GoodsType":"",
  "Barcode":"123456789",

```

```

        "UnitType": "шт",
        "Quantity": "1",
        "Price": "20000.03",
        "Amount": "19000.03",
        "DiscountPrc": null,
        "DiscountSum": "1000",
        "VATRate": "20",
        "IsPriceIncludeVAT": true,
        "SumVAT": "3800.01",
        "IsExcise": false,
        "OtherParametrs": null
    },
    {
        "VendorCode": "АртикулТовару2",
        "Name": "НазваТовару2",
        "CdUKDZED": "",
        "CdDKPP": "",
        "GoodsType": "",
        "Barcode": "123456789",
        "UnitType": "шт",
        "Quantity": "1",
        "Price": "20000.03",
        "Amount": "19000.03",
        "DiscountPrc": null,
        "DiscountSum": "1000",
        "VATRate": "20",
        "IsPriceIncludeVAT": true,
        "SumVAT": "3800.01",
        "IsExcise": false,
        "OtherParametrs": null
    }
]
}

payDataJSON=
{
    "SumCash": "20000",
    "SumPayByCard": "19000",
    "SumPayByCredit": null,
    "SumPayByCertificate": null,
    "SumPayCheck": "38000.10"
}

```

Приклад XML-рядка необов'язкових параметрів OtherParametrs:

```

<?xml version="1.0" encoding="UTF-8"?>
<Table>
  <Record
    VAT ="20"
    IsExcisable="False"
    Nomenclature="СТЕР Пазл 104 Віні Пух"
    UKTZED=""
    VendorCode="82106"
    Barcode=""
    Dimension="шт"
    Discount="1"
    DiscountAmount="0.56"
    PriceIncludeVAT="1"
    AmountVAT="9.24"
  />
</Table>

```

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

ТОВ "Тест-гідрометцентр"

Тест-гідромет

УКРАЇНА, м. Київ, Адреса

ІД 24234227

ФН ПРРО 4000007619

КАСІР Лабораторний П. П.

Чек №494010

| | |
|--|---------------------|
| НазваТовару1 | |
| 1 X 20000,03 | 20000,03 О |
| Знижка | -1000,00 |
| НазваТовару2 | |
| 1 X 20000,03 | 20000,03 |
| Знижка | -1000,00 |
| ----- | |
| Сума, грн | 38000,06 |
| ПДВ О=20% | 3166,67 |
| ----- | |
| Заокруглення | 0,04 |
| До сплати | 38000,10 |
| ГОТІВКА, грн | 20000,00 |
| Решта, грн | 999,90 |
| КАРТКА | 19000,00 |
| Ingenico | |
| Термінал | 51K700J8 |
| ОПЛАТА | |
| Сума | 19000,00 грн. |
| MASTERCARD | |
| ЕПЗ: | XXXXXXXXXXXX6883 |
| Код авторизації: | 680148 |
| RRN: | 048867173620 |
| ДАТА:9/6/2021 | ЧАС:1:07:28 PM |
| ЧЕК: | 224 |
| Касир: | |
| Держатель ЕПЗ: Підпис власника карти не потрібен | |
| 494010 | 9/6/2021 1:07:31 PM |



ФІСКАЛЬНИЙ ЧЕК
ОНЛАЙН
Cashalot

Друк

Надіслати на Email

Готово

6. Визначення фіскального номеру чеку

З метою визначення номеру фіскального чека, зареєстрованого раніше в ФСКО, за локальним номером, зареєстрованим в касі при продажу виконується метод

"GetReceiptFiscalNumberByLocalNumber". В якості параметру передається параметр:

```
{
    Фіскальний номер РРО ,
    Локальний номер чеку
}
```

Специфікація методу:

```

/// <summary>
/// Отримати фінансовий номер чека по локальному номеру
/// </summary>
/// <param name="rroFiscalNumber">Фінансовий номер РРО</param>
/// <param name="checkLocalNumber">Локальний номер чека</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","
/// "Values":{"ReceiptFiscalNumber":"203366"}}
/// </returns>
public string GetReceiptFiscalNumberByLocalNumber( string
rroFiscalNumber, string checkLocalNumber );

```

Приклад:

