

Специфікація драйверу ПРРО CASHALOT

Загальна інформація

Драйвер ПРРО CASHALOT використовується для інтеграції ПРРО з розрахунково-обліковими системами.

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.

Поняття і визначення

ДПС - Державна податкова служба.

КЕП – кваліфікований електронний підпис.

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

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

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

Фіскальний номер чека – номер, призначений чеку Сервером ФСКО.

ФСКО - фіскальний сервер контролюючого органу.

Підключення драйверу ПРРО CASHALOT

Підключення ПРРО до розрахунково-облікової системи відбувається за допомогою розробленої бібліотеки драйверу ПРРО CASHALOT. Встановлене ліцензійне програмне забезпечення Cashalot включає в комплект поставки бібліотеку драйверу **CashaLotApiBridge.dll**, шлях до якої необхідно вказати для його коректного налаштування.

Для інтеграції Cashalot в ПРРО за допомогою драйверу ПРРО CASHALOT необхідно створити функції, що будуть транслювати виклик методів із бібліотеки Cashalot (CashaLotApiBridge.dll).

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

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

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

Для використання драйверу ПРРО CASHALOT у користувача повинен бути відкритий доступ до наступних адрес:

- 1) <http://fs.tax.gov.ua> - фіскальний сервер контролюючого органу;
- 2) <https://my.cashalot.org.ua> - кабінет користувача Cashalot.

Команди та порядок взаємодії з драйвером ПРРО CASHALOT

Функціонал ПРРО, який реалізовано в драйвері ПРРО CASHALOT, забезпечується наступними методами:

- [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 драйверу 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 АПІ
    public Dictionary<string, string> Values { get; set; }
}
```

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

Порядок роботи ПРРО з інтегрованим драйвером ПРРО CASHALOT

Після підключення та налаштування драйверу ПРРО готова до роботи.

Перед початком роботи необхідно провести операцію відкриття зміни.

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

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

Деякі методи Cashalot, вказані у розділі [Загальна інформація](#), виконуються неявно під час обробки запитів від ПРРО і можуть повертати відповідні службові повідомлення (перехід у

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 запити і після підтвердження, відправляє відповідне XML повідомлення на Сервер ФСКО.

Cashalot **Новая транзакция**

Службове внесення Службова видача

Касир Сума, грн

Тестовий платник 3
Київ Ромашка
УКРАЇНА, ХАРКІВСЬКА ОБЛ., м. ХАРКІВ КИЇВСЬКИЙ
Р-Н, вул. Громадянська, буд 18
ПН 123456789019
ФН РРРО 4000032790

Службове внесення

Службове внесення 500,00

885493 15.12.2020 14:10:16
СЛУЖБОВИЙ ЧЕК

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 запити і після підтвердження, відправляє відповідне XML повідомлення на Сервер ФСКО.

The screenshot shows a window titled "Cashalot Нова транзакція". It has two radio buttons: "Службове внесення" (unselected) and "Службова видача" (selected). Below the buttons, there are input fields for "Касир" (CASHIER) containing "ТЕСТОВИЙ ПЛАТНИК 3 (ТЕСТОВИ)" and "Сума, грн" (SUM, UAH) containing "100". There is also a "Коментар" (COMMENT) field. At the bottom, there are two buttons: "Далі" (Next) and "Відмінити" (Cancel).

The screenshot shows a receipt with the following text:

Тестовий платник 3
Кіоск Ромашка
УКРАЇНА, ХАРКІВСЬКА ОБЛ., М. ХАРКІВ КИЇВСЬКИЙ
Р-Н, вул. Громадянська, буд 18
ПН 123456789019
ФН ПРРО 4000032790

Службова видача

Службова видача	100,00
-----------------	--------

885606 15.12.2020 14:15:29
СЛУЖБОВИЙ ЧЕК

Buttons: "Друкувати звіт" (Print report) and "Готово" (Done).

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	
UnitType	одиниця виміру. За замовчуванням: "шт"	"шт"
Quantity	кількість	"1"
Price	ціна (за одиницю товару)	"2000.00"
Amount	вартість (сумарна, за всі одиниці товару)	"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
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.	<?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>

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 оплати готівкою	"1950.00"
SumPayByCard	Сума безготівкової оплати	"0.00"
SumPayByCredit	Сума оплати в кредит	"0.00"
SumPayByCertificate	Сума оплати сертифікатом, картою знижок та ін.	"0.00"
SumPayCheck	Загальна сума до сплати з урахуванням всіх видів оплати та знижок	"1900.00"

для якої значення також мають бути представлені у типі ([string](#)).

Специфікація методу:

