

## **StopPark API**

**Version 1.2**

**2021-06**

**Київ  
2021**

## Зміст

<b>Функції.....</b>	<b>3</b>
• <b>ManDataDB (StopParkAPI.dll).....</b>	<b>3</b>
• <b>ConvertMifare (StopParkAPI.dll).....</b>	<b>4</b>
• <b>CalculationTicket (StopParkAPI.dll).....</b>	<b>5</b>
• <b>CalculationCart (StopParkAPI.dll).....</b>	<b>6</b>
• <b>ConfirmationPayment (StopParkAPI.dll).....</b>	<b>7</b>
• <b>CheckTicket (StopParkAPI.dll).....</b>	<b>8</b>
<b>Опис таблиць StopPark PRO v1.3 .....</b>	<b>9</b>
<b>Налаштування API StopParkAPI.dll.....</b>	<b>11</b>
<b>Взаємодія зі зчитувачами CS-03M-Park або УП-ММ-01USB .....</b>	<b>11</b>
<b>Додаток 1.....</b>	<b>12</b>
<b>Додаток 2.....</b>	<b>13</b>
<b>Додаток 3.....</b>	<b>14</b>
<b>Додаток 4.....</b>	<b>15</b>
<b>Додаток 5.....</b>	<b>16</b>
<b>Додаток 6.....</b>	<b>17</b>
<b>Додаток 7.....</b>	<b>18</b>

**StopParkAPI.dll** є програмним інтерфейсом для взаємодії з системою StopPark PRO v1.3, **Mifare.dll** є залежністю для програмного інтерфейсу **StopParkAPI.dll**

Для коректної роботи ці два файли (**StopParkAPI.dll**, **Mifare.dll**) повинні знаходитись в одній директорії.

### Функції

- **ManDataDB (StopParkAPI.dll)**

ManDataDB	
Опис	Функція використовується для взаємодії з базою даних StopPark
Вхідні дані	Стандартний SQL запит що відповідає структурі БД приведений в додатку 1. Запит в функцію надається в форматі string.
Обмеження	В запиті заборонено використання : "*", "create", "alter", "drop";
Приклади вхідних даних	
1) INSERT INTO Ticket(Ticket, BAR, TimeIn, TimeCount, Status) VALUES('Ticlet', '{barcode}', '{time_in}', '{time_in}', 1)" 2) SELECT TimeIn, TimeCount FROM Ticket WHERE BAR = {barcode}	
Приклади вихідних даних	
1) None 2) 20-11-27 09:24:30, 20-11-27 09:24:30	

**Приклад** програми для тестування бібліотеки приведений в додатку 2.

- **ConvertMifare (StopParkAPI.dll)**

Призначений для роботи зі зчитувачами CS-03M-Park або УП-MM-01USB в складі системи StopPark з метою коректної передачі UID карток Mifare в систему StopPark.

ConvertMifare	
Опис	Функція використовується для конвертування UID Mifare карток для роботи з StopPark
Вхідні дані	String UID довжиною 4 або 7 байт.
Обмеження	String повинен включати лише UID картки.
Приклади вхідних даних	
HEX : 3E 30 8F FD	
Приклади вихідних даних	
String : 00E728E211	

Приклад програми для тестування бібліотеки приведений в додатку 3.

- **CalculationTicket (StopParkAPI.dll)**

CalculationTicket	
Опис	Функція використовується для розрахунку вартості погодинного паркування за допомогою штрихкоду талону.
Вхідні дані	String штрихкод талону.
Обмеження	String повинен включати лише штрихкод.
Приклади вхідних даних	
String : 100818220918000049	
Приклади вихідних даних	
PayData	

Приклад програми для тестування бібліотеки приведений в додатку 4.

#### Опис структури PayData

- double price – поле в якому зберігається сума оплати;
- int hour – поле в якому зберігається кількість годин паркування, якщо використовується метод **CalculationCart** то буде порожньою
- int mounth - поле в якому зберігається кількість місяців абонементу, якщо використовується метод **CalculationTicket** то буде порожньою

- **CalculationCart (StopParkAPI.dll)**

CalculationCart	
Опис	Функція використовується для розрахунку вартості погодинного паркування за допомогою UID картки довжиною 4 або 7 байт.
Вхідні дані	String UID довжиною 4 або 7 байт.
Обмеження	String повинен включати лише UID картки.
Приклади вхідних даних	
HEX : 3E 30 8F FD	
Приклади вихідних даних	
PayData	

**Приклад** програми для тестування бібліотеки приведений в додатку 5.

#### Опис структури PayData

- double price – поле в якому зберігається сума оплати;
- int hour – поле в якому зберігається кількість годин паркування, якщо використовується метод **CalculationCart** то буде порожньою
- int mounth - поле в якому зберігається кількість місяців абонементу, якщо використовується метод **CalculationTicket** то буде порожньою

