

PcPos .NET API

طبقه بندی: عادی

تهیه و تدوین: معاونت فناوری اطلاعات

این سند توسط شرکت پرداخت الکترونیک سداد تولید شده است و به عنوان دارایی با ارزش سازمان محسوب می شود.
هرگونه کپی برداری و یا استفاده دیگر از این سند بدون کسب مجوز کتبی از این شرکت ممنوع می باشد. ضمناً دارنده این سند از هرگونه افشاء یا انتشار سند به اشخاص دیگر جهت هرگونه استفاده و بهره برداری بدون اطلاع این شرکت پرهیز نماید.

تاریخچه مستند

شماره ویرایش	تاریخ صدور	شرح	نام و نام خانوادگی
۱,۰	۱۳۹۵/۱۲/۱۵	تدوین	علی گیتی نژاد
۱,۱	۱۳۹۶/۰۴/۲۱	ویرایش	احمد بابایی کفشگر
۱,۲	۱۳۹۶/۰۸/۱۵	ویرایش	احمد بابایی کفشگر
۱,۳	۱۳۹۶/۰۸/۲۹	ویرایش	احمد بابایی کفشگر
۱,۴	۱۳۹۶/۰۹/۰۹	ویرایش	احمد بابایی کفشگر
۱,۵	۱۳۹۶/۱۲/۱۵	ویرایش	احمد بابایی کفشگر
۱,۶	۱۳۹۷/۰۲/۲۲	اضافه شدن استعلام شماره شبا و تغییرات تسهیم	احمد بابایی کفشگر
۱,۷	۱۳۹۷/۰۲/۲۶	اصلاح خصوصیات، اضافه شدن کلاس ها	احمد بابایی کفشگر
۱,۸	۱۳۹۷/۰۶/۱۲	اضافه شدن OrderId	احمد بابایی کفشگر
۱,۹	۱۳۹۷/۰۶/۲۶	اضافه شدن گرفتن خصوصیات Config در سرویس REST اضافه شدن گرفتن مشخصات پوز	احمد بابایی کفشگر
۱,۱۰	۱۳۹۷/۰۷/۰۳	حذف RequestId اضافه شدن اصلاح نمونه کدها تبدیل کدهای برگشتی از سوئیچ به پیام فارسی	احمد بابایی کفشگر
۱,۱۱	۱۳۹۷/۰۸/۲۷	اضافه شدن شماره کارت به صورت Mask در CardInfo توضیحات دستگاه ها و تراکنش های چند حسابی	احمد بابایی کفشگر
۱,۱۲	۱۳۹۷/۰۸/۳۰	خرید بسته حمایت غذایی	احمد بابایی کفشگر
۱,۱۳	۱۳۹۷/۱۰/۰۱	تغییرات خصوصیات کلاس ها خرید بسته امنیت غذایی	احمد بابایی کفشگر
۱,۱۴	۱۳۹۸/۰۳/۰۵	اضافه شدن MultiSaleId، اصلاح سند طبق تغییرات رابط کاربری	احمد بابایی کفشگر
۱,۱۵	۱۳۹۸/۰۴/۱۰	اضافه شدن قابلیت های پرداخت با تعرفه، جستجو، تسهیم با ورود شماره شبا، و تغییر گزارش تراکنش های پوز	احمد بابایی کفشگر
۱,۱۶	۱۳۹۸/۰۶/۳۰	کم شدن حداقل زمان پاسخگویی، ارسال نسخه DLL به دستگاه در تراکنش ها، اضافه شدن تنظیمات مربوط به تراکنش، اصلاحات جزئی	احمد بابایی کفشگر
۱,۱۷	۱۳۹۸/۱۲/۰۶	اصلاح کنترل خطاها، تغییر مدلها، بهبود ارتباط پوز مجیک، استعلام تراکنش مجیک	احمد بابایی کفشگر
۱,۱۸	۱۳۹۹/۰۲/۱۶	اضافه شدن متن انتهای رسید.	احمد بابایی کفشگر
۱,۱۹	۱۳۹۹/۰۶/۱۲	اصلاحات نامگذاری متدها و خصوصیات. اضافه شدن پرداخت نشان اعتباری. اصلاحات داخلی جواب تراکنش.	احمد بابایی کفشگر
۱,۲۰	۱۴۰۱/۰۳/۰۳	اضافه شدن اطلاعات تسهیم تراکنش در پاسخ	احمد بابایی کفشگر
۱,۲۱	۱۴۰۲/۰۷/۳۰	اصلاح url استعلام iban	احمد بابایی کفشگر

فهرست مطالب

۱ هدف.....	۸
۲ دامنه کاربرد.....	۸
۳ مسئولیت ها.....	۸
۳,۱ محیط برنامه نویسی مورد نیاز برای توسعه ی نرم افزار با استفاده از DLL.....	۱۰
۳,۲ قابلیت ها.....	۱۰
۴ خصوصیات.....	۱۱
۴,۱ Ip.....	۱۱
۴,۲ Port.....	۱۱
۴,۳ RetryTimeout.....	۱۱
۴,۴ ResponseTimeout.....	۱۱
۴,۵ Amount.....	۱۲
۴,۶ PcPosDllVersion.....	۱۲
۴,۷ SerialNo.....	۱۲
۴,۸ ConnectionType.....	۱۲
۴,۹ ComPortName.....	۱۳
۴,۱۰ Field1-8.....	۱۳
۵ متدها.....	۱۴
۵,۱ ConvertIbans.....	۱۴
۵,۲ ConvertIbans.....	۱۵
۵,۳ SetSaleId.....	۱۶
۵,۴ SetMultiSaleId.....	۱۶
۵,۵ SetOrderId.....	۱۷
۵,۶ SetBillInfo.....	۱۷
۵,۷ SetFixedDuty.....	۱۷
۵,۸ SetMultiAccountData.....	۱۸
۵,۹ SetMultiAccountsData.....	۱۹
۵,۱۰ SetMultiAccountData.....	۲۱

۲۳.....	SetAdvertisement	۵,۱۱
۲۳.....	SetCardInfo	۵,۱۲
۲۳.....	SetSearchParam	۵,۱۳
۲۴.....	GetAvailableSerialPorts	۵,۱۴
۲۴.....	AsyncSaleTransaction	۵,۱۵
۲۴.....	SyncSaleTransaction	۵,۱۶
۲۵.....	AsyncBillPaymentTransaction	۵,۱۷
۲۵.....	AsyncInquiredIdentifiedTransaction	۵,۱۸
۲۶.....	AsyncGovernmentInquiredIdentifiedTransaction	۵,۱۹
۲۷.....	AsyncFixedDuty	۵,۲۰
۲۷.....	AsyncCommodityBasket	۵,۲۱
۲۸.....	AsyncFoodSafety	۵,۲۲
۲۸.....	AsyncCardInfo	۵,۲۳
۲۹.....	AsyncTransactionLogs	۵,۲۴
۲۹.....	AsyncSearch	۵,۲۵
۳۰.....	AsyncGetAccount	۵,۲۶
۳۰.....	AbortOperation	۵,۲۷
۳۱.....	StartGettingCardInfo	۵,۲۸
۳۱.....	StopGettingCardInfo	۵,۲۹
۳۱.....	PcPosDiscovery	۵,۳۰
۳۲.....	SearchPcPosByIp	۵,۳۱
۳۲.....	SearchPcPosByMerchantName	۵,۳۲
۳۲.....	SearchPcPosByTerminalId	۵,۳۳
۳۳.....	SearchPcPosByMerchantId	۵,۳۴
۳۴.....	۶ کنترل خطا	
۳۴.....	PcPosBadInputException	۶,۱
۳۴.....	MaximumValueException	۶,۲
۳۴.....	MinimumValueException	۶,۳

۳۶	۷	رویدادها
۳۶	۷,۱	OnSaleResult
۳۶	۷,۲	OnBillPaymentResult
۳۶	۷,۳	OnInquiredIdentifiedSaleResult
۳۶	۷,۴	OnGovernmentInquiredIdentifiedSaleResult
۳۷	۷,۵	OnCardInfoResult
۳۷	۷,۶	OnLogResult
۳۷	۷,۷	OnPosDeviceInfoResult
۳۷	۷,۸	OnAccountInfoResult
۳۸	۷,۹	OnSearchPcPos
۳۹	۸	کلاس ها
۳۹	۸,۱	PosResult
۴۰	8.2	ExtraDataResult
۴۰	۸,۳	PosCardInfoResult
۴۱	۸,۴	PosLogsResult
۴۱	۸,۵	PosLogResult
۴۲	۸,۶	PosDevice
۴۲	۸,۷	DiscoveryResult
۴۲	۸,۸	SaleIdProvider
۴۳	۸,۹	SaleIdData
۴۳	۸,۱۰	AccountResult
۴۳	۸,۱۱	Account
۴۴	۹	تنظیمات رابط توسعه
۴۵	۹,۱	سریال
۴۶	۹,۲	شبکه
۴۶	۹,۳	تراکنش خرید
۴۶	۹,۴	تراکنش پرداخت قبض
۴۶	۹,۵	تراکنش خرید با استعلام

۹,۶	تراکنش با استعلام دولتی.....	۴۷
۱۰	نحوه استفاده از DLL در Visual Studio.....	۴۸
۱۰,۱	ایجاد پروژه C#.....	۴۸
۱۱	روال استفاده.....	۴۹
۱۱,۱	در فروشگاه‌های حالت خاص (افق کوروش).....	۴۹

۱ هدف

ارایه مستندات API پی سی پوز میباشد.

۲ دامنه کاربرد

این مستند به منظور آشنایی برنامه نویسان جهت استفاده از API پی سی پوز دستگاه های برند بلوبرد، بایتل و مجیک و دیگر دستگاه های شرکت پرداخت الکترونیک سداد ارایه شده است.

برند	مدل
بلوبرد ^۱	P3500, CT280, MT280
بایتل ^۲	IC3300, IC3100
مجیک ^۳	X8, X5

۳ مسئولیت ها

در راه اندازی سرویس پی سی پوز همکاری واحدهای پشتیبانی و بخش فناوری اطلاعات جهت راه اندازی الزامی میباشد.

^۱ BlueBird

^۲ Bitel

^۳ Magic

درباره PCPOS .Net API

نسخه 5.3.531.1632 رابط توسعه نرم افزاری PC POS .NET API به منظور توسعه نرم افزار و اتصال به دستگاه پوز طراحی شده است. ارتباط بین دستگاه پوز و رابط برنامه نویسی (DLL) از طریق شبکه یا درگاه سریال میسر می باشد. در ارتباط از طریق شبکه، ابتدا شبکه جهت شناسایی پوزهای در دسترس جستجو می شود. سپس دستگاه مورد نظر جهت ادامه فرآیند خرید مورد استفاده قرار میگیرد. در ارتباط سریال هم کفایت بعد از مقدار دهی شماره درگاه سریال ادامه فرآیند تراکنش انجام شود.

رابط برنامه نویسی امکانات زیر را در اختیار کاربران قرار می دهد:

- امکان شناسایی پوزهای موجود در شبکه
- امکان ارسال تراکنش از طریق بستر سریال
- امکان ارسال تراکنش و دریافت نتیجه
- امکان لغو تراکنش از سمت رابط برنامه نویسی
- امکان لغو تراکنش از سمت پوز
- ارایه امکان خرید تک حسابی و چند حسابی
- ارایه امکان خرید تک و چند شناسه ای برای پرداخت های دولتی
- امکان خرید بسته حمایت غذایی و بسته امنیت غذایی
- امکان چاپ متن انتهای رسید دستگاه POS
- امکان پرداخت قبض
- امکان دریافت ۱۳ تراکنش آخر ثبت شده در دستگاه POS
- امکان استعلام تراکنش در پوزهای مجیک
- انتقال رمز گذاری شده اطلاعات بین پوز و رابط توسعه نرم افزاری پی سی پوز
- جلوگیری از جابجایی پوز با استفاده از شماره سریال پوز
- امکان تنظیم تنظیمات تراکنش و اتصالات در فایل xml

۳,۱ محیط برنامه نویسی مورد نیاز برای توسعه‌ی نرم افزار با استفاده از DLL

محیط مورد نیاز برای Build کردن ماژول DLL، Microsoft Visual Studio نسخه 2013 به بالا بوده و زبان برنامه نویسی مورد استفاده در سورس این نرم افزار زبان C# می باشد. نسخه Dot Net Framework مورد استفاده 2.0 می باشد. نمونه کد نحوه‌ی استفاده از DLL در مسیر <installDir>\Sample Code آورده شده است.

۳,۲ قابلیت ها

Device Type	Connection Type	Library Support
BlueBird P3500, CT280, MT280	LAN Serial	خرید (عادی، شناسه دار با استعلام، دولتی، بسته حمایت غذایی) پرداخت قبض تسهیم چندحسابی تا ۱۰ حساب پرداخت/خرید دولتی چندشناسه‌ای شناسایی پوزهای موجود در شبکه گزارش تراکنش‌های آخر
Bitel IC-3300, IC-3100	Serial	~~
Magic X8, X5	Serial	خرید (عادی، شناسه دار ۱۵ رقمی) بدون تسهیم (تک حسابی)

۴ خصوصیات

این قسمت خصوصیات کلاس PcPosBusiness توضیح داده شده است. در مثال‌های ارائه شده فرض بر این است که یک نمونه از این کلاس همانند زیر ایجاد شده است:

```
PcPosBusiness PcPos = new PcPosBusiness();
```

توجه شود در زمان بسته شدن نرم افزار یا اتمام کار حتماً متد **Dispose** از این شیء فراخوانی شود تا اگر منابع تخصیص داده شده آزاد شوند.

Ip ۴,۱

Prototype:	public string Ip
Function:	با استفاده از این خاصیت آدرس آی پی دستگاه پوز جهت ارسال و دریافت تعریف می گردد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}
Notes:	
Example:	PcPos.Ip = "192.168.100.1";

Port ۴,۲

Prototype:	public int Port
Function:	با استفاده از این خاصیت درگاه دستگاه پوز جهت ارسال و دریافت تعریف می گردد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}
Notes:	مقدار پیش فرض درگاه ۸۸۸۸ می باشد.
Example:	PcPos.Port = 8888;

RetryTimeout ۴,۳

Prototype:	public int[] RetryTimeout
Function:	تعداد و حداکثر مدت زمان جهت اتصال به پایانه را تعیین می نماید. بعد از گذشت این مدت تراکنش Timeout شده و خطای آن را بر می گرداند.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	از نسخه 5.3.xxx به بعد این خصوصیت کاربردی ندارد.
Example:	var ConnectionTimeout = 5000; PcPos.RetryTimeout = new int[] { ConnectionTimeout, ConnectionTimeout, ConnectionTimeout };

ResponseTimeout ۴,۴

Prototype:	public int[] ResponseTimeout
Function:	تعداد و حداکثر مدت زمان انتظار جهت دریافت پاسخ تراکنش از پوز را تعیین می نماید. بعد از گذشت این مدت تراکنش، Timeout شده و خطای آن را بر می گرداند. دفعات تکرار توسط طول آرایه و مدت زمان انتظار توسط مقدار آرایه در هر اندیس تعیین می شود.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
مهرکیت:	این متد توسط شرکت پرداخت الکترونیک سداد تولید شده است. هرگونه کپی برداری و یا استفاده دیگر از این سند بدون کسب مجوز کتبی از این شرکت ممنوع می باشد.
صفحه ۱۱ از ۵۰	

Notes:	از نسخه 5.3.xxx به بعد فقط اولین اندیس این خصوصیت کاربرد دارد.
Example:	<pre>var ConnectionTimeout = 5000; var ResponseTimeout = 180000; PcPos.ResponseTimeout = new int[] { ResponseTimeout, ConnectionTimeout, ConnectionTimeout };</pre>

Amount ۴,۵

Prototype:	public string Amount
Function:	مبلغ تراکنش را تعیین می نماید.
Be Applicable In:	All
Notes:	
Example:	PcPos.Amount = "10000";

PcPosDllVersion ۴,۶

Prototype:	public static string PcPosDllVersion
Function:	نسخه DLL مورد استفاده را تعیین می نماید.
Be Applicable In:	All
Notes:	
Example:	

SerialNo ۴,۷

Prototype:	public string PcPosSerialNo
Function:	شماره سریال پوز مورد استفاده را تعیین می نماید. در صورتی که شماره سریال پوز تغییر یابد خطا کلید تبادلی اعلام می شود.
Be Applicable In:	BlueBird {P3500, CT280, MT280}
Notes:	مقداردهی اشتباه این مشخصه در انجام تراکنش تاثیرگذار است.
Example:	

ConnectionType ۴,۸

Prototype:	public PcPosConnectionType ConnectionType
Function:	نوع ارتباط بین رایانه و دستگاه پوز را تعیین می نماید. کاربر می تواند از طریق درگاه سریال یا شبکه با دستگاه ارتباط برقرار نماید.
Be Applicable In:	All
Notes:	<pre>PcPosConnectionType.Serial = 0 PcPosConnectionType.Lan = 1</pre>
Example:	<pre>PcPos.PcPosMode = PcPosConnectionType.Serial; // 0 PcPos.PcPosMode = PcPosConnectionType.Lan; // 1</pre>

ComPortName ۴,۹

Prototype:	<code>public string ComPortName</code>
Function:	جهت تعیین درگاه سریال استفاده می شود.
Be Applicable In:	All
Notes:	
Example:	<code>PcPos.ComPortName = "COM2";</code>

Field1-8 ۴,۱۰

Prototype:	<code>public string Field1-8</code>
Function:	جهت تعیین اطلاعات تراکنش برای ارسال به درگاه پرداخت.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}
Notes:	اطلاعات اختیاری میتواند رشته ای از عدد و حرف باشد. تعداد این خصوصیت ۸ عدد میباشد که خصوصیات ۱ تا ۵ رزرو شده اند و توسط برنامه نویسی نهایی قابل استفاده نمیشوند.
Example:	<code>PcPos.Field1 = "16513219687";</code>

۵ متدها

قبل از مطالعه این بخش به نکات زیر توجه شود:

۱. در ارتباط سریال به کابل ارتباطی توجه شود. هر یک از برندها کابل مخصوص به خود را برای ارتباط با پایانه دارند.
۲. تنظیمات اتصال (سریال و شبکه) و چک کردن الزامات تراکنش در فایل SadadConfig.xml قرار دارد. در صورت عدم وجود این فایل در مسیر رابط توسعه (DLL) تنظیمات پیش فرض در نظر گرفته میشود.
۳. به ازاء هر پایانه (پوز) میتوانیم یک یا چند حساب داشته باشیم، بنابراین پایانه میتواند به صورت تک حسابی یا چندحسابی تنظیم شود. این تنظیمات در مرکز (سوئیچ پرداخت شرکت خدمات الکترونیک سداد) انجام می‌پذیرد و به ازاء هر تراکنش، این مشخصه بررسی می‌شود، لذا مشتری (دارنده پوز) با درخواست کتبی تعیین میکند یک یا چند حساب به ازاء هر پایانه وجود داشته باشد. حال اگر پایانه تک حسابی باشد و از پی‌سی‌پوز درخواست خرید بدون ورود تسهیم یا اگر پایانه چند حسابی باشد و اطلاعات تسهیم صحیح به پایانه فرستاده شود، تراکنش با موفقیت انجام می‌پذیرد؛ در غیر این صورت خطای شماره ۳۰ (فرمت اشتباه است) برگشت داده میشود. برای کار با پایانه (پوز)هایی که به صورت چند حسابی در سیستم پرداخت شرکت خدمات الکترونیک سداد تنظیم شده اند لطفا قسمت‌های ۵،۱، ۵،۲، ۵،۸، ۵،۹ و ۵،۱۰ مطالعه شود.

۴. دلایل خطای شماره ۳۰:

۱. اگر پایانه (پوز) تک حسابی باشد و از پی‌سی‌پوز اطلاعات تسهیم فرستاده شود.
 ۲. اگر پایانه (پوز) چند حسابی باشد و از پی‌سی‌پوز اطلاعات تسهیم پر نشود.
 ۳. اگر پایانه (پوز) چند حسابی باشد و از پی‌سی‌پوز اطلاعات تسهیم عمدا یا سهوا اشتباه پر شود. مثلا شماره اندیس حساب اشتباه وارد شود، یا اندیسی وارد شود که وجود ندارد، یا شماره شبای حسابی وارد شود که در سوئیچ پرداخت ثبت نشده است، یا مجموع درصد برابر با ۱۰۰ نباشد، و یا مجموع مبالغ تسهیم با مبلغ تراکنش برابر نباشد.
- موارد متعددی پیش آمده است که مشتری درخواست افزایش حساب و تبدیل دستگاه به چندحسابی به شرکت پرداخت الکترونیک سداد نداده است ولی در تراکنش اطلاعات تسهیم را می‌فرستد یا بالعکس.

۵،۱ ConvertIbans

Prototype:	<code>public List<int> ConvertIbans(string terminalId, string merchantId, List<string> ibans, string url = null)</code>	
Function:	تبدیل لیست شماره شبا به لیست RowIndex.	
Parameter:	terminalId	کد یا شماره پایانه (۸ رقمی)
Parameter:	merchantId	کد یا شماره پذیرنده (۱۵ رقمی)
Parameter:	ibans	لیستی از نوع string که شامل شماره حساب شبا (۲۶ رقمی با IR) می‌باشد.
Parameter:	url	آدرس ارسال و دریافت اطلاعات
Return:	لیستی از نوع int برمیگرداند که به ازاء هر مقدار از ورودی iban یک شماره اختصاص میدهد که همان RowIndex برای تسهیم چند حسابی میباشد. اعداد بالای ۰ معتبر میباشند و اعداد منفی بیانگر عدم وجود شماره شبا در سیستم میباشد.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}	
Notes:	اجرای این متد نیاز به اتصال اینترنت دارد.	

محرکیت:

این سند توسط شرکت پرداخت الکترونیک سداد تولید شده است. هرگونه کپی برداری و یا استفاده دیگر از این سند بدون کسب مجوز کتبی از این شرکت ممنوع می‌باشد.

	توسعه دهندگان میتوانند برای راحتی کار مشتریان خود شماره ردیف و شبا را در تنظیمات نرم افزار خود ذخیره کنند تا نیاز به اتصال به اینترنت را به حداقل برسانند. در صورتیکه مقدار url معادل null باشد مقدار پیشفرض در نظر گرفته میشود. امکان برگشت خطا با استثنائات وب مثلا WebException وجود دارد. اگر شماره پذیرنده کمتر از ۱۵ رقم باشد با قرار دادن عدد صفر به ابتدا طول آن را به ۱۵ رقم رساند: 0000000000000301
Example:	<pre>var idxs = new List<string>(); foreach (var s in txtIbanDevide.Text.Split(',')) { var iban = s.Split(':')[0]; idxs.Add(iban); } if (PcPos == null) PcPos = new PcPosBusiness(); var mmpds = PcPos.ConvertIbans(terminalId, merchantId, idxs);</pre>

ConvertIbans ۵,۲

Prototype:	<pre>public List<MultiAccountPosDevider> ConvertIbans(string terminalId, string merchantId, List<MultiAccountPosDeviderEx> multiAccountValueEx, string url = null)</pre>	
Function:	تبدیل لیست شماره شبا به همراه اطلاعات تسهیم به لیست از نوع MultiAccountPosDevider.	
Parameter:	terminalId	کد یا شماره پایانه (۸ رقمی)
Parameter:	merchantId	کد یا شماره پذیرنده (۱۵ رقمی)
Parameter:	multiAccountValueEx	لیستی از نوع MultiAccountPosDeviderEx که شامل شماره حساب شبا و اطلاعات تسهیم می باشد.
Parameter:	url	آدرس ارسال و دریافت اطلاعات
Return:	لیستی از نوع MultiAccountPosDevider برمیگرداند که شامل اطلاعات تسهیم می باشد. در مقدار Index اعداد بالای ۰ معتبر میباشند و اعداد منفی بیانگر عدم وجود شماره شبا در سیستم میباشد.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}	
Notes:	<p>اجرای این متد نیاز به اتصال اینترنت دارد.</p> <p>توسعه دهندگان میتوانند برای راحتی کار مشتریان خود شماره ردیف و شبا را در تنظیمات نرم افزار خود ذخیره کنند تا نیاز به اتصال به اینترنت را به حداقل برسانند. در صورتیکه مقدار url معادل null باشد مقدار پیشفرض در نظر گرفته میشود. امکان برگشت خطا با استثنائات وب مثلا WebException وجود دارد. اگر شماره پذیرنده کمتر از ۱۵ رقم باشد با قرار دادن عدد صفر به ابتدا طول آن را به ۱۵ رقم رساند: 0000000000000301</p>	
Example:	<pre>var mmpdxx = new List<MultiAccountPosDeviderEx>(); foreach (var s in txtIbanDevide.Text.Split(',')) { var mmpdx = new MultiAccountPosDeviderEx() { Iban = s.Split(':')[0], Value = s.Split(':')[1] }; mmpdxx.Add(mmpdx); } if (PcPos == null) PcPos = new PcPosBusiness();</pre>	

	<code>var mmpds = PcPos.ConvertIbans(terminalId, merchantId, mmpdxs);</code>
--	--

SetSaleId ۵,۳

Prototype:	<code>public void SetSaleId(string saleId)</code>	
Function:	شناسه خرید یا استعلام را تعیین می نماید.	
Parameter:	saleId	حداکثر طول شناسه خرید ۴۰ بایت می باشد.
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}	
Notes:	در صورت ورود داده نامعتبر <code>PcPosBadInputException</code> به عنوان خطا برگشت داده می شود. برای خالی کردن مقدار <code>SaleId</code> از تابع <code>ClearSaleId</code> استفاده شود.	
Example:	<code>PcPos.SetSaleId("21700913001001590");</code>	

SetMultiSaleId ۵,۴

Prototype:	<code>public void SetMultiSaleId(SaleIdProvider multiSaleId)</code>	
Function:	شناسه های خرید یا استعلام را برای پرداخت دولتی تعیین می نماید.	
Parameter:	multiSaleId	کلاس تعیین شناسه های استعلامی
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}	
Notes:	<p>در صورت ورود داده نامعتبر <code>PcPosBadInputException</code> به عنوان خطا برگشت داده می شود.</p> <p>برای خالی کردن مقدار <code>SaleId</code> از تابع <code>ClearMultiSaleId</code> استفاده شود.</p> <p>تعداد شناسه های مجاز همزمان ۱۰ شناسه می باشد.</p> <p>برای استفاده از این تابع کلاس مجزایی معین شده است که تقریباً همانند متد <code>SetMultiAccount</code> {۵,۸}، {۵,۹}، {۵,۱۰}، <code>Error! Reference source not found</code> استفاده می شود. یعنی هم می شود با شماره ردیف حساب در سوئیچ پرداخت و هم با ورود شماره شباه اقدام به پر کردن داده های تراکنش کرد.</p> <p>در صورت استفاده از شماره ردیف حساب، باید شماره ردیف ثبت شده برای حساب دولتی در سوئیچ پرداخت شرکت پرداخت الکترونیک سداد، شناسه مورد نظر، و مبلغ واریزی هر شناسه و حساب تعیین گردد.</p> <p>برای گرفتن شماره ردیف حساب می توان از متد <code>ConvertIbans</code> {۵,۱} استفاده کرده و در تنظیمات نرم افزار ذخیره کرد تا مجدداً استفاده نمایند.</p> <p>برای معرفی شناسه های خرید باید <code>RowType = Private</code> باشد.</p> <p>لطفاً ۸,۸ و ۸,۹ مطالعه شود.</p>	
Example:	<pre> SaleIdProvider multiSaleId = new SaleIdProvider(FormatType.Bcd); decimal sum = 0; foreach (DataGridViewRow row in dataGridViewMultiSaleId.Rows) { try { var idx = int.Parse(row.Cells[0].Value.ToString()); var saleId = row.Cells[1].Value.ToString(); var amount = decimal.Parse(row.Cells[2].Value.ToString()); sum += amount; } } </pre>	

	<pre> var data = new SaleIdData(idx, saleId, amount, RowType.Private); multiSaleId.SaleIds.Add(data); } catch (Exception) { } } PcPos.Amount = sum.ToString(); PcPos.SetMultiSaleId(multiSaleId); </pre>
--	---

SetOrderId ۵,۵

Prototype:	public bool SetOrderId(string orderId)	
Function:	شناسه تراکنش را تعیین می نماید.	
Parameter:	orderId	حداکثر طول شناسه تراکنش ۹۹۹ حرف می باشد.
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}	
Notes:	در صورت ورود داده نامعتبر PcPosBadInputException به عنوان خطا برگشت داده می شود. برای خالی کردن مقدار OrderId از تابع ClearOrderId استفاده شود.	
Example:	PcPos.SetOrderId("96541615961");	

SetBillInfo ۵,۶

Prototype:	public bool SetSaleId(string billId, string payId)	
Function:	شناسه پرداخت و شناسه قبض را تعیین می نماید.	
Parameter:	billId	شناسه قبض
Parameter:	payId	شناسه پرداخت
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}	
Notes:	در صورت ورود داده نامعتبر PcPosBadInputException به عنوان خطا برگشت داده می شود. برای خالی کردن مقدار BillInfo از تابع ClearBillInfo استفاده شود.	
Example:	PcPos.SetBillInfo("12345678900123", "987654321");	

SetFixedDuty ۵,۷

Prototype:	public bool SetFixedDuty(int providerId, int serviceCode, int count)	
Function:	داده مربوط به تعرفه را تعیین می نماید.	
Parameter:	providerId	شناسه تامین کننده (سه رقم)
Parameter:	serviceCode	شناسه سرویس (چهار رقم)
Parameter:	count	تعداد (سه رقم)

مهرکیت:	این سند توسط شرکت پرداخت الکترونیک سداد تولید شده است. هرگونه کپی برداری و یا استفاده دیگر از این سند بدون کسب مجوز کتبی از این شرکت ممنوع می باشد.
صفحه ۱۷ از ۵۰	

Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.
Be Applicable In:	BlueBird {P3500, CT280, MT280} DEMO
Notes:	در صورت ورود داده نامعتبر PcPosBadInputException به عنوان خطا برگشت داده می شود. برای خالی کردن مقدار FixedDuty از تابع ClearFixedDuty استفاده شود.
Example:	PcPos.SetBillInfo(1234, 74, 5);

SetMultiAccountData ۵,۸

Prototype:	public bool SetMultiAccountData(List<MultiAccountPosDevider> multiAccountValue, MultiAccountMode MultiAccountMode, out int errorNo)	
Function:	مبالغ و حساب های خرید چند حسابی و همچنین نوع تسهیم را تعیین می نماید.	
Parameter:	multiAccountValue	لیستی از نوع MultiAccountPosDevider که مبلغ یا درصد سهم حساب با اندیس مورد نظر را تعیین می نماید.
Parameter:	multiAccountMode	نوع تسهیم در تراکنش را تعیین می نماید.
Parameter:	errorNo	در صورت وجود خطا شماره خطا در این مشخصه قرار می گیرد.
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}	
Notes:	<p>در صورت ورود داده نامعتبر PcPosBadInputException به عنوان خطا برگشت داده می شود.</p> <p>فرض تسهیم توسط کاربر/برنامه نویس بر این مدل می باشد:</p> <p>مبلغ ورودی: ۱۲۰۰۰ ریال</p> <p>درصدی به صورت <Index>:<Percent> مقداردهی میشود: "1:20,2:30:3:50"</p> <p>مبلغی به صورت <Index>:<Amount> مقداردهی میشود: "1:5000,2:6000,3:1000"</p> <p>شماره های ۱ و ۲ و ۳ اندیس حساب هایی هستند که در سوئیچ پرداخت تعیین شده است. این توافق در زمان انعقاد قرارداد مشتری با شرکت انجام می پذیرد و با درخواست دوباره مشتری قابل تغییر است. همچنین برای جلوگیری از خطای پیچیدگی استفاده و تغییر اندیس میتوان از شماره شبا (IBAN) با ورود IR ابتدای آن استفاده نمود که باید به ۵,۱۰ مراجعه نمود.</p> <p>مقادیر قابل پذیرش برای multiAccountMode داده های عددی زیر می باشد:</p> <p>۱. AmountOld = 3</p> <p>۲. PercentOld = 2</p> <p>۳. PercentNew = 4</p> <p>۴. AmountNew = 5</p> <p>برای خالی کردن مقدار MultiAccount از تابع ClearMultiAccount استفاده شود.</p>	
Example:	<pre>if (!string.IsNullOrEmpty(txtMultiAccount.Text)) { var listTimeOutDic = new List<MultiAccountPosDevider>(); foreach (var s in txtMultiAccount.Text.Split(',')) { var mmpd = new MultiAccountPosDevider() { Value = s.Split(':')[0], Index = int.Parse(s.Split(':')[1]) }; } }</pre>	

	<pre> listTimeOutDic.Add(mmpd); } if (cmbxDivideType.SelectedItem.ToString().Contains("1")) // قدیم درصدی { if (!PcPos.SetMultiAccountData(listTimeOutDic, MultiAccountMode.PercentOld, out multiAccountErrorCode)) { MessageBox.Show("Invalid input for Multi-Merchant data."); return; } } else if (cmbxDivideType.SelectedItem.ToString().Contains("2")) // قدیم مبلغی { if (!PcPos.SetMultiAccountData(listTimeOutDic, MultiAccountMode.AmountOld, out multiAccountErrorCode)) { MessageBox.Show("Invalid input for Multi-Merchant data."); return; } } else if (cmbxDivideType.SelectedItem.ToString().Contains("3")) // جدید درصدی { if (!PcPos.SetMultiAccountData(listTimeOutDic, MultiAccountMode.PercentNew, out multiAccountErrorCode)) { MessageBox.Show("Invalid input for Multi-Merchant data."); return; } } else if (cmbxDivideType.SelectedItem.ToString().Contains("4")) // جدید مبلغی { if (!PcPos.SetMultiAccountData(listTimeOutDic, MultiAccountMode.AmountNew, out multiAccountErrorCode)) { MessageBox.Show("Invalid input for Multi-Merchant data."); return; } } } </pre>
--	---

SetMultiAccountsData ۵,۹

Prototype:	<pre> public bool SetMultiAccountData (string terminalId, string merchantId, List<MultiAccountPosDeviderEx> multiAccountValueEx, MultiAccountMode, out int pErrorNo) </pre>	
Function:	مبالغ و حساب‌های خرید چند حسابی و همچنین نوع تسهیم را تعیین می‌نماید.	
Parameter:	terminalId	کد یا شماره پایانه
Parameter:	merchantId	کد یا شماره پذیرنده

Parameter:	multiAccountValue	لیستی از نوع MultiAccountPosDeviderEx که مبلغ یا درصد سهم حساب با شماره شبای مورد نظر را تعیین می نماید.
Parameter:	multiAccountMode	نوع تسهیم در تراکنش را تعیین می نماید.
	errorNo	در صورت وجود خطا شماره خطا در این مشخصه قرار می گیرد.
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}	
Notes:	<p>در صورت ورود داده نامعتبر MinimumValueException، PcPosBadInputException یا MaximumValueException به عنوان خطا برگشت داده می شود.</p> <p>فرض تسهیم توسط کاربر/برنامه نویس بر این مدل میباشد:</p> <p>مبلغ ورودی: ۱۲۰۰۰ ریال</p> <p>درصدی به صورت <IBAN>:<Percent> مقداردهی میشود:</p> <p>"IR350170000000106928854003:20, IR350170000000106928854006:30: IR430170000002171291343008:50"</p> <p>مبلغی به صورت <IBAN>:<Amount> مقداردهی میشود:</p> <p>"IR350170000000106928854003:5000, IR350170000000106928854006:6000, IR430170000002171291343008:1000"</p> <p>اجرای این متد نیاز به اتصال اینترنت دارد.</p> <p>توسعه دهندگان میتوانند برای راحتی کار مشتریان خود شماره ردیف و شبا را در تنظیمات نرم افزار خود ذخیره کنند تا نیاز به اتصال به اینترنت را به حداقل برسانند.</p> <p>برای خالی کردن مقدار MultiAccount از تابع ClearMultiAccount استفاده شود.</p>	
Example:	<pre>int multiAccountErrorCode = 0; if (!string.IsNullOrEmpty(txtMultiAccount.Text)) { var terminalId = "C0000301"; var merchantId = "M000000000000152"; var mmpdxs = new List<MultiAccountPosDeviderEx>(); foreach (var s in txtMultiAccount.Text.Split(',')) { var mmpd = new MultiAccountPosDeviderEx() { Iban = s.Split(':')[0], Index = int.Parse(s.Split(':')[1]) }; mmpdxs.Add(mmpd); } if (cmbxDivideType.SelectedItem.ToString().Contains("1")) // قدیم درصدی { PcPos.SetMultiAccountData(terminalId, merchantId, mmpdxs, MultiAccountMode.PercentOld, out multiAccountErrorCode); } else if (cmbxDivideType.SelectedItem.ToString().Contains("2")) // قدیم مبلغی { </pre>	

	<pre> PcPos.SetMultiAccountData(terminalId, merchantId, mmpdxs, MultiAccountMode.AmountOld, out multiAccountErrorCode); } else if (cmbxDivideType.SelectedItem.ToString().Contains("3")) // جدید درصدی { PcPos.SetMultiAccountData(terminalId, merchantId, mmpdxs, MultiAccountMode.PercentNew, out multiAccountErrorCode); } else if (cmbxDivideType.SelectedItem.ToString().Contains("4")) // جدید مبلغی { PcPos.SetMultiAccountData(terminalId, merchantId, mmpdxs, MultiAccountMode.AmountNew, out multiAccountErrorCode); } } </pre>
--	---

SetMultiAccountData ۵,۱۰

Prototype:	<pre> public bool SetMultiAccountData (List<MultiAccountPosDividerEx> multiAccountValueEx, MultiAccountMode, out int pErrorNo) </pre>	
Function:	مبالغ و حساب‌های خرید چند حسابی و همچنین نوع تسهیم را تعیین می‌نماید.	
Parameter:	multiAccountValue	لیستی از نوع MultiAccountPosDividerEx که مبلغ یا درصد سهم حساب با شماره شبای مورد نظر را تعیین می‌نماید.
Parameter:	multiAccountMode	نوع تسهیم در تراکنش را تعیین می‌نماید.
	errorNo	در صورت وجود خطا شماره خطا در این مشخصه قرار می‌گیرد.
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی‌گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel { IC3300 , IC3100 }	
Notes:	<p>در صورت ورود داده نامعتبر MinimumValueException، PcPosBadInputException یا MaximumValueException به عنوان خطا برگشت داده می‌شود.</p> <p>فرض تسهیم توسط کاربر/برنامه نویس بر این مدل می‌باشد:</p> <p>مبلغ ورودی: ۱۲۰۰۰ ریال</p> <p>درصدی به صورت <IBAN>:<Percent> مقداردهی میشود:</p> <p>"IR350170000000106928854003:20, IR350170000000106928854006:30: IR430170000002171291343008:50"</p> <p>مبلغی به صورت <IBAN>:<Amount> مقداردهی میشود:</p> <p>"IR350170000000106928854003:5000, IR350170000000106928854006:6000, IR430170000002171291343008:1000"</p> <p>برای خالی کردن مقدار MultiAccount از تابع ClearMultiAccount استفاده شود.</p>	
Example:	<pre> if (!string.IsNullOrEmpty(txtMultiAccount.Text)) { var listTimeOutDicEx = new List<MultiAccountPosDividerEx>(); foreach (var s in txtMultiAccount.Text.Split(',')) { </pre>	

```

        if (txtMultiAccount.Text.StartsWith("IR",
StringComparison.InvariantCultureIgnoreCase))
        {
            var mmpd = new MultiAccountPosDividerEx()
            {
                Iban = s.Split(':')[0],
                Value = s.Split(':')[1]
            };
            listTimeOutDicEx.Add(mmpd);
        }
    }

    int multiAccountErrorCode = 0;
    if (cmbxDivideType.SelectedItem.ToString().Contains("1")) //
    قدیم درصدی
    {
        if (txtMultiAccount.Text.StartsWith("IR",
StringComparison.InvariantCultureIgnoreCase))
        {
            if (!PcPos.SetMultiAccountData(listTimeOutDicEx,
MultiAccountMode.PercentOld, out multiAccountErrorCode))
            {
                MessageBox.Show("Invalid input for Multi-Merchant
data.");
                return;
            }
        }
    }
    else if (cmbxDivideType.SelectedItem.ToString().Contains("2")) //
    قدیم مبلغی
    {
        if (txtMultiAccount.Text.StartsWith("IR",
StringComparison.InvariantCultureIgnoreCase))
        {
            if (!PcPos.SetMultiAccountData(listTimeOutDicEx,
MultiAccountMode.AmountOld, out multiAccountErrorCode))
            {
                MessageBox.Show("Invalid input for Multi-Merchant
data.");
                return;
            }
        }
    }
    else if (cmbxDivideType.SelectedItem.ToString().Contains("3")) //
    جدید درصدی
    {
        if (txtMultiAccount.Text.StartsWith("IR",
StringComparison.InvariantCultureIgnoreCase))
        {
            if (!PcPos.SetMultiAccountData(listTimeOutDicEx,
MultiAccountMode.PercentNew, out multiAccountErrorCode))
            {
                MessageBox.Show("Invalid input for Multi-Merchant
data.");
                return;
            }
        }
    }
    else if (cmbxDivideType.SelectedItem.ToString().Contains("4")) //
    جدید مبلغی
    {

```

	<pre> if (txtMultiAccount.Text.StartsWith("IR", StringComparison.InvariantCultureIgnoreCase)) { if (!PcPos.SetMultiAccountData(listTimeOutDicEx, MultiAccountMode.AmountNew, out multiAccountErrorCode)) { MessageBox.Show("Invalid input for Multi-Merchant data."); return; } } } </pre>
--	--

SetAdvertisement ۵,۱۱

Prototype:	public bool SetAdvertisement(string advertisementData)	
Function:	متن برای چاپ انتهای رسید از پوز را مشخص میکند.	
Parameter:	advertisementData	طول متن تا ۵۰۰ کاراکتر میباشد.
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100} Should ask from support center.	
Notes:	برای خالی کردن مقدار Advertisement از تابع ClearAdvertisement استفاده شود.	
Example:	<pre> var sb = new StringBuilder(); sb.AppendLine("نام: علی"); sb.AppendLine("نام: محمدی"); sb.AppendLine("کدملی: 1234567890"); sb.AppendLine("سفارش شماره: 9876543210"); PcPos.SetAdvertisement(sb.ToString()); </pre>	

SetCardInfo ۵,۱۲

Prototype:	public bool SetCardInfo(string cardId)	
Function:	مشخصات Hash شده کارت را مقداردهی می نماید.	
Parameter:	cardInfo	طول رشته Hash شده ۶۴ میباشد.
Return:	در صورت وجود خطا مقدار false و در غیر این صورت true برمی گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}	
Notes:	برای خالی کردن مقدار CardInfo از تابع ClearCardInfo استفاده شود.	
Example:	<pre> var hashedInfo = "532c8f979678475185d63fa5b36b7830b4ee56ceb178e45136a58ae79f378273"; PcPos.SetBillInfo(hashedInfo); </pre>	

SetSearchParam ۵,۱۳

Prototype:	public void SetSearchParam(Dictionary<SearchParam, string> param)	
Function:	موارد جستجو را مقداردهی می نماید.	
Parameter:	param	پارامترهای جستجو
Return:	فاقد مقدار برگشتی می باشد.	

مهرکیت:	این سند توسط شرکت پرداخت الکترونیک سداد تولید شده است. هرگونه کپی برداری و یا استفاده دیگر از این سند بدون کسب مجوز کتبی از این شرکت ممنوع می باشد.
صفحه ۵۰ از ۳۳	

Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	
Example:	PcPos.SetSearchParam(new Dictionary<SearchParam, string>() { { SearchParam.OrderId, txtSearchOrderId.Text } });

GetAvailableSerialPorts ۵,۱۴

Prototype:	public [static] List<string> GetAvailableSerialPorts()
Function:	درگاه‌های سریال موجود در رایانه را برمی‌گرداند.
Parameter:	فاقد پارامتر می‌باشد.
Return:	فاقد مقدار برگشتی می‌باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	
Example:	cmbSerialPort.Items.Clear(); var objArr = new List<object>(PcPosBusiness.GetAvailableSerialPorts().ToArray()); cmbSerialPort.Items.AddRange(objArr.ToArray());

AsyncSaleTransaction ۵,۱۵

Prototype:	public void AsyncSaleTransaction()
Function:	این متد جهت ارسال درخواست تراکنش به صورت آسنکرون به پایانه می‌باشد. در حالت آسنکرون بایستی EventHandler<PosResult> OnSaleResult جهت دریافت نتیجه‌ی تراکنش پیش‌تر تعریف شده باشد. در صورتیکه کد پاسخ {ResponseCode} مقدار "۰۰" باشد تراکنش موفق و در غیر این صورت تراکنش ناموفق می‌باشد. توسعه دهندگان باید قبل از بررسی مشخصه ResponseCode برای تراکنش‌های موفق باید مشخصه صحت ارتباط رایانه با دستگاه پوز {PcPosStatus} را نیز چک کنند.
Parameter:	فاقد پارامتر می‌باشد.
Return:	فاقد مقدار برگشتی می‌باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد OnSaleResult را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد.
Example:	PcPos.Amount = txtPayAmount.Text; //set result call back PcPos.OnSaleResult += PcPosSaleResult; //send request PcPos.AsyncSaleTransaction();

SyncSaleTransaction ۵,۱۶

Prototype:	public PosResult SyncSaleTransaction()
Function:	این متد جهت ارسال درخواست تراکنش به صورت سنکرون به پایانه می‌باشد. در حالت سنکرون نتیجه‌ی تراکنش از طریق مقدار برگشتی در دسترس می‌باشد.

مهرکیت:	این سند توسط شرکت پرداخت الکترونیک سداد تولید شده است. هرگونه کپی برداری و یا استفاده دیگر از این سند بدون کسب مجوز کتبی از این شرکت ممنوع می‌باشد.
صفحه ۵۰ از ۲۴	

Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}
Notes:	
Example:	<pre>PcPos.Amount = txtPayAmount.Text; //send request var result = PcPos.SyncSaleTransaction();</pre>

۵,۱۷ AsyncBillPaymentTransaction

Prototype:	<code>public void AsyncBillPaymentTransaction()</code>
Function:	این متد جهت ارسال درخواست پرداخت قبض به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>EventHandler<PosResult> OnBillPaymentResult</code> جهت دریافت نتیجه ی تراکنش پیش تر تعریف شده باشد. در صورتیکه کد پاسخ <code>{ResponseCode}</code> مقدار "۰۰" باشد تراکنش موفق و در غیر این صورت تراکنش ناموفق می باشد. توسعه دهندگان باید قبل از بررسی مشخصه <code>ResponseCode</code> برای تراکنش های موفق باید مشخصه صحت ارتباط رایانه با دستگاه پوز <code>{PcPosStatus}</code> را نیز چک کنند.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد <code>OnBillPaymentResult</code> را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد. پیش از فراخوانی این تابع شناسه ی قبض و خرید را از طریق <code>SetBillInfo</code> مشخص نمایید.
Example:	<pre>PcPos.Amount = BillAmount.Text; //set result call back PcPos.OnBillPaymentResult += PcPosPayBillResult; //send request PcPos.SetBillInfo(BillId.Text, BillPayId.Text); PcPos.AsyncBillPaymentTransaction();</pre>

۵,۱۸ AsyncInquiredIdentifiedTransaction

Prototype:	<code>public void AsyncInquiredIdentifiedTransaction()</code>
Function:	این متد جهت ارسال درخواست خرید شناسه دار با استعلام به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>EventHandler<PosResult> OnInquiredIdentifiedSaleResult</code> جهت دریافت نتیجه ی تراکنش پیش تر تعریف شده باشد. در صورتیکه کد پاسخ <code>{ResponseCode}</code> مقدار "۰۰" باشد تراکنش موفق و در غیر این صورت تراکنش ناموفق می باشد. توسعه دهندگان باید قبل از بررسی مشخصه <code>ResponseCode</code> برای تراکنش های موفق باید مشخصه صحت ارتباط رایانه با دستگاه پوز <code>{PcPosStatus}</code> را نیز چک کنند.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}

Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد <code>EventHandler<PosResult></code> <code>OnInquiredIdentifiedSaleResult</code> را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد. پیش از فراخوانی این تابع شناسه‌ی خرید را از طریق <code>SetSaleId</code> مشخص نمایید.
Example:	<pre>//set result call back PcPos.OnInquiredIdentifiedSaleResult += PcPos_OnInquiredSaleResult; PcPos.SetSaleId(txtInquiredSaleId.Text); //send request PcPos.AsyncInquiredIdentifiedTransaction();</pre>

۵.۱۹ AsyncGovernmentInquiredIdentifiedTransaction

Prototype:	<code>public void AsyncGovernmentInquiredIdentifiedTransaction()</code>
Function:	این متد جهت ارسال درخواست خرید شناسه دار با اعلام به صورت آسنکرون به پایانه می‌باشد. در حالت آسنکرون بایستی <code>EventHandler<PosResult></code> <code>OnGovernmentInquiredIdentifiedSaleResult</code> جهت دریافت نتیجه‌ی تراکنش پیش‌تر تعریف شده باشد. در صورتیکه کد پاسخ <code>{ResponseCode}</code> مقدار "۰۰" باشد تراکنش موفق و در غیر این صورت تراکنش ناموفق می‌باشد. توسعه دهندگان باید قبل از بررسی مشخصه <code>ResponseCode</code> برای تراکنش‌های موفق باید مشخصه صحت ارتباط رایانه با دستگاه پوز <code>{PcPosStatus}</code> را نیز چک کنند.
Parameter:	فاقد پارامتر می‌باشد.
Return:	فاقد مقدار برگشتی می‌باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد <code>OnGovernmentInquiredIdentifiedSaleResult</code> را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد. از نسخه ۵.۳.۰ به بعد پرداخت دولتی چند شناسه ای تا ۱۰ شناسه و مبلغ میسر شده است. بنابراین پیش از فراخوانی این تابع شناسه‌ی خرید را از طریق <code>SetSaleId</code> یا <code>SetMultiSaleId</code> مشخص نمایید.
Example:	<pre>//set result call back PcPos.OnGovernmentInquiredIdentifiedSaleResult += PcPos_OnGovernmentInquiredIdentifiedSaleResult; if (radioOneGovSaleId.Checked) { if (!string.IsNullOrEmpty(txtSaleAmountInquiredSaleOrganization.Text)) PcPos.Amount = txtSaleAmountInquiredSaleOrganization.Text; //set transaction id PcPos.SetSaleId(txtSaleIdInquiredSaleOrganization.Text); } else { SaleIdProvider multiSaleId = new SaleIdProvider(FormatType.Ascii); decimal sum = 0; foreach (DataGridViewRow row in dataGridViewMultiSaleId.Rows) { try { var idx = int.Parse(row.Cells[0].Value.ToString());</pre>

	<pre> var saleId = row.Cells[1].Value == null ? string.Empty : row.Cells[1].Value.ToString(); var amount = decimal.Parse(row.Cells[2].Value.ToString()); sum += amount; var data = new SaleIdData(idx, saleId, amount, RowType.Private); multiSaleId.SaleIds.Add(data); } catch (Exception) { } } PcPos.Amount = sum.ToString(); PcPos.SetMultiSaleId(multiSaleId); } //send request PcPos.AsyncGovernmentInquiredIdentifiedTransaction(); </pre>
--	--

۵,۲۰ AsyncFixedDuty

Prototype:	<code>public void AsyncFixedDuty()</code>
Function:	این متد جهت ارسال درخواست خرید با تعرفه به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>EventHandler<PosResult> OnSaleResult</code> جهت دریافت نتیجه ی تراکنش پیش تر تعریف شده باشد. در صورتیکه کد پاسخ <code>{ResponseCode}</code> مقدار "۰۰" باشد تراکنش موفق و در غیر این صورت تراکنش ناموفق می باشد. توسعه دهندگان باید قبل از بررسی مشخصه <code>ResponseCode</code> برای تراکنش های موفق باید مشخصه صحت ارتباط رایانه با دستگاه پوز <code>{PcPosStatus}</code> را نیز چک کنند.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280} DEMO
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد <code>OnSaleResult</code> را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد.
Example:	<pre> PcPos.SetFixedDuty((int)numericProviderId.Value, (int)numericServiceCode.Value, (int)numericFixedDutyCount.Value); PcPos.SetOrderId(txtInquiredOrderId.Text); //set result call back PcPos.OnSaleResult += PcPosSaleResult; //send request PcPos.AsyncFixedDuty(); </pre>

۵,۲۱ AsyncCommodityBasket

Prototype:	<code>public void AsyncCommodityBasket()</code>
Function:	این متد جهت ارسال درخواست خرید بسته سبد کالا به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>EventHandler<PosResult> OnSaleResult</code> جهت دریافت نتیجه ی تراکنش پیش تر

	تعریف شده باشد. در صورتیکه کد پاسخ {ResponseCode} مقدار "۰۰" باشد تراکنش موفق و در غیر این صورت تراکنش ناموفق می باشد. توسعه دهندگان باید قبل از بررسی مشخصه ResponseCode برای تراکنش های موفق باید مشخصه صحت ارتباط رایانه با دستگاه پوز {PcPosStatus} را نیز چک کنند.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد OnSaleResult را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد.
Example:	<pre>PcPos.Amount = txtPayAmount.Text; //set result call back PcPos.OnSaleResult += PcPosSaleResult; //send request PcPos.AsyncCommodityBasket();</pre>

۵,۲۲ AsyncFoodSafety

Prototype:	<code>public void AsyncFoodSafety()</code>
Function:	این متد جهت ارسال درخواست خرید بسته امنیت غذایی به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>OnSaleResult <EventHandler<PosResult></code> جهت دریافت نتیجه ی تراکنش پیش تر تعریف شده باشد. در صورتیکه کد پاسخ {ResponseCode} مقدار "۰۰" باشد تراکنش موفق و در غیر این صورت تراکنش ناموفق می باشد. توسعه دهندگان باید قبل از بررسی مشخصه ResponseCode برای تراکنش های موفق باید مشخصه صحت ارتباط رایانه با دستگاه پوز {PcPosStatus} را نیز چک کنند.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد OnSaleResult را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد.
Example:	<pre>PcPos.Amount = txtPayAmount.Text; //set result call back PcPos.OnSaleResult += PcPosSaleResult; //send request PcPos.AsyncFoodSafety();</pre>

۵,۲۳ AsyncCardInfo

Prototype:	<code>public void AsyncCardInfo()</code>
Function:	این متد جهت ارسال درخواست "گرفتن اطلاعات کارت" یا "تراکنش با نشان پرداخت" به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>OnPcPosCardInfoResult <EventHandler<PosCardInfoResult></code> جهت دریافت نتیجه ی تراکنش پیش تر تعریف شده باشد. در صورتیکه عملیات موفق باشد و مقدار خصوصیت Toggle برابر ۱ باشد مشخصات Hash شده برگردانده میشود و در غیر

	این صورت مقدار مشخصات کارت خالی فرستاده میشود و در صورتیکه مقدار خصوصیت Toggle برابر ۲ باشد مشخصات تراکنش نشان اعتباری فرستاده میشود.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد OnPcPosCardInfoResult را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد. توسعه دهندگانی که گرفتن برخی اطلاعات دستگاه مثل شماره درگاه (TerminalId) یا شماره پذیرنده برای انجام تراکنش هایشان مهم است میتوانند با این متد کار پرداخت یا خرید را آغاز کنند.
Example:	<pre> PcPos.Amount = txtPayAmount.Text; if (checkBoxCreateRandomOrderId.Checked) { txtOrderId.Text = GetUnixTime(); } PcPos.SetOrderId(txtOrderId.Text); //set result call back PcPos.OnPcPosCardInfoResult += PcPos_OnPcPosCardInfoResult; //send request PcPos.AsyncCardInfo(); </pre>

۵,۲۴ AsyncTransactionLogs

Prototype:	<code>public void AsyncTransactionLogs()</code>
Function:	این متد جهت ارسال درخواست گزارش تراکنش های انجام شده به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>OnLogResult <PosLogsResult> EventHandler</code> جهت دریافت گزارش تراکنش پیش تر تعریف شده باشد. در صورتیکه عملیات موفق باشد لیستی از تراکنش های انجام شده برگردانده میشود و در غیر این صورت یک لیست خالی فرستاده میشود.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد OnLogResult را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد.
Example:	<pre> PcPos.Count = "05" PcPos.TransactionType = LogTransactionType.Successful; //set result call back PcPos.OnLogResult += PcPos_OnLogResult; //send request PcPos.AsyncTransactionLogs(); </pre>

۵,۲۵ AsyncSearch

Prototype:	<code>public void AsyncSearch()</code>
Function:	این متد جهت ارسال درخواست جستجوی تراکنش های انجام شده بر اساس شناسه تراکنش (OrderId) به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>OnLogResult <PosLogsResult> EventHandler</code> جهت دریافت گزارش تراکنش پیش تر تعریف شده باشد. در صورتیکه عملیات موفق باشد و

	در صورتیکه تراکنش با شناسه تراکنش (OrderId) فرستاده شده یکی باشد لیستی از تراکنشهای انجام شده برگردانده میشود و در غیر این صورت یک لیست خالی فرستاده میشود.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد OnLogResult را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد.
Example:	<pre>PcPos.SetSearchParam(new Dictionary<SearchParam, string>() { { SearchParam.OrderId, txtSearchOrderId.Text } }); //set result call back PcPos.OnLogResult += PcPos_OnLogResult; //send request PcPos.AsyncSearch();</pre>

۵,۲۶ AsyncGetAccount

Prototype:	<code>public void AsyncGetAccount()</code>
Function:	این متد جهت ارسال درخواست گزارش لیست حسابهای متصل به پایانه به همراه شماره ردیف متناظر با آن در سوئیچ پرداخت، به صورت آسنکرون به پایانه می باشد. در حالت آسنکرون بایستی <code>EventHandler<AccountResult> OnAccountInfoResult</code> جهت دریافت گزارش لیست حسابهای متصل به پایانه می باشد. در صورتیکه عملیات موفق باشد لیستی از حسابها برگردانده می شود و در غیر این صورت یک لیست خالی فرستاده می شود.
Parameter:	فاقد پارامتر می باشد.
Return:	فاقد مقدار برگشتی می باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد OnAccountInfoResult را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد. ممکن است حسابهای دولتی در این لیست نباشد. لذا برای استعلام شماره شبا و شماره ردیف متناظر با آن میتوانید از <code>ConvertIbans</code> ^{۵,۱} استفاده کنید که نیاز به اتصال اینترنت دارد.
Example:	<pre>//set result call back PcPos.OnAccountInfoResult += PcPos_OnAccountInfoResult; //send request PcPos.AsyncGetAccount();</pre>

۵,۲۷ AbortOperation

Prototype:	<code>public async Task<FrameExchangeResponse> AbortPcPosOperation()</code>
Function:	جهت لغو فرآیند تراکنش استفاده می شود.
Parameter:	فاقد مقدار پارامتر می باشد.
Return:	
Be Applicable In:	All
Notes:	
Example:	<code>PcPos.AbortPcPosOperation();</code>

مهرکیت:	این سند توسط شرکت پرداخت الکترونیک سداد تولید شده است. هرگونه کپی برداری و یا استفاده دیگر از این سند بدون کسب مجوز کتبی از این شرکت ممنوع می باشد.
صفحه ۳۰ از ۵۰	

StartGettingCardInfo ۵,۲۸

Prototype:	<code>public void StarGettingCardInfo()</code>
Function:	جهت گرفتن مشخصات کارت به صورت مداوم استفاده میشود. این متد برای فروشگاه‌های خاص در نظر گرفته شده است.
Parameter:	فاقد مقدار پارامتر می‌باشد.
Return:	فاقد مقدار برگشتی می‌باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	پیش از فراخوانی این تابع باید متدی را برای رویداد OnCardInfoResult را مقداردهی کرد تا جواب برگشتی به متد معرفی شده ارجاع داد.
Example:	<code>PcPos.OnCardInfoResult += PcPos_OnCardInfoResult; PcPos.StarGettingCardInfo();</code>

StopGettingCardInfo ۵,۲۹

Prototype:	<code>public void StopGettingCardInfo()</code>
Function:	جهت لغو گرفتن مشخصات کارت به صورت مداوم استفاده میشود. این متد برای فروشگاه‌های خاص در نظر گرفته شده است.
Parameter:	فاقد مقدار پارامتر می‌باشد.
Return:	فاقد مقدار برگشتی می‌باشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	
Example:	<code>PcPos.StopGettingCardInfo();</code>

PcPosDiscovery ۵,۳۰

Prototype:	<code>public Job<List<PosDevice>> SearchPcPos(int port, int timeOut)</code>
Function:	جهت شناسایی خودکار پوزهای موجود در شبکه مورد استفاده قرار می‌گیرد.
Parameter:	Port درگاه مورد استفاده جهت شناسایی و اسکن خودکار درگاه‌های موجود در شبکه را تعیین می‌نماید. مقدار پیش فرض ۱۰۰ در نرم افزار پوز تنظیم شده است.
Parameter:	TimeOut حداکثر زمان جستجوی شبکه جهت شناسایی پوزهای را تعیین مینماید.
Return:	لیست دستگاه‌های پیدا شده را بر می‌گرداند.
Be Applicable In:	BlueBird {P3500, CT280, MT280}
Notes:	از کلاس PcPosDiscovery جهت شناسایی خودکار پوزهای موجود در شبکه استفاده می‌شود. پس از پایان زمان Timeout پوزهای یافت شده را در قالب رویداد OnSearchPcPos اطلاع رسانی می‌نماید.
Example:	<code>var SearchPos = new PcPosDiscovery(); SearchPos.OnSearchPcPos += btnSearch_OnSearchPcPos; SearchPos.SearchPcPos(100, 5000);</code>

SearchPcPosByIp ۵,۳۱

Prototype:	<code>public Job<List<PosDevice>> SearchPcPosByIp(int port, int timeout, string Ip)</code>	
Function:	جهت شناسایی خودکار پوزهای موجود در شبکه مورد با IP مورد نظر، استفاده می شود.	
Parameter:	Port	درگاه مورد استفاده جهت شناسایی و اسکن خودکار درگاههای موجود در شبکه را تعیین می نماید. مقدار پیش فرض ۱۰۰ در نرم افزار پوز تنظیم شده است.
Parameter:	Timeout	حداکثر زمان جستجوی شبکه جهت شناسایی پوزهای را تعیین مینماید.
Return:	لیست دستگاههای پیدا شده را بر می گرداند.	
Be Applicable In:	BlueBird {P3500, CT280, MT280}	
Notes:	از کلاس PcPosDiscovery جهت شناسایی خودکار پوزهای موجود در شبکه استفاده می شود. پس از پایان زمان Timeout پوزهای یافت شده را در قالب رویداد OnSearchPcPos اطلاع رسانی می نماید.	
Example:		

SearchPcPosByMerchantName ۵,۳۲

Prototype:	<code>public Job<List<PosDevice>> SearchPcPosByMerchantName(int port, int timeout, string merchantName)</code>	
Function:	جهت شناسایی خودکار پوزهای موجود در شبکه مورد با نام مشتری مورد نظر، استفاده می شود.	
Parameter:	Port	درگاه مورد استفاده جهت شناسایی و اسکن خودکار درگاههای موجود در شبکه را تعیین می نماید. مقدار پیش فرض ۱۰۰ در نرم افزار پوز تنظیم شده است.
Parameter:	Timeout	حداکثر زمان جستجوی شبکه جهت شناسایی پوزهای را تعیین مینماید.
Return:	لیست دستگاههای پیدا شده را بر می گرداند. ^{۸۶}	
Be Applicable In:	BlueBird {P3500, CT280, MT280}	
Notes:	از کلاس PcPosDiscovery جهت شناسایی خودکار پوزهای موجود در شبکه استفاده می شود. پس از پایان زمان Timeout پوزهای یافت شده را در قالب رویداد OnSearchPcPos اطلاع رسانی می نماید.	
Example:		

SearchPcPosByTerminalId ۵,۳۳

Prototype:	<code>public Job<List<PosDevice>> SearchPcPosByTerminalId(int port, int timeout, string TerminalId)</code>	
Function:	جهت شناسایی خودکار پوزهای موجود در شبکه مورد با شماره پایانه مورد نظر، استفاده می شود.	
Parameter:	Port	درگاه مورد استفاده جهت شناسایی و اسکن خودکار درگاههای موجود در شبکه را تعیین می نماید. مقدار پیش فرض ۱۰۰ در نرم افزار پوز تنظیم شده است.
Parameter:	Timeout	حداکثر زمان جستجوی شبکه جهت شناسایی پوزهای را تعیین مینماید.

Return:	لیست دستگاه‌های پیدا شده را بر می گرداند. ^{۸۶}
Be Applicable In:	BlueBird {P3500, CT280, MT280}
Notes:	از کلاس PcPosDiscovery جهت شناسایی خودکار پوزهای موجود در شبکه استفاده می شود. پس از پایان زمان Timeout پوزهای یافت شده را در قالب رویداد OnSearchPcPos اطلاع رسانی می نماید.
Example:	

SearchPcPosByMerchantId ۵,۳۴

Prototype:	<code>public Job<List<PosDevice>> SearchPcPosByMerchantId(int port, int timeout, string MerchantId)</code>	
Function:	جهت شناسایی خودکار پوزهای موجود در شبکه مورد با شماره مشتری مورد نظر، استفاده می شود.	
Parameter:	Port	درگاه مورد استفاده جهت شناسایی و اسکن خودکار درگاه‌های موجود در شبکه را تعیین می نماید. مقدار پیش فرض ۱۰۰ در نرم افزار پوز تنظیم شده است.
Parameter:	Timeout	حداکثر زمان جستجوی شبکه جهت شناسایی پوزهای را تعیین مینماید.
Return:	لیست دستگاه‌های پیدا شده را بر می گرداند. ^{۸۶}	
Be Applicable In:	BlueBird {P3500, CT280, MT280}	
Notes:	از کلاس PcPosDiscovery جهت شناسایی خودکار پوزهای موجود در شبکه استفاده می شود. پس از پایان زمان Timeout پوزهای یافت شده را در قالب رویداد OnSearchPcPos اطلاع رسانی می نماید.	
Example:		

۶ کنترل خطا

در صورت وجود خطا یا ورود داده نامعتبر در ورودی توابع استثنائات باید کنترل شوند. برای این منظور کلاس‌هایی زیر در نظر گرفته شده است که در این قسمت توضیحات مورد نیاز آن وجود دارد.

۶.۱ PcPosBadInputException

Prototype:	<code>public class PcPosBadInputException</code>	
Function:	جهت ارسال خطاهای مربوط به ورود داده‌های نامعتبر استفاده می‌شود. این کلاس غیر از ورودی‌های استاندارد پارامتر <code>responseCode</code> را به عنوان کد خطا در ورودی دریافت می‌کند.	
Parameter:	<code>responseCode</code>	کد خطا در قالب یک داده شمارشی.
Method:	<code>string GetPcPosError()</code>	کد خطا را در قالب رشته برمی‌گرداند.
Be Applicable In:	All	
Notes:	لیست خطاهای احتمالی در رابط برنامه نویسی در <code>enumeration</code> به نام <code>SadadErrorCodes</code> موجود است.	
Example:		

۶.۲ MaximumValueException

Prototype:	<code>public class PcPosBadInputException</code>	
Function:	جهت ارسال خطاهای مربوط به ورود داده‌های بیش از حد مجاز استفاده می‌شود. این کلاس غیر از ورودی‌های استاندارد پارامتر <code>responseCode</code> را به عنوان کد خطا در ورودی دریافت می‌کند.	
Parameter:	<code>responseCode</code>	کد خطا در قالب یک داده شمارشی.
Method:	<code>string GetPcPosError()</code>	کد خطا را در قالب رشته برمی‌گرداند.
Be Applicable In:	All	
Notes:	لیست خطاهای احتمالی در رابط برنامه نویسی در <code>enumeration</code> به نام <code>SadadErrorCodes</code> موجود است.	
Example:		

۶.۳ MinimumValueException

Prototype:	<code>public class PcPosBadInputException</code>	
Function:	جهت ارسال خطاهای مربوط به ورود داده‌های کمتر از حد مجاز استفاده می‌شود. این کلاس غیر از ورودی‌های استاندارد پارامتر <code>responseCode</code> را به عنوان کد خطا در ورودی دریافت می‌کند.	
Parameter:	<code>responseCode</code>	کد خطا در قالب یک داده شمارشی.
Method:	<code>string GetPcPosError()</code>	کد خطا را در قالب رشته برمی‌گرداند.
Be Applicable In:	All	

Notes:	لیست خطاهای احتمالی در رابط برنامه نویسی در enumeration ی به نام SadadErrorCodes موجود است.
Example:	

۷ رویدادها

OnSaleResult ۷,۱

Prototype:	<code>public event EventHandler<PosResult> OnSaleResult</code>
Function:	نتیجه تراکنش در مد آسنکرون از طریق رویداد OnSaleResult تعیین می گردد.
Parameter:	<code>PosResult e^{۸,۱}</code>
Be Applicable In:	All
Notes:	
Example:	<code>PcPos.OnSaleResult += PcPos_OnSaleResult;</code>

OnBillPaymentResult ۷,۲

Prototype:	<code>public event EventHandler<PosResult> OnBillPaymentResult</code>
Function:	نتیجه تراکنش در مد آسنکرون از طریق رویداد OnBillPaymentResult تعیین می گردد.
Parameter:	<code>PosResult e^{۸,۱}</code>
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	
Example:	<code>PcPos.OnBillPaymentResult += PcPos_OnBillPaymentResult;</code>

OnInquiredIdentifiedSaleResult ۷,۳

Prototype:	<code>public event EventHandler<PosResult> OnInquiredIdentifiedSaleResult</code>
Function:	نتیجه تراکنش در مد آسنکرون از طریق رویداد OnInquiredIdentifiedSaleResult تعیین می گردد.
Parameter:	<code>PosResult e^{۸,۱}</code>
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	
Example:	<code>PcPos.OnInquiredIdentifiedSaleResult += PcPos_OnInquiredIdentifiedSaleResult;</code>

OnGovernmentInquiredIdentifiedSaleResult ۷,۴

Prototype:	<code>public event EventHandler<PosResult> OnGovernmentInquiredIdentifiedSaleResult</code>
Function:	نتیجه تراکنش در مد آسنکرون از طریق رویداد PcPosResult تعیین می گردد.
Parameter:	<code>PosResult e^{۸,۱}</code>
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bite1 {IC3300, IC3100}
Notes:	
Example:	<code>PcPos.OnGovernmentInquiredIdentifiedSaleResult += PcPos_GovernmentInquiredIdentifiedSaleResult;</code>

OnCardInfoResult ۷,۵

Prototype:	<code>public event EventHandler<PosCardInfoResult> OnCardInfoResult</code>
Function:	نتیجه درخواست در مد آسنکرون از طریق رویداد OnCardInfoResult تعیین می گردد.
Parameter:	<code>PosCardInfoResult e^{۸,۱}</code>
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}
Notes:	
Example:	<code>PcPos.OnCardInfoResult += PcPos_OnCardInfoResult;</code>

OnLogResult ۷,۶

Prototype:	<code>public event EventHandler<PosLogsResult> OnLogResult</code>
Function:	نتیجه درخواست در مد آسنکرون از طریق رویداد OnLogResult تعیین می گردد.
Parameter:	<code>PosLogsResult e^{۸,۴}</code>
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}
Notes:	
Example:	<code>PcPos.OnLogResult += PcPos_OnLogResult;</code>

OnPosDeviceInfoResult ۷,۷

Prototype:	<code>public event EventHandler<PosDevice> OnPosDeviceInfoResult</code>
Function:	نتیجه درخواست پس از اتمام زمان آن از طریق این رویداد به کاربر اطلاع داده می شود.
Parameter:	<code>PosDevice posDevice^{۸,۶}</code>
Be Applicable In:	BlueBird {P3500, CT280, MT280}
Notes:	
Example:	<code>PcPos.OnPosDeviceInfoResult += PcPos_OnPosDeviceInfoResult;</code>

OnAccountInfoResult ۷,۸

Prototype:	<code>public event EventHandler<AccountResult> OnAccountInfoResult</code>
Function:	نتیجه درخواست پس از اتمام زمان آن از طریق این رویداد به کاربر اطلاع داده می شود.
Parameter:	<code>AccountResult accountResult^{۸,۱۰}</code>
Be Applicable In:	BlueBird {P3500, CT280, MT280}
Notes:	
Example:	<code>PcPos.OnAccountInfoResult += PcPos_OnAccountInfoResult;</code>

OnSearchPcPos ۷,۹

Prototype:	<code>public event EventHandler<DiscoveryResult> OnSearchPcPos</code>
Function:	نتیجه‌ی جستجو پس از اتمام زمان آن از طریق این رویداد به کاربر اطلاع داده می شود.
Parameter:	<code>DiscoveryResult</code> OnSearchPcPos ^{۸,۶}
Be Applicable In:	BlueBird {P3500, CT280, MT280}
Notes:	
Example:	<code>SearchPos.OnSearchPcPos += SearchPos_OnSearchPcPos ;</code>

۸ کلاس ها

در وروی هر رخداد پارامترهایی با انواع مختلف معرفی شده است که در این قسمت توضیحات مورد نیاز آنها آمده است:

۸.۱ PosResult

Prototype:	<code>public class PosResult</code>		
Function:	نتیجه‌ی تراکنش در خصوصیات مورد انتظار آن در این کلاس نمایش داده میشود.		
Property:	Amount	string	مبلغ تراکنش
Property:	ApprovalCode	string	کد تأییدیه تراکنش
Property:	CardNo	string	شماره کارت تراکنش به صورت Mask
Property:	Merchant	string	شماره/کد پذیرنده دستگاه پوز
Property:	OptionalField	string	مقادیر اختیاری ^۱
Property:	PacketType	string	نوع بسته‌های ارسالی و دریافتی به/از دستگاه پوز ^۲
Property:	PcPosStatus	string	متن پیام کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	PcPosStatusCode	int	کدهای کنترلی ارتباط بین رایانه و دستگاه پوز ^۳
Property:	ProcessingCode	string	کد پردازشی تراکنش
Property:	ResponseCode	string	کد پاسخ تراکنش ^۴
Property:	ResponseCodeMessage	string	متن پیام کد پاسخ تراکنش
Property:	Rrn	string	شماره مرجع تراکنش ^۵
Property:	Terminal	string	شماره/کد پایانه دستگاه پوز
Property:	TransactionDate	string	تاریخ انجام تراکنش
Property:	TransactionNo	string	شماره پیگیری تراکنش ^۶
Property:	TransactionTime	string	زمان انجام تراکنش
Property:	OrderId	string	شناسه تراکنش
Property:	ExtraData	List <ExtraDataResult ^{۸,۲} >	اطلاعات اضافه
Be Applicable In:	All		
Notes:	<p>۱. برخی پیام‌های دریافت شده از دستگاه پوز و یا درگاه پرداخت در این خصوصیت برگشت داده میشود.</p> <p>۲. برای توسعه دهندگان نهایی کاربرد ندارد.</p> <p>۳. این کدها برای کنترل ارسال دریافت اطلاعات از رایانه به پایانه و بالعکس میباشد. توسعه دهندگان قبل از چک کردن مشخصه ResponseCode باید این مشخصه را بررسی نمایند:</p> <p>a. ۰: پاسخی دریافت نشد</p> <p>b. ۱: خطا حداکثر دفعات تکرار</p> <p>c. ۳: لغو تراکنش توسط کاربر پوز</p> <p>d. ۲: لغو تراکنش توسط پی‌سی پوز</p>		

	<p>e. ۴: ارتباط موفق با دستگاه</p> <p>f. ۵: خطای داخلی</p> <p>g. ۶: خطای درخواست همزمان</p> <p>h. در غیر اینصورت: خطای ناشناخته</p> <p>۴. شامل کدهایی است که برای جواب تراکنش استفاده شده است. این جواب ها از درگاه پرداخت یا دستگاه پوز دریافت میشوند که در سند لیست خطاها توضیحات هر کد خطا آمده است.</p> <p>۵. تراکنشی که کد مرجع نداشته باشد حتما مشکلی داشته است و انجام نشده است.</p> <p>۶. شماره ای که به صورت شمارشی اضافه میشود. در صورت بروز رسانی دستگاه و رسیدن به رقم ۹۹۹۹۹۹، صفر می شود.</p>
--	--

ExtraDataResult ۸,۲

Prototype:	public class ExtraDataResult		
Function:	نتیجه‌ی اطلاعات تسهیم تراکنش در صورت وجود در این کلاس نمایش داده میشود.		
Property:	Amount	string	مبلغ ^۱
Property:	IBAN	string	شماره حساب ^۲
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		
Notes:	<p>۱. رشته عددی به طول نهایت ۱۲ رقم.</p> <p>۲. شماره حساب به صورت رشته عددی ۱۳ رقمی. استاندارد حال حاضر این خصوصیت به صورت شماره شبا نمایش میدهد.</p>		

PosCardInfoResult ۸,۳

Prototype:	public class PosCardInfoResult		
Function:	نتیجه‌ی اطلاعات کارت در این کلاس نمایش داده میشود.		
Property:	CardInfo	string	شماره کارت را به صورت درهم شده ^۱
Property:	CardInfoN16	string	تبدیل درهم شماره کارت به عدد ۱۶ رقمی ^۲
Property:	CardNo	string	شماره کارت تراکنش به صورت Mask
Property:	MerchantId	string	شماره/کد پذیرنده دستگاه پوز
Property:	PcPosStatus	string	متن پیام کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	PcPosStatusCode	int	کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	ResponseCode	string	کد پاسخ تراکنش ^۳
Property:	ResponseCodeMessage	string	متن پیام کد پاسخ تراکنش
Property:	TerminalId	string	شماره/کد پایانه دستگاه پوز
Property:	OrderId	string	شناسه تراکنش
Property:	Toggle	int	تعیین کننده نوع جواب ^۴
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		

Notes:	<p>۳. درهم (Hash) کردن با الگوریتم SHA-2 انجام میگیرد.</p> <p>۴. با استفاده از تابعی به نام ToN16Hash که در کلاس Common رابط برنامه نویسی تعبیه شده است انجام میشود.</p> <p>۵. شامل کدهایی است که برای جواب تراکنش استفاده شده است. این جواب ها از درگاه پرداخت یا دستگاه پوز دریافت میشوند که در سند لیست خطاها توضیحات هر کد خطا آمده است.</p> <p>۶. مقدار این خصوصیت میتواند ۱ (جواب مشخصات کارت کشیده شده روی دستگاه پوز توسط مشتری) یا ۲ (جواب تراکنش نشان اعتباری) باشد.</p>
--------	--

۸,۴ PosLogsResult

Prototype:	public class PosLogsResult		
Function:	گزارش تراکنش ها در این کلاس نمایش داده میشود.		
Property:	Items	List<PosLogResult ^{۸,۵} >	لیست اطلاعات تراکنش ها
Property:	PcPosStatus	string	متن پیام کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	PcPosStatusCode	int	کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	ResponseCode	string	کد پاسخ تراکنش ^۱
Property:	ResponseCodeMessage	string	متن پیام کد پاسخ تراکنش
Property:	OrderId	string	شناسه تراکنش
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		
Notes:	<p>۱. شامل کدهایی است که برای جواب تراکنش استفاده شده است. این جواب ها از درگاه پرداخت یا دستگاه پوز دریافت میشوند که در سند لیست خطاها توضیحات هر کد خطا آمده است.</p>		

۸,۵ PosLogResult

Prototype:	public class PosLogResult		
Function:	اطلاعات تراکنش در این کلاس نمایش داده میشود.		
Property:	Amount	string	مبلغ تراکنش
Property:	CardNo	string	شماره کارت تراکنش
Property:	PcPosStatus	string	متن پیام کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	PcPosStatusCode	int	کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	ResponseCode	string	کد پاسخ تراکنش (شامل کدهایی است که برای جواب تراکنش استفاده شده است. این جواب ها از درگاه پرداخت یا دستگاه پوز دریافت میشوند که در سند لیست خطاها توضیحات هر کد خطا آمده است.)
Property:	Rrn	string	شماره مرجع تراکنش
Property:	TransactionDate	string	تاریخ انجام تراکنش

Property:	TransactionNo	string	شماره پیگیری تراکنش
Property:	TransactionTime	string	زمان انجام تراکنش
Property:	TransactionType	string	نوع تراکنش*
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		
Notes:	<p>* با شماره های ۰ و ۱ و ۲ مشخص میشود:</p> <ul style="list-style-type: none"> ۰: تراکنش های ناموفق ۱: تراکنش های موفق ۲: همه 		

PosDevice ۸,۶

Prototype:	public class PosDevice		
Function:	نتیجه ی اطلاعات دستگاه پوز در این کلاس نمایش داده میشود.		
Property:	Header	string	عنوان بسته دریافت شده
Property:	IpAdress	string	آدرس IP دستگاه پوز
Property:	MerchantId	int	شماره/کد پذیرنده دستگاه پوز
Property:	MerchantName	string	عنوان/نام پذیرنده دستگاه پوز
Property:	Port	string	درگاه قابل ارتباط با دستگاه
Property:	TerminalId	string	شماره/کد پایانه دستگاه پوز
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		
Notes:			

DiscoveryResult ۸,۷

Prototype:	public class DiscoveryResult		
Function:	نتیجه ی جستجوی دستگاه پوز در شبکه LAN در این کلاس نمایش داده میشود.		
Property:	Devices	List<PosDevice>	لیست دستگاه های یافت شده
Be Applicable In:	BlueBird {P3500, CT280, MT280}		
Notes:			

SaleIdProvider ۸,۸

Prototype:	public class SaleIdProvider		
Function:	تولید شناسه دولتی برای ارسال پوز.		
Property:	Count	int	تعداد شناسه های ساخته شده
Property:	SaleIds	List<SaleIdData>	لیست شناسه های دولتی
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		
Notes:			

SaleIdData ۸,۹

Prototype:	public class SaleIdData		
Function:	تولید شناسه دولتی برای ارسال پوز.		
Property:	Amount	decimal	مبلغ شناسه
Property:	Index	int	ردیف شماره حساب/شبابی متناظر با شناسه در سوئیچ پرداخت شرکت الکترونیک سداد
Property:	IBAN	string	شماره شبای متناظر با شناسه در سوئیچ پرداخت شرکت پرداخت الکترونیک سداد
Property:	SaleId	string	شناسه دولتی
Property:	RowType	RowType	نوع حساب Public/Private که همیشه مقدار Private یعنی ۱ میباشد.
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		
Notes:	پیشنهاد میشود یکی از مقادیر شماره شبا (IBAN) یا شماره ردیف (Index) پر شود.		

AccountResult ۸,۱۰

Prototype:	public class AccountResult		
Function:	نتیجه‌ی اطلاعات دستگاه پوز در این کلاس نمایش داده میشود.		
Property:	Accounts	List<Account ^{۸,۱۱} >	لیست حساب‌های متصل به پوز
Property:	PcPosStatus	string	متن پیام کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	PcPosStatusCode	int	کدهای کنترلی ارتباط بین رایانه و دستگاه پوز
Property:	OrderId	string	شناسه درخواست
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		
Notes:			

Account ۸,۱۱

Prototype:	public class Account		
Function:	نتیجه‌ی اطلاعات دستگاه پوز در این کلاس نمایش داده میشود.		
Property:	Iban	string	شماره شبا
Property:	RowNo	int	شماره ردیف متناظر با شماره شبا در سوئیچ پرداخت
Be Applicable In:	BlueBird {P3500, CT280, MT280}, Bitel {IC3300, IC3100}		
Notes:			

۹ تنظیمات رابط توسعه

فایل SadadConfig.xml در مسیر فایل رابط نرم افزاری موجود است. تنظیمات مربوط به اتصالات و تراکنش ها در این فایل وجود دارد.

ساختار این فایل به صورت زیر میباشد:

```
<?xml version="1.0" encoding="utf-8" ?>
<Config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Modems>
    <Serial>
      <Name>COM1</Name>
      <!-- BlueBird = 0, Magic = 1, Bitel = 2, Pax = 3, Castle = 4 -->
      <DeviceType>2</DeviceType>
      <!-- 1200, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 115200 -->
      <BaudRate>115200</BaudRate>
      <!-- None, Odd, Even, Mark, Space -->
      <Parity>None</Parity>
      <!-- 5 to 8 -->
      <DataBits>8</DataBits>
      <!-- None, One, Two, OnePointFive -->
      <StopBits>One</StopBits>
      <!-- None, XOnXOff, RequestToSend, RequestToSendXOnXOff -->
      <Handshake>None</Handshake>
      <!-- 10 to 180000 or higher -->
      <Timeout>180000</Timeout>
    </Serial>
    <Lan>
      <IP>172.24.34.186</IP>
      <Port>8888</Port>
      <!-- 10 to 180000 or higher -->
      <Timeout>180000</Timeout>
    </Lan>
  </Modems>
  <Transactions>
    <Sale>
      <IsAmountRequired>true</IsAmountRequired>
      <IsSaleIdRequired>false</IsSaleIdRequired>
      <IsOrderIdRequired>false</IsOrderIdRequired>
      <IsMultiAccount>false</IsMultiAccount>
      <MaxIban>10</MaxIban>
      <MinAmount>1000</MinAmount>
    </Sale>
    <Bill>
      <IsAmountRequired>false</IsAmountRequired>
      <IsBillIdRequired>true</IsBillIdRequired>
      <IsPayIdRequired>true</IsPayIdRequired>
      <IsOrderIdRequired>false</IsOrderIdRequired>
      <MinAmount>1000</MinAmount>
    </Bill>
    <Inquiry>
      <IsAmountRequired>false</IsAmountRequired>
      <IsSaleIdRequired>true</IsSaleIdRequired>
      <IsOrderIdRequired>false</IsOrderIdRequired>
      <MinAmount>1000</MinAmount>
    </Inquiry>
    <Government>
      <IsAmountRequired>true</IsAmountRequired>
      <IsSaleIdRequired>true</IsSaleIdRequired>
      <IsOrderIdRequired>false</IsOrderIdRequired>
      <IsMultiSaleId>false</IsMultiSaleId>
    </Government>
  </Transactions>
</Config>
```

```

    <MaxSaleId>10</MaxSaleId>
    <MinAmount>1000</MinAmount>
  </Government>
</Transactions>
<WebServices>
  <URLs>
    <URL>
      <Name>AddData</Name>
      <Address>http://172.24.34.67:9150/api/TerminalAddData/Import</Address>
    </URL>
  </URLs>
  <Magic>
    <IsMagicUsedWithWebService>true</IsMagicUsedWithWebService>
  </Magic>
</WebServices>
</Config>

```

در قسمت Modem میتوان یک یا چند مودم از نوع LAN (اتصال شبکه) یا Serial (اتصال سریال) اضافه کرد. در قسمت تراکنش ها نیز هر یک خصوصیات بعنوان قانون برای بررسی محدودیت ها در نظر گرفته شده اند. خصوصیات هر یک از آنها در جداول زیر آمده است:

۹,۱ سریال

Title	مقدار پیش فرض	مقادیر قابل قبول	توضیحات
Name	COM1	--	نام یا عنوان پورت سریال
DeviceType	0	BlueBird = 0, Magic = 1, Bitel = 2, Pax = 3, Castle = 4	نوع دستگاه
BaudRate	115200	1200, 2400, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 115200	سرعت اتصال
Parity	None	None, Odd, Even, Mark, Space	نوع بیت توازن
DataBits	8	5 to 8	اندازه بیت داده
StopBits	One	None, One, Two, OnePointFive	اندازه بیت استاپ
Handshake	None	None, XOnXOff, RequestToSend, RequestToSendXOnXOff	نوع Handshake
Timeout	180000	10 to 180000 or higher	زمان پاسخدهی به میلی ثانیه

۹,۲ شبکه

Title	مقدار پیش فرض	مقادیر قابل قبول	توضیحات
IP	COM1		آدرس اتصال
Port	8888	8888	پورت اتصال
Timeout	180000	10 to 180000 or higher	زمان پاسخدهی به میلی ثانیه

۹,۳ تراکنش خرید

Title	مقدار پیش فرض	مقادیر قابل قبول	توضیحات
IsAmountRequired	True	True, False	ضرورت ورود مبلغ
IsSaleIdRequired	False	True, False	ضرورت ورود شناسه خرید
IsOrderIdRequired	False	True, False	ضرورت ورود شناسه تراکنش
IsMultiAccount	False	True, False	چند حسابی بودن دستگاه
MaxIban	10	1 to 10	محدودیت نهایت تعداد حساب
MinAmount	1000	1000 to 999999999999	محدودیت حداقل مبلغ

۹,۴ تراکنش پرداخت قبض

Title	مقدار پیش فرض	مقادیر قابل قبول	توضیحات
IsAmountRequired	False	True, False	ضرورت ورود مبلغ
IsBillIdRequired	True	True, False	ضرورت ورود مبلغ
IsPayIdRequired	True	True, False	ضرورت ورود مبلغ
IsOrderIdRequired	False	True, False	ضرورت ورود شناسه تراکنش
MinAmount	1000	1000 to 999999999999	محدودیت حداقل مبلغ

۹,۵ تراکنش خرید با استعلام

Title	مقدار پیش فرض	مقادیر قابل قبول	توضیحات
-------	---------------	------------------	---------

مهرکفیت:	این سند توسط شرکت پرداخت الکترونیک سداد تولید شده است. هرگونه کپی برداری و یا استفاده دیگر از این سند بدون کسب مجوز کتبی از این شرکت ممنوع می باشد.
صفحه ۴ از ۵۰	

IsAmountRequired	True	True, False	ضرورت ورود مبلغ
IsSaleIdRequired	False	True, False	ضرورت ورود شناسه خرید
IsOrderIdRequired	False	True, False	ضرورت ورود شناسه تراکنش
MinAmount	1000	1000 to 999999999999	محدودیت حداقل مبلغ

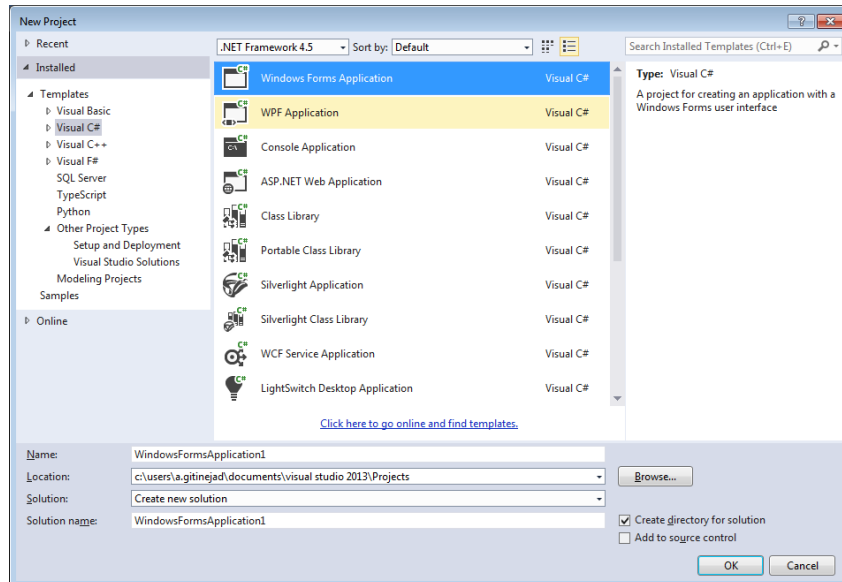
۹,۶ تراکنش با استعلام دولتی

Title	مقدار پیش فرض	مقادیر قابل قبول	توضیحات
IsAmountRequired	True	True, False	ضرورت ورود مبلغ
IsSaleIdRequired	False	True, False	ضرورت ورود شناسه خرید
IsOrderIdRequired	False	True, False	ضرورت ورود شناسه تراکنش
IsMultiSaleId	False	True, False	ضرورت چند شناسه ای
MaxIban	10	1 to 10	محدودیت نهایت تعداد حساب
MinAmount	1000	1000 to 999999999999	محدودیت حداقل مبلغ

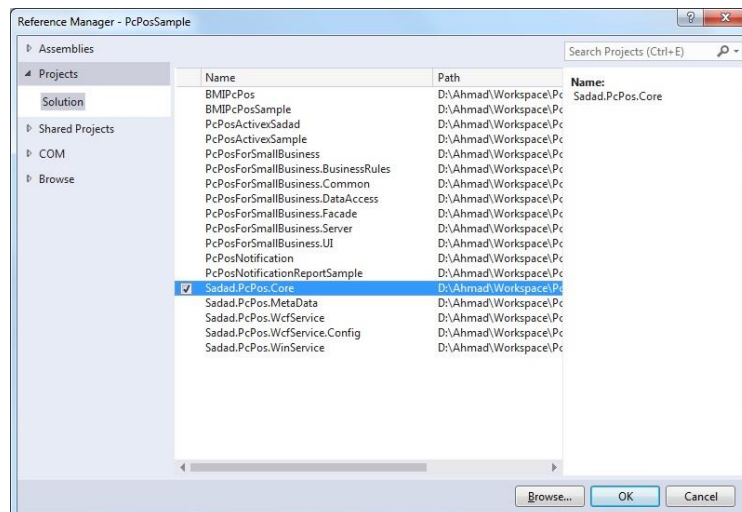
۱۰ نحوه استفاده از DLL در Visual Studio

۱۰,۱ ایجاد پروژه C#

در منوی فایل ویژوال استودیو گزینه New Project را انتخاب نمایید، سپس از لیست پروژه‌های ویژوال استودیو Visual C# و سپس گزینه‌ی Windows Form Application را انتخاب نمایید:



سپس پروژه را PcPosSample نام گذاری نمایید. بر روی پروژه کلیک راست نموده گزینه‌ی Add و سپس Reference را انتخاب نمایید.



جهت استفاده از DLL فضای نام Sadad.PcPos.Core را به برنامه اضافه نماید.

۱۱ روال استفاده

۱۱,۱ در فروشگاه‌های حالت خاص (افق کوروش)

۱. ارسال درخواست مشخصات کارت

۲. دریافت مشخصات کارت به صورت Hash

۳. ارسال داده‌های تراکنش (خرید)

۴. دریافت اطلاعات تراکنش

