

Специфікація рішень API ПРРО Cashalot

Актуально починаючи з версії ПРРО Cashalot 1.00.21

Загальна інформація

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

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)/X-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.

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.

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.

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)-206-72-25.

Варіант 1 – драйвер ПРРО Cashalot для BAS 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.

В драйвері CashalotApiDrv реалізована обробка наступних функцій:

1. ПолучитьНомерВерсии
2. ПолучитьОшибку
3. ПолучитьПараметры
4. УстановитьПараметр
5. Подключить
6. Отключить
7. ТестУстройства
8. ОткрытьСмену
9. ОткрытьЧек
10. НапечататьФискальнуюСтроку
11. НапечататьФискСтроку
12. ЗакрытьЧек
13. ОтменитьЧек
14. НапечататьЧекВнесенияВыемки
15. НапечататьОтчетБезГашения
16. НапечататьОтчетСГашением
17. ЗакрытьСмену

При роботі з ПРРО Cashalot рекомендовано кожну операцію в BAS супроводжувати встановленням параметрів налаштувань (фіскальний номер ПРРО та шлях до ПРРО Cashalot):

1. УстановитьПараметр
2. Операція (наприклад, реєстрація чеку)

1. Отримання номеру версії драйверу

Для отримання номеру версії драйверу використовується функція

СТРОКА ПолучитьНомерВерсии()

Приклад:

Версия = ОбъектДрайвера.ПолучитьНомерВерсии();

2. Отримання опису помилки

Для отримання опису помилки використовується функція

ЧИСЛО ПолучитьОшибку(ОписаниеОшибки)

Параметри функції зазначені в Таблиці 2.1.

Таблиця 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]);
```

3. 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);
```

4. Встановлення налаштувань драйверу

Для встановлення значень параметрів налаштувань 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
"DeviceIDFnRRO"	Фіскальний номер ПРРО
"PathToCashalotDir"	Шлях до каталогу з встановленим ПРРО Cashalot

Приклад:

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

5. Підключення фіскального реєстратора ПРРО Cashalot

Для підключення ПРРО Cashalot 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:

```
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.Подключить(ПараметрыПодключения.ИДУстройства);
```

6. Відключення фіскального реєстратора ПРРО Cashalot

Для відключення ПРРО Cashalot використовується функція

БУЛЕВО Отключить(ИДУстройства)

Параметри функції зазначені в Таблиці 6.1.

Таблиця 6.1 Параметри функції відключення пристрою.

Ім'я	Тип	Опис
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Функція повертає ідентифікатор відключеного ПРРО.

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.Отключить(ПараметрыПодключения.ИДУстройства);
```

7. Перевірка ПРРО Cashalot

Для перевірки ПРРО шляхом пробного підключення і опитування пристрою з поточними значеннями параметрів, встановлених функцією «УстановитьПараметр» використовується функція

БУЛЕВО ТестУстройства (Описание, ПризнакДемоРежима)

Параметри функції зазначені в Таблиці 7.1.

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("", "");
```

8. Відкриття зміни

При відкритті зміни застосовується функція

БУЛЕВО ОткрытьСмену(ИДУстройства)

Параметри функції зазначені в Таблиці 8.1. 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 8.1 Параметри функції відкриття зміни.

Ім'я	Тип	Опис
ИДУстройства	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);
```

9. 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, XML таблица);

або

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

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

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.

Функція **Закри́тьЧек** 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. Порухення послідовності функцій призведе до помилки реєстрації чеків.

Заокруглення суми по чеку реалізовано наступним чином. В функцію **Закри́тьЧек** 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:

ТОВ "Тест-гідрометцентр"	
Тест-гідромет	
УКРАЇНА, м. Київ, Адреса	
ІД 24234227	
ФН ПРРО 4000007619	
КАС/ІР Гідрометр 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
<input type="button" value="Друкувати звіт"/>	
<input type="button" value="Готово"/>	

У випадку, якщо параметр

СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСУчетомОкругления не визначений (null) 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.

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

Таблиця 9.1 Параметри функцій для формування чеку.

Функція	Параметри функцій		
	Ім'я	Тип	Опис
ОткрытьЧек	ИДУстройства	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	ИДУстройства	СТРОКА	Ідентифікатор пристрою
	Наименование	СТРОКА	Найменування товару
	КодУКТЗЕД	СТРОКА	Код УКТЗЕД
	Артикул	СТРОКА	Артикул
	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 (от 0 до 99,99)
	ПризнакСуммаВключаетНДС	БУЛЕВО	Ознака, що сума включає ПДВ
	Сумма НДС	ЧИСЛО	ПДВ в процентах (от 0 до 99,99)
	ПризнакАкцизногоНалога	БУЛЕВО	Ознака, що сума включає ПДВ
	НомерСтроки	БУЛЕВО	Номер рядку (не використовується)
	ДопРеквизиты	СТРОКА	Додаткові параметри (XML таблиця)
НапечататьФискСтроку	ИДУстройства	СТРОКА	Ідентифікатор пристрою
	Наименование	СТРОКА	Найменування товару
	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 (от 0 до 99,99)
ДопРеквизиты	СТРОКА	Додаткові параметри	
ЗакрытьЧек	ИДУстройства	СТРОКА	Ідентифікатор пристрою
	НаличнаяОплата	ЧИСЛО	Сума оплати готівкою
	ОплатаКартой	ЧИСЛО	Сума оплати картою
	ОплатаКредитом	ЧИСЛО	Сума оплати кредитом
	ОплатаСертификатом	ЧИСЛО	Сума оплати сертифікатом

СуммаОплатыВсегоЧекаОбщ аяПоВсемВидамОплатыСУче томОкругления	ЧИСЛО	Сума до оплати, що враховує всі форми оплати, знижки та заокруглення
ФискальныйНомерЧекаПоКот оромуПроизводитсяВозврат	СТРОКА	Фіскальний номер чеку, за яким виконується повернення
НомерЧекаЧисловой	ЧИСЛО [OUT]	Номер нового чеку
ФискальныйНомерСозданног оЧека	СТРОКА [OUT]	Фіскальний номер нового чеку

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
UKTZED	Код УКТЗЕД
VENDORCODE	Артикул
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

Приклад текстового XML з додатковими параметрами:

```
<?xmlversion="1.0"encoding="UTF-8"?>
<Table>
  <VAT="20"
```

```
IsExcisable="False"  
Nomenclature="СТЕРПазл1000Гавайи.ОстровОаху"  
UKTZED=""  
VendorCode="79078"  
Barcode="1234567890"  
Dimension="шт"  
Discountamount="1.83"  
IsPriceIncludeVAT="1"  
AmountVAT="30.2"/>  
</Table>
```

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[0];  
ФискальныйЧек = ВходныеПараметры[1];  
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 = 240,00;  
СуммаБезналичнойОплаты1 = 0;  
СуммаБезналичнойОплаты2 = 0;  
СуммаБезналичнойОплаты3 = 0;  
СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСучетомОкругления = 239,60;  
Ответ = ОбъектДрайвера.ЗакретьЧек(ИДУстройства, СуммаНаличнойОплаты,  
СуммаБезналичнойОплаты1, СуммаБезналичнойОплаты2, СуммаБезналичнойОплаты3,  
СуммаОплатыВсегоЧекаОбщаяПоВсемВидамОплатыСучетомОкругления, НомерЧекаВозвратаФискальный,  
НомерСозданногоЧека, ФискальныйНомерСозданногоЧека);
```

Для запису рядку чеку можна використати *НапечататьФискСтроку*:

```
ИДУстройства = "4000001234";  
Наименование = "Товар1";
```

```

НомерСекции = 1;
Количество = 5;
Цена = 50,45;
Сумма = 239,64;
СтавкаНДС = 50,45;
ДопРеквизиты = "<?xmlversion=""1.0""encoding=""UTF-8""?>
<Table>
    <VAT=""20""
    IsExcisable=""False""
    Nomenclature=""Товар1""
    UKTZED=""000001""
    VendorCode=""79078""
    Barcode=""1234567890""
    Dimension=""шт""
    DiscountAmount=""1.83""
    PriceIncludeVAT=""1""
    AmountVAT=""30.2"" />
</Table>";
Ответ = ОбъектДрайвера.НапечататьФискСтроку(ИДУстройства, Наименование, Количество, Цена,
Сумма, НомерСекции, СтавкаНДС, ДопРеквизиты);

```

10. 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
ИДУстройства	СТРОКА	Ідентифікатор пристрою

Приклад:

```

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

```

11. 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);

12. Службове внесення

Для виконання операції службового внесення використовується функція

БУЛЕВО НапечататьЧекВнесенияВыемки (ИДУстройства, Сумма)

Параметри функції зазначені в Таблиці 12.1.

Таблиця 12.1 Параметри функції службового внесення.

Ім'я	Тип	Опис
ИДУстройства	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)*);

13. 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 13.1.

Таблиця 13.1 Параметри функції службового внесення.

Ім'я	Тип	Опис
ИДУстройства	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)*);

14. Формування Х-звіту

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

БУЛЕВО НапечататьОтчетБезГашения(**ИДУстройства**)

Параметри функції зазначені в Таблиці 14.1.

Таблиця 14.1 Параметри функції Х-звіту.

Ім'я	Тип	Опис
ИДУстройства	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*);

15. Формування Z-звіту

Для формування Z-звіту використовується функція

БУЛЕВО НапечататьОтчетСГашением(ИДУстройства)

Параметри функції зазначені в Таблиці 15.1.

Таблиця 15.1 Параметри функції Z-звіту.

Ім'я	Тип	Опис
ИДУстройства	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*);

16. Закриття зміни

Для виконання операції службового внесення використовується функція

БУЛЕВО ЗакрытьСмену(ИДУстройства)

Параметри функції зазначені в Таблиці 16.1.

Таблиця 16.1 Параметри функції закриття зміни.

Ім'я	Тип	Опис
ИДУстройства	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 = ЗакрытьСмену(*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 17.

Таблиця 17. Порядок виклику функцій при роботі з касою.

Касова операція	Послідовність функцій драйверу
Відкриття зміни	УстановитьПараметр ОткрытьСмену

Фіскалізація чеків	УстановитьПараметр ОткрытьЧек НапечататьФискальнуюСтроку <i>або</i> НапечататьФискСтроку ЗакрытьЧек <i>або</i> ОтменитьЧек
Повернення товару	УстановитьПараметр ОткрытьЧек НапечататьФискальнуюСтроку <i>або</i> НапечататьФискСтроку ЗакрытьЧек <i>або</i> ОтменитьЧек
X-звіт	УстановитьПараметр НапечататьОтчетБезГашения
Z-звіт	УстановитьПараметр НапечататьОтчетСГашением
Закриття зміни	УстановитьПараметр ЗакрытьСмену
Внесення грошей	УстановитьПараметр НапечататьЧекВнесенияВыемки
Видача грошей	УстановитьПараметр НапечататьЧекВнесенияВыемки
Службові операції	ПолучитьОшибку <i>або</i> ПолучитьПараметры <i>або</i> ТестУстройства

Варіант 2а – API ПРРО Cashalot з COM інтерфейсом CashalotApi32.dll, CashalotApi64.dll

Підключення ПРРО Cashalot до розрахунково-облікової платформи або системи відбувається за допомогою бібліотек **CashalotApi32.dll** та **CashalotApi64.dll** (в залежності від розрядності платформи, далі **CashalotApi**). Вказані бібліотеки розміщені в кореновому каталозі ПРРО Cashalot, і шлях до них необхідно вказати при встановленні параметрів (див. Встановлення налаштувань бібліотеки).

Бібліотеку необхідно зареєструвати. В меню Windows Старт натисніть “Виконати” (Run) та введіть рядок:

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

(для 32-бітної платформи)

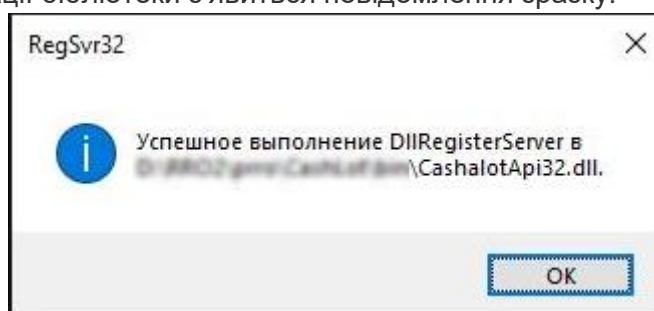
```
regsvr32 <Повний шлях до ПРРО Cashalot>\CashalotApi64.dll
```

(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:

1. SetParameter
2. GetVersion
3. CashInOutcome
4. OpenShift
5. FiscalizeCheck
6. FiscalizeReturnCheck
7. GetReceiptLocalNumberByFiscalNumber
8. GetReceiptFiscalNumberByLocalNumber
9. GetReceiptXML
10. ShowReceipt
11. GetLastError
12. PrintXReport
13. CloseShift
14. PrintZReport

Функції

OpenShift

FiscalizeCheck

FiscalizeReturnCheck

CloseShift

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

Приклади використання COM-функцій наведено у вигляді

```
Ret = GetVersion()
```

1. SetParameter – Встановлення налаштувань API бібліотеки

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

Параметр **Name** може приймати значення:

Значення	Опис
DeviceIDFnRRO	Фіскальний номер ПРРО
PathToCashalotDir	Шлях до каталогу з встановленим ПРРО Cashalot

Відповідь

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

Приклад:

```
param = "PathToCashalotDir"  
paramVal = "C:\Users\User\AppData\Roaming\Cashalot"  
Ret = SetParameter(param, paramVal)  
param = "DeviceIDFnRRO"  
paramVal = "4000007621"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.SetParameter(param, paramVal)
```

Приклад відповіді:

```
Ret = True;
```

2. 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";
```

3. 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
FiscalNumberRRO	String	Ідентифікатор пристрою

Відповідь

Структура **CashalotApiRetVal**, що містить значення відповіді Description, JsonVal, Return, ShiftID.

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 = "";
```

4. 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 такі параметри номенклатури:

Таблиця полів елементів масиву ReceiptLst

Поле	Значення
VendorCode	Артикул
Name	Найменування товару
CdUKDZED	Код УКТЗЕД
CdDKPP	Код ДКПП
GoodsType	Тип номенклатури: 1 – товар 2 – 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)
DiscountSum	Сума знижки на позицію
VATRate	ПДВ в процентах (от 0 до 99,99)
IsPriceIncludeVAT	Ознака, що ціна враховує ПДВ
SumVAT	ПДВ в процентах (от 0 до 99,99)
IsExcise	Ознака підакцизного товару
OtherParametrs	Додаткові параметри у вигляді XML таблиці

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

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

Атрибути XML-рядка OtherParams

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 PRICEINCLUDEVAT	Ознака, що ціна враховує ПДВ
AMOUNTVAT	Сума ПДВ, у форматі <Гривні, Копійки>
ISEXCISABLE	Ознака підакцизного товару, True/False

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

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, помилки, локальний та фіскальний номери чеку
ReceiptFiscalNum	String	Фіскальний номер чеку
ReceiptLocalNum	String	Локальний номер чеку
Return	Bool	Успішність операції

Приклад:

```

JSONGoodsLstData = "{\"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, \"OtherParametrs\": 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, \"OtherParametrs\": null}]}\"
JSONPayData =
\"{\"SumCash\": \"366\", \"SumPayByCard\": null, \"SumPayByCredit\": null, \"SumPayByCertificate\": null, \"SumPayCheck\": \"362,34\"}\"
App = CreateObject(\"AddIn.CashaLotApi\")
Ret = App.App.FiscalizeCheck(\"4000007621\", JSONGoodsLstData, JSONPayData)

```

Приклад відповіді:

```

ret.JsonVal =
{"Ret": true, "ErrorString": "", "Values": {"ReceiptFiscalNumber": "391929", "ReceiptLocalNumber": "502"}}
ret.ReceiptFiscalNum = 391929
ret.ReceiptLocalNum = 502
ret.Return = true

```

5. 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 у випадку формування чеку продажу

Відповідь

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

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

Приклад:

```
JSONGoodsLstData = "{\"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, \"OtherParams\": null}, {\"VendorCode\": \"АртикулТовару2\", \"Name\": \"НазваТовару2\", \"CdUKDZED\": \"\", \"CdDKPP\": \"\", \"GoodsType\": \"\", \"Barcode\": \"999999999\", \"UnitType\": \"шт\", \"Quantity\": \"1\", \"Price\": \"183\", \"Amount\": \"181,17\", \"DiscountPrc\": \"\", \"DiscountSum\": \"1.83\", \"VATRate\": \"20\", \"IsPriceIncludeVAT\": true, \"SumVAT\": \"30.2\", \"IsExcise\": false, \"OtherParams\": null}]}\"
JSONPayData =
\"{\"SumCash\": \"366\", \"SumPayByCard\": null, \"SumPayByCredit\": null, \"SumPayByCertificate\": null, \"SumPayCheck\": \"362,34\"}\"
ReceiptFiscalNum = 391929
App = CreateObject(\"AddIn.CashaLotApi\")
Ret = App.FiscalizeReturnCheck(ReceiptFiscalNum, JSONGoodsLstData, JSONPayData, ReceiptFiscalNum)
```