- **ConfirmationPayment (StopParkAPI.dll)**

<b>ConfirmationPayment</b>	
Опис	Функція використовується для підтвердження факту оплати як для талонів так і для карток.
Вхідні дані	String носій унікального ідентифікатора(картка або талон), String ім'я оператора, Double сума оплати, Int кількість одиниць.
Обмеження	
Приклади вхідних даних	
("34F435324", "Test", 234.34, 23)	
Приклади вихідних даних	
String None (в разі успішного виконання)	

**Приклад** програми для тестування бібліотеки приведений в додатку 6.

#### Опис структури PayData

- double price – поле в якому зберігається сума оплати;
- int hour – поле в якому зберігається кількість годин паркування, якщо використовується метод **CalculationCart** то буде порожньою
- int mounth - поле в якому зберігається кількість місяців абонементу, якщо використовується метод **CalculationTicket** то буде порожньою

- **CheckTicket (StopParkAPI.dll)**

<b>CheckTicket</b>	
Опис	Функція використовується для перевірки валідності талону.
Вхідні дані	String носій унікального ідентифікатора(талону).
Обмеження	
Приклади вхідних даних	
("573472863274000033")	
Приклади вихідних даних	
Ticket.dateTime, Ticket.ticketType (в разі успішного виконання)	

**Приклад** програми для тестування бібліотеки приведений в додатку 7.

#### Опис структури Ticket

- string dateTime – поле в якому зберігається дата в'їзду;
- TicketType ticketType – поле в якому зберігається тип талону.

#### Опис перерахування TicketType

- validTicket – талон валідний та є в базі даних ПО-04;
- validTicketAndNotInTheDatabase – талон валідний, але його немає в базі ПО-04;
- notValidTicket – талон не валідний;
- exception – виключна ситуація.



## Опис таблиць StopPark PRO v1.3

Опис даних що містяться в таблицях зображених в додатку 1.

### Events :

- 1) Events: текстове поле, має містити слово «Events»
- 2) Id: містить порядковий id запису
- 3) EventName: містить дію що була виконана «відкриття» або «проїзд»
- 4) DateTime: містить час створення запису
- 5) Terminal: містить ID стійки на якій була виконана дія
- 6) Direction: містить напрямок руху при події «проїзд», може містити «В'їзд» або «Виїзд»
- 7) Reason: містить причину «відкриття», може бути «автоматично» або «вручну»
- 8) FreePlaces: містить кількість вільних місць на момент події
- 9) Card: містить номер картки або квитка
- 10) GosNom: містить номер авто, якщо підключена ALPR камера

### Ticket :

- 1) Ticket: текстове поле, має містити слово «Ticket»
- 2) ID: містить порядковий id запису
- 3) BAR: містить номер квитка
- 4) TypeTarif: містить ID тарифу за яким був оплачений
- 5) PriceTarif: містить вартість за одиницю тарифу
- 6) Summ: містить суму оплати за паркування
- 7) SummDopl: містить суму в разі доплати
- 8) TimeIn: містить час в'їзду на паркування
- 9) TimeOut: містить час виїзду з паркування
- 10) TimeCount: містить час оплати квитка
- 11) TimeDopl: містить час доплати квитка
- 12) Status: містить статус квитка може бути: 1(в'їхав), 5(оплачено), 10(виїхав)

### Payment:

- 1) Id: містить порядковий id запису
- 2) Payment: текстове поле, має містити слово «Talon payment»
- 3) Type:
- 4) Kassa: містить id каси оплати
- 5) Operator: містить Ім'я та Прізвище оператора який провів оплату
- 6) DTime: містить час оплати
- 7) TalonID: містить номер картки або квитка
- 8) Status: містить «5»(оплачено)
- 9) TarifType: містить id тарифа за яким проводиться оплата
- 10) Tarif: містить вартість одного періоду тарифу
- 11) TarifKol: містить кількість одиниць до оплати
- 12) DTIn: містить час в'їзду на паркування

- 13) DTOut: містить час виїзду з паркування
- 14) Summa: містить суму оплати за паркування

**Card:**

- 1) Card: текстове поле, має містити слово «Card»
- 2) ID: містить порядковий id запису
- 3) Type: містить тип картки
- 4) CardID: містить номер картки
- 5) DTReg: містить дату реєстрації картки
- 6) DTEnd: містить дату закінчення картки
- 7) DTIn: містить час в'їзду на паркування
- 8) DTOut: містить час виїзду з паркування
- 9) DriveName: містить Ім'я водія
- 10) DriveSName: містить По-батькові водія
- 11) DriveFam: містить Прізвище водія
- 12) DrivePhone: містить номер телефону водія
- 13) CarGosNom: містить номер авто водія
- 14) CarModel: містить модель авто водія
- 15) CarColor: містить колір авто водія
- 16) Status: містить статус картки водія