```

Object[] parameters = { edFiscalNumberRRO.Text.Trim(), checkLocalNumber};
var ret = CallCashalotMethod( "GetReceiptFiscalNumberByLocalNumber",
parameters );

```

Визначений фінансовий номер чека використовується, зокрема, в якості додаткового параметра при фінансовізації чеку повернення товару.

7. Проведення операції повернення товару

Метод повернення товару схожий на метод фінансовізації чека. Додатково потрібно вказати в якості параметру номер фінансового чека товару, який було продано.

Для виконання повернення товару чи товарів з фінансового чека з відомим (визначеним) фінансовим номером використовується метод **"FiscalizeReturnCheck"**, для якого формується чек повернення аналогічно операції [проведення розрахункової операції](#) із зазначенням тих товарів, які реєструються для повернення: створюються **strJSONData** для списку товарів повернення, **payDataJSON** для повної вартості покупок.

Додатково зазначається параметр:

| Параметр | Значення | Приклад | Примітка |
|-------------------------|--|----------------------------|--------------------------|
| strFiscalNumReturnCheck | Фінансовий номер чека по якому здійснюється повернення | "275301"
"19255.3.7284" | онлайн-чек
офлайн-чек |

Специфікація методу:

```

/// <summary>
/// Фіскалізувати видатковий чек (повернення)
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="strJSONData">
/// Параметри з інформацією про строки чеку (товар з характеристиками,
кількість, сума і тд.), у вигляді JSON строки
/// Приклад:
/// { "ReceiptLst": [ { "VendorCode": "АртикулТовару1", "Name": "НазваТовару1", "CdUKDZED": "", "CdDKPP": "", "GoodsType": "", "Barcode": "123456789", "UnitType": "шт", "Quantity": "1", "Price": "183", "Amount": "181,17", "DiscountPrc": "", "DiscountSum": "1.83", "VATRate": "20", "IsPriceIncludeVAT": true, "SumVAT": "30.2", "IsExcise": false, "OtherParameters": null }, { "VendorCode": "АртикулТовару2", "Name": "НазваТовару2", "CdUKDZED": "", "CdDKPP": "", "GoodsType": "", "Barcode": "9999999999", "UnitType": "шт", "Quantity": "1", "Price": "183", "Amount": "181,17", "DiscountPrc": "", "DiscountSum": "1.83", "VATRate": "20", "IsPriceIncludeVAT": true, "SumVAT": "30.2", "IsExcise": false, "OtherParameters": null } ] }
/// </param>
/// <param name="payDataJSON">
/// Інформація про оплату у вигляді JSON строки
/// Приклад:
/// { "SumCash": "366", "SumPayByCard": null, "SumPayByCredit": null, "SumPayByCertificate": null, "SumPayCheck": "362,34" }
/// </param>
/// <param name="strFiscalNumReturnCheck"> Фіскальний номер чеку по якому здійснюється повернення</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// { "Ret": true, "ErrorString": "", "Values": { "ReceiptFiscalNumber": "203350", "ReceiptLocalNumber": "1141" } }
/// </returns>
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 );
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 );
}

```

```
strJSONData=
{"ReceiptLst":[{"VendorCode":"АртикулТовару2","Name":"НазваТовару2","CdUKDZED":"","Barcode":"999999999999","UnitType":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":false,"OtherParametersOtherParameters
```

Для відображення X-Звіту необхідно викликати метод **"ShowXReport"** з параметром:

```
Object[] parameters = { edFiscalNumberRRO.Text.Trim(),
receiptFiscalnumber};
var ret = CallCashalotMethod( "GetReceiptLocalNumberByFiscalNumber",
parameters );
```

10. Отримання XML-структури чеку

Для отримання XML-структури фіскалізованого чеку використовується функція "GetReceiptXML" з параметрами

```
{
Фіскальний номер РРО,
Фіскальний номер чеку
}
```

Специфікація методу:

```
/// <summary>
/// Показати чек для друку або відправки на емейл, по фіскальному номеру чека та номеру РРО
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="checkLocalNumber">Фіскальний номер чека</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON з параметром Base64Str1251ReceiptXML - закодована в Base64 строка XML в кодуванні Win-1251
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{"Base64Str1251ReceiptXML":"...."}}
/// </returns>
public string GetReceiptXML(string rroFiscalNumber, string checkFiscalNumber );
```

11. Надрукувати періодичний звіт по датам

Надрукувати періодичний звіт по датам Для цієї операції використовується метод **Get** PrintPReportDate з параметрами:

