



# گروه پرداخت های نوین

---

## فهرست

۳	هدف این نوشته
۳	ترمینولوژی
۴	نیازمندی های امنیتی و بانکی
۴	نیازمندی های امنیتی
۴	نیازمندی های بانکی
۵	بسترسازی سمت فروشنده
۵	۱. انجام تراکنش خرید
۷	۲. انجام تراکنش پرداخت قبض
۹	۳. برگشت تراکنش خرید
۱۰	پشتیبانی مشتریان
۱۱	ضمیمه الف: پیاده سازی
۱۱	تکنولوژی های مورد استفاده:

۱۱	.....	نیازمندی استفاده از Web Methods
۱۲	.....	توجه:
۱۳	.....	Web Methods ب: شرح
۱۳	.....	متد ورود (login)
۱۴	.....	متد خروج (logout)
۱۴	.....	متد درخواست تسویه فروشنده (merchantSettlementRequest)
۱۵	.....	متد تایید تراکنش (verifyTransaction)
۱۶	.....	متد تایید تراکنش امن (secureVerifyTransaction)
۱۷	.....	متد برگشت تراکنش (reverseTransaction)
۱۸	.....	متد گزارش تراکنش (reportTransaction)
۲۱	.....	متد گزارش جزئیات تراکنش (detailReportTransaction)
۲۲	.....	ضمیمه ج: پارامترهای تبادلی میان سایت فروشنده و سایت صدور رسید دیجیتالی
۲۲	.....	تراکنش خرید اینترنتی
۲۳	.....	تراکنش پرداخت قبض

# هدف این نوشتار

این نوشته قدم های لازم برای ایجاد بستر پرداخت با استفاده از رسید دیجیتالی را در سمت فروشنده بر می شمارد. ساختار این نوشته به صورت ذیل می باشد.

## ترمینولوژی

سیستم پرداخت الکترونیکی با استفاده از کد تایید انتقال، از پنج جزء تشکیل یافته است:

- **خریدار:** موجودیتی که تقاضای خرید سرویس یا کالا دارد.
- **فروشنده:** موجودیتی که سرویس یا کالا را در اختیار خریدار قرار می دهد.
- **گردآورنده (پذیرنده) (Acquirer):** موجودیتی واسط میان شبکه عمومی (شامل خریدار و فروشنده) و شبکه بین بانکی (شامل بانک های نگه دارنده سپرده های فروشنده و خریدار).
- **بانک های عضو شتاب:** موجودیت هایی که سپرده های خریدار و فروشنده را نزد خود نگه می دارند.
- **سایت صدور رسید دیجیتالی:** سایتی متعلق به بانک می باشد که در آن خریدار شماره کارت و رمز آن را وارد می نماید و انتقال مبلغ خرید به سپرده فروشنده را تایید می نماید. در صورت موفقیت آمیز بودن انتقال، یک رسید دیجیتالی برای آن انتقال صادر می نماید. (در واقع این کار می تواند به عنوان بخشی از وظایف ماشین پذیرنده در نظر گرفته شود)

همچنین در این نوشته ترم های زیر نیز به کار برده خواهند شد:

- **رسید دیجیتالی (Reference Number):** یک سلسله ای از کاراکترها که می تواند تا ۲۰ حرف باشد و بانک به عنوان رسید پس از انجام یک انتقال به خریدار (و در نهایت به فروشنده) ارائه می دهد.
- **شماره رزرو (Reservation Number):** کدی که فروشنده برای هر تراکنش خریدار در نظر می گیرد و خریدار می تواند توسط آن کد، خرید خود را پیگیری کند. در واقع مشخصه تراکنش است در سمت فروشنده. این کد می تواند تا ۵۰ حرف باشد و می تواند ترکیبی باشد از عدد و حروف.
- **شماره سپرده خریدار (Buyer Deposit Number):** شماره سپرده ای که خریدار از آن مبلغ خرید را به سپرده فروشنده انتقال داده است.
- **کد فروشنده (Merchant ID):** کدی است که بانک برای هر فروشنده اختصاص می دهد.
- **نام کاربری فروشنده (Merchant Username):** نام کاربری فروشنده که توسط بانک به هر فروشنده اختصاص داده می شود.
- **سپرده:** در بعضی از ترمینولوژی های بانکی بدان حساب گویند.
- **تراکنش:** یک عملیات مالی، که در این نوشته مصداق آن یک خرید می باشد.

# نیازمندی‌های امنیتی و بانکی

## نیازمندی‌های امنیتی

- سایت بانک دارای گواهینامه‌های معتبر می‌باشد. پس ارتباط خریدار با بانک و فروشنده با بانک می‌تواند در بستر SSL انجام شود. اگر فروشنده نیز دارای گواهینامه معتبری باشد، ارتباط بین خریدار و فروشنده نیز در بستر SSL و به صورت امن خواهد بود. اینکه فروشنده دارای گواهینامه معتبر باشد، اجباری نیست، بلکه بهتر است اینگونه باشد.
- نیازمندی امنیتی دیگر اینست که فروشنده از هیچ کدام از اطلاعات مالی خریدار (مانند شماره کارت، رمز کارت، میزان موجودی و ....) مطلع نشود. به همین خاطر فروشنده از خریدار هیچ نوع اطلاعات مالی و بانکی دریافت نمی‌کند و تمامی این اطلاعات توسط خریدار در سایت بانک وارد می‌شود.
- برای اطمینان از هویت فروشنده در هنگام درخواست برگشت خوردن سند، کلمه عبوری به فروشنده داده می‌شود که در هنگام برگشت زدن یک تراکنش باید به ماشین درگاه پرداخت داده شود. (این کلمه عبور در هنگام راه اندازی فروشنده برای بار اول از طرف بانک به فروشنده داده می‌شود)

## نیازمندی‌های بانکی

فروشنده باید سپرده‌ای را نزد یکی از بانک‌های عضو شتاب افتتاح نماید و آن را به عنوان "سپرده فروشنده" به بانک معرفی نماید. بانک نیز به وی یک کد اختصاص خواهد که در هر تراکنش فروشنده خود را با ارائه این شماره (که اختصاراً به آن کد فروشنده می‌گوییم) به بانک معرفی می‌کند.

## بسترسازی سمت فروشنده

در سمت فروشنده ۲ تراکنش قابل پیاده سازی می باشد.

- خرید اینترنتی کالا
- پرداخت قبوض

برای هر کدام از این تراکنش ها، فروشنده باید اطلاعات مختلفی را به بانک ارسال کند که در قسمت بعدی توضیح داده خواهد شد. نکته مهم در تراکنش پرداخت قبض این است که این تراکنش قابلیت برگشت جزئی ندارد.

### ۱. انجام تراکنش خرید

فروشنده خریدار را به نقطه ای می رساند که آماده دریافت پول و نهایی کردن خرید می باشد. در این نقطه فروشنده باید خریدار را به سایت صدور رسید دیجیتالی **redirect** کند. تمامی پارامترهایی که به این سایت به صورت **POST** می دهد عبارتند از: (تمامی پارامتر های باید با نام نوشته شده در زیر به سایت بانک ارسال شود، توجه کنید که بزرگی و کوچکی حروف نیز مهم است).

- Amount
- ResNum
- MID
- RedirectURL
- language

نشانی ای که فروشنده باید این فرم را به آن **submit** کند را باید از بانک دریافت کند ولی به طور مثال باید چیزی شبیه به مورد زیر باشد:

<https://fanava.shaparak.ir/cardmanager/controller>

توجه داشته باشید که فروشنده در صورتی نیاز دارد شماره رزرو را به سایت صدور رسید دیجیتالی پاس دهد که تکنولوژی **session management** آن **cookie based** نباشد. در غیر این صورت با استفاده از اطلاعات **session** می تواند رفت خریدار به سایت صدور رسید دیجیتالی و برگشت وی را تعقیب نماید. حتی اگر فروشنده از **Cookie** استفاده می نماید ما توصیه می کنیم باز هم فروشنده شماره رزرو را پر کرده و برای بانک ارسال نماید. بانک با این شماره رزرو هیچ کار خاصی انجام نمی دهد و فقط آن را همراه با نتیجه تراکنش برای فروشنده بر می گرداند. فروشنده با دادن و گرفتن شماره رزرو می تواند خریدار را دنبال کند و تشخیص دهد که یک جواب خاص از جانب بانک مربوط به کدام درخواست فروشنده می شده است. (اشکال استفاده از **Cookie** ها در این قسمت این است که **Cookie** امکان دارد در پروسه رفت و برگشت به سایت بانک، **Expire** شود).<sup>۱</sup>

همچنین این ارتباط باید در بستر **SSL** قرار بگیرد (اجباری است).

در سایت صدور رسید دیجیتالی، خریدار اطلاعات کارت خود را باید وارد کند:

- شماره کارت (PAN)
- کلمه عبور (PIN)
- تاریخ انقضای کارت (ExpDate)
- شماره CVV۲

---

<sup>۱</sup> شرح کامل پارامترهای اجباری و اختیاری که فروشنده در اختیار بانک می گذارد در ضمیمه ج آورده شده است.

البته موارد بالا در سایت صدور رسید دیجیتالی پیاده سازی شده و فروشنده هیچ کار خاصی در موارد پارامترهای بالا ندارد.

سایت صدور رسید دیجیتالی پس از اتمام انتقال وجه، خریدار را دوباره به سایت فروشنده redirect می کند (به طوری که رفت و آمد خریدار از سایت فروشنده به سایت صدور رسید دیجیتالی و سپس به سایت فروشنده در همان session سایت فروشنده صورت می گیرد). فروشنده پارامترهای پاس شده از سایت صدور رسید دیجیتالی را از بدنه فرم (POST) باید دریافت کند.

پارامترها عبارتند از:

- redirectURL
- MID
- ResNum
- RefNum
- State
- Language
- CardPanHash

فروشنده می تواند بر اساس وضعیت تراکنش، موفقیت آمیز بودن تراکنش را تشخیص دهد. شرح کامل پارامترها در ضمیمه ج آورده شده است.

**پارامتر CardPanHash** با استفاده از روش **SHA۲۵۶** شماره **Pan** کارت خریدار را تبدیل به رشته ای **Hash** شده می کند و برای فروشنده ارسال می نماید.

اگر خرید موفقیت آمیز نبود فروشنده موظف است خطای به وجود آمده را با توجه به فیلد وضعیت تراکنش برای خریدار شرح دهد و به او بگوید دقیق چه اتفاقی روی داده است. اگر وضعیت تراکنش ok بود، به این معنی است که مقداری پول از کارت خریدار به سپرده فروشنده منتقل شده است ولی برای تایید مقدار منتقل شده و همچنین عدم برگشت به صورت سیستمی مبلغ واریز شده، فروشنده باید توسط وب سرویس ای که در اختیار او قرار داده شده تراکنش خرید را تایید کند. برای تایید خرید فروشنده باید مقدار **RefNum** دریافت شده را ابتدا در پایگاه داده خود ذخیره کند و سپس متد تایید وب سرویس را صدا بزند. شرح متد های وب سرویس و نحوه اجرای آنها در ضمیمه ب آورده شده است.

**نکته بسیار مهم:** پس از اجرای متد تایید، فروشنده باید نتیجه را بررسی کرده و تصمیم گیرد که خریدار مبلغ مناسب را واریز نموده است یا خیر. در صورت درست بودن انتقال، فروشنده رسید دیجیتالی را باید در پایگاه داده خود ذخیره کند و پس از آن می تواند سرویس خود را به خریدار ارایه کند. در صورت درست نبودن مبلغ انتقالی، باید همچنان رسید دیجیتالی را ذخیره کند و حتما باید درخواست برگشت آن خرید را به بانک بدهد. (هرچند که عملا خریدار نباید بتواند مبلغ نادرستی را انتقال دهد)

با مقایسه مبلغ دریافت شده و مبلغ فاکتوری که فروشنده خود در اختیار دارد ۳ حالت زیر ممکن است به وجود آید:

- اگر این دو مبلغ برابر باشند، فروشنده می تواند سرویس خود را ارایه نماید.
- اگر مبلغ پرداختی کمتر از مبلغ فاکتور فروشنده باشد، فروشنده می تواند پس از اعلام به خریدار، کل سند را برگشت بزند.
- اگر مبلغ پرداختی بیشتر از مبلغ فاکتور نزد فروشنده باشد، فروشنده می تواند پس از اعلام به خریدار، ما به التفاوت سندها را برگشت بزند.

## نکات:

۱. تاکید می‌شود که مصرف شدگی رسید دیجیتالی در سمت فروشنده تعیین و نگهداری می‌شود و نه در سمت بانک. بانک تنها اعتبار و مبلغ برگشت نخورده رسید دیجیتالی را گزارش می‌دهد. بدین ترتیب بانک می‌تواند مشخصات یک رسید دیجیتالی را چندین بار به فروشنده گزارش دهد بدون آن‌که وضعیت مصرف‌شدگی آن تغییر کند. حسن این روش در این است که اگر فروشنده‌ای یک رسید دیجیتالی را برای اعتبارسنجی به بانک بدهد و بانک نیز نتیجه را برای فروشنده ارسال دارد ولی این جواب به هر دلیلی به دست فروشنده نرسد، رسید دیجیتالی اعتبار خود را از دست نخواهد داد و فروشنده می‌تواند دوباره تقاضای اعتبارسنجی نماید و در صورت مثبت بودن نتیجه آن را در پایگاه داده خود ذخیره کرده و وضعیت رسید تراکنش را به مصرف شده تغییر دهد.
۲. در صورتی که جواب تابع تایید تراکنش، به هر دلیلی به دست فروشنده نرسد (Timeout شود، مشکل شبکه پیش آمده باشد و ...) فروشنده باید به تعداد مشخصی مجدداً سعی نماید. دقت شود تکرار در صورتی باید انجام شود که جواب به دست فروشنده نرسیده باشد نه اینکه نتیجه آن در فیلد resultCode مشخص شده باشد. بعد از تعداد مشخصی تلاش از جانب فروشنده اگر هنوز جواب دریافت نشد، فروشنده باید سعی کند تراکنش را به طور کامل برگشت بزند.
۳. در صورتی که تراکنش به هر دلیلی در مدت زمان مشخصی (این زمان در بانک تعیین می‌شود و فروشنده باید از بانک درخواست کند این زمان به او اعلام شود) از جانب فروشنده تایید نشد، بانک اقدام به برگشت زدن تراکنش خواهد کرد.
۴. در این روش پرداخت، امکان اینکه یک رسید دیجیتالی در دو فروشنده‌ی مختلف استفاده شود، وجود ندارد.
۵. مسولیت جلوگیری از Double Spending بر عهده فروشنده است و در صورت ضعفی در پیاده‌سازی سایت فروشنده، ضرر آن متوجه خود اوست.
۶. امنیت این بخش از سیستم به کمک SSL و ACL تامین شده است.
۷. مدیریت ریسک این بخش با استفاده از سقف‌های برداشت برای موجودیت خریدار و الگوهای فروش برای فروشنده صورت می‌گیرد. خریدار با تعیین سقف انتقال سپرده‌های خود می‌تواند میزان ریسک سپرده‌های خود را مدیریت نماید. همچنین فروشنده با معرفی الگوی فروش خود به بانک، حداکثر مبالغ فروش خود را می‌تواند معرفی نماید.

## ۲. انجام تراکنش پرداخت قبض

فروشنده کاربرش را به نقطه‌ای می‌رساند که آماده پرداخت قبض می‌باشد. بهتر است فروشنده اطلاعات قبض مورد نظر را به کاربرش نمایش داده و از او بخواهد که با زدن دکمه صفحه این قبض را در سایت بانک پرداخت کند. در این نقطه فروشنده باید کاربر را به سایت صدور رسید دیجیتالی redirect کند. تمامی پارامترهایی که به این سایت به صورت POST می‌دهد عبارتند از: (تمامی پارامترهای باید با نام نوشته شده در زیر به سایت بانک ارسال شود، توجه کنید که بزرگی و کوچکی حروف نیز مهم است).

- billId (شناسه قبض)
- paymentId (شناسه پرداخت قبض)
- returnUrl (آدرس سایت فروشنده که بعد از اتمام عملیات کاربر باید به آنجا برگردد)
- language (زبان مورد استفاده کاربر)

نشانی‌ای که فروشنده باید این فرم را به آن submit کند را باید از بانک دریافت کند ولی به طور مثال باید چیزی شبیه به مورد زیر باشد:

[https://bank\\_site.ir/card\\_service/billController.html](https://bank_site.ir/card_service/billController.html)

پارامترهای ارسالی بین فروشنده و بانک در ضمیمه ج توضیح داده شده است.  
توجه کنید که این ارتباط باید در بستر SSL قرار بگیرد (اجباری است).

در سایت صدور رسید دیجیتالی، کاربر اطلاعات کارت خود را که شامل موارد زیر است، باید وارد کند:

- شماره کارت (PAN)
- کلمه عبور (PIN)
- تاریخ انقضای کارت (ExpDate)
- شماره CVV۲

البته موارد بالا در سایت صدور رسید دیجیتالی پیاده سازی شده و فروشنده هیچ کار خاصی در موارد پارامترهای بالا ندارد.

سایت صدور رسید دیجیتالی پس پرداخت قبض، کاربر را دوباره به سایت فروشنده redirect می کند (به طوری که رفت و آمد خریدار از سایت فروشنده به سایت صدور رسید دیجیتالی و سپس به سایت فروشنده در همان session سایت فروشنده صورت می گیرد). فروشنده پارامترهای پاس شده از سایت صدور رسید دیجیتالی را از بدنه فرم (POST) باید دریافت کند.  
پارامترها عبارتند از:

- redirectUrl
- refNum
- billId
- paymentId
- state
- language
- CardPanHash

پارامتر **CardPanHash** با استفاده از روش **SHA۲۵۶** شماره **Pan** کارت خریدار را تبدیل به رشته ای **Hash** شده می کند و برای فروشنده ارسال می نماید.

فروشنده می تواند بر اساس وضعیت تراکنش، موفقیت آمیز بودن تراکنش را تشخیص دهد. شرح کامل پارامترها در ضمیمه ج آورده شده است.

اگر وضعیت تراکنش ok بود، به این معنی است که قبض ارسال شده از فروشنده به بانک پرداخت شده است. توجه کنید که به دلیل اینکه فرم پرداخت قبض از کاربر به سایت بانک ارسال می شود کاربر می تواند اطلاعات قبض را تغییر دهد و قبض دیگری را پرداخت کند، برای کنترل این روند فروشنده پس از گرفتن جواب ok باید با استفاده از متد گزارشگیری وب سرویس مطمئن شود که کاربر قبض مورد نظر فروشنده را پرداخت کرده است، نه قبض دیگری را.

### نکات:

۱. امنیت این بخش از سیستم به کمک SSL و ACL تامین شده است.
۲. مدیریت ریسک این بخش با استفاده از سقفهای برداشت برای موجودیت خریدار و الگوهای فروش برای فروشنده صورت می گیرد. خریدار با تعیین سقف انتقال سپرده های خود می تواند میزان ریسک سپرده های خود را مدیریت نماید. همچنین فروشنده با معرفی الگوی فروش خود به بانک، حداکثر مبالغ فروش خود را می تواند معرفی نماید.



۳. فروشنده باید قسمتی در سایت خود فراهم کند تا کاربر بتواند پرداخت قبضه‌هایی که بابت آنها جوابی دریافت نکرده است را به فروشنده اطلاع رسانی کند. این حالت زمانی رخ می‌دهد که پس از پرداخت قبضه ارتباط کاربر با سایت بانک قطع شود و کاربر نتواند فرم مربوطه را به اطلاع فروشنده برساند. بنابراین کاربر باید بتواند به سایت فروشنده مراجعه کند و فروشنده را از قبض پرداخت شده آگاه کند. فروشنده پس از دریافت این درخواست، باید با استفاده از متد گزارش‌گیری وب سرویس، پرداخت شدن قبضه مورد نظر را تایید کند و پس از آن می‌تواند سرویس خود را به مشتری ارائه کند.

### ۳. برگشت تراکنش خرید

- فروشنده می‌تواند امکان لغو خرید را پیاده‌سازی کند. این امکان ممکن است به دو صورت مورد نیاز واقع شود:
- فروشنده لغو یک خرید را لازم بداند (برای مثال ممکن است فروشنده دیگر کالا یا سرویس مورد نظر خریدار را برای تحویل نداشته باشد).
  - خریدار مایل به لغو خرید باشد.

در هر یک از دو حالت بالا، فروشنده می‌تواند یک خرید را به دو صورت برگشت زند:

- ۱) برگشت کامل (Full Reverse): سند به صورت کامل برگشت خورده سپرده خریدار به مبلغ خرید بستانکار می‌شود و سپرده فروشنده به مبلغ خرید بدهکار. برای این کار فروشنده باید با استفاده از پایگاه داده خود، رسید دیجیتالی مربوط به Reservation Number مورد نظر خود را استخراج نماید و سپس با چک کردن وضعیت برگشت‌خورده‌گی آن مطمئن شود که قبلاً برگشت جزئی یا کامل نخورده باشد (البته این چک برای صرفه‌جویی در ارتباطات شبکه‌ای است و گرنه این چک در سمت بانک نیز صورت می‌گیرد). وی سپس متد برگشت وب سرویس را با پارامترهای ذیل فرا خواهد خواند. مقدار برگشتی این متد کدی است که نتیجه برگشت را نشان می‌دهد. فروشنده سپس پایگاه داده خود را به روز می‌نماید. شرح خروجی این متد در ضمیمه ب آورده شده است.
- ۲) برگشت ناقص (Partial Reverse): فروشنده بخشی از مبلغ خرید را برگشت می‌زند. برای این کار فروشنده بر اساس Business Rule های خود مبلغ برگشتی را محاسبه می‌کند و با استفاده از پایگاه داده خود، رسید دیجیتالی مربوط به Reservation Number مورد نظر خود را استخراج می‌نماید. سپس فروشنده باید وضعیت برگشت‌خورده‌گی رسید دیجیتالی را چک کند و تنها در صورتی آن را برای برگشت جزئی استفاده نماید که یا پیشتر تایید شده باشد و یا مبلغ برگشتی به اضافه مبالغ برگشت‌های جزئی پیشین از کل مبلغ سند بیشتر نباشد (البته این چک کردن نیز برای صرفه‌جویی در ارتباطات شبکه‌ای است چراکه این چک در سمت بانک نیز صورت می‌گیرد). در این صورت فروشنده می‌تواند همان متد برگشت وب سرویس را فراخواند. مقدار برگشتی این متد فیلدی دارد به نام resultCode. این کد، کدی است که نتیجه برگشت را نشان می‌دهد. در این حالت فروشنده باید پایگاه داده خود را به روز نماید و وضعیت رسید دیجیتالی مربوط به آن Reservation Number را برگشت خورده نماید تا برگشت‌های جزئی بعدی را مدیریت نماید.

### نکات:

- ۱) دوباره تاکید می‌شود که لزومی ندارد وضعیت برگشت‌خورده‌گی رسید دیجیتالی پیش از صدور درخواست آن، توسط فروشنده چک شود؛ چراکه بانک قبل از اجرای دستور فروشنده، مبلغ برگشت نخورده رسید دیجیتالی را محاسبه می‌کند و در صورت کفایت این مبلغ، دستور برگشت را اجرا می‌نماید.
- ۲) دستور برگشت الزاماً باید از سوی فروشنده صادر شود و بانک برای آن که مطمئن گردد که این دستور از سوی فروشنده صادر شده است، نام کاربری فروشنده و کلمه عبور وی را چک می‌کند. در نتیجه اگر سیاست فروشنده به

گونه‌ای است که خریدار می‌تواند فرایند برگشت را خود آغاز کند، باید نام کاربری و کلمه عبور خود را در پیاده‌سازی سایت خود، hard code نماید.

۳) امنیت این بخش از سیستم به کمک SSL، ACL و Merchant Username and Password تامین شده است.

## پشتیبانی مشتریان

از آن‌جا که سایت شما در حال ارائه خدمات ۲۴\*۷ است، لازم است که پشتیبانی تماس فیزیکی ۲۴\*۷ نیز در سایت شما تعبیه گردد.

# ضمیمه الف: پیاده سازی

## تکنولوژی‌های مورد استفاده:

برای پیاده سازی این سیستم از تکنولوژی Web Services استفاده شده است و ارتباطی نقطه به نقطه بین سایت فروشنده و بانک برقرار می‌گردد. در این مدل سرویس‌هایی که از جانب بانک ارائه می‌شود در قالب توابعی در اختیار فروشنده قرار می‌گیرد تا در سایت خود از آنان استفاده نماید. فروشنده می‌تواند این توابع را به صورت مسقیم از متن برنامه‌ی سایت خود صدا زند.

انتقال اطلاعات در لایه پایین‌تر مطابق پروتکل SOAP<sup>۲</sup> خواهد بود. خود پروتکل SOAP برای دسته بندی و مدیریت داده‌ها از استاندارد XML استفاده می‌کند. در لایه ترانسپورت نیز انتقال داده‌ها بر عهده پروتکل HTTP و یا HTTPS می‌باشد. حسن استفاده از web service در این است که فروشنده می‌تواند از هر تکنولوژی‌ای برای پیاده‌سازی سایت خود استفاده نماید. مثال‌هایی از نحوه consume کردن و فراخواندن این web methodها برای عموم تکنولوژی‌های موجود تهیه شده است و فروشنده می‌تواند از آنان برای سایت خود بهره گیرد.

## نیازمندی استفاده از Web Methods

- برای استفاده از هر یک از web methodها تحت تکنولوژی PHP، باید ابتدا PHP Module ی را به نام NuSOAP به سیستم فروشنده اضافه نمایید و مسیر آن را در کد خود (مطابق نمونه کد داده شده) مشخص نمایید.
- تحت تکنولوژی ASP باید ابتدا SOAPSDK را بر روی سیستم فروشنده نصب نمایید. این SDK را می‌توانید از <http://www.microsoft.com/downloads> دریافت نمایید.
- در تکنولوژی ASP.NET می‌توان از پشتیبانی خود NET. برای صدا زدن web method استفاده کرد.
- در تکنولوژی دلفی نیز می‌توان از پشتیبانی خود زبان برنامه نویسی استفاده کرد (البته در نسخه های خاصی از دلفی این پشتیبانی وجود دارد). اگر نسخه دلفی شما webservice را به صورت محلی پشتیبانی نمی‌کند، می‌توان از بسته نرم افزاری SOAPSDK استفاده نمایید. SOAPSDK به صورت یک ActiveX در می‌آید که در اکثر زبانهای برنامه نویسی می‌توان از آن استفاده کرد.
- در تکنولوژی Perl استفاده از بسته نرم افزاری SOAP-Lite توصیه می‌شود.
- برای تکنولوژی جاوا باید، یک wrapper بر روی interface متدها ایجاد کنید و باید آن را (به همراه دیگر jar های مرتبط با استفاده از web services) در classpath خود قرار دهید.

---

<sup>۲</sup>Simple Object Access Protocol

## توجه:

در تکنولوژی **.net framework** ممکن است پارامترهایی با عنوان Specified در برنامه دیده شود. در صورتی که مقدار Specified مربوط به یک پارامتر برابر true باشد، مقدار مورد نظر به وب سرویس ارسال خواهد شد و در غیر این صورت ارسال نخواهد شد.

## ضمیمه ب: شرح Web Methods

۶ متد (Web Method) در اختیار فروشندگان قرار می‌گیرد که شرح پارامترهای ورودی و خروجی هر یک در زیر آمده است. توجه داشته باشید که نام type مقادیر ورودی و برگشتی، نام‌های عامی می‌باشند که در هر زبان برنامه‌نویسی ممکن است تفاوت یابند. همچنین مقدار برگشتی متد با نام خود آن متد مشخص شده است.

### متد ورود (login)

متدی است برای ورود فروشنده به سیستم که باید قبل از صدا زدن دیگر متد ها مورد استفاده قرار گیرد. این متد نام کاربری و کلمه عبور فروشنده را دریافت می‌کند و یک رشته به عنوان شناسه جلسه کاری کاربر که در سیستم SESSION\_ID نامیده می‌شود برمی‌گرداند. که این رشته باید در دیگر سرویس ها به عنوان پارامتر ورودی ارسال شود.

### نکات

- صدا زدن این متد قبل از هر بار صدا زدن دیگر سرویس ها اجباری نیست و در صورتی که جلسه کاری فروشنده به اتمام نرسیده باشد فروشنده همچنان می‌تواند از این SESSION\_ID استفاده کند.
- در صورتی که در هنگام صدا زدن دیگر سرویس ها خطای WsClientAddressException دریافت شد، فروشنده باید ابتدا یکبار دیگر متد لاگین را فراخوانی کند تا SESSION\_ID جدید بگیرد و سپس به فراخوانی دیگر متد ها بپردازد.

این متد یک پارامتر از نوع LoginRequest می‌گیرد که مقادیر آن در زیر تشریح شده اند:

نام	نوع	شرح
username	String	نام کاربری فروشنده است که از بانک دریافت کرده است.
password	String	کلمه عبور فروشنده است که از بانک دریافت کرده است. فروشنده می‌تواند این مقدار را توسط یک web application که آدرس آن را بانک در اختیارش گذاشته است تغییر دهد.

مقدار خروجی این متد رشته ای است که باید فروشنده آنرا سمت خود نگه دارد و برای فراخوانی دیگر متدها از آن استفاده کند. این رشته در این مستند SESSION\_ID نامیده می‌شود.

خطاهای رخ داده در این متد:

- WsInvalidCredentialException: در صورتی رخ می‌دهد که نام کاربری یا کلمه عبور فروشنده نادرست باشد.
- WsBlockUserException: در صورتی رخ می‌دهد که فروشنده به دلیل تلاش های زیاد برای ورود به سیستم با کلمه عبور نادرست قفل شده باشد. در این صورت فروشنده یک مدت زمان خاص – که در بانک تعیین می‌شود – نمی‌تواند با سیستم کار کند ولی بعد از آن قفل فروشنده برداشته شده و می‌تواند از سیستم استفاده کند. این ممکن است به دلیل تلاش افراد خرابکار در سیستم اتفاق بیفتد. البته به دلیل اینکه نام کاربری فروشنده قفل شده است جای هیچ نگرانی وجود ندارد.

