



## شرکت کارت اعتباری ایران کیش

نام مستند: مستند راهنمای استفاده از سامانه‌ی واسط پایانه‌های PCPOS

شماره مستند:

سطح محرمانگی: عادی

محل نگهداری: شرکت کارت اعتباری ایران کیش

تعداد صفحات: 26

تاریخ آخرین اصلاح: 1395/05/18

شماره بازنگری: 7

نام فایل: DEVELOPERSGUIDE

تهیه کننده: اداره تحلیل و مستندسازی    تأیید کننده: اداره تولید نرم افزار    تصویب کننده: معاون فناوری اطلاعات

امضاء:

امضاء:

امضاء:

تاریخ:

تاریخ:

تاریخ:

این مستند متعلق به شرکت کارت اعتباری ایران کیش است و هیچ کس بدون اجازه کتبی حق انتشار، کپی برداری و افشای آن را ندارد.

این مستند متعلق به شرکت کارت اعتباری ایران کیش است و هیچ کس بدون اجازه کتبی حق انتشار، کپی برداری و افشای آن را ندارد.



## فهرست مطالب

1	مقدمه	5.....
2	نیازمندی ها	5.....
3	قابلیت ها و ویژگی ها	5.....
4	نحوه انجام عملیات	7.....
4.1	کلاس SERIALINGENICO	7.....
4.1.1	متد InitiateService	7.....
4.1.2	متد Sale	9.....
4.1.3	متد SaleWithPaymentId	9.....
4.1.4	متد SpecialPayment	10.....
4.1.5	متد CreditRemaining	10.....
4.1.6	متد MultiplexPayment و متد MultiplexPaymentWithPaymentId	10.....
4.1.7	متد BilPayment	12.....
4.1.8	پرداخت چند حسابی با استفاده از سوئیچ Vas	14.....
4.1.9	متد ResetService	16.....
4.1.10	متد TerminateService	16.....
4.1.11	متد Dispose	17.....
4.1.12	نحوه دریافت پاسخ از متدهای غیر همزمان	17.....
4.1.13	خروجی متدهای غیر همزمان	17.....
4.1.14	کلاس ResponseReceivedEventArgs	17.....
4.1.15	نمونه خروجی متدهای همزمان و غیر همزمان	18.....
4.1.16	نوع شمارشی SerialPortParity	18.....
4.1.17	نوع شمارشی ReadyState	19.....
4.1.18	نوع شمارشی SerialPortStopBit	19.....
4.1.19	پاسخ تراکنش و کدهای خطا	19.....
5	نحوه استفاده از کتابخانه حاضر در نرم افزار های وب و تنظیمات نمونه برای مرورگر IE	19.....
5.1.1	استفاده از JavaScript و ActiveX جهت پیاده سازی	20.....
6	برقراری ارتباط با پایانه فروش از طریق وب سرویس و عامل واسط (WINDOWS SERVICE)	21.....
6.1	نصب سرویس عامل (WINDOWS SERVICE)	21.....



22	دسترسی به سرویس WCF	6.2
22	کلمه عبور PASSPHRASE	6.3
22	شرح متدهای پیش بینی شده و نحوه استفاده از سرویس WCF	6.4
22	شرح متدها	6.4.1
22	ساختار بندی پایانه فروشگاهی	6.4.2
23	دریافت ساختار بندی فعلی	6.4.3
23	انجام تراکنش توسط وب سرویس	6.4.4



## 1 مقدمه

مستند جهت تعیین مشخصات کتابخانه انجام تراکنش های مالی از طریق پایانه های فروشگاه های PCPOS شرکت کارت اعتباری ایران کیش تدوین گردیده است.

لازم به توضیح است این مستند مطابق با کتابخانه نسخه 1.0.0.21 تدوین و منتشر گردیده است.

**نکته :** این راهکار برای پایانه های فروش برند AMP(KICE) فقط برای نسخه 1.0.0.18 و بالاتر عملیاتی می باشد.

## 2 نیازمندی ها

نرم افزار فوق جهت اجرای صحیح و بدون مشکل نیازمند موارد ذیل می باشد :

- NET Framework نسخه ی 2.0.50727 و یا بالاتر بر روی سیستم کاربر (Client) نصب و آماده به کار باشد.
- سیستم دارای درگاه COM (سریال) باشد (در صورت عدم وجود یک درگاه سریال که در تنظیمات نرم افزار مشخص گردیده باشد، کاربر با پیغام خطای مناسب روبرو خواهد شد).
- جهت استفاده از کتابخانه Kiccc.Ing.PcPos بر روی سامانه های مبتنی بر وب، توصیه می گردد در صورت امکان از مرورگر Internet Explorer استفاده گردد. (برای استفاده بر روی Mozilla Firefox باید Plug-in اجرای ActiveX بر روی مرورگر از طریق سایت Mozilla.org نصب گردد.) بدلیل اینکه این کتابخانه جهت ارائه سرویسهای خود نیاز به دسترسی به منابع client (درگاه سریال) دارد و انجام این کار توسط سامانه های مبتنی بر وب بر خلاف الزامات امنیتی بوده، لذا جهت ایجاد این دسترسی استفاده از افزونه ActiveX الزامی است.
- نرم افزار دارای یک مکانیزم درونی جهت برقراری ارتباط با مرورگر ها مطابق با استانداردهای امنیتی و شناسایی خود به عنوان یک نرم افزار امن و بی خطر بوده و به طور معمول به هنگام استفاده از آن بر روی سامانه های مبتنی بر وب، بسته به نسخه ی مرورگر مورد استفاده یا به طور خودکار و یا پس از پرسش از کاربر (تنها برای بار اول) فعال و قابل استفاده می گردد؛ چنانچه این نرم افزار به هر دلیلی (اعم از تغییر تنظیمات توسط خود کاربر و یا نسخ متفاوت مرورگر) امکان اجرای برنامه وجود نداشت، تنظیمات Internet Explorer به گونه ای انجام شده باشد که اجازه ی نصب و اجرای ActiveX را به کاربر بدهد. (برای توضیحات بیشتر قسمت تنظیمات IE را ببینید).
- تنظیمات Internet Explorer به گونه ای انجام شده باشد که اجازه ی اجرای JavaScript را به کاربر بدهد.

## 3 قابلیت ها و ویژگی ها

✓ ارسال فرامین و دریافت پاسخ از پایانه فروشگاه های

✓ قابلیت انجام تراکنشهای پایانه های چندحسابی

✓ پردازش داده های دریافتی

✓ مدیریت درگاه ارتباطی



- ✓ تولید خروجی یکپارچه و قابل فهم با فرمت XML برای برنامه‌های استفاده کننده
- ✓ معماری با قابلیت استفاده هم در محیط Microsoft.Net (به صورت Native .Net Assembly) و هم در محیط COM (برای نرم افزارهای تحت ویندوز و وب خارج از محیط .Net )
- ✓ قابلیت انعطاف جهت تغییر تنظیمات مربوط به پایانه ی فروش به صورت پویا
- ✓ نرم افزار قادر به انجام تراکنش به صورت غیر همزمان (Asynchronous) بوده تا برنامه‌های استفاده کننده به هنگام درخواست انجام تراکنش، دچار هنگ و یا Freeze شدگی نشوند؛ بدین معنی که نرم افزار پس از دریافت درخواست انجام تراکنش از برنامه‌ی درخواست کننده آن را انجام داده پس از انجام آن، نرم افزار درخواست کننده را از نتیجه ی تراکنش مطلع می سازد.
- متدهایی که در ادامه مستند نام آنها با کلمه Begin شروع می‌شود، همگی غیرهمزمان هستند و عملکردشان مشابه با متدهای هم‌نام خود که کلمه Begin ابتدای نام آنها وجود ندارد، است. نتیجه عملیات توابع غیرهمزمان از طریق رویداد ResponseReceived به نرم افزار بازمی‌گردد.
- ✓ همچنین نرم افزار قادر به انجام تراکنش به صورت همزمان (Synchronous) جهت فراهم آوردن امکان توقف تا زمان انجام کامل تراکنش نیز می باشد؛ بدین معنی که کنترل اجرای برنامه‌ی استفاده کننده پس از صدا زدن توابع همزمان انجام تراکنش‌ها به واسط نرم افزاری منتقل شده و پس از اجرای کامل تراکنش کنترل به برنامه ی صدا کننده برگردانده می‌شود.
- ✓ نرم افزار همچنین مکانیزمی جهت صرف نظر کردن از تراکنش‌هایی که زمان انجام آنها از مدت مشخصی تعیین شده توسط کاربر تجاوز نموده است را دارا بوده و به همین دلیل قابلیت اتکای زیادی را برای کاربران ایجاد می‌نماید.
- ✓ نرم افزار همچنین دارای خصوصیتی است که وضعیت آن و وضعیت کار با ترمینال را به نرم افزار استفاده کننده اطلاع می دهند.
- ✓ کتابخانه قابلیت بکارگیری در خصوص پایانه های Multi Merchant را دارا می باشد.
- ✓ امکان برقراری ارتباط از طریق وب سرویس با پایانه فروش از طریق عامل واسط (WINDOWS SERVICE)



✓ و ...

#### 4 نحوه انجام عملیات

این کتابخانه دارای دو فضای نام Kiccc.Ing.PcPos و Kiccc.Ing.PcPos.Serial است.

فضای نام Kiccc.Ing.PcPos دارای سه کلاس به نامهای AmountList، MultiplexPayment و

ResponseReceivedEventArgs است.

فضای نام Kiccc.Ing.PcPos.Serial دارای کلاس SerialIngenico است.

##### 4.1 کلاس SerialIngenico

این کلاس وظیفه ارتباط با پایانه فروشگاهی را داشته و شامل تمامی متدهای مورد استفاده جهت تعامل با پایانه فروشگاهی می باشد.

##### 4.1.1 متد InitiateService

```
Void InitiateService();

void InitiateService(string serialNo, string acceptorId, string terminalId, string
comPort, int baudRate, int dataBits, SerialPortStopBit stopBit,
SerialPortParity parity, int timeOut = 200);

void InitiateService(string serialNo, string acceptorId, string terminalId, string
comPort, int baudRate, int dataBits, int stopBit,
int parity, int timeOut = 200);
```

این تابع جهت راه اندازی و آغاز به کار سرویس واسط پایانه‌های فروش PCPos به کار می رود. پیش از هر گونه استفاده از نرم افزار و یا متدها و اعضای کلاس، این تابع باید صدا زده شود.

##### 4.1.1.1 پارامترهای ورودی

نام پارامتر	نوع	مقادیر ورودی	مقدار پیش فرض	توضیحات
serialNo	string			شماره سریال پایانه
acceptorId	string			کد پذیرنده
terminalId	string			کد پایانه
comPort	string	"COM1" "COM2" "COM3" ...	COM1	نام درگاه COM ای که POS به آن متصل است.



		"COM256"		
<b>baudRate</b>	int	75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57500, 115200, 128000	115200	نرخ انتقال اطلاعات بین PC و POS بر حسب bps
<b>dataBits</b>	int	4, 5, 6, 7, 8	8	تعداد بیت های انتقال داده
<b>stopBit</b>	SerialPortStopBit	None = 0 One = 1 Two = 2 OnePointFive = 3	One	بیت پایانی
<b>parity</b>	SerialPortParity	None = 0 Odd = 1 Even = 2 Mark = 3 Space = 4	None	نوع زوجیت (Parity)
<b>timeOut</b>	int	int	200	مدت زمان انتظار نرم افزار قبل از بستن درگاه COM به صورت خودکار (به ثانیه)

