Система реєстрації електронних касових чеків "E-Receipt" ("Е-Чек")

Порядок обміну повідомленнями

# Поняття і визначення

**Клієнт** – Система виписки електронних касових чеків, що використовується продавцем.

**Сервер** – Сервер реєстрації касових чеків.

**Локальний номер чека** – Номер, призначений чеку системою продавця.

**Фіскальний номер чека** – Номер, призначений чеку сервером.

# Типи фіскальних чеків

1. Реалізація товару
2. Повернення товару
3. Відкриття зміни
4. Закриття зміни
5. Одержання коштів
6. Здавання коштів

# Порядок локальної нумерації документів

Локальна нумерація документів ведеться окремо кожним клієнтом. Нумерація здійснюється в межах:

* юридичної/фізичної особи продавця;
* господарської одиниці;
* номера каси.

Локальна нумерація документів ведеться наскрізь, незалежно від типу документа.

# Порядок фіскальної нумерації документів

Фіскальна нумерація документів ведеться сервером. Нумерація здійснюється в межах:

* юридичної/фізичної особи продавця;
* господарської одиниці;
* номера каси.

Фіскальна нумерація документів ведеться наскрізь, незалежно від типу документа.

# Загальний порядок роботи

1. Відкриття зміни;
2. Реєстрація фіскальних касових чеків;
3. Реєстрація Z-звіту і закриття зміни.

# Відкриття зміни

1. Клієнт створює XML документ чека із призначеним локальним номером і типом «Відкриття зміни», засвідчує його ЕЦП продавця і надсилає на сервер.
2. Сервер здійснює необхідні перевірки.
3. У разі помилки, сервер надсилає у відповідь XML квитанції.
4. У разі успішної обробки, сервер зберігає документ в БД і надсилає у відповідь XML квитанції щодо успішної реєстрації документа.
5. Сервер встановлює для каси стан «зміну відкрито».

# Реєстрація фіскального касового чека

1. Клієнт створює XML документ чека із призначеним локальним номером, засвідчує його ЕЦП продавця і надсилає на сервер.
2. Сервер здійснює необхідні перевірки.
3. У разі помилки, сервер надсилає у відповідь XML квитанції.
4. У разі успішної обробки, сервер зберігає документ в БД, призначає документу фіскальний номер і надсилає у відповідь XML квитанції щодо успішної реєстрації документа, що містить фіскальний номер.

# Реєстрація Z-звіту і закриття зміни

1. Клієнт створює XML документ Z-звіту із призначеним локальним номером, засвідчує його ЕЦП продавця і надсилає на сервер.
2. Сервер здійснює необхідні перевірки.
3. У разі помилки, сервер надсилає у відповідь XML квитанції.
4. У разі успішної обробки, сервер зберігає документ в БД, і надсилає у відповідь XML квитанції щодо успішної реєстрації документа.
5. Клієнт реєструє фіскальний касовий чек із типом «Закриття зміни».
6. Сервер встановлює для каси стан «зміну закрито».

# Зауваження щодо реалізації

1. Кодування текстових полів документів XML: Windows-1251.
2. Тип HTTP-запиту – POST.
3. Заголовок “Content-Type”: “application/octet-stream”.
4. Використання компресії Gnu Zip (GZip) або Zip (deflate). Вихідні документи оброблюються алгоритмом GZip або Zip для зменшення розміру повідомлення.
5. У разі використання компресії, заголовок “Content-Encoding” вхідного повідомлення повинний мати значення “gzip” або “deflate”.
6. Повідомлення містить XML документ, засвідчений ЕЦП відправника.
7. Вхідне повідомлення може містити серйозні помилки, що не дозволяють його подальшу обробку. Наприклад, повідомлення не являє собою підписаний блок або містить більше ніж одну ЕЦП. У випадку грубого порушення вимог до формату вхідного повідомлення, у відповідь надсилається код статусу обробки HTTP 400 “Bad Request”. У тілі відповіді надається інформація щодо причини помилки. Наприклад, “Content-Type must be 'application/octet-stream'”, “Message not contains signed data” і таке інше.
8. Сервер оброблює Url:
   * “<адреса>/doc“ – одержання документів (чеків, Z-звітів тощо)
   * “<адреса>/cmd“ – одержання команд
9. Команди надсилаються у форматі JSON. Див. розділ «Команди».

## Фіскальний чек. Типи документів

Елемент “DOCTYPE”

Значення:

0 - Фіскальний чек;

1 - Відкриття зміни;

2 - Закриття зміни.