## متد خروج (logout)

متدی است برای منقضی کردن جلسه کاری فروشنده، که فروشنده به منظور اطمینان از منقضی شدن جلسه کاری اش باید این متد را صدا بزند.

ورود این متد یک پارامتر از نوع `WsContext` است که باید درون آن مقدار `SESSION_ID` دریافت شده از متد ورود را قرار دهید. نحوه قرار دادن مقدار `SESSION_ID` درون این پارامتر به این شکل است که باید نام دقیق `"SESSION_ID"` و مقدار `SESSION_ID` درون `WsContext` قرار گیرد. برای مثال اگر از جاوا استفاده می کنید کد آن به این شکل می شود:

```
String loginMethodResult = "۲b۹e۹۴۴۹-f۷۲۲-۴۰ac-۸c۳۷-۵۰۰d۹e۱c۳e۴۰";//login response
```

```
WsContext context = new WsContext();
```

```
context.addData("SESSION_ID", loginMethodResult);
```

در کد بالا مقدار `WsContext.SESSION_ID` در کد نیز تعریف شده است و بهتر است به جای نوشتن `"SESSION_ID"` از عبارت `WsContext.SESSION_ID` استفاده شود.

این متد خروجی ندارد.

**توجه:**

در صورت موجود نبودن متد `"addDate"`، باید یک `new HashMap` کرده و `SESSION_ID` و مقدار آن را به `HashMap`، `add` کنید. سپس `Map` فوق را درون `WsContext` قرار دهید. برای مثال اگر از جاوا استفاده می کنید کد آن به این شکل می شود:

```
String loginMethodResult = "۲b۹e۹۴۴۹-f۷۲۲-۴۰ac-۸c۳۷-۵۰۰d۹e۱c۳e۴۰";//login response
```

```
WsContext context = new WsContext();
```

```
HashMap<String, String> data = new HashMap<String, String>();
```

```
data.put("SESSION_ID", loginMethodResult);
```

```
context.setData(data);
```

خطاهای رخ داده در این متد:

- `WsInvalidSessionException`: در صورتی رخ می دهد که مقدار `SESSION_ID` اشتباه باشد یا قبلاً `EXPIRE` شده باشد.

## متد درخواست تسویه فروشنده (merchantSettlementRequest)

متدی است برای تسویه ترمینال های فروشنده. این متد زمان تسویه ترمینال های فروشنده را به زمان حال تغییر می دهد تا در زمان تسویه خودکار، عملیات تسویه آن انجام شود.

ورود این متد یک پارامتر از نوع `WsContext` است که باید درون آن مقدار `SESSION_ID` دریافت شده از متد ورود را قرار دهید. نحوه قرار دادن مقدار `SESSION_ID` درون این پارامتر به این شکل است که باید نام دقیق `"SESSION_ID"` و مقدار `SESSION_ID` درون `WsContext` قرار گیرد. برای مثال اگر از جاوا استفاده می کنید کد آن به این شکل می شود:

```
String loginMethodResult = "۲b۹e۹۴۴۹-f۷۲۲-۴۰ac-۸c۳۷-۵۰۰d۹e۱c۳e۴۰";//login response
```

```
WsContext context = new WsContext();
```

```
context.addData("SESSION_ID", loginMethodResult);
```

در کد بالا مقدار `WsContext.SESSION_ID` در کد نیز تعریف شده است و بهتر است به جای نوشتن `"SESSION_ID"` از عبارت `WsContext.SESSION_ID` استفاده شود.

**توجه:**

در صورت موجود نبودن متد "addDate"، باید یک new HashMap، new کرده و مقدار آن را به add، HashMap کنید. سپس Map فوق را درون WsContext قرار دهید. برای مثال اگر از جاوا استفاده می کنید کد آن به این شکل می شود:

```
String loginMethodResult = "۲b۹e۹۴۴۹-f۷۲۲-۴۰ac-۸c۳۷-۵۰۰d۹e۱c۳e۴۰"; //login response
WsContext context = new WsContext();
HashMap<String, String> data = new HashMap<String, String>();
data.put("SESSION_ID", loginMethodResult);
context.setData(data);
```

این متد خروجی ندارد.

خطاهای رخ داده در این متد:

- WsInvalidSessionException: در صورتی رخ می دهد که مقدار SESSION\_ID اشتباه باشد یا قبلا EXPIRE شده باشد.
- RequestAlreadyInProcessException: در صورتی رخ می دهد که درخواست تسویه قبلا ثبت شده باشد یا زمان تسویه همه ترمینال های فروشنده رسیده باشد.

## متد تایید تراکنش (verifyTransaction)

متدی است برای تایید تراکنش تا فروشنده از مقدار خرید انجام شده با خبر شود و همچنین باعث شود تراکنش به صورت سیستمی برگشت نخورد و مقدار خرید از حساب فروشنده به کارت مشتری واریز نشود. ورودی این متد دو پارامتر است که اولی از نوع WsContext می باشد و دومی از نوع VerifyRequest. نحوه پر کردن پارامتر اول مانند متد خروج است. پارامتر دوم یک لیست از نوع String دریافت می کند که مقدار شماره پیگیری (RefNum) هایی هستند که فروشنده می خواهد آنرا تایید کند. اگر بعد از هر خرید یکبار این متد فراخوانی شود، این لیست حاوی یک رکورد بیشتر نمی باشد ولی فروشنده در زمان هایی که خرید های بسیار زیادی در واحد زمان دارد می تواند تعدادی خرید انجام دهد و بعد از انجام شدن یک مقدار خاص یا گذشت یک زمان خاص تمامی آنها را برای سایت ارسال کند.

خروجی این متد یک پارامتر است از نوع VerifyResponse که تنها حاوی یک لیست از VerifyResponseResult ها می باشد. مقادیر VerifyResponseResult در جدول زیر تشریح شده اند:

نام	نوع	شرح
amount	BigDecimal	مقدار خرید انجام شده که تایید شده است.
refNum	String	مقدار شماره پیگیری که ارسال شده بود.
verificationError	VerificationError	نوع خطای رخ داده برای تراکنشی که مقدار RefNum آن در بالا آمده است. این متد بصورت دسته ای است و توسط این متد می توان چند شماره پیگیری را برای تایید ارسال کرد و به ازای هر RefNum باید خطای آن مشخص شود. بنابراین VerificationError به صورت enum تعریف شده و مقادیر زیر را دارد: • TRANSACTION_NOT_FOUND_ERROR: این

خطا هنگامی رخ می دهد که تراکنش اصلی پیدا نشود یا شماره پیگیری ارسال شده اشتباه باشد.		
<ul style="list-style-type: none"> <li>TRANSACTION_IS_NOT_VERIFIABLE: این خطا هنگامی رخ می دهد که تراکنش قبل از تایید شدن توسط سیستم برگشت خورده باشد.</li> <li>INVALID_SESSION_EXCEPTION: این خطا به این معنی است که جلسه کاری کاربر در سمت سرور منقضی شده و فروشنده باید دوباره درخواست خود را تکرار کند.</li> <li>SYSTEM_ERROR: این خطا به معنی خطای داخلی سیستم بوده و در صورت تکرار فرایند و دریافت مجدد این خطا باید به پشتیبانی بانک اطلاع داده شود.</li> </ul>		

خطاهای رخ داده در این متد:

- WsValidationException: در صورتی که مقادیر ورودی نامعتبر باشند.
- WsInvalidSessionException: در صورتی که SESSION\_ID ارسال شده منقضی شده باشد.

## متد تایید تراکنش امن (secureVerifyTransaction)

متدی است برای تایید تراکنش تا فروشنده از مقدار خرید انجام شده با خبر شود و همچنین باعث شود تراکنش به صورت سیستمی برگشت نخورد و مقدار خرید از حساب فروشنده به کارت مشتری واریز نشود. در تایید تراکنش امن به غیر از شماره پیگیری، شماره خرید نیز دریافت می شود.

ورودی این متد دو پارامتر است که اولی از نوع WsContext می باشد و دومی از نوع SecureVerifyRequest. نحوه پر کردن پارامتر اول مانند متد خروج است. پارامتر دوم یک لیست از نوع SecureVerifyInfo دریافت می کند که این نوع شامل دو مقدار شماره پیگیری (RefNum) و شماره خرید (ResNum) می باشد که فروشنده می خواهد آن را تایید کند. اگر بعد از هر خرید یکبار این متد فراخوانی شود، این لیست حاوی یک رکورد بیشتر نمی باشد ولی فروشنده در زمان هایی که خرید های بسیار زیادی در واحد زمان دارد می تواند تعدادی خرید انجام دهد و بعد از انجام شدن یک مقدار خاص یا گذشت یک زمان خاص تمامی آنها را برای سایت ارسال کند.

خروجی این متد یک پارامتر است از نوع SecureVerifyResponse که تنها حاوی یک لیست از SecureVerifyResponseResult ها می باشد. مقادیر SecureVerifyResponseResult در جدول زیر تشریح شده اند:

نام	نوع	شرح
amount	BigDecimal	مقدار خرید انجام شده که تایید شده است.
refNum	String	مقدار شماره پیگیری که ارسال شده بود.
resNum	String	مقدار شماره خرید که ارسال شده بود.
verificationError	VerificationError	نوع خطای رخ داده برای تراکنشی که مقدار RefNum آن در بالا آمده



<p>است.</p> <p>این متد بصورت دسته ای است و توسط این متد می توان چند شماره پیگیری را برای تایید ارسال کرد و به ازای هر RefNum باید خطای آن مشخص شود. بنابراین VerificationError به صورت enum تعریف شده و مقادیر زیر را دارد:</p> <ul style="list-style-type: none"> <li>TRANSACTION_NOT_FOUND_ERROR: این خطا هنگامی رخ می دهد که تراکنش اصلی پیدا نشود یا شماره پیگیری ارسال شده اشتباه باشد.</li> <li>TRANSACTION_IS_NOT_VERIFIABLE: این خطا هنگامی رخ می دهد که تراکنش قبل از تایید شدن توسط سیستم برگشت خورده باشد.</li> </ul>		
--	--	--

خطاهای رخ داده در این متد:

- WsValidationException: در صورتی که مقادیر ورودی نامعتبر باشند.
- WsInvalidSessionException: در صورتی که SESSION\_ID ارسال شده منقضی شده باشد.

## متد برگشت تراکنش (reverseTransaction)

توسط این متد فروشنده می تواند یک خرید را به صورت کامل یا ناقص برگشت بزند. در صورت برگشت کامل مقدار خرید شده به کارت کاربر برگشت زده خواهد شد و در صورت برگشت ناقص مقداری از خرید انجام شده که فروشنده در خواست آنرا داده است به کارت کاربر واریز خواهد شد.

فروشنده می تواند به دلیل سیاست های خودش این مورد را در سایت خود پیاده سازی کند. ورودی این متد دو پارامتر است که پارامتر اول از نوع WsContext است که نحوه پر کردن آن در متد خروج آمده است. پارامتر دوم از نوع ReverseRequest است که مقادیر آن در جدول زیر تشریح شده اند:

نام	نوع	شرح
mainTransactionRefNum	String	شماره پیگیری خرید انجام شده که فروشنده می خواهد آنرا برگشت بزند.
reverseTransactionResNum	String	شماره کدی که فروشنده باید برای تراکنش برگشت همانند خرید تعیین کند تا تکرار صورت نگیرد. این پارامتر می تواند ترکیبی از حرف و عدد حداکثر تا ۵۰ حرف می باشد.
amount	BigDecimal	مقداری که فروشنده می خواهد به کارت کاربر برگشت بزند. حتما باید از مقدار خرید کوچکتر یا مساوی با آن باشد.

خروجی این متد شماره پیگیری این سند است که باید توسط فروشنده ذخیره شود.

خطاهای رخ داده در این متد:

- **WsAmountConstraintViolationException**: در صورتی که مبلغ برگشتی به اضافه ی مبلغ های برگشت خورده قبلی بیش از مبلغ اصلی تراکنش شود.
- **WsAuthenticationException**: در صورت عدم دسترسی به این سرویس این خطا رخ می دهد. با بانک تماس بگیرید.
- **WsClientAddressException**: در صورتی که فروشنده از آدرس IP غیر از آدرسی که به بانک اعلام کرده بخواهد این سرویس را فراخوانی کند.
- **WsPaymentReverseException**: در صورتی که امکان برگشت ناقص وجود نداشته باشد و فروشنده بخواهد تراکنشی را برگشت ناقص بزند.
- **WsSystemMalFunctionException**: در صورتی که خطایی در شبکه شتاب بوجود آید و امکان واریز مبلغ به کارت خریدار وجود نداشته باشد و یا خطایی در زیر سیستم های مرتبط رخ داده باشد.
- **WsValidationException**: در صورتی که پارامترهای ارسالی معتبر نباشند.
- **WsInsufficientFundsException**: در صورتی که موجودی فروشنده کمتر از مقدار برگشتی باشد.
- **WebServiceException**: در صورتی که خطایی رخ داده باشد ولی در گروه خطاهای بالا ننگنجد.

## متد گزارش تراکنش (reportTransaction)

متدی است برای گزارش گیری از خرید های انجام شده. توسط این متد فروشنده می تواند از آخرین وضعیت تراکنش باخبر شود و سیاست لازم برای برخورد با مشتری خود را بکار گیرد.

ورودی این متد ۲ پارامتر از نوع **WsContext** و **ReportRequest** است که نحوه پر کردن **WsContext** در متد خروج آمده است. پارامتر **ReportRequest** کلاسی است که از تعدادی پارامتر دیگر تشکیل شده که شرح آن در جدول زیر آمده است. تمامی پارامتر های این کلاس اختیاری هستند بجز **onlyReversed** و **length.offset** که حتما باید ارسال شوند.

نام	نوع	شرح
transactionType	TransactionType	یک نوع شمارشی است (enum) که نوع تراکنش را مشخص می کند. می تواند PURCHASE (خرید) و یا BILL_PAYMENT (پرداخت قبض) باشد.
transactionState	TransactionState	یک نوع شمارشی است (enum) که وضعیت تراکنش را مشخص می کند. می تواند SUCCESS (تراکنش انجام شده) یا NOT_VERIFIED (تراکنشی که انجام شده ولی هنوز تایید نشده) و یا FAILED (تراکنشی که انجام نشده و یا بعد از انجام به دلیل عدم تایید برگشت سیستمی خورده است) باشد.
amountMin	BigDecimal	حداقل مبلغ را نشان می دهد. در صورت تعیین این پارامتر در خروجی همه تراکنش ها مبلغی بیش یا مساوی با این مقدار دارند.
amountMax	BigDecimal	حداکثر مبلغ را نشان می دهد. در صورت تعیین این پارامتر در خروجی همه تراکنش ها مبلغی کمتر یا مساوی با این مقدار دارند.