Приклад відповіді:

```
ret.JsonVal =
{"Ret": true, "ErrorString": "", "Values": {"ReceiptFiscalNumber": "391930", "ReceiptLocalNumber": "503"}}
ret.ReceiptFiscalNum = 391930
ret.ReceiptLocalNum = 503
ret.Return = true
```

6. 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
FiscalNumberRRO	String	Ідентифікатор пристрою
ReceiptFiscalNumber	Number	Фіскальний номер чеку

Відповідь

	String	Локальний номер чеку
--	--------	----------------------

Якщо номер чеку не знайдено, то повертається пустий рядок.

Приклад:

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

Приклад відповіді:

Ret = "503"

7. GetReceiptFiscalNumberByLocalNumber – Отримання фіскального номеру чеку за його локальним номером

Функція дозволяє визначити фіскальний номеру чеку за відомим локальним 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
FiscalNumberRRO	String	Ідентифікатор пристрою
ReceiptLocalNumber	Number	Локальний номер чеку

Відповідь

	String	Фіскальний номер чеку
--	--------	-----------------------

Якщо номер чеку не знайдено, то повертається пустий рядок.

Приклад:

```
FiscalNumberRRO = "4000007621"  
ReceiptLocalNumber = "502"  
App = CreateObject("AddIn.CashaLotApi")  
Ret = App.GetReceiptFiscalNumberByLocalNumber(FiscalNumberRRO,  
ReceiptLocalNumber)
```

Приклад відповіді:

Ret = 391929

8. 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
---------------------	--------	-----------------------

Відповідь

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

Приклад:

```
FiscalNumberRRO = "4000007621"
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>F312B173-A8ED-43D3-905A-1BE298816EC8</UID>
    <TIN>24234227</TIN>
    <ORGNM>ТОВ "Тест-гідрометцентр"</ORGNM>
    <POINTNM>Тест-гідромет</POINTNM>
    <POINTADDR>УКРАЇНА, м. Київ, Адреса</POINTADDR>
    <ORDERDATE>01062021</ORDERDATE>
    <ORDERTIME>153801</ORDERTIME>
    <ORDERNUM>503</ORDERNUM>
    <CASHDESKNUM>3</CASHDESKNUM>
    <CASHREGISTERNUM>4000007621</CASHREGISTERNUM>
    <CASHIER>Лабораторний П. П.</CASHIER>
    <VER>1</VER>
    <ORDERTAXNUM>391930</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>
    <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>
    <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>

```

9. 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
FiscalNumberRRO	String	Ідентифікатор пристрою
ReceiptFiscalNumber	Number	Фіскальний номер чеку

Відповідь

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

Приклад:

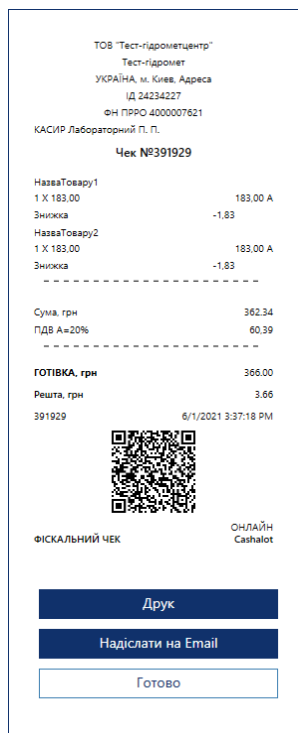
```

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

```

Приклад відповіді:

```
Ret = True;
```



10. ServiceInput – Службове внесення грошей в касу

Для виконання службового внесення в касу використовують функцію **ServiceInput**

Параметри

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

Відповідь

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

Приклад:

```
DeviceID = "4000007621"
Amount = "100"
App = CreateObject("AddIn.CashaLotApi")
Ret = App.ServiceInput (DeviceID, Amount)
```

11. ServiceOutput – Службова видача грошей з каси

Для виконання службової видачі з каси використовують функцію **ServiceOutput**

Параметри

Ім'я	Тип	Опис
DeviceID	String	Ідентифікатор пристрою

Amount	Number	Додатне значення суми видачі у форматі <Грн, Коп>
--------	--------	---

Відповідь

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

Приклад:

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

Приклад відповіді:

```
Ret = True;
```

12. 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. "
```

13. PrintXReport - Формування X-звіту

Параметри

Ім'я	Тип	Опис
DeviceID	String	Ідентифікатор пристрою

Відповідь

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

Приклад:

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

Приклад відповіді:

```
Ret = True;
```

14. CloseShift - Закриття зміни

Параметри

Ім'я	Тип	Опис
DeviceID	String	Ідентифікатор пристрою

Відповідь

Структура **CashalotApiRetVal**, що містить значення відповіді Description, JsonVal, Return, ShiftID.

Приклад:

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

Приклад відповіді:

```
Ret.Description = ""
Ret.JsonVal = {"Ret":true,"ErrorString":"","Values":{"ShiftID":"364641e3-72c6-4fb7-9eac-32c970066b4c"}}
Ret.Return = True
Ret.ShiftID = "3ecc284d-b536-44c7-9055-b550f43d773a"
```

15. PrintZReport – Формування Z-звіту

Виклик функції **PrintZReport()** призводить до виконання послідовності операцій аналогічних до виклику функції **CloseShift()**: служба 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
DeviceID	String	Ідентифікатор пристрою

Відповідь

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

Приклад:

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

Приклад відповіді:

```
Ret = True;
```

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

встановити параметри

DeviceIDFnRRO

PathToCashalotDir

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 сформуванати 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 по вказаному фіскальному номеру чеку.

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 скриньку, 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

Параметр **Amount** приймає додатне значення.

10. Службова видача грошей з каси

Для виконання службової видачі з каси використовується функція

ServiceOutput

Параметр **Amount** приймає додатне значення.

11. Формування X-звіту

X-звіт формується в результаті виконання функції

PrintXReport

12. 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**.

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.

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.

Порядок роботи обліково-розрахункової системи з API

2. 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="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 ServiceInput( string rroFiscalNumber, string sumVal );
```

Приклад:

```
Object[] parameters = {"111111111111", /* фіскальний 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.

3. 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

4. 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

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

Параметр	Значення	Приклад
VendorCode	Артикул товару	"24563"
Name	Найменування товару	"Бунчук гетьманський, сувенірний"
CdUKDZED	Код УКДЗЕД	"1806903910"
CdDKPP	Код ДКПП	"47.00.69"
GoodsType	Тип товару: 1 - товар; 2 - послуга. 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
OtherParametrs	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="СТЕП Пазл 104 Винни Пух" UKTZED="" VendorCode="82106" Barcode="" Dimension="шт" Discount="1" DiscountAmount="0.56" PriceIncludeVAT="1" AmountVAT="9.24"/></Table>

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

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, картою 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**).

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 НБУ² 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:

² <https://zakon.rada.gov.ua/laws/main/v0025500-18> п.4.

<i>SumInit</i>	<i>SumPayCheck</i>	<i>SumDelta</i>
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.

Специфікація методу:

```

/// <summary>
/// Фіскалізувати чек
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер PPO</param>
/// <param name="strJSONData">
///Параметри з інформацією про строки чеку(товар з характеристиками, кількість
сумма 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,"OtherParams":null},{
"VendorCode":"АртикулТовару2","Name":"НазваТовару2","CdUKDZED":"","CdDKPP":"","GoodsType":"","Barcode":"999999999","UnitType":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":false,"OtherParams":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 rroFiscalNumber, 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 PPO
    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,
    "OtherParameters": 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,
    "OtherParameters": null
  }
]
}

```

```

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

```

Приклад XML-рядка необов'язкових параметрів *OtherParameters*:

```

<?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
КАСИР Лабораторний П. П.


Чек №383875

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

383875 5/28/2021 10:50:45 AM



ФІСКАЛЬНИЙ ЧЕК ОНЛАЙН
Cashalot

Друк

Надіслати на Email

Готово

5. 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 фіскальний номер чека по локальному номеру
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний 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 фіскальний номер чека використовується, зокрема, в якості додаткового параметра при фіскалізації чеку повернення товару.