## Коди помилок

/// <summary>

///     Коди помилок

/// </summary>

public enum ErrorCode {

    /// <summary>

    ///     OK

    /// </summary>

**OK** = 0,

    /// <summary>

    ///     РРО не зареєстрований

    /// </summary>

**CashRegisterAbsent** = 1,

    /// <summary>

    ///     Відсутній доступ до РРО для оператора

    /// </summary>

**OperatorAccessToCashRegisterNotGranted** = 2,

    /// <summary>

    ///     В документі зазначено реєстраційний код платника, що не дорівнює реєстраційному коду господарської одиниці

    /// </summary>

**InvalidTin** = 3,

    /// <summary>

    ///     Зміну для РРО наразі відкрито

    /// </summary>

**ShiftAlreadyOpened** = 4,

    /// <summary>

    ///     Зміну для РРО наразі не відкрито

    /// </summary>

**ShiftNotOpened** = 5,

    /// <summary>

    ///     Останній документ, зареєстрований перед закриттям зміни, повинний бути Z-звітом

    /// </summary>

**LastDocumentMustBeZForm** = 6,

    /// <summary>

    ///     Некоректний локальний номер чека

    /// </summary>

**CheckLocalNumberInvalid** = 7

}

## Команди

### Запит доступних об'єктів

Запит повинний містити JSON, підписаний ЕЦП оператора.

Формат JSON запиту:

{

    Command = “**Objects”**

}

У відповідь повертається перелік доступних оператору господарських одиниць і РРО.

Формат JSON відповіді:

internal class ResponseObjects {

    public TaxObjectItem[] TaxObjects { get; set; }

}

public class TaxObjectItem {

    public string Address { get; set; }

    public string Guid { get; set; }

    public string Name { get; set; }

    public long Tin { get; set; }

    public CashRegisterItem[] CashRegisters { get; set; }

}

public class CashRegisterItem {

    public long NumFiscal { get; set; }

    public int NumLocal { get; set; }

}

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

### Запит стану РРО

Запит повинний містити JSON, підписаний ЕЦП оператора.

Формат JSON запиту:

{

    Command = “CashRegisterState**”,**

NumFiscal = <Фіскальний номер РРО>

}

Формат JSON відповіді:

{

State = <0-зміну не відкрито, 1-зміну відкрито>,

ShiftId = <Ідентифікатор зміни>,

Name = <П.І.Б. оператора, що відкрив зміну>,

IssuerId = <Ідентифікатор видавця сертифікату оператора>,

Serial = <Серійний номер сертифікату оператора>,

NextLocalNum = <Наступний внутрішній номер документу>

}

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

### Запит чека

Запит повинний містити JSON.

Формат JSON запиту:

{

    Command = “**Check”,**

NumFiscal = <Фіскальний номер чека>

}

У відповідь надається XML документ чека, засвідчений ЕЦП сервера.

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

### Запит Z-звіту

Запит повинний містити JSON, підписаний ЕЦП оператора.

Формат JSON запиту:

{

    Command = “**ZForm”,**

NumFiscal = <Фіскальний номер Z-звіту>

}

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

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

### Запит переліку змін за період

Запит повинний містити JSON, підписаний ЕЦП оператора.

Формат JSON запиту:

{

    Command = “**Shifts”,**

NumFiscal = <Фіскальний номер РРО>,

From = <Дата і час початку періоду>,

To = <Дата і час завершення періоду>

}

Дата і час представлені текстом у форматі ISO 8601.

Формат JSON відповіді:

{

    ShiftId = <Ідентифікатор зміни>,

Opened = <Дата і час відкриття зміни>,

OpenName = <П.І.Б. оператора, що відкрив зміну>,

OpenIssuerId = <Ідентифікатор видавця сертифікату оператора>,

OpenSerial = <Серійний номер сертифікату оператора>,

Closed = <Дата і час закриття зміни>,

CloseName = <П.І.Б. оператора, що закрив зміну>,

CloseIssuerId = <Ідентифікатор видавця сертифікату оператора>,

CloseSerial = <Серійний номер сертифікату оператора>

}

### Запит переліку документів зміни

Запит повинний містити JSON, підписаний ЕЦП оператора.

Формат JSON запиту:

{

    Command = “**Documents”,**

    ShiftId = <Ідентифікатор зміни>

}

Формат JSON відповіді:

{

    ShiftId = <Ідентифікатор зміни>,

NumFiscal = <Фіскальний номер документа>,

DocType = <Тип документа (“**Check**”, “**ZForm**”)>,

CheckDocType = <Тип чека (“**Check**”, “**OpenShift**”, “**CloseShift**”)>

}

# Адреса тестового сервера

<http://217.76.198.145/er>