```
/// <summary>
/// Фіскалізувати чек
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="strJSONData">
/// Параметри з інформацією про строки чеку(товар з характеристиками, кількість сумми
/// і тд.), у вигляді JSON строки
/// Приклад:
/// {"ReceiptLst":[{"VendorCode":"АртикулТовару1","Name":"НазваТовару1","CdUKDZED":"","
/// CdDKPP":"","GoodsType":"","Barcode":"123456789","UnitType":"шт","Quantity":"1","Pr
/// ice":"183","Amount":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","I
/// sPriceIncludeVAT":true,"SumVAT":"30.2","IsExcise":false,"OtherParams":null},{
/// VendorCode":"АртикулТовару2","Name":"НазваТовару2","CdUKDZED":"","CdDKPP":"","GoodsType
///":"","Barcode":"999999999","UnitType":"шт","Quantity":"1","Price":"183","Amount":"1
/// 81,17","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,"SumPayCheck":"362,34"}
/// </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 = "999999999";
lstGoodsInCheck.ReceiptLst.Add( chInfo );//додати рядок чека з товаром №2

PayInfoRsp payInfo = new PayInfoRsp();
payInfo.SumCash = "366"; //Сума до сплати без урахування знижки
payInfo.SumPayCheck = "362,34"; //Загальна сума до сплати з урахуванням всіх
видів оплати та знижок
//Конвертація структур до формату JSON рядка
string strJSONData = JsonConvert.SerializeObject( lstGoodsInCheck );
string payDataJSON = JsonConvert.SerializeObject( payInfo );
//Параметри функції
Object[] parameters =
{
    edFiscalNumberRR0.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": "",
      "Barcode": "123456789",
      "UnitType": "шт",
      "Quantity": "1",
      "Price": "183",
      "Amount": "181,17",
      "DiscountPrc": "",
      "DiscountSum": "1.83",
      "VATRate": "20",
      "IsPriceIncludeVAT": true,
      "SumVAT": "30.2",
      "IsExcise": null,
      "OtherParameters": null
    },
    {
      "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": null,
      "OtherParameters": null
    }
  ]
}

payDataJSON=
{
  "SumCash": "366",
  "SumPayByCard": null,
  "SumPayByCredit": null,
```

```
"SumPayByCertificate":null,  
"SumPayCheck": "362,34"  
}
```

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 Кешалот
ТОВ Кешалот
УКРАЇНА, м. Київ, вул. Новосолоньківська 9
ПІН 242342255555
ФІН ПРРО 4000007015
КАСІР Лабіринтський П. П.
Чек № 203516
Чайник BINATONE AEJ-1001, 2.2л
1 X 1140,00 1140,00 А
Знижка -57,00

Сума, грн 1083,00
ПДВ А=20% 180,50

ГОТІВКА, грн 1083,00
203516 12/15/2020 2:21:26 PM

ФІСКАЛЬНИЙ ЧЕК ОНЛАЙН
Cashalot
Друкувати звіт
Готово

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.

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 PPO ,  
    Локальний номер чеку  
}
```

Специфікація методу:

```
/// <summary>  
/// Отримати фіскальний номер чека по локальному номеру  
/// </summary>  
/// <param name="rroFiscalNumber">Фіскальний номер PPO</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 в якості додаткового параметра при фіскалізації чеку повернення товару.

Повернення товару

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

В JSON рядку **payDataJSON** додатково зазначається параметр

NumReturnCheck	Фіскальний номер чека по якому здійснюється повернення
----------------	---

Специфікація методу:

```

/// <summary>
/// Фіскалізувати видатковий чек(повернення)
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</param>
/// <param name="strJSONData">
///Праметри з інформацією про строки чеку(товар з характеристиками, кількість сума
і тд.), у вигляді JSON строки
///Приклад:
///{"ReceiptLst":[{"VendorCode":"АртикулТовару1","Name":"НазваТовару1","CdUK
DZED":"","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":"999999999","UnitType":"шт","Quantity":"1","Price":"183
","Amount":"181,17","DiscountPrc":"","DiscountSum":"1.83","VATRate":"20","IsPriceIn
cludeVAT":true,"SumVAT":"30.2","IsExcise":false,"OtherParametrs":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="strLocalNumReturnCheck">Номер чеку по якому здійснюється
повернення</param>
/// <returns>
/// Функція повертає структуру у вигляді JSON
/// Приклад:
///
{"Ret":true,"ErrorString":"","Values":{"ReceiptFiscalNumber":"203350","ReceiptLocal
Number":"1141"}}
/// </returns>
public string FiscalizeReturnCheck( string rroFiscalNumber, string strJSONData,
string payDataJSON, String strLocalNumReturnCheck);

```

Приклад:

```

Object[] parameters = { edFiscalNumberRRO.Text.Trim(), fiscNum };
var ret = CallCashaLotMethod( "GetReceiptLocalNumberByFiscalNumber", parameters );
if ( !ret.Ret )
{
    MessageBox.Show( String.IsNullOrEmpty( ret.ErrorString ) ? "Не вдалося отримати
локальний номер чека" : ret.ErrorString );
}
else
{
    var vals = ret.Values.ToArray();
    for ( int i = 0; i < vals.Length; i++ )
    {
        if ( vals[i].Key.ToLower() == "receiptlocalnumber" )
        {
            edLocalCheckNum.Text = vals[i].Value;
        }
    }
}
// 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,
        eckNum.Text/*Локальний 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":"","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}]}

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

```

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

Створення X-Звіту

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

```

{
    Фіскальний номер РРО
}

```

Специфікація методу:

ТОБ Тернопіль
ТОБ Тернопіль

УКРАЇНА, м. Київ, вул. Незалежності № 8
ПІН 242342255555
ФН ПРРО 4000007015

X - звіт

Продаж

ЗАГАЛЬНА СУМА	1083.00
ГОТІВКА, грн	1083.00
КАРТКА	0.00
Кількість чеків	1

Обіг А	1083.00
ОБІГ	1083.00
ПДВ А=20%	180.50
ЗАГАЛЬНА СУМА ПОДАТКУ	180.50

Повернення

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

ОБІГ	0.00
ЗАГАЛЬНА СУМА ПОДАТКУ	0.00

Службове внесення 0.00
Службова видача 0.00

15.12.2020 14:22:23
СЛУЖБОВИЙ ЧЕК

Друкувати звіт

Готово

Закриття зміни та створення Z-Звіту

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

```
{
  Фіскальний номер РРО
}
```

Специфікація методу:

```
/// <summary>
/// Закриття зміни
/// </summary>
/// <param name="rroFiscalNumber">Фіскальний номер РРО</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 РРО*/};
```



```
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 Чешалот
ТОВ Чешалот
УКРАЇНА, м. Київ, вул. Пилипетівська, 8
ПН 242342255555
ФН ПРРО 4000007015

Z - звіт

Продаж

ЗАГАЛЬНА СУМА	1083.00
ГОТІВКА, грн	1083.00
КАРТКА	0.00
Кількість чеків	1

Обіг А	1083.00
ОБІГ	1083.00
ПДВ А=20%	180.50
ЗАГАЛЬНА СУМА ПОДАТКУ	180.50

Повернення

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

ОБІГ	0.00
ЗАГАЛЬНА СУМА ПОДАТКУ	0.00

Службове внесення 500.00
Службова видача 1583.00

РЕГІСТРИ ДЕННИХ ПІДСУМКІВ ОБІГІВ
ФІСКАЛЬНИЙ ЗВІТНИЙ ЧЕК ДІЙСНИЙ
203526 15.12.2020 14:24:56
ФІСКАЛЬНИЙ ЧЕК

Друкувати звіт

Готово

Приклад реалізації функції для виклику методів драйверу ПРРО CASHALOT

```
//шлях до каталогу ПРРО Cashalot  
public string edCashaLotPath = "C:\\ProgramData\\Cashalot\\Cashalot";  
  
//Фіскальний номер РРО  
public string edFiscalNumberRRO = "111111111111";  
  
//завантажена бібліотека  
private Assembly asmCashaLotD11;
```

```

//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 АПІ
    public Dictionary<string, string> Values { get; set; }
}

/// <summary>
/// Функція для виклику метода з АПІ Cashalot
/// </summary>
/// <param name="methodName"> Назва методу </param>
/// <param name="parameters"> Список параметрів методу</param>
/// <returns></returns>
public CashalotApiRsp CallCashalotMethod(string methodName, Object[] parameters )
{
    CashalotApiRsp ret = new CashalotApiRsp();

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

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