

راهنمای راه اندازی درگاه پرداخت اینترنتی

تغییرات نسخ این سند:

شماره نسخه	تاریخ انتشار	تغییرات نسخه
2.0	1392/08/25	نسخه اولیه سند پرداخت اینترنتی فن آوا کارت
2.1	1393/03/01	شیوه ارجاع مشتری به درگاه پرداخت بصورت شفاف تری توضیح داده شده است.
2.2	1393/07/15	پشتیبانی از تقسیم وجوه به درگاه پرداخت اینترنتی افزوده شد.
3.0	1394/03/07	شرح بیشتری در خصوص خطاهای درگاه پرداخت به مستند اضافه شد.
3.1	1395/11/12	تغییر URL های درگاه و وب سرویس و نحوه ی اتصال به درگاه جدید
3.2	1395/12/15	تغییر URL (Fanava) به (FCP)

فهرست

3..... محدود

3..... پیش نیازها

5..... نیازمندیهای امنیتی و بانکی

5..... نیازمندیهای امنیتی

5..... نیازمندیهای بانکی

5..... تراکنش خرید اینترنتی به روش ساده

9..... تراکنش پرداخت قبض به روش ساده

1..... برگشت تراکنش خرید

3..... شرح متد های بکار رفته در سرویس دبی درگاه پرداخت اینترنتی

3..... سرویس MerchantLogin

4..... متد ورود (login)

4..... متد خروج (logout)

5..... متد درخواست تسویه فروشنده (merchantSettlementRequest)

6..... متد تایید تراکنش (verifyTransaction)

8..... متد تایید تراکنش امن (secureVerifyTransaction)

10..... متد برگشت تراکنش (reverseTransaction)

11..... ضمیمه یک، که خطای های برگشتی سرویس ها

برای راحتی کاربران و پرهیز از مشکلات ناشی از برخی راه حل های دشوار که پیاده سازی درگاه پرداخت اینترنتی را برای پذیرندگان اینترنتی دشوار می سازد، شرکت فن آواکارت از فناوری وب سرویس **(SOAP) Web Services** برای برقراری خدمات پرداخت استفاده نموده است. با بکارگیری این فناوری، امکان پیاده سازی سرویس های عملیات پرداخت برای پذیرندگان اینترنتی با سهولت زیاد فراهم می شود و ضمن آن، استقلال این فناوری از بستر پیاده سازی و استفاده از استاندارد **HTTP** برای تبادل ارتباطات و بهره گیری از گواهینامه امنیتی **SSL** معتبر، امکان مجتمع سازی امن این خدمات را روی بستر استاندارد امکان پذیر می کند. در سامانه درگاه پرداخت اینترنتی فن آواکارت، از سوی فن آواکارت متد هایی در اختیار پذیرنده قرار می گیرد، که در سایت خود از آنها استفاده نماید.

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.

محدوده

هدف از ارائه این سند، معرفی سرویس های بهم پیوسته ایست که مجموعاً خدمات درگاه پرداخت اینترنتی فن آواکارت از منظر پذیرنده اینترنتی را شکل می دهند. محدوده این سند، شرح کلی عملیات لازم برای انجام عملیات پرداخت درخواستی از سوی پذیرنده روی درگاه پرداخت اینترنتی فن آواکارت است و در کنار آن، تمامی متدهای مربوط به این عملیات شرح داده شده اند.

پیش نیازها

برای استفاده از این سرویس و اتصال به سرور پرداخت، لازم است در ابتدا **IP** وب سرور پذیرنده یا هر سرور دیگری که برای استفاده توسط وب سرور، عملیات مالی ساماندهی می کند و آدرس صفحه **Domain** ارجاع دهنده از وب سایت پذیرنده به درگاه پرداخت اینترنتی را به همراه درخواست استفاده از سرویس به آدرس ایمیل زیر ارسال فرمائید:

IPG@fanavacard.ir

چنانچه این درخواست به همراه اطلاعات سرور ارسال نشود، فن آواکارت اجازه دسترسی به درگاه پرداخت اینترنتی را نخواهد داد. در صورتی که درخواست مزبور از سوی فن آواکارت مورد موافقت قرار گیرد، پذیرنده اطلاعات ذیل را از طریق ایمیل اعلام شده دریافت خواهد نمود.

شماره پذیرنده : **Merchant ID**

نام کاربری : **Username**

رمز عبور : **Password**

لازم است اطمینان حاصل کنید که پورت های 443 و 80 روی سرور پذیرنده باز هستند و می توانند روی این دو پورت اطلاعات را ارسال و دریافت نمایند.

ترمیولوژی

خریدار: موجودیتی که تقاضای خرید سرویس یا کالا را دارد.

فروشنده: موجودیتی که سرویس یا کالا را در اختیار خریدار قرار میدهد.

پذیرنده (Acquirer): موجودیتی واسط میان شبکه عمومی (شامل خریدار و فروشنده) و شبکه بین بانکی (شامل بانکهای نگهدارنده سپرده های فروشنده و خریدار).

بانکهای عضو شتاب : موجودیت هایی که سپرده های خریدار و فروشنده را نزد خود نگه میدارند.

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 میشود. (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 24 حرف باشد و بانک به عنوان رسید پس از انجام یک انتقال به خریدار (و در نهایت به فروشنده) ارائه میدهد.

شماره رزرو (Reservation Number): کدی که فروشنده برای هر تراکنش خریدار در نظر میگیرد و خریدار میتواند توسط آن کد، خرید خود را پیگیری کند. در واقع مشخصه تراکنش است در سمت فروشنده. این کد می تواند تا 24 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 ترمینولوژیهای بانکی بدان حساب می گویند.

تراکنش: یک عملیات مالی، که در این نوشته مصداق آن یک خرید میباشد.

نیازمندیهای امنیتی وبانکی

نیازمندیهای امنیتی

سایت بانک دارای گواهینامه های معتبر می باشد. پس ارتباط خریدار با بانک و فروشنده با بانک می تواند 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 این است 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 نمی کند و تمامی این اطلاعات توسط خریدار در سایت بانک وارد می شود.

برای اطمینان از هویت فروشنده در هنگام درخواست تأیید و یا برگشت خوردن سند، کلمه عبوری به فروشنده داده می شود 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 می شود.

نیازمندیهای بانکی

فروشنده باید سپرده ای 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 داد که در هر تراکنش فروشنده خود را با ارائه این شماره (که اختصاراً به آن کد فروشنده میگوییم) به بانک معرفی می کند.

تراکنش خرید اینترنتی به روش ساده

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

سایت صدور رسید دیجیتالی فن آوا کارت :

<https://fcp.shaparak.ir/ipgw//payment/simple/>

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 12 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 50 حرف داشته باشد.
MID	بله	کد فروشنده که بانک به فروشنده اختصاص می دهد.	می تواند ترکیبی از عدد و - باشد.
goodReferenceId	خیر	شناسه خرید دارنده کارت	ترکیبی است از عدد و حروف که می تواند حداکثر 30 حرف داشته باشد.
merchantData	خیر	شناسه خرید فروشنده	ترکیبی است از عدد و حروف که می تواند حداکثر 50 حرف داشته 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	بله	زبان نشان داده شده به کاربر در صفحه خرید است.	می تواند مقادیر fa برای پارسی و en برای انگلیسی داشته باشد.

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

- شماره کارت (PAN)
- کلمه عبور (PIN)
- تاریخ انقضای کارت (ExpDate)
- شماره CVV2

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

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.

شرح	Not approve	Approve	نوع	نام
وضعیت انجام تراکنش است. یا "ok" و یا "Canceled By User" می تواند باشد.	✓	✓	String	State
کدی است که فروشنده به کالای خود اختصاص داده و باید برای بانک ارسال کند.	✓	✓	String	resNum
شماره پیگیری تراکنش انجام شده است.		✓	String	refNum
		✓	String	CustomerRefNum
کد فروشنده که بانک به فروشنده اختصاص می دهد.	✓	✓	String	MID
زبان نشان داده شده به کاربر در صفحه خرید است.	✓	✓	String	Language
Hash شماره کارت خریدار		✓	String	CardPanHash
آدرس صفحه دریافت اطلاعات فروشنده است که بعد از خرید انتظار اطلاعات از بانک را دارد.		✓	String	redirectURL
شناسه خرید دارنده کارت	✓	✓	String	goodReferenceld
Mask شماره کارت خریدار	✓	✓		CardPanMask
شناسه خرید فروشنده	✓	✓	String	merchantData

فروشنده میتواند بر اساس وضعیت تراکنش، موفقیت آمیز بودن تراکنش را تشخیص دهد. 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 ارائه 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 اعمال خریدار نباید بتواند مبلغ نادرستی را انتقال دهد) با مقایسه مبلغ دریافت شده و مبلغ فاکتوری که فروشنده خود در اختیار دارد 9 حالت زیر ممکن است به وجود آید:

- اگر این دو مبلغ برابر باشند، فروشنده می تواند سرویس خود را ارائه 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:

1. تاکید میشود که یکتا بودن 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 مصرف شدگی آن تغییر کند. حسن این روش در این است که اگر فروشنده ای 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 می تواند 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.

2. 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 مجددا 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.

9. در صورتی که تراکنش به هر دلیلی در مدت زمان مشخصی (این زمان در بانک تعیین می شود و فروشنده باید از بانک درخواست کند این زمان به او اعلام شود و مقدار `Default 15` دقیقه می باشد.) از جانب فروشنده تایید نشد، بانک اقدام به برگشت زدن تراکنش خواهد کرد.

0. در این روش پرداخت، امکان اینکه یک رسید دیجیتالی در دو فروشنده ی مختلف استفاده شود، وجود ندارد.
2. مسئولیت جلوگیری از Double Spending بر عهده فروشنده است و در صورت ضعفی در پیاده سازی سایت فروشنده، ضرر آن متوجه خود اوست.
6. امنیت این بخش از سیستم به کمک SSL و ACL تامین شده است.
7. 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 می تواند میزان ریسک سپردههای خود را مدیریت نماید. همچنین فروشنده با معرفی الگوی فروش خود به بانک، حداکثر مبلغ فروش خود را می تواند معرفی نماید.

تراکنش پرداخت قبض به روش ساده

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 6 تا 13 حرف.
paymentId	بله	شناسه پرداخت قبض می باشد.	رشته ای از اعداد باید باشد به طول 6 تا 13 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	بله	زبان نشان داده شده به کاربر در صفحه خرید است.	می تواند مقادیر fa برای پارسی و en برای انگلیسی داشته باشد.
resNum	بله	کدی است که برای تراکنش	ترکیبی است از عدد و حروف که می تواند حداکثر 50 حرف داشته باشد.
MID	بله	کد فروشنده	می تواند ترکیبی از عدد و - 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.

شرح	Not Approve	Approve	نوع	نام
Canceled By و یا "ok" وضعیت انجام تراکنش است. یا " می تواند باشد. User	✓	✓	String	State
آدرسی که فروشنده به بانک ارسال کرده بود.		✓	String	redirectURL
کدی است که اگر ارسال شده بود بر گردانده می شود.	✓	✓	String	resNum
شماره پیگیری تراکنش انجام شده است.		✓	String	RefNum
		✓	String	CustomerRefNum
شناسه قبض دریافت شده از سایت فروشنده می باشد.	✓	✓	String	billId
شناسه پرداخت دریافت شده از سایت فروشنده می باشد.	✓	✓	String	paymentId
کد فروشنده دریافت شده از سایت فروشنده میباشد. بدون خط تیره	✓	✓	String	MID
زبان نشان داده شده به کاربر در صفحه خرید است.	✓	✓	String	language
فعلا خالی ارسال می شود		✓	String	CardPanHash

- نکته 1: در صورتی که تراکنش با موفقیت انجام شده باشد، State برابر با ok خواهد بود و فروشنده می تواند سرویس خود را به کاربر ارائه کند.
 - نکته 2: 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 3: مواردی که از نظر اجباری بودن خیر هستند اگر از سمت فروشنده مقداری POST نشده بود نباید خطا داده شود.
- نشانی ای که فروشنده باید این فرم را به آن submit کند:

<https://fcp.shaparak.ir/ipgw//payment/bill/>

- کلمه عبور (PIN)
- تاریخ انقضای کارت (ExpDate)
- شماره CVV2

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 میگیرد). 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

فروشنده میتواند بر اساس وضعیت تراکنش، موفقیت آمیز بودن تراکنش را تشخیص دهد. 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:

1. امنیت این بخش از سیستم به کمک SSL و ACL تامین شده است.
2. مدیریت ریسک این بخش با استفاده از سقفهای برداشت برای موجودیت خریدار و الگوهای فروش برای فروشنده صورت میگیرد. 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.
9. فروشنده باید قسمتی در سایت خود فراهم کند تا کاربر بتواند پرداخت قبض هایی که بابت آنها جوابی دریافت نکرده است را به فروشنده اطلاع رسانی کند. این حالت زمانی رخ می دهد که پس از پرداخت قبض 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 گزارش گیری

وب سرویس، پرداخت شدن قبض مورد نظر را تایید کند و پس از آن می تواند سرویس خود را به مشتری ارائه کند.

برگشت تراکنش خرید

فروشنده میتواند امکان لغو خرید را پیاده سازی کند. این امکان ممکن است به دو صورت مورد نیاز واقع شود:

- فروشنده لغو یک خرید را الزم 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:

(1) 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 برگشت خوردگی آن مطمئن شود که قابل برگشت جزئی یا کامل نخورده باشد (البته این چک برای صرفه جویی در ارتباطات شبکه ای 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 می نماید.

(2) 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 برگشت خوردگی 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 صرفه جویی در ارتباطات شبکه ای 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:

دوباره تاکید میشود که لزومی ندارد وضعیت برگشت خوردگی 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.

دستور برگشت الزاما 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 گونه ای است که خریدار میتواند فرایند برگشت را خود آغاز کند، باید نام کاربری و کلمه عبور خود را در پیاده سازی سایت خود، `hard code` نماید.

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

شرح متد های بکار رفته در سرویس دهبی درگاه پرداخت اینترنتی

آدرس وب سرویس پرداخت شرکت فن آواکارت:

<https://fcp.shaparak.ir/ref-payment/jax/merchantAuth?wsdl>

سرویس MerchantLogin

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 می کند و یک رشته به عنوان شناسه جلسه کاری کاربر که در سیستم SESSION_ID نامیده می شود برمی گرداند. 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 می تواند 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 پردازد. این متد یک پارامتر از نوع LoginRequest می گیرد 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 می تواند 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 می شود.

خطاهای رخ داده در این متد:

- **WsInvalidCredentialException**: در صورتی رخ می دهد که نام کاربری یا کلمه عبور فروشنده نادرست باشد.
- **WsBlockUserException**: در صورتی رخ می دهد 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 می شود – نمی تواند با سیستم کار کند ولی بعد از آن قفل فروشنده برداشته شده و می تواند 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.

ورودی 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 = "2b9e9449-f722-40ac-8c37-500d9e1c3e40";//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.

همچنین در **WsContext** می توان پارامتر زبان را با عنوان **"LANGUAGE"** قرار داد که مقدار آن می تواند **"en"** یا **"fa"** باشد و بهتر است به جای نوشتن **"LANGUAGE"** از **WsContext.LANGUAGE** استفاده شود. ارسال این پارامتر اختیاری است و در صورتی که ارسال نشود از زبان پیش فرض برنامه استفاده می شود. این متد خروجی ندارد.

توجه:

در صورت موجود نبودن متد "addData" ، باید یک new HashMap، new 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 = "2b9e9449-f722-40ac-8c37-500d9e1c3e40";//login response
WsContext context = new WsContext();
HashMap<String, String> data = new HashMap<String, String>();
data.put("SESSION_ID", loginMethodResult);
```

خطاهای رخ داده در این متد:

- **WsInvalidSessionException**: در صورتی رخ می دهد که مقدار SESSION_ID اشتباه باشد یا قبلا EXPIRE شده باشد.
- **WebServiceRuntimeException**: در صورت بروز خطاهای ناشناخته رخ می دهد.

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 = "2b9e9449-f722-40ac-8c37-500d9e1c3e40";//login response
WsContext context = new WsContext();
context.addData("SESSION_ID", loginMethodResult);
```


توجه:

در صورت موجود نبودن متد "addData" ، باید یک HashMap، new کرده 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:

این متد خروجی ندارد.

```
String loginMethodResult = "2b9e9449-f722-40ac-8c37-500d9e1c3e40";//login response
WsContext context = new WsContext();
HashMap<String, String> data = new HashMap<String, String>();
data.put("SESSION_ID", loginMethodResult);
```

خطاهای رخ داده در این متد:

- **WsInvalidSessionException**: در صورتی رخ می دهد که مقدار SESSION_ID اشتباه باشد یا قبلا EXPIRE شده باشد.
- **WsDuplicateTransactionException**: در صورتی رخ می دهد که درخواست تسویه قبلا ثبت شده باشد یا زمان تسویه همه ترمینال های فروشنده رسیده باشد.
- **WebServiceRuntimeException**: در صورت بروز خطاهای ناشناخته رخ می دهد.

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	<p>نوع خطای رخ داده برای تراکنشی که مقدار RefNum آن در بالا آمده 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"> TRANSACTION_NOT_FOUND_ERROR: این خطا هنگامی رخ می دهد که تراکنش اصلی پیدا نشود یا شماره پیگیری ارسال شده اشتباه باشد. TRANSACTION_IS_NOT_VERIFIABLE: این خطا برگشت خورده باشد. 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. VALIDATION_EXCEPTION: این خطا در صورتی رخ میدهد که مقادیر ورودی نامعتبر باشند.

<ul style="list-style-type: none"> • 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. 		
---	--	--

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.

مقدار شماره خرید که ارسال شده بود.	String	resNum
<p>نوع خطای رخ داده برای تراکنشی که مقدار RefNum آن در بالا آمده 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"> • TRANSACTION_NOT_FOUND_ERROR: این خطا هنگامی رخ می دهد که تراکنش اصلی پیدا نشود یا شماره پیگیری ارسال شده اشتباه باشد. • TRANSACTION_IS_NOT_VERIFIABLE: این خطا هنگامی رخ می دهد که تراکنش قبل از تایید شدن توسط سیستم برگشت خورده باشد. • 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. • VALIDATION_EXCEPTION: این خطا در صورتی رخ میدهد که مقادیر ورودی نامعتبر باشند. • 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. 	VerificationError	verificationError

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
<code>mainTransactionRefNum</code>	String	شماره پیگیری خرید انجام شده که فروشنده می خواهد آنرا برگشت بزند.
<code>reverseTransactionResNum</code>	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 50 حرف باشد.
<code>amount</code>	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 بخواند این سرویس را فراخوانی کند.
- **WsPaymentReverseException**: در صورتی که امکان برگشت ناقص وجود نداشته باشد و فروشنده بخواند 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.
- **WsTransactionNotFoundException**: در صورتی که رکوردی برای شماره پیگیری داده شده یافت نشود این خطا رخ میدهد.
- **WsDuplicateTransactionException**: این خطا در صورتی رخ میدهد که تراکنش دیگری قبلا با شماره پیگیری داده شده انجام شده باشد.
- **WsInvalidCredentialException**: در صورتی که Session فروشنده منقضی شده باشد این خطا رخ میدهد.
- **WebServiceException**: در صورتی که خطایی رخ داده باشد ولی در گروه خطاهای بالا ننگند.

ضمیمه یک، کدهای بازرگانی سرویس ها

erSucceed	سرویس با موفقیت اجراء شد.
erAAS_UseridOrPassIsRequired	کد کاربری و رمز الزامی هست
erAAS_InvalidUseridOrPass	کد کاربری یا رمز صحیح نمی باشد
erAAS_InvalidUserType	نوع کاربر صحیح نمی باشد
erAAS_UserExpired	کاربر منقضی شده است
erAAS_UserNotActive	کاربر غیر فعال هست
erAAS_UserTemporaryInactive	کاربر موقتا غیر فعال شده است
erAAS_UserSessionGenerateError	خطا در تولید شناسه لاگین
erAAS_UserPassMinLengthError	حداقل طول رمز رعایت نشده است
erAAS_UserPassMaxLengthError	حداکثر طول رمز رعایت نشده است
erAAS_InvalidUserCertificate	برای کاربر فایل سرتیفیکیت تعریف نشده است
erAAS_InvalidPasswordChars	کاراکترهای غیر مجاز در رمز
erAAS_InvalidSession	شناسه لاگین معتبر نمی باشد
erAAS_InvalidChannelId	کانال معتبر نمی باشد
erAAS_InvalidParam	پارامترها معتبر نمی باشد

erAAS_NotAllowedToService	کاربر مجوز سرویس را ندارد
erAAS_SessionIsExpired	شناسه لاگین معتبر نمی باشد
erAAS_InvalidData	داده ها معتبر نمی باشد
erAAS_InvalidSignature	امضاء دیتا درست نمی باشد
erAAS_InvalidToken	توکن معتبر نمی باشد
erAAS_InvalidSourceIp	آدرس آی پی معتبر نمی باشد
erMts_ParamIsNull	پارامترهای ورودی خالی می باشد
erMts_UnknownError	خطای ناشناخته
erMts_InvalidAmount	مبلغ معتبر نمی باشد
erMts_InvalidBillId	شناسه قبض معتبر نمی باشد
erMts_InvalidPayId	شناسه پرداخت معتبر نمی باشد
erMts_InvalidEmailAddLen	طول ایمیل معتبر نمی باشد
erMts_InvalidGoodsReferenceIdLen	طول شناسه خرید معتبر نمی باشد
erMts_InvalidMerchantGoodsReferenceIdLen	طول شناسه خرید پذیرنده معتبر نمی باشد
erMts_InvalidMobileNo	فرمت شماره موبایل معتبر نمی باشد
erMts_InvalidPorductId	طول یا فرمت کد محصول معتبر نمی باشد
erMts_InvalidRedirectUrl	طول یا فرمت آدرس صفحه رجوع معتبر نمی باشد
erMts_InvalidReferenceNum	طول یا فرمت شماره رفرنس معتبر نمی باشد
erMts_InvalidRequestParam	پارامترهای درخواست معتبر نمی باشد
erMts_InvalidReserveNum	طول یا فرمت شماره رزرو معتبر نمی باشد
erMts_InvalidSessionId	شناسه لاگین معتبر نمی باشد
erMts_InvalidSignature	طول یا فرمت امضاء دیتا معتبر نمی باشد
erMts_InvalidTerminal	کد ترمینال معتبر نمی باشد
erMts_InvalidToken	توکن معتبر نمی باشد
erMts_InvalidTransType	نوع تراکنش معتبر نمی باشد
erMts_InvalidUniqueId	کد یکتا معتبر نمی باشد
erMts_InvalidUseridOrPass	رمز یا کد کاربری معتبر نمی باشد
erMts_RepeatedBillId	پرداخت قبض تکراری می باشد
erMts_AASError	خطای سرور کنترل کاربران
erMts_SCMEError	خطای سرور مدیریت کانال
erScm_ParamIsNull	پارامترهای ارسالی خالی می باشند
erScm_ParamItemIsNull	آیتم های پارامترهای ارسالی خالی می باشد
erScm_InvalidAcceptor	پذیرنده معتبر نمی باشد
erScm_InvalidTerminal	ترمینال معتبر نمی باشد

erScm_NotBelongTerminalToAccept or	ترمینال متعلق به پذیرنده نمی باشد
erScm_InvalidAmount	مبلغ معتبر نمی باشد
erScm_InvalidLocalDate	زمان محلی معتبر نمی باشد
erScm_InvalidTraceNo	شماره پیگیری معتبر نمی باشد
erScm_InvalidTransType	نوع تراکنش معتبر نمی باشد
erScm_InvalidBillId	شناسه قبض معتبر نمی باشد
erScm_InvalidPayId	شناسه پرداخت معتبر نمی باشد
erScm_InvalidToken	توکن معتبر نمی باشد
erScm_InvalidVocherType	نوع وچر معتبر نمی باشد
erScm_InvalidReserveNum	فرمت کد رزرو معتبر نمی باشد
erScm_InvalidCardData	اطلاعات کارت معتبر نمی باشد
erScm_InvalidCardDataPan	فرمت شماره کارت معتبر نمی باشد
erScm_InvalidCardDataCvv2	فرمت سی وی وی کارت معتبر نمی باشد
erScm_InvalidCardDataPin2	فرمت رمز کارت معتبر نمی باشد
erScm_InvalidCardDataExpireData	فرمت تاریخ انقضاء کارت معتبر نمی باشد
erScm_InvalidSignature	امضاء صحیح نمی باشد
erScm_InvalidUniqueId	کد یکتای داده ها معتبر نمی باشد
erScm_InvalidRedirectUrl	فرمت آدرس بازنشانی صحیح نمی باشد
erScm_InvalidDiagKey	کد تراکنش معتبر نمی باشد
erScm_InvalidTransData	دیتای تراکنش معتبر نمی باشد
erScm_InvalidLanguge	زبان معتبر نمی باشد
erScm_NotBindUserIdToANYAccepto r	کاربری با هیچ پذیرنده ای مرتبط نشده است
erScm_BindUserIdToManyTerminal	کاربر با چندین پذیرنده مرتبط هست
erScm_TransactionDonebefore	تراکنش تکراری
erScm_InvalidReferenceNum	کد رفرنس معتبر نمی باشد
erScm_InvalidSessionId	کد لاگین معتبر نمی باشد
erScm_OrgTransNotApproved	تراکنش اصلی موفق نبوده است
erScm_OrgTransVerifiedBefore	تراکنش اصلی قبلا تایید شده است
erScm_AmountNotEqualWithOrgTran sAmount	مبلغ ارسالی با مبلغ تراکنش برابر نمی باشد
erScm_ResNumNotEqualWithOrgTran sResNum	کد رزرو ارسالی با کد رزرو تراکنش معتبر نمی باشد
erScm_OrgTransReversedBefore	تراکنش قبلا ریورس شده است
erScm_InvalidTransKey	کد تراکنش معتبر نمی باشد
erScm_CompleteTransBefore	تراکنش قبلا تکمیل شده است

erScm_QueueStoreError	خطای صف جهت تعیین تکلیف
erScm_DecideTransNotFound	تراکنش تعیین تکلیف پیدا نشد
erScm_InvalidParam	پارامترهای ارسالی معتبر نمی باشد
erScm_NotApproveFromSwitch	پاسخ موفق از مرکز تراکنش دریافت نگردید
erScm_TimeoutFromSwitch	پاسخی از مرکز تراکنش دریافت نگردید
erScm_NotConnectSwitch	عدم برقراری ارتباط با مرکز تراکنش
erScm_ProcessException	خطای پردازش تراکنش
erScm_InvalidUserIdOrPass	کد یا رمز کاربر صحیح نمی باشد
erScm_TimeMismatch	عدم هماهنگی زمان تراکنش
erScm_WaitTimeForTransIsFinished	فرصت انجام تراکنش به پایان رسیده است
erScm_TransRepeatCountIsLimited	تعداد مجاز تکرار تراکنش بیش از حد مجاز می باشد
erScm_TransactionCompletebefore	تراکنش قبلا تعیین تکلیف شده است
erScm_AcceptorHaveManyTerminal	پذیرنده دارای چندین ترمینال اینترنتی می باشد
erScm_InvalidAcceptorType	پذیرنده از نوع اینترنتی نمی باشد
erScm_InquiryNotApprove	جواب استعلام موفقیت آمیز نبود
erScm_InvalidProductId	فرمت کد محصول معتبر نمی باشد
erScm_InvalidRequestParam	داده های درخواستی معتبر نمی باشد
erScm_InvalidMobileNo	فرمت کد موبایل معتبر نمی باشد
erScm_BillPaymentIsRepeated	پرداخت قبض تکراری می باشد
erScm_ReverseNotAllowed	ریوس تراکنش مجاز نمی باشد
erScm_TransactionNotPermitted	تراکنش معتبر نمی باشد
erScm_InvalidParamLength	طول پارامتر معتبر نمی باشد
erScm_InvalidGoodsReferenceIdLen	فرمت شناسه خرید معتبر نمی باشد
erScm_InvalidMerchantGoodsReferenceIdLen	فرمت شناسه خرید پذیرنده معتبر نمی باشد
erScm_InvalidEmailAddLen	فرمت ایمیل معتبر نمی باشد
erScm_InvalidSourceIP	آی پی مجاز نمی باشد
erScm_InvalidBillList	لیست قبوض خالی هست
erScm_RepeatedBillId	شناسه قبض تکراری می باشد
erScm_AcceptorWebInfoNotDefined	پارامترهای اینترنتی پذیرنده تعریف نشده است
erScm_Repeat	تکرار باید انجام گیرد
erScm_InvalidChannelId	کد کانال معتبر نمی باشد
erScm_InvalidUserId	کد کاربر معتبر نمی باشد

erScm_InvalidAcceptorStatus	پذیرنده فعال نمی باشد
erScm_InvalidTerminalStatus	ترمینال فعال نمی باشد
erScm_UseridOrPassIsRequired	کد کاربر و رمز اجباریست
erScm_InvalidUserType	نوع کاربر معتبر نمی باشد
erScm_UserExpired	کاربر منقضی شده است
erScm_UserNotActive	کاربر غیر فعال هست
erScm_UserTemporaryInactive	کاربر موقتا غیر فعال هست
erScm_UserSessionGenerateError	خطا در صدور مجوز لاگین
erScm_UserPassMinLengthError	خطا در حداقل طول رمز
erScm_UserPassMaxLengthError	خطا در حداکثر طول رمز
erScm_InvalidPasswordChars	کاراکتر های نامجاز در رمز
erScm_InvalidSession	مجوز لاگین معتبر نمی باشد
erScm_NotAllowedToService	کاربر مجوز انجام سرویس را ندارد
erScm_SessionIsExpired	مجوز لاگین منقضی شده است
erScm_InvalidData	دیتا معتبر نمی باشد
erScm_InvalidUser	کاربر معتبر نمی باشد
erScm_InvalidUserCertificatePass	خطای رمز سرتیفیکیت
erScm_InvalidUserSourceIp	آی پی مجاز نمی باشد
erScm_AASError	خطای سرور
erScm_SCMEError	خطای سرور
erScm_UnknownError	خطای ناشناخته
erScm_MerchantNotHaveOrManySomeProduct	پذیرنده یا دارای این محصول نمی باشد یا چند محصول مشابه دارد
erScm_VerifyUserNotEqualWithTransUser	کد کاربر تایید کننده با کد کاربر درخواست کننده تراکنش متفاوت هست
erScm_TransAmountNotEqualWithSrcAmnt	مبلغ تراکنش با مبلغ سرویس فروش وچر متفاوت هست
erScm_DBException	خطای ناشناخته بانک اطلاعاتی