**config\_new:**

- 1) work\_start (INT) - ; поле яке відповідає за час початку погодинної тарифікації в змінному тарифі
- 2) work\_end (INT) - поле яке відповідає за час закінчення погодинної тарифікації в змінному тарифі
- 3) cost\_weekend (INT) – Встановлення ціни на паркування в вихідні дні;
- 4) FREE\_TIME (INT) - поле яке відповідає за фрітайм;
- 5) cam\_ver (INT) – поле для встановлення режиму роботи з камерами (0 – вимкнені, затримки при друці немає; 1- робота з камерами НомерОК; 2- робота з камерами TVT)
- 6) cost\_hour (INT) - Встановлення ціни за годину паркування в змінному тарифі;
- 7) APB (INT) – Встановлення режиму AntiPassBack для проїзду камерами (0 – вимкнено; 1 – ввімкнено) ;
- 8) INTERVAL (INT) - поле яке відповідає за інтервал часу для виїзду.

## Налаштування API StopParkAPI.dll

Перед використанням функції ManDataDB наведеної вище необхідно виконати налаштування API для роботи з базою даних пульта оператора StopPark. Для цього необхідно використати наведений нижче код.

```
API aPI = new API();
//setting the parking ip address
aPI.Address = "192.168.31.249";
//setting the parking port
aPI.Port = 101;
```

А саме створити об'єкт, та задати змінним значення, а саме aPI.Address повинен містити IP адресу пульта оператора StopPark в локальній мережі(!!!), aPI.Port повинен містити значення порту пульта оператора StopPark, значення цього порту 101 і не змінюється вручну, дане значення може змінитися лише при використанні Port Forwarding в локальній мережі.

Після виконання даних дій можливе коректне використання функції ManDataDB для взаємодії з базою даних StopPark.

## Взаємодія зі зчитувачами CS-03M-Park або УП-ММ-01USB

Зчитувач УП-ММ-01USB після встановлення драйверів в системі відображається в якості віртуального послідовного порту. CS-03M-Park же має можливість лише фізичного під'єднання через порт RS232.

Обидва зчитувача мають однакові конфігурації для використання. Сам послідовний порт повинен конфігурований за наступними параметрами:

Швидкість обміну даними .....	9600 бод
Біти даних .....	8
Стоп біт .....	1
Парність .....	Ні
Керування потоком.....	Ні

Для зчитування UID карти необхідно на зчитувач передати символ «n». Після чого якщо карта знаходиться в межах поля зчитувача буде переданий UID картки у вигляді ;0492387A662180? або ;3E308FFD?.

## Додаток 1

Events			
	Event	text	
	id	integer	
	EventName	text	
	DateTime	text	
	Terminal	integer	
	Direction	text	
	Reason	text	
	FreePlaces	integer	
	Card	text	
	GosNom	text	
Add field			

Ticket			
	Ticket	text	
	ID	integer	
	BAR	text	
	TypeTarif	integer	
	PriceTarif	integer	
	Summ	integer	
	SummDopl	integer	
	Timeln	text	
	TimeOut	text	
	TimeCount	text	
	TimeDopl	text	
	Status	integer	
Add field			

Card			
	Card	text	
	ID	integer	
	Type	integer	
	CardID	text	
	DTReg	text	
	DTEnd	text	
	DTIn	text	
	DTOut	text	
	DriveName	text	
	DriveSName	text	
	DriveFam	text	
	DrivePhone	text	
	CarGosNom	text	
	CarModel	text	
	CarColor	text	
	Status	integer	
	TarifType	integer	
	TarifPrice	integer	
	TarifSumm	integer	
Add field			

Payment			
	id	integer	
	Payment	text	
	Type	integer	
	Kassa	integer	
	Operator	text	
	DTime	text	
	TalonID	text	
	Status	integer	
	TarifType	integer	
	Tarif	integer	
	TarifKol	integer	
	DTIn	text	
	DTOut	text	
	Summa	integer	
Add field			

	Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Значение по умолчанию
1	work_start	INTEGER							NULL
2	work_end	INTEGER							NULL
3	cost_night	INTEGER							NULL
4	cost_weekend	INTEGER							NULL
5	FREE_TIME	INTEGER							NULL
6	cam_ver	INTEGER							NULL
7	cost_hour	INTEGER							NULL
8	APB	INTEGER							NULL
9	INTERVAL	INTEGER							NULL