```
{
Фіскальний номер РРО ,
Фіскальний номер чеку
}
```

Специфікація методу:

PrintPReportDate

```
/// </summary>
/// <param name="FiscalNumberRRO">Фіскальний номер РРО</param>
/// <param name="dtBeg">Початок періоду</param>
/// <param name="dtEnd">Кінець періоду</param>
/// <param name="isShort">Ознака скорочений- TRUE/ Повний - FALSE</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{}}
/// </returns>
public string PrintPReportDate(string rroFiscalNumber, string dateBeg, string dateEnd, string isShort)
```

12. Відображення фіскалізованого чеку

Після фіскалізації чек відображається на екрані. Проте, іноді необхідно відобразити чек повторно. Для цього можна скористатися функцією **"ShowReceipt"** з параметрами

```
{  
    Фіскальний номер PPO,  
    Фіскальний номер чеку  
}
```

Специфікація методу:

```
/// <summary>  
/// Показати чек для друку або відправки на емейл, по фіскальному номеру чека та номеру PPO  
/// </summary>  
/// <param name="rroFiscalNumber">Фіскальний номер PPO</param>  
/// <param name="checkLocalNumber">Фіскальний номер чека</param>  
/// <returns>  
/// Функція повертає структуру у вигляді JSON  
/// Приклад:  
/// {"Ret":true,"ErrorString":"","Values":{}}  
/// </returns>  
public string ShowReceipt(string rroFiscalNumber, string checkFiscalNumber );
```

13. Закриття зміни та створення Z-Звіту

Для закриття зміни необхідно викликати метод **"CloseShift"** з параметром:

```
{  
    Фіскальний номер PPO  
}
```

Специфікація методу:

```
/// <summary>  
/// Закриття зміни  
/// </summary>  
/// <param name="rroFiscalNumber">Фіскальний номер PPO</param>  
/// <returns>  
/// Функція повертає структуру у вигляді JSON  
/// Приклад:  
/// {"Ret":true,"ErrorString":"","Values":{}}  
/// </returns>  
public string CloseShift( string rroFiscalNumber );
```

Приклад:

```
Object[] parameters = {"11111111111111" /*фіскальний номер PPO*/};  
var ret = CallCashalotMethod( "CloseShift", parameters );
```

Перед створенням Z-Звіту та закриттям зміни, описаним в даному розділі, автоматично виконується [метод службового вилучення наявних грошових коштів з каси](#).

Також виконується графічне відображення Z-Звіту для можливості подальшого друку.

ТОВ "Тест-гідрометцентр"

Тест-гідромет

УКРАЇНА, м. Київ, Адреса

ІД 24234227

ФН ПРРО 4000007619

Z - звіт

Продаж

| | |
|-----------------------|----------|
| ЗАГАЛЬНА СУМА | 38000.10 |
| ГОТІВКА, грн | 19000.10 |
| КАРТКА | 19000.00 |
| Кількість чеків | 1 |
| Об'єкт О | 19000.03 |
| ПДВ О=20% | 3166.67 |
| ЗАГАЛЬНА СУМА ПОДАТКУ | 3166.67 |

Повернення

| | |
|-----------------------|------|
| ЗАГАЛЬНА СУМА | 0.00 |
| ГОТІВКА, грн | 0.00 |
| КАРТКА | 0.00 |
| Кількість чеків | 0 |
| ЗАГАЛЬНА СУМА ПОДАТКУ | 0.00 |

Службове внесення

0.00

Службова видача

19000.10

РЕГІСТРИ ДЕННИХ ПІДСУМКІВ ОБНУЛЕНІ

ФІСКАЛЬНИЙ ЗВІТНИЙ ЧЕК ДІЙСНИЙ

494012 06.09.2021 13:10:47

ФІСКАЛЬНИЙ ЧЕК

Cashalot

Друк

Готово

NEW

4. Ломбардні операції

Для фіскалізації ломбардних послуг введено системні номенклатури, які необхідно використовувати при виклику методу фіскалізації чеку `Fiskalizecheck()`. Повернення товару при ломбардних операціях не підтримується.

Також необхідно вказати тип документу `DocType = 4` в JSON-структурі параметрів чеку.

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" | + |

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

Напрямок руху необхідно враховувати при обчисленні кінцевого підсумку в чеку: від позицій типу 2 віднімаються позиції типу 3, результат вказується в **Paydata** з врахуванням знаку.

Приклад даних для ломбардної операції

```
Itmdata = "{ \"ReceiptLst\": [{ \"VendorCode\": \"ЛО-00001\", \"Name\": \"Повернення  
страхування\", \"DocNumber\": \"1/a12 від  
15.08.21\", \"CdUKDZED\": null, \"CdDKPP\": null, \"GoodsType\": \"3\", \"Barcode\": null, \"UnitType\":  
\"шт\", \"Quantity\": \"1\", \"Price\": \"100\", \"Amount\": \"100\", \"DiscountPrc\": null, \"DiscountSum\":  
null, \"IsPriceIncludeVAT\": false, \"VATRate\": null, \"SumVAT\": null, \"IsExcise\": false, \"OtherParams\": null }], \"Comment\": \"ТЕСТОВА ЛОМБАРДНА ОПЕРАЦІЯ\", \"DocType\": \"4\" }";  
Paydata =  
"{ \"SumCash\": \"100\", \"SumPayByCard\": null, \"SumPayByCredit\": null, \"SumPayByCertificate\":  
null, \"SumPayCheck\": \"100\" }";
```

NEW

5. Приклад використання API для ломбардних операцій

```
LstCheckInfoRsp lstGoodsInCheck = new LstCheckInfoRsp();  
lstGoodsInCheck.ReceiptLst = new List<CheckInfo>();  
CheckInfo chInfo = new CheckInfo();  
chInfo.VendorCode = "ЛО-00002";  
chInfo.Name = "Надання кредиту";  
chInfo.Quantity = "1"; //кількість  
chInfo.Price = "200"; //ціна (обов'язково)  
chInfo.Amount = "200"; //сумма ((кількість * ціну) - дисконт)  
chInfo.GoodsType = "3"; // мінусова операція  
  
//додати строку чеку з товаром №1  
lstGoodsInCheck.ReceiptLst.Add( chInfo );  
  
CheckInfo chInfo2 = new CheckInfo( chInfo );  
chInfo2.VendorCode = "ЛО-00009";  
chInfo2.Name = "Повернення кредиту";  
chInfo2.Quantity = "1"; //кількість  
chInfo2.Price = "100"; //ціна (обов'язково)  
chInfo2.Amount = "100"; //сумма ((кількість * ціну) - дисконт)  
chInfo2.GoodsType = "2"; // плюсова операція  
  
//додати строку чеку з товаром №2  
lstGoodsInCheck.ReceiptLst.Add( chInfo2 );  
lstGoodsInCheck.DocType = 4; // Вказуємо що тип чека ломбардний!!!
```

```

PayInfoRsp payInfo = new PayInfoRsp();
payInfo.SumCash = "-100"; //Сума готівки наданої покупцем(мінусова так як до
видачі 100грн )

payInfo.SumPayCheck = "-100";//Сума оплати загальна(мінусова так як до видачі
100грн)

//Конвертація структур до формату джейсон строки
string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
string payDataJSON = JsonConvert.SerializeObject( payInfo );

//Параметри функції
Object[] parameters =
{
    edFiscalNumberRRO.Text.Trim(), // фіскальний номер РРО
    strJSONData, // структура інформації зі строками чека
    payDataJSON // структура інформації про оплату по чеку
};

var ret = CallCashaLotMethod( "FiscalizeCheck", parameters );
if ( ret?.Ret != true )
{
    MessageBox.Show( String.IsNullOrEmpty( ret?.ErrorString ) ? "Не вдалося
фіскалізувати чек" : ret?.ErrorString );
}
else if ( ret?.Values != null )
{
    //отримання параметру 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} "
        );
}

```

6. Приклад реалізації функції для виклику API методів ПРРО Cashalot

//Структура інформації строки чеку(містить всі дані для створення товару в Cashalot)

```

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 замовчуванням
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 Пазл 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;
        //Відображати додаткову інформацію в Z-звіті у печатній формі чеку
        public String PrintPReportDate { get; set; } = String.Empty;
        //Штрихкод акцизної марки
        public String ExciseStampBarcode { get; set; } = String.Empty;
    };

    //Структура з масивом срок чеку
    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 = "11111111111111";

    //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 метода з API 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 PPO" );

    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;  
}
```