حداقل تاریخ را نشان می دهد. در صورت تعیین این پارامتر در خروجی همه تراکنش ها پس از این تاریخ انجام شده اند. این تاریخ به تقویم میلادی باید باشد. به حالت: yyyy-mm-ddThh:MM:ss+۰۰:۰۰	Date	timeMin
حداکثر تاریخ را نشان می دهد. در صورت تعیین این پارامتر در خروجی همه تراکنش ها پیش از این تاریخ انجام شده اند. این تاریخ به تقویم میلادی باید باشد. به حالت: yyyy-mm-ddThh:MM:ss+۰۰:۰۰	Date	timeMax
اگر true تنظیم شود تنها تراکنش هایی را در خروجی نشان می دهد که توسط فروشنده برگشت ناقص خورده اند. به صورت پیش فرض این مقدار false است.	boolean	onlyReversed
شماره پیگیری تراکنش انجام شده است. در صورت تنظیم این مقدار و صحیح بودن آن تنها یک تراکنش برگردانده خواهد شد.	String	refNum
شماره پیگیری مخصوص مشتری برای تراکنش انجام شده است. در صورت تنظیم این مقدار و صحیح بودن آن تنها یک تراکنش برگردانده خواهد شد.	String	customerRefNum
شماره کدی که فروشنده کالای خود را شناسایی کرده است می باشد. در صورت تنظیم این مقدار و صحیح بودن آن تنها یک تراکنش برگردانده خواهد شد. در صورتی که transactionType بر روی BILL_PAYMENT تنظیم شده است این مقدار باید خالی (null) باشد.	String	resNum
مجموعه ای است از انواع قبض هایی که فروشنده می خواهد دریافت کند. با تنظیم این مقدار تنها تراکنش های پرداخت قبضی برگردانده خواهند شد که قبضشان از نوع گفته شده باشد. مقادیر آن عبارتند از: .GAS .ELECTRICITY .WATER .UNKNOWN .IMMOBILE_PHONE (تلفن ثابت)، .MOBILE_PHONE (تلفن همراه)، .MUNICIPALITY_DUE (قبض شهرداری با کد ۷)، .MUNICIPALITY_۷ (قبض شهرداری با کد ۷)، .TAX (امور مالیاتی) و .CUSTOM در صورتی که transactionType بر روی PURCHASE تنظیم شده است این مقدار باید خالی (null) باشد.	Set<BillType>	billTypes
شناسه قبض می باشد. در صورتی که transactionType بر روی PURCHASE تنظیم شده است این مقدار باید خالی (null) باشد.	String	billId
شناسه پرداخت قبض می باشد. در صورتی که transactionType بر روی PURCHASE تنظیم شده است این مقدار باید خالی (null) باشد.	String	paymentId

باشد.		
یک نوع شمارشی است (enum) که به منظور مرتب سازی خروجی باید مورد استفاده قرار گیرد. می تواند مقادیر TRANSACTION_TIME (برای مرتب سازی بر اساس زمان انجام تراکنش) و یا AMOUNT (برای مرتب سازی بر اساس مبلغ تراکنش) را داشته باشد. در صورت عدم تعیین این مقدار مرتب بودن تراکنش های برگشتی تضمین نمی شود.	OrderField	orderField
یک نوع شمارشی است (enum) که به منظور تعیین نحوه مرتب سازی خروجی مورد استفاده قرار می گیرد. می تواند مقادیر ASC (افزایشی) و یا DESC (کاهشی) را داشته باشد.	OrderType	orderType
به منظور اعمال صفحه بندی برای نمایش تراکنش ها باید مورد استفاده قرار گیرد. در صورت تنظیم این مقدار تراکنش هایی نمایش داده می شوند که از شماره اندیسی که تنظیم شده است به بعد شروع شوند. اندیس از صفر شروع می شود.	long	offset
تعداد رکورد های برگشتی را مشخص می کند. در صورت تعیین این مقدار در خروجی تنها همین تعداد تراکنش وجود دارد.	short	length

در خروجی این متد دو پارامتر وجود دارد. اولی از نوع <ReportResponseResult> List است که در زیر تشریح شده است و دومی تعداد کل رکوردهایی است که - با فیلتر داده شده در ورودی - در پایگاه داده وجود دارد. پارامتر اول فهرستی است از نوع ReportResponseResult که هر کدام از موجودیت های این فهرست مشخصه های زیر را دارند که در جدول زیر تشریح شده اند.

نام	نوع	شرح
id	long	شماره کد تراکنش می باشد که برای مشاهده جزئیات تراکنش باید مورد استفاده قرار گیرد.
transactionType	TransactionType	نوع تراکنش را نشان می دهد. برای اطلاعات بیشتر به همین نوع در ورودی متد مراجعه کنید.
transactionState	TransactionState	وضعیت تراکنش را نشان می دهد. برای اطلاعات بیشتر به همین نوع در ورودی متد مراجعه کنید.
amount	BigDecimal	مبلغ واقعی تراکنش انجام شده را نشان می دهد.
time	Date	زمان واقعی انجام تراکنش را نشان می دهد. به تقویم میلادی می باشد. به حالت: yyyy-mm-ddThh:MM:ss+۰۰:۰۰
refNum	String	شماره پیگیری تراکنش را نشان می دهد. در صورت عدم انجام این تراکنش این مقدار خالی (null) است.

String	resNum	شماره کدی که فروشنده کالای خود را شناسایی کرده را نشان می دهد.
BillType	billType	نوعی شمارشی (enum) است که نوع قبض پرداخت شده را نشان می دهد. در صورتی که تراکنش از نوع خرید کالا باشد این مقدار خالی (null) است. برای اطلاعات بیشتر به همین نوع در ورودی متد مراجعه کنید.
String	billId	شناسه قبض را نشان می دهد. در صورتی که تراکنش از نوع خرید کالا باشد این مقدار خالی (null) است.
String	paymentId	شناسه پرداخت قبض را نشان می دهد. در صورتی که تراکنش از نوع خرید کالا باشد این مقدار خالی (null) است.

خطاهایی که در هنگام فراخوانی این متد ممکن است رخ دهد عبارتند از:  
WsValidationException: در صورتی که داده های وارد شده معتبر نباشند.

### متد گزارش جزئیات تراکنش (detailReportTransaction)

این متد برای گزارش گیری از یک تراکنش خاص باید مورد استفاده قرار گیرد. در واقع این متد هنگامی خروجی دارد که تراکنشی به صورت کامل یا ناقص برگشت خورده باشد. ورودی های این متد همانند متد گزارش گیری است با این تفاوت که در ورودی یک پارامتر به نام mainTransactionId از نوع Long وجود دارد که می تواند خالی باشد ولی در صورت تنظیم شدن آن تنها جزئیات تراکنشی که شناسه آن در این فیلد تنظیم شده است برگردانده خواهد شد. برای تنظیم کردن شناسه تراکنش از خروجی متد گزارش تراکنش باید استفاده کنید (فیلد id).

در خروجی این متد دو پارامتر وجود دارد که اولی از نوع <DetailReportResponseResult> List است و دومی از نوع Long که پارامتر اول در زیر تشریح شده است و پارامتر دوم تعداد کل رکوردهای یافت شده با در نظر گرفتن فیلتر ورودی را نشان می دهد. توجه کنید که کل رکوردهای یافت شده برگردانده نمی شوند و تنها به تعداد length که در ورودی دریافت شده است رکورد برگردانده می شود.

نام	نوع	شرح
id	long	شماره کد تراکنش می باشد.
amount	BigDecimal	مبلغ تراکنش فعلی را نشان می دهد. یعنی اگر برگشت خورده باشد، مبلغ تراکنش برگشت را نشان می دهد.
time	Date	زمان واقعی انجام تراکنش را نشان می دهد. به تقویم میلادی می باشد. به حالت: yyyy-mm-ddThh:MM:ss+۰۰:۰۰
refNum	String	شماره پیگیری تراکنش را نشان می دهد. در صورت عدم انجام این تراکنش این مقدار خالی (null) است.
resNum	String	شماره کدی که فروشنده تراکنش برگشت خود را شناسایی کرده را

نشانی می دهد.		
کد تراکنش اصلی (خرید یا پرداخت قبض) را نشان می دهد.	Long	mainTransactionId
یک نوع شمارشی (enum) می باشد که نوع تراکنش برگشت را نشان می دهد. مقادیر CUSTOM_REVERSED (نشان دهنده تراکنش برگشت ناقص است) و SYSTEM_REVERSED (نشان دهنده تراکنش برگشت سیستمی است) را دارد.	ReverseType	reverseType

نکته: در صورتی که مقادیر ورودی هیچ تراکنشی را مشخص نکنند این متد خطا نمی دهد و تنها خروجی خالی برمیگرداند.

خطاهایی که در هنگام فراخوانی این متد ممکن است رخ دهد عبارتند از:  
WsValidationException: در صورتی که داده های وارد شده معتبر نباشند.

## ضمیمه ج: پارامترهای تبادلی میان سایت فروشنده و

### سایت صدور رسید دیجیتالی

#### تراکنش خرید اینترنتی

پارامترهایی که سایت فروشنده به سایت صدور رسید دیجیتالی برای تراکنش خرید باید به صورت POST ارائه کند عبارت است از:

نام	اجباری	شرح	نوع
Amount	بله	مبلغ کالا که باید از کارت مشتری برداشت شود و به سپرده فروشنده ارسال شود.	عددی است که حداکثر می تواند ۱۲ رقم داشته باشد. واحد آن ریال است.
ResNum	بله	کدی است که فروشنده به کالای خود اختصاص داده و باید برای بانک ارسال کند.	ترکیبی است از عدد و حروف که می تواند حداکثر ۵۰ حرف داشته باشد.
MID	بله	کد فروشنده که بانک به فروشنده اختصاص می دهد.	می تواند ترکیبی از عدد و - باشد.
RedirectURL	بله	آدرس صفحه دریافت اطلاعات فروشنده است که بعد از خرید انتظار اطلاعات از بانک را دارد.	آدرسی کامل url سایت فروشنده است که باید به صورت HTML Encoded برای بانک ارسال شود.
language	خیر	زبان نشان داده شده به کاربر در صفحه خرید است.	می تواند مقادیر Local برای پارسی و Foreign برای انگلیسی داشته باشد.

پارامترهایی که سایت صدور رسید دیجیتالی به سایت فروشنده به صورت **POST** برمی گرداند در زیر تشریح شده اند. توجه کنید که این پارامترها را از url query string دریافت نکنید و این پارامترها در بدنه POST وجود دارند.

نام	نوع	شرح
State	String	وضعیت انجام تراکنش است. یا "ok" و یا "Canceled By User" می تواند باشد.
ResNum	String	کدی است که فروشنده به کالای خود اختصاص داده و باید برای بانک ارسال کند.
RefNum	String	شماره پیگیری تراکنش انجام شده است.
MID	String	کد فروشنده که بانک به فروشنده اختصاص می دهد.
language	String	زبان نشان داده شده به کاربر در صفحه خرید است.
CardPanHash	String	شماره کارت خریدار، با روش SHA۲۵۶، Hash شده و برای فروشنده ارسال می شود.

## تراکنش پرداخت قبض

پارامترهایی که سایت فروشنده به سایت صدور رسید دیجیتالی برای تراکنش پرداخت قبض باید به صورت POST ارائه کند عبارت است از:

نام	اجباری	شرح	نوع
billId	بله	شناسه قبض می باشد.	رشته ای از اعداد باید باشد به طول ۶ تا ۱۳ حرف.
paymentId	بله	شناسه پرداخت قبض می باشد.	رشته ای از اعداد باید باشد به طول ۶ تا ۱۳ حرف.
redirectUrl	بله	آدرس صفحه دریافت اطلاعات فروشنده است که بعد از خرید انتظار اطلاعات از بانک را دارد.	آدرسی کامل url سایت فروشنده است که باید به صورت HTML Encoded برای بانک ارسال شود.
language	خیر	زبان نشان داده شده به کاربر در صفحه خرید است.	می تواند مقادیر Local برای پارسی و Foreign برای انگلیسی داشته باشد.

پارامترهایی که سایت صدور رسید دیجیتالی به سایت فروشنده به صورت **POST** برمی گرداند در زیر تشریح شده اند. توجه کنید که این پارامترها را از url query string دریافت نکنید و این پارامترها در بدنه POST وجود دارند.

نام	نوع	شرح
state	String	وضعیت انجام تراکنش است. یا "ok" و یا "Canceled By User" می تواند باشد.
redirectUrl	String	آدرسی که فروشنده به بانک ارسال کرده بود.
refNum	String	شماره پیگیری تراکنش انجام شده است.
billId	String	شناسه قبض دریافت شده از سایت فروشنده می باشد.
paymentId	String	شناسه پرداخت دریافت شده از سایت فروشنده می باشد.
language	String	زبان نشان داده شده به کاربر در صفحه خرید است.
CardPanHash	String	شماره کارت خریدار، با روش SHA۲۵۶، Hash شده و برای فروشنده ارسال می شود.

	شود.	
--	------	--

- **نکته ۱:** در صورتی که بانک شماره پیگیری تهی به فروشنده برگرداند، به معنای این است که مشکلی در انتقال توسط خریدار بوجود آمده است.
- **نکته ۲:** در صورتی که تراکنش با موفقیت انجام شده باشد، State برابر با ok خواهد بود و فروشنده می تواند سرویس خود را به کاربر ارایه کند.
- **نکته ۴:** سیستم نسبت به حروف بزرگ و کوچک حساس است و لذا نام متغیرها را به همین ترتیبی که در این مستند ذکر شده است ارسال دارید.
- **نکته ۵:** کارتهای شتاب دو شماره رمز (PIN) دارند. یک شماره رمز برداشت وجه از ATM و خرید از طریق سامانه های POS و شماره رمز دیگر برای خریدهای اینترنتی استفاده می شود. در حال حاضر این دو شماره رمز یکسان هستند و همانی است که کاربر با آن کار می کند. از کار افتادن کارت به خاطر ورود شماره رمز اشتباه نیز به دو قسمت تبدیل می شود. از کار افتادن کارت در استفاده از ATM و POS و دیگری از کار افتادن کارت برای خرید اینترنتی. این دو مفهوم از هم مجزا هستند و می توانند به صورت مجزا از یکدیگر اتفاق بیفتند. توجه به این نکته ضروری است که اگر کاربر PIN خود را از طریق ATM عوض کند، فقط PIN اول عوض می شود و PIN خرید اینترنتی همان قبلی باقی خواهد ماند. برای تغییر PIN خرید اینترنتی باید به شعبه مراجعه کرد و یا در ATM بانک صادر کننده نسبت به تغییر آن اقدام نمود.

### **آدرس سایت بانک برای انتقال وجه و صدور رسید دیجیتالی**

آدرسی که فروشنده باید فرم را به آن ارسال کند باید از بانک طرف قرار داد دریافت شود.

### **آدرس Web Service Provider**

برای فراخواندن web method نیز باید نشانی آن را از بانک طرف قرارداد دریافت کنید.