#### 4.1.1.2 پارامترهای خروجی

ندارد

در زیر نمونه کدی از نحوه ساخت شی از کلاسهای SerialIngenico و چگونگی فراخوانی متد Sale از کلاس SeialIngenico آمده است.

```
// ساخت نمونه
Kiccc.Ing.PcPos.Serial.SerialIngenico _serialIngObject = new
Kiccc.Ing.PcPos.Serial.SerialIngenico();

// مقدار دهی پارامترهای اولیه و اجباری جهت تعامل با پایانه فروشگاهی
_serialIngObject.AcceptorId = "*****";
_serialIngObject.BaudRate = 115200;
_serialIngObject.ComPort = "COM1";
_serialIngObject.DataBits = 8;
_serialIngObject.Parity = SerialPortParity.None;
_serialIngObject.SerialNo = "*****";
_serialIngObject.StopBit = SerialPortStopBit.One;
_serialIngObject.TerminalId = "*****";
_serialIngObject.TimeOut = 200;

// قبل از هرگونه تراکنشی متد زیر می بایست فراخوانی گردد
_serialIngObject.InitiateService();
```





```
// انجام تراکنش خرید با مبلغ نمونه 1000 و ذخیره پاسخ در متغیر
var _resp = _serialIngObject.Sale("1000");

// آزاد سازی منابع مورد استفاده توسط کتابخانه
_serialIngObject.Dispose();
```

**4.1.2 متد Sale**

```
string Sale(string amount);
```

این تابع جهت آغاز یک تراکنش خرید کالا و خدمات به کار می رود. با صدا کردن متد Sale، نرم افزار دستور انجام تراکنش را به دستگاه POS صادر نموده و پایانه‌ی فروش منتظر کاربر برای کشیدن کارت می‌گردد. لازم به ذکر است تابع BeginSale با همان مشخصات تابع Sale و تنها به صورت غیرهمزمان برای استفاده در موارد لزوم نیز موجود است.

**4.1.2.1 پارامترهای ورودی**

توضیحات	مقادیر ورودی	نوع	نام پارامتر
مبلغ خرید به ریال	Integer Value In String Format	string	amount

**4.1.2.2 پارامترهای خروجی**

مفهوم	مقادیر خروجی	نوع خروجی
پاسخ تراکنش / عملیات درخواست شده	XML	string

**4.1.3 متد SaleWithPaymentId**

```
string SaleWithPaymentId(string amount, string paymentId);
```

این تابع جهت آغاز یک تراکنش خرید با شناسه به کار می رود. با صدا کردن متد SaleWithPaymentId، نرم افزار دستور انجام تراکنش را به دستگاه POS صادر نموده و پایانه‌ی فروش منتظر کاربر برای کشیدن کارت می‌گردد. لازم به ذکر است تابع BeginSaleWithPaymentId با همان مشخصات تابع SaleWithPaymentId و تنها به صورت غیرهمزمان برای استفاده در موارد لزوم نیز موجود است.

**4.1.3.1 پارامترهای ورودی**

توضیحات	مقادیر ورودی	نوع	نام پارامتر
مبلغ خرید به ریال	Integer Value In String Format	string	amount
شناسه خرید	Integer Value In String Format	string	paymentId

**4.1.3.2 پارامترهای خروجی**

مفهوم	مقادیر خروجی	نوع خروجی
پاسخ تراکنش / عملیات درخواست شده	XML	string

**4.1.4 متد SpecialPayment**

```
string SpecialPayment(string amount);
```

این تابع جهت آغاز یک تراکنش خرید خاص به کار می رود. با صدا کردن متد SpecialPayment، نرم افزار دستور انجام تراکنش را به دستگاه POS صادر نموده و پایانه ی فروش منتظر کاربر برای کشیدن کارت می گردد. لازم به ذکر است تابع BeginSpecialPayment با همان مشخصات تابع SpecialPayment و تنها به صورت غیرهمزمان برای استفاده در موارد لزوم نیز موجود است.

**4.1.4.1 پارامترهای ورودی**

توضیحات	مقادیر ورودی	نوع	نام پارامتر
مبلغ خرید به ریال	Integer Value In String Format	string	amount

**4.1.4.2 پارامترهای خروجی**

مفهوم	مقادیر خروجی	نوع خروجی
پاسخ تراکنش / عملیات درخواست شده	XML	string

**4.1.5 متد CreditRemaining**

```
string CreditRemaining();
```

این تابع جهت آغاز یک تراکنش مانده اعتباری به کار می رود. با صدا کردن متد CreditRemaining، نرم افزار دستور انجام تراکنش را به دستگاه POS صادر نموده و پایانه ی فروش منتظر کاربر برای کشیدن کارت می گردد. لازم به ذکر است تابع BeginCreditRemaining با همان مشخصات تابع CreditRemaining و تنها به صورت غیرهمزمان برای استفاده در موارد لزوم نیز موجود است.

**4.1.6 متد MultiplexPayment و متد MultiplexPaymentWithPaymentId**

```
string MultiplexPayment(MultiplexPayment multiplexPayment);
string MultiplexPaymentWithPaymentId(MultiplexPayment multiplexPayment)
```

این تابع جهت آغاز یک تراکنش پرداخت چندتایی به کار می رود. با صدا کردن این تابع نرم افزار دستور انجام تراکنش را به دستگاه POS صادر نموده و پایانه ی فروش منتظر کاربر برای کشیدن کارت می گردد. پایانه پس از انجام عملیات، نتیجه ی تراکنش را در جواب تابع به نرم افزار صداکننده بازمی گرداند. این تابع همزمان (Synchronous) می باشد.



لازم به ذکر است تابع `BeginMultiplexPayment` با همان مشخصات تابع `MultiplexPayment` و تنها به صورت غیرهمزمان برای استفاده در موارد لزوم نیز موجود است.

**نکته:** متد `MultiplexPaymentWithPaymentId` مشابه متد `MultiplexPayment` می باشد با این تفاوت که قابلیت ارسال تراکنش با شناسه 18 کاراکتری را در مشخصه **`SpecialPaymentId`** دارا می باشد (دقت بفرمایید که مشخصه **`PaymentId`** در هر دو متد اجباری می باشد).

#### 4.1.6.1 پارامترهای ورودی

ورودی این متد شی ای از کلاس `MultiplexPayment` است.

#### 4.1.6.2 پارامترهای خروجی

مفهوم	مقادیر خروجی	نوع خروجی
پاسخ تراکنش / عملیات درخواست شده	XML	String

#### 4.1.6.3 کلاس `AmountList`

به کمک نمونه سازی از این کلاس می توان صفی (queue) از حسابهای پایانه های چندحسابی را ایجاد کرد.

##### 4.1.6.3.1 متد `Add`

```
Void Add(string amount)
```

توسط این متد باید مبالغ به ترتیبی که حسابهای پایانه های چندحسابی (تسهیم) بر روی پایانه تعریف شده اند به شی ساخته شده از کلاس `AmountList` اضافه گردند.

##### 4.1.6.3.2 پارامترهای ورودی

توضیحات	مقادیر ورودی	نوع	نام پارامتر
مبلغ خرید به ریال	Integer Value In String Format	string	amount

##### 4.1.6.3.3 پارامترهای خروجی

ندارد

##### 4.1.6.3.4 متد `Remove`

```
string Remove();
```



در صورت فراخوانی این متد توسط شی ساخته شده از کلاس AmountList، آخرین مبلغ اضافه شده به آن شی حذف می‌شود.

#### 4.1.6.3.5 پارامترهای ورودی

ندارد

#### 4.1.6.3.6 پارامترهای خروجی

خروجی این متد رشته‌ایست که بیانگر مبلغ حذف شده از شی است.

### 4.1.7 متد BillPayment

```
string BillPayment(string billid, string billpaymentid)
```

این تابع جهت آغاز یک تراکنش پرداخت قبض به کار می‌رود. با صدا کردن متد BillPayment، نرم‌افزار دستور انجام تراکنش را به دستگاه POS صادر نموده و پایانه‌ی فروش منتظر کاربر برای کشیدن کارت می‌گردد. لازم به ذکر است تابع BeginBillPayment با همان مشخصات تابع BillPayment و تنها به صورت غیرهمزمان برای استفاده در موارد لزوم نیز موجود است.

#### 4.1.7.1 پارامترهای ورودی

نام پارامتر	نوع	مقادیر ورودی	توضیحات
billid	string	Integer Value In String Format	شناسه قبض
billpaymentid	string	Integer Value In String Format	شناسه پرداخت قبض

#### 4.1.7.2 پارامترهای خروجی

نوع خروجی	مقادیر خروجی	مفهوم
string	XML	پاسخ تراکنش / عملیات درخواست شده

### 4.1.7.3 کلاس MultiplexPayment

شی ساخته شده از این کلاس حاوی کلیه اطلاعات ضروری انجام یک پرداخت چندتایی است. اطلاعات لیست مبالغ تسهیم در قالب شی‌ای از جنس AmountList از ویژگیهای این کلاس است.



نام پارامتر	نوع	مقادیر ورودی	طول	توضیحات
BankCode	string	Integer Value In String Format	2	کد بانک مرجع
Organization	string	Integer Value In String Format	2	کد سازمان
ServiceCode	string	Integer Value In String Format	1	کد خدمت
Financialyear	string	Integer Value In String Format	2	سال مالی
FiscalPeriod	string	Integer Value In String Format	3-1	تعداد روز سپری شده از سال
SequenceCode	string	Integer Value In String Format	3-1	کد مسلسل
PaymentId	string	Integer Value In String Format	1-10	شناسه پرداخت یکتا به ازای هر پایانه
SpecialPaymentId	string	Integer Value In String Format	1-18	شناسه پرداخت یکتا به ازای هر پایانه
Amounts	AmountList	Queue	حد اکثر 10 حساب	ترتیب ورود مبالغ با ترتیب تعریف حسابها ارتباط مستقیم دارد

در زیر نمونه کدی از نحوه ساخت شی از این کلاس آمده است:

```
// نمونه سازی
Kiccc.Ing.PcPos.MultiplexPayment _mpObj = new MultiplexPayment();

_mpObj.BankCode = "***";
_mpObj.Organization = "***";
_mpObj.ServiceCode = "*";
_mpObj.Financialyear = "***";
_mpObj.FiscalPeriod = "****";
_mpObj.SequenceCode = "****";
_mpObj.PaymentId = "*****";

// مبلغی که می بایست به حساب اول واریز گردد
_mpObj.Amounts.Add("250");
// مبلغی که می بایست به حساب دوم واریز گردد
_mpObj.Amounts.Add("250");
// مبلغی که می بایست به حساب سوم واریز گردد
_mpObj.Amounts.Add("250");
// مبلغی که می بایست به حساب چهارم واریز گردد
_mpObj.Amounts.Add("250");

// انجام عملیات خرید تسهیم
_serialIngObject.MultiplexPayment(_mpObj);
```

**4.1.8 پرداخت چند حسابی با استفاده از سوئیچ Vas**

```
string MultiplexPaymentWithVas(VasMultiplexPayment multiplexPayment)
```

این تابع جهت آغاز یک تراکنش پرداخت چندتایی به کار می رود. با صدا کردن این تابع نرم افزار دستور انجام تراکنش را به دستگاه POS صادر نموده و پایانه ی فروش منتظر کاربر برای کشیدن کارت می گردد. پایانه پس از انجام عملیات، نتیجه ی تراکنش را در جواب تابع به نرم افزار صداکننده بازمی گرداند. این تابع همزمان (Synchronous) می باشد. لازم به ذکر است تابع BeginMultiplexPaymentWithVas با همان مشخصات تابع MultiplexPaymentWithVas و تنها به صورت غیرهمزمان برای استفاده در موارد لزوم نیز موجود است.

**نکته:** برای استفاده از قابلیت های این متد می بایست مقدمات لازم از طرف فراهم کننده خدمات پرداخت قبلا مهیا تا فرایند انتقال تراکنش مالی میسر گردد.

**4.1.8.1 پارامترهای ورودی**

ورودی این متد شی ای از کلاس VasMultiplexPayment است.

**4.1.8.2 پارامترهای خروجی**

مفهوم	مقادیر خروجی	نوع خروجی
پاسخ تراکنش / عملیات درخواست شده	XML	String

**4.1.8.3 کلاس VasMultiplexPayment**

شی ساخته شده از این کلاس حاوی کلیه اطلاعات ضروری انجام یک پرداخت چندتایی است. اطلاعات لیست مبالغ تسهیم در قالب شی ای از جنس VasAmountList از ویژگی های این کلاس است.

توضیحات	طول	مقادیر ورودی	نوع	نام پارامتر
شناسه پرداخت یکتا به ازای هر پایانه	1	0 تا 9	int	* Index
شناسه پرداخت یکتا به ازای هر پایانه	1-18	Integer Value In String Format	string	** PaymentId
ترتیب ورود مبالغ با ترتیب تعریف حسابها ارتباط مستقیم دارد	حد اکثر 10 حساب		VasAmountList	Amounts

\*- اندیس متناظر با تنظیمات ترمینال پند حسابی (در حال حاضر می بایست با عدد 1 مقدار دهی شود)

\*\* - شناسه پرداخت ، برای استفاده از این قابلیت می بایست فراهم کننده خدمات پرداخت مجوز لازم برای استفاده را اعطا نماید.

**4.1.8.4 کلاس AmountList**

به کمک نمونه‌سازی از این کلاس می‌توان لیستی از حسابهای پایانه‌های چندحسابی را ایجاد کرد.

**4.1.8.4.1 Add متد**

```
void Add(long amount, int order)
```

توسط این متد باید مبالغ مرتبط با حسابهای متناظر تعریف شده بروی سوئیچ با ترتیب (پارامتر order) صحیح در زمان تعریف اضافه گردد.

**نکته:** حساب اصلی پذیرنده چند حسابی می‌بایست دارای کمترین ارزش عددی (0) باشد و به ترتیب سایر حسابهای با ترتیب قرار گیری متناظرشان اضافه گردد.

**نکته:** برای تمامی حسابهای متناظر می‌بایست مبلغ درج گردد، ولی مبلغ می‌تواند (0) باشد.

**4.1.8.4.2 پارامترهای ورودی**

نام پارامتر	نوع	مقادیر ورودی	توضیحات
<b>Amount</b>	long	Integer Value	مبلغ خرید به ریال
<b>Order</b>	int	Integer Value	ترتیب قرارگیری حساب ها با اندیس شروع صفر

**4.1.8.4.3 پارامترهای خروجی**

ندارد

**4.1.8.4.4 Remove متد**

```
bool Remove(long amount, int order)
```

برای حذف یک حساب با ترتیب خاص از لیست مبالغ از این متد استفاده می‌شود.

**4.1.8.4.5 پارامترهای ورودی**

نام پارامتر	نوع	مقادیر ورودی	توضیحات
<b>Amount</b>	long	Integer Value	مبلغ خرید به ریال
<b>Order</b>	int	Integer Value	ترتیب قرارگیری حساب ها با اندیس شروع صفر



## 4.1.8.4.6 پارامترهای خروجی

خروجی این متد مقدار منطقی است که بیانگر نتیجه انجام عملیات است.

## 4.1.9 متد ResetService

```
void ResetService();
```

این تابع جهت راه اندازی مجدد سرویس با تنظیمات انجام شده در هنگام راه اندازی اولیه و مقادیر ویژگیهای شی ساخته شده در حافظه، به کار می رود. در مواردی که سرویس از حالت پاسخگویی خارج شده باشد و یا کاربر مایل باشد پیش از اتمام تراکنش و یا زمان Time out آن، سرویس را از حالت Busy به حالت Active ببرد و یا به هر علتی سرویس به حالت Inactive رفته باشد، با صدا کردن این تابع، پورت COM یکبار باز و بسته شده و کلیه ی مکانیزم های داخلی سرویس به حالت عادی باز خواهد گشت و سرویس واسط مجدداً قابل استفاده خواهد گردید. این متد در واقع عملیات متدهای Dispose و InitiateService را به ترتیب انجام می دهد.

## 4.1.9.1 پارامترهای ورودی

ندارد

## 4.1.9.2 پارامترهای خروجی

ندارد

## 4.1.10 متد TerminateService

```
void TerminateService();
```

این تابع جهت خاتمه دادن به سرویس واسط پایانه های فروش به کار می رود. پس از صدا زدن این تابع، بدون توجه به شرایط فعلی سرویس و تراکنش، درگاه COM بسته شده، کلیه ی Resource های مورد استفاده توسط سرویس آزاد گشته، و تحویل سیستم می گردند.

با هر درخواست برای انجام تراکنش، سرویس واسط به صورت خودکار منابعی را از سیستم کاربر دریافت، درگاه COM مذکور را باز و آن را تا رسیدن پاسخ تراکنش از POS یا تمام شدن زمان Timeout (که مدت زمانیست که نرم افزار قبل از بستن درگاه سریال انتظار میکشد) باز نگه داشته و پس از دریافت پاسخ نیز آنرا می بندد. از این تابع در پایان استفاده از سرویس و یا موارد خاصی که به هر دلیلی اعم از ایجاد مشکلات فنی در پایانه، بسته شدن صفحه ی مرورگر قبل از رسیدن پاسخ پایانه ی فروش و .... برای بستن درگاه COM استفاده می شود. شایان ذکر است که در شرایط عادی که کاربر تا رسیدن پاسخ تراکنش یا تمام شدن زمان انتظار صبر می کند نیازی به استفاده از این تابع نیست و تنها باید پس از انجام کلیه ی عملیات دلخواه جهت آزادسازی منابع آن را صدا نمود.

## 4.1.10.1 پارامترهای ورودی

ندارد



**4.1.10.2 پارامترهای خروجی**

ندارد

**4.1.11 متد Dispose**

```
void Dispose();
```

با فراخوانی این متد کلیه منابع مورد استفاده توسط نرم افزار آزاد می شوند.

**4.1.11.1 پارامترهای ورودی**

ندارد

**4.1.11.2 پارامترهای خروجی**

ندارد

**4.1.12 نحوه دریافت پاسخ از متدهای غیر همزمان**

پاسخ توابع غیرهمزمان از طریق این رویداد قابل دسترسی است. هنگامی که کاربر درخواست انجام عملیاتی را از طریق یکی از توابع غیرهمزمان (Asynchronous) از پایانه ی فروش می نماید، سرویس وارد حالت مشغول (Busy) گردیده و تا زمان آمدن پاسخ از ترمینال و یا فرارسیدن مهلت Timeout منتظر می ماند؛ به محض اتفاق افتادن یکی از این 2 حالت، رویداد ResponseReceived رخ داده و از طریق خصوصیت Result نتیجه ی عملیات درخواستی را به نرم افزار صدا کننده منتقل نموده و پس از اتمام فراخوانی، وضعیت سرویس را به حالت Ready تغییر می دهد.

شایان ذکر است پیش از صدا نمودن توابع غیرهمزمان می بایست این رویداد Subscribe شده باشد.

**4.1.13 خروجی متدهای غیرهمزمان**

جدول زیر بیانگر مفهوم خروجی متدهای غیرهمزمان (که با کلمه Begin شروع می شوند) است.

مفهوم	مقادیر خروجی	نوع خروجی
درخواست تراکنش با موفقیت برای POS فرستاده شد؛ منتظر رویداد ResponseReceived باشید.	true	bool
در حین ارسال درخواست تراکنش برای POS خطایی رخ داده است. ضمناً متناسب با هر خطای رخ داده استثنائی رخ می دهد.	false	

**4.1.14 کلاس ResponseReceivedEventArgs**

ویژگی Result از این کلاس جهت دریافت پاسخ متدهای غیرهمزمان بکار می رود.

توضیحات	مقادیر ورودی	نوع	نام پارامتر
نتیجه ی عملیات درخواستی از پایانه در قالب XML	XML Formatted String	string	Result

**4.1.15 نمونه خروجی متدهای همزمان و غیر همزمان**

در زیر نمونه ای از خروجی متدهای همزمان و غیر همزمان آورده شده است .

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Response>
<RRN>015672010112</RRN>
<RespCode>00</RespCode>
<SerialNo>009038707292</SerialNo>
<TransactionDate>1394/04/30</TransactionDate>
<TransactionTime>13:59:34</TransactionTime>
<Amount>000000001000</Amount>
<TerminalId>03001567</TerminalId>
<TraceNo>000012</TraceNo>
<Pan>603799*****1222</Pan>
</Response>
```

نام پارامتر	توضیحات
RRN	کد ارجاع انجام تراکنش
RespCode	کد وضعیت انجام تراکنش
SerialNo	شماره سریال پایانه فروشگاهی
TransactionDate	تاریخ انجام تراکنش
TransactionTime	زمان انجام تراکنش
Amount	مبلغ
TerminalId	شماره پایانه
TraceNo	شماره سند / تراکنش
Pan	شماره کارت

**4.1.16 نوع شمارشی SerialPortParity**

دارای مقادیر زیر است:

مقدار حالت	نام حالت
0	None
1	Odd
2	Even
3	Mark
4	Space

**4.1.17 نوع شمارشی ReadyState**

دارای مقادیر زیر است:

مقدار حالت	نام حالت
0	Ready
1	Busy
2	Fault

**4.1.18 نوع شمارشی SerialPortStopBit**

دارای مقادیر زیر است:

مقدار حالت	نام حالت
0	None
1	One
2	Two
3	OnePointFive

**4.1.19 پاسخ تراکنش و کدهای خطا**

کد	پاسخ
-1	CanceledByUserBeforeTrans
-100	InvalidData
-200	InvalidTransaction
-201	InvalidSerial
-202	InvalidTerminal
-203	InvalidAcceptor
-999	UnknownError
-998	Timeout
00	تراکنش موفق

**5 نحوه استفاده از کتابخانه حاضر در نرم افزار های وب و تنظیمات نمونه برای مرورگر IE**

نرم افزار Internet Explorer به صورت پیش فرض از اجرا شدن ActiveX ها و قطعات کد به دلایل امنیتی خودداری می نماید، به همین علت و در جهت رفع این مشکل مکانیزم های خاصی در درون نرم افزار واسط پایانه های فروش هایپرکام به منظور معرفی آن به عنوان یک نرم افزار ایمن و مطمئن به مرورگرها تعبیه و پیاده سازی گردیده است که میزان این مشکلات را به حداقل رسانده و عموماً در نسخ متفاوت مرورگر IE و تحت تنظیمات مختلف، کمترین عملیات برای استفاده از آن مورد نیاز است.



- در اینجا 3 حالت متفاوتی که به هنگام استفاده از نرم افزار ممکن است اتفاق بیفتد بررسی می گردد :
1. در نسخ قدیمی IE مکانیزم های داخلی باعث اجرای صحیح و بدون مشکل نرم افزار می شوند که در اینگونه موارد نیاز به هیچ تنظیمی موجود نیست.
  2. چنانچه کاربر تنظیمات پیش فرض IE را تغییر نداده باشد، به هنگام اولین استفاده از نرم افزار، مستطیل زردرنگ باریکی در بالای صفحه ظاهر شده و از کاربر جهت اجرای نرم افزار اجازه می گیرد که در اکثریت مواقع نیز همین سناریو اتفاق می افتد.
  3. چنانچه کاربر تنظیمات پیش فرض را تغییر داده باشد، نیاز است تا از طریق تغییر تنظیمات به کاربر جهت استفاده از نرم افزار اجازه داده شود.
- شایان ذکر است که نرم افزار Internet Explorer پس از اعمال تغییرات نیاز به شروع مجدد دارد.

### 5.1.1 استفاده از JavaScript و ActiveX جهت پیاده سازی

نمونه کد زیر به زبان JavaScript جهت پیاده سازی و استفاده از کتابخانه راهکار PCPOS در نرم افزارهای محیط وب نوشته شده و با مرورگرهای IE سازگار می باشد. (در رابطه با سایر مرورگرها می بایست Plugin مربوطه نصب گردد) قطعه کد ارائه شده در زیر متد خرید (Sale) را پیاده سازی نموده است.

```
function Sale() {  
    var posDriver = new ActiveXObject("Kiccc.Ing.PcPos.Serial.SerialIngenico");  
  
    if (posDriver != null) {  
        posDriver.InitiateService_3("90*****402", "0*****76", "0*****67",  
"COM1",115200, 8, 1, 0, 200);  
  
        alert(posDriver.Sale("1000"));  
    }  
    else {  
        alert("failed");  
    }  
    return false;  
}
```

قطعه کد ارائه شده در زیر متد خرید تسهیم (MultiplexPayment) را پیاده سازی نموده است.  
نکته: پیاده سازی نمونه زیر برای یک پرداخت تسهیم 4 حسابی انجام شده است.

```
function MultiplexPayment() {  
    var posDriver = new ActiveXObject("Kiccc.Ing.PcPos.Serial.SerialIngenico");  
    var multiplex = new ActiveXObject("Kiccc.Ing.PcPos.MultiplexPayment");  
  
    if (posDriver != null && multiplex != null) {  
  
        posDriver.InitiateService_3("90*****402", "0*****76", "0*****67",  
"COM1",115200, 8, 1, 0, 200);  
  
        multiplex.BankCode = "***";  
        multiplex.Organization = "***";  
    }  
}
```



```
multiplex.ServiceCode = "*";
multiplex.Financialyear = "***";
multiplex.FiscalPeriod = "***";
multiplex.SequenceCode = "***";
multiplex.PaymentId = "90*****91";
multiplex.Amounts.Add("***");
multiplex.Amounts.Add("***");
multiplex.Amounts.Add("***");
multiplex.Amounts.Add("***");

alert(posDriver.MultiplexPayment(multiplex));

}
else {
alert("failed");
}

return false;
}
```

## 6 برقراری ارتباط با پایانه فروش از طریق وب سرویس و عامل واسط (Windows Service)

در این راهکار ، ابتدا یک ویندوز سرویس در رایانه مقصد نصب می گردد ، بعد از شروع بکار سرویس مذکور یک سرویس WCF توسط سرویس عامل بروی رایانه مقصد میزبانی و از طریق متدهای پیش بینی شده امکان ساختار بندی و برقراری ارتباط با پایانه فروش را فراهم می کند .

- در این راهکار سرویس عامل از طریق پورت USB و یا پورت Serial با پایانه فروش ارتباط برقرار می نماید.
- امکان ساختار بندی تنظیمات ارتباطی با پایانه فروش توسط متدهای سرویس WCF و یا از طریق ابزار کمکی ساختار بندی امکان پذیر می باشد.
- تمامی ورودی و خروجی در متدها پیش بینی شده به صورت XML String می باشد.

### 6.1 نصب سرویس عامل (Windows Service)

برای نصب سرویس واسط کفایت بسته نرم افزاری ارائه شده توسط این شامل یک فایل جادوگر نصب می باشد اجرا گردد.

- برای استفاده از این راهکار می بایست بسته Net Framework 3.5. بروی رایانه مقصد نصب باشد. در صورتی که این بسته موجود نباشد ، جادوگر نصب سرویس عامل اقدام به نصب این بسته نرم افزاری می نماید.
- نصب بسته فوق الذکر نیازمند دسترسی Administrator می باشد.
- بعد از اتمام مراحل نصب سرویس عامل به صورت خودکار Start و آماده بهره برداری می شود.

**6.2 دسترسی به سرویس WCF**

در صورتی که نصب بسته نرم افزاری سرویس عامل به درستی انجام گیرد سرویس WCF در آدرس زیر میزبانی و در دسترس خواهد بود

<http://localhost:50000/KicccPosDriver>

**6.3 کلمه عبور Passphrase**

برای تعامل با پایانه فروش در این راهکار علاوه بر پارامترهای استاندارد پرداخت و یا ساختار بندی ، پارامتری با عنوان کلمه عبور (Passphrase) دیده شده است که بصورت پیش فرض به شرح زیر می باشد.

- لازم به ذکر است که امکان تغییر کلمه عبور از طریق متد های پیش بینی شده وجود دارد.

Passphrase = KicccPcPosAgent

**6.4 شرح متدهای پیش بینی شده و نحوه استفاده از سرویس WCF****6.4.1 شرح متدها**

ردیف	نام متد	شرح متد
1	SetConfig	ساختار بندی پایانه فروشگاهی
2	GetConfig	دریافت ساختار بندی فعلی
3	TerminateService	قطع ارتباط سرویس عامل با پایانه فروش
4	ResetService	راه اندازی مجدد ارتباط سرویس عامل با پایانه فروشگاهی
5	Transaction	انجام تراکنش
6	ChangePassPhrase	تغییر کلمه عبور

**6.4.2 ساختار بندی پایانه فروشگاهی**

```
string SetConfig(string driverconfig);
```

برای انجام تنظیمات ارتباطی و پارامترهای مشخص کننده حساب پشت پایانه فروشگاهی از این متد استفاده می گردد.

پارامترهای ورودی و خروجی هر دو از نوی رشته ای و به صورت XML می باشند.

شرح و مشخصات پارامترها قبلا در این مستند توضیح داده شده است.

- در رابطه با پارامترهای نوع شمارش می بایست مقدار عددی هر متغیر به صورت رشته به سرویس ارجاع گردد که قبلا در این مستند توضیح داده شده است.

**6.4.2.1 نمونه یک درخواست پیکر بندی صحیح**

```
?>xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DriverConfig>
<PassPhrase>KicccPcPosAgent</PassPhrase>
<SerialNo>2151047993</SerialNo>
<AcceptorId>222490000460</AcceptorId>
<TerminalId>03005709</TerminalId>
<ComPort>COM3</ComPort>
<BaudRate>115200</BaudRate>
<Parity>0</Parity>
<DataBits>8</DataBits>
<StopBit>1</StopBit>
<TimeOut>220</TimeOut>
</DriverConfig>
```

پاسخ اجرای متد فوق بصورتی یک رشته XML میباشد که نمونه پاسخ موفق و نا موفق به شرح زیر ارائه می گردد

**6.4.2.2 پاسخ موفق**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<SucceedResponse>
<Description>Configuration Updated Successfully</Description>
</SucceedResponse>
```

**6.4.2.3 پاسخ نا موفق**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<FailureResponse>
<Description>Invalid StopBit</Description>
</FailureResponse>
```

**6.4.3 دریافت ساختار بندی فعلی**

```
string GetConfig(string passphrase);
```

برای دریافت اطلاعات ساختار بندی فعلی از این متد استفاده می شود .

ورودی و خروجی های هردو از نوع رشته می باشد.

پارامتر ورودی کلمه عبور و خروجی متد مشابه خروجی متد ساختار بندی پایانه فروشگاهی است.

**6.4.4 انجام تراکنش توسط وب سرویس**

```
string Transaction(string transactionconfig);
```

برای انجام تراکنش می بایست از این متد استفاده نمود ، ورودی و خروجی ها هر دو از نوع XML String می باشد.

در زیر به ازای هر تراکنش نمونه از ورودی صحیح قرارداد شده است .



- خروجی متد در صورت انجام تراکنش مطابق با بند 4.1.14 و در صورت بروز خطا مطابق با بند 6.4.2.3 می باشد.

#### 6.4.4.1 انواع تراکنش قابل انجام توسط سرویس و شرح مقادیر تگ <TransactionType>

ردیف	شرح	مقدار
1	تراکنش خرید	1
2	تراکنش خرید با شناسه 18 کاراکتری	2
3	تراکنش مانده گیری اعتباری	3
4	تراکنش خرید اعتباری خاص	4
5	تراکنش چند حسابی	5
6	تراکنش پرداخت قبض	6
7	تراکنش خرید با شناسه 99 کاراکتری و اطلاعات قابل چاپ بروی پرفراژ	7

#### 6.4.4.2 تراکنش خرید

```
?>xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TransactionRequest>
<PassPhrase>KicccPcPosAgent</PassPhrase>
<TransactionType>0</TransactionType>
<Amount>1000</Amount>
</TransactionRequest>
```

#### 6.4.4.3 تراکنش خرید با شناسه

```
?>xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TransactionRequest>
<PassPhrase>KicccPcPosAgent</PassPhrase>
<TransactionType>1</TransactionType>
<Amount>1000</Amount>
<PaymentId>12713803</PaymentId>
</TransactionRequest>
```

#### 6.4.4.4 تراکنش پرداخت قبض

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TransactionRequest>
<PassPhrase>KicccPcPosAgent</PassPhrase>
```





```
<TransactionType>6</TransactionType>
<BillId>12</BillId>
<BillPaymentId></BillPaymentId>
</TransactionRequest>
```

#### 6.4.4.5 تراکنش چند حسابی

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TransactionRequest>
<PassPhrase>KicccPcPosAgent</PassPhrase>
<TransactionType>***</TransactionType>
<Multiplex>
  <PaymentId>*****</PaymentId>
  <Organization>*****</Organization>
  <ServiceCode>***</ServiceCode>
  <SequenceCode>***</SequenceCode>
  <Financialyear>***</Financialyear>
  <FiscalPeriod>***</FiscalPeriod>
  <BankCode>***</BankCode>
  <SpecialPaymentId>***</SpecialPaymentId>
  <Amounts>
    <Amount>2500</Amount>
    <Amount>2500</Amount>
    <Amount>2500</Amount>
    <Amount>2500</Amount>
  </Amounts>
</Multiplex>
</TransactionRequest>
```

##### 6.4.4.5.1 Tag ، TransactionType در پرداخت چند حسابی

برای انجام تراکنش چند حسابی بدون شناسه 18 رقمی مقدار این Tag می بایست 2 باشد، لازم به توضیح است که وجود Tag، SpecialPaymentId در این حالت لزومی ندارد.

برای انجام تراکنش چند حسابی با شناسه 18 رقمی مقدار این Tag می بایست 5 باشد و وجود Tag ، SpecialPaymentId اجباری است.

مبالغ مرتبط با حسابها به شکل فوق و با رعایت ترتیب تعریف حسابها می بایست ارسال گردد.

#### 6.4.4.6 قطع ارتباط سرویس عامل با پایانه فروش

```
string TerminateService(string passphrase);
```



برای قطع ارتباط سرویس عامل با پایانه فروش از این متد استفاده می شود.

پارامترهای ورودی از نوع رشته ای و شامل کلمه عبور می باشد.

پارامتر خروجی از نوع رشته ای ، به صورت XML String و مطابق با بندهای 6.4.2.2 و 6.4.2.3 است.

#### 6.4.4.7 راه اندازی مجدد ارتباط سرویس عامل با پایانه فروش

```
string ResetService(string passphrase);
```

برای راه اندازی مجدد ارتباط سرویس عامل با پایانه فروش می بایست از این متد استفاده نمایید.

پارامتر ورودی از نوع رشته ای و شامل کلمه عبور می باشد.

پارامتر خروجی از نوع رشته ای ، به صورت XML String و مطابق با بندهای 6.4.2.2 و 6.4.2.3 است.

#### 6.4.4.8 تغییر کلمه عبور

```
string ChangePassPhrase(string changePassPhraseParams);
```

برای تغییر کلمه عبور می بایست از متد فوق استفاده نمایید .

پارامتر ورودی از نوع رشته ای و به صورت XML String می باشد. در زیر نمونه از درخواست صحیح تغییر کلمه عبور

ارائه شده است.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ChangePassPhrase>
<OldPassPhrase>MyOldPassphrase</OldPassPhrase>
<NewPassPhrase>MyNewPassphrase</NewPassPhrase>
</ChangePassPhrase>
```

پارامتر خروجی از نوع رشته ای ، به صورت XML String و مطابق با بندهای 6.4.2.2 و 6.4.2.3 است.