6. Проведення операції повернення товару

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

Для виконання повернення товару чи товарів з фіскального чека з відомим (визначеним) фіскальним 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	Фіскальний 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 PPO</param>
/// <param name="strJSONData">
/// Параметри з інформацією про строки чеку(товар з характеристиками, кількість
сумма 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":"","Good
sType":"","Barcode":"9999999999","UnitType":"шт","Quantity":"1","Price":"183","Amou
nt":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVA
T":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":nu
ll,"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 );
}
}

```

приклад JSON рядків:

```

strJSONData=
{"ReceiptLst":[{"VendorCode":"АртикулТовару2","Name":"НазваТовару2","CdUKDZED":"","Barcode":"
9999999999","UnitType":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountPrc":"","
DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":false,
"OtherParametrs":null},{VendorCode":"АртикулТовару2","Name":"НазваТовару2","CdUKDZED":"","Ba
rcode":"9999999999","UnitType":"шт","Quantity":"1","Price":"183","Amount":"181,17","DiscountP
rc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise
":false,"OtherParametrs":null}]}}

payDataJSON=
{"SumCash":"366","SumPayByCard":null,"SumPayByCredit":null,"SumPayByCertificate":null,"SumPay
Check":"362,34","NumReturnCheck":"180"}

```

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

7. Створення X-Звіту

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

```

{
    Фіскальний номер РРО
}

```

Специфікація методу:

```

/// <summary>
/// Відобразити X-Звіт(денний звіт без гасіння)
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
/// {"Ret":true,"ErrorString":"","Values":{}}
/// </returns>
public string ShowXReport( string rroFiscalNumber );

```

Приклад:

```

Object[] parameters = {"11111111111111111111"/*фіскальний номер РРО*/};

```

```
var ret = CallCashalotMethod( "ShowXReport", parameters );
```

Результат виконання операції створення X-Звіту відображається на екрані.

8. Визначення локального номеру чека по фіскальному номеру

В деяких випадках виникає необхідність визначення локального номеру чека, зареєстрованого в касі, по відомому фіскальному номеру чека. Для цієї операції використовується метод **GetReceiptLocalNumberByFiscalNumber** з параметрами:

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

Специфікація методу:

```
/// <summary>  
/// Отримати фіскальний номер чека по локальному номеру  
/// </summary>  
/// <param name="rroFiscalNumber">Фіскальний номер PPO</param>  
/// <param name="receiptFiscalNumber">Фіскальний 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(), receiptFiscalNumber};  
var ret = CallCashalotMethod( "GetReceiptLocalNumberByFiscalNumber", parameters );
```

ТОВ "Тест-гідрометцентр"
Тест-гідромет
УКРАЇНА, м. Київ, Адреса
ІД 24234227
ФН ПРРО 4000007619

X - звіт

Продаж

ЗАГАЛЬНА СУМА	38000.10
ГОТІВКА, грн	19000.10
КАРТКА	19000.00
Кількість чеків	1
Об'єкт А	38000.06
ПДВ А=20%	6333.34
ЗАГАЛЬНА СУМА ПОДАТКУ	6333.34

Повернення

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

Службове внесення 100.00
Службова видача 100.00

28.05.2021 10:52:18
СЛУЖБОВИЙ ЧЕК Cashalot

Друк
Готово

9. Отримання XML-структури чеку

Для отримання XML-структури фіскалізованого чеку використовується функція "**GetReceiptXML**" 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 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 );
```

10. 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 );
```

11. 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 = {"11111111111111111111"} /*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

Обіг А	38000.06
ПДВ А=20%	6333.34
ЗАГАЛЬНА СУМА ПОДАТКУ	6333.34
Повернення	
ЗАГАЛЬНА СУМА	0.00
ГОТІВКА, грн	0.00
КАРТКА	0.00
Кількість чеків	0

ЗАГАЛЬНА СУМА ПОДАТКУ	0.00
Службове внесення	100.00
Службова видача	19100.10

РЕГІСТРИ ДЕННИХ ПІДСУМКІВ ОБНУЛЕНІ
ФІСКАЛЬНИЙ ЗВІТНИЙ ЧЕК ДІЙСНИЙ
383880 28.05.2021 10:53:15
ФІСКАЛЬНИЙ ЧЕК Cashalot

Друк

Готово

Приклад реалізації функції для виклику API методів ПРРО Cashalot

```
//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 Api
private Type asmCashalotClass;

//екземпляр класу Api
private object instCashalotApi;

//клас відповіді api 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 API
    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 РРО" );

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

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

    if ( asmCashalotDll != null )
    {
        if( asmCashalotClass == null)
```

```

asmCashalotClass = asmCashalotDll.GetType( "CashalotApiBridge.BridgeApi" );

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

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

```