## Додаток 2

Демо ПЗ для тестування бібліотеки ManDataDB:

```
using System;
using StopParkAPI;
using System.IO.Ports;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            Program pr = new Program();
            API aPI = new API();

            aPI.Address = "192.168.31.249";
            aPI.Port = 101;
            var response = aPI.ManDataDB("Select TimeIn From Ticket");
            Console.WriteLine(response);
            Console.ReadLine();
        }
    }
}
```

### Додаток 3

Демо ПЗ для тестування бібліотеки ConvertMifare:

```
using System;
using StopParkAPI;
using System.IO.Ports;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            API aPI = new API();

            //setting the parking ip address
            aPI.Address = "192.168.31.249";
            //setting the parking port
            aPI.Port = 101;
            //executing a sql command and getting the result
            var response = aPI.ManDataDB("Select TimeIn From Ticket");
            Console.WriteLine(response);

            string uid1 = "0492387A662180";
            string uid2 = "0437117A662180";
            string uid3 = "044F75B2792280";
            string uid4 = "047F7C12374E80";
            string uid5 = "00000016D86912";

            //convert uid card mifare
            var res = aPI.ConvertMifare(uid5);
            Console.WriteLine(res);
            Console.ReadLine();
        }
    }
}
```

## Додаток 4

Демо ПЗ для тестування бібліотеки CalculationTicket:

```
using System;
using StopParkAPI;
using System.IO.Ports;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            API aPI = new API();

            //setting the parking ip address
            aPI.Address = "192.168.31.249";
            //setting the parking port
            aPI.Port = 101;
            //executing a sql command and getting the result
            var response = aPI.ManDataDB("Select TimeIn From Ticket");
            //Console.WriteLine(response);

            string uid1 = "0492387A662180";
            string uid2 = "0000003E308FFD";
            string uid3 = "0000004754E570";
            string uid4 = "0000007714D770";

            var res = aPI.CalculationTicket ("100818220918000049");

            aPI.ConfirmationPayment("100818220918000049", "Test", res.price,
res.hour);
            //convert uid card mifare
            var res = aPI.ConvertMifare(uid3);
            Console.WriteLine(res);
            Console.ReadLine();
        }
    }
}
```

## Додаток 5

Демо ПЗ для тестування бібліотеки CalculationCart:

```
using System;
using StopParkAPI;
using System.IO.Ports;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            API aPI = new API();

            //setting the parking ip address
            aPI.Address = "192.168.31.249";
            //setting the parking port
            aPI.Port = 101;
            //executing a sql command and getting the result
            var response = aPI.ManDataDB("Select TimeIn From Ticket");
            //Console.WriteLine(response);

            string uid1 = "0492387A662180";
            string uid2 = "0000003E308FFD";
            string uid3 = "0000004754E570";
            string uid4 = "0000007714D770";

            var res = aPI.CalculationCart("34F435324");

            //convert uid card mifare
            var res = aPI.ConvertMifare(uid3);
            Console.WriteLine(res);
            Console.ReadLine();
        }
    }
}
```



## Додаток 6

Демо ПЗ для тестування бібліотеки ConfirmationPayment:

```
using System;
using StopParkAPI;
using System.IO.Ports;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            API aPI = new API();

            //setting the parking ip address
            aPI.Address = "192.168.31.249";
            //setting the parking port
            aPI.Port = 101;
            //executing a sql command and getting the result
            var response = aPI.ManDataDB("Select TimeIn From Ticket");
            //Console.WriteLine(response);

            string uid1 = "0492387A662180";
            string uid2 = "0000003E308FFD";
            string uid3 = "0000004754E570";
            string uid4 = "0000007714D770";

            var res = aPI.CalculationCart("34F435324");

            aPI.ConfirmationPayment("34F435324", "Test", res.price,
res.mounth);
            //convert uid card mifare
            var res = aPI.ConvertMifare(uid3);
            Console.WriteLine(res);
            Console.ReadLine();
        }
    }
}
```

## Додаток 7

Демо ПЗ для тестування бібліотеки CheckTicket:

```
using System;
using StopParkAPI;
using System.IO.Ports;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            API aPI = new API();

            //setting the parking ip address
            aPI.Address = "192.168.31.249";
            //setting the parking port
            aPI.Port = 101;
            //executing a sql command and getting the result
            var response = aPI.ManDataDB("Select TimeIn From Ticket");
            //Console.WriteLine(response);

            var res = aPI.CheckTicket("737775870386000036");
            Console.WriteLine(res.dateTime);
            Console.WriteLine(res.ticketType);
            Console.ReadLine();
        }
    }
}
```