

مستندات راه اندازی و پیاده سازی درگاه پرداخت اینترنتی

ویرایش ۳,۵

شماره نسخه	تاریخ انتشار	تغییرات نسخه	ویرایشگر
۲,۰	۱۳۹۲/۰۸/۲۵	نسخه اولیه سند پرداخت اینترنتی فن آوا کارت	معصومه فرخی
۲,۱	۱۳۹۳/۰۳/۰۱	شیوه ارجاع مشتری به درگاه پرداخت به صورت شفاف تری توضیح داده شده است.	معصومه فرخی
۲,۲	۱۳۹۳/۰۷/۱۵	پشتیبانی از تقسیم وجوه به درگاه پرداخت اینترنتی افزوده شد.	معصومه فرخی
۳,۰	۱۳۹۴/۰۳/۰۷	شرح بیشتری در خصوص خطاهای درگاه پرداخت به مستند اضافه شد.	معصومه فرخی
۳,۱	۱۳۹۵/۱۱/۱۲	تغییر URL درگاه و وبسرویس و نحوه اتصال به درگاه جدید	معصومه فرخی
۳,۲	۱۳۹۵/۱۲/۱۵	تغییر URL (Fanava) به (FCP)	معصومه فرخی
۳,۳	۱۳۹۶/۰۳/۰۳	اضافه شدن متد گزارش گیری تراکنش	معصومه فرخی
۳,۴	۱۳۹۶/۰۴/۱۴	ویرایش مجدد	آرمان مهاجر میلانی
۳,۵	۱۳۹۷/۰۴/۰۳	اصلاح نام متد گزارش گیری و پارامترهای ارسالی	آرمان مهاجر میلانی

فهرست

۳	محدوده.....
۳	پیش نیازها.....
۵	نیازمندی های امنیتی وبانگی.....
۵	نیازمندی های امنیتی.....
۵	نیازمندی های بانگی.....
۵	تراکنش خرید اینترنتی به روش ساده.....
۹	تراکنش پرداخت قبض به روش ساده.....
۱۱	نکات:.....
۱۲	برگشت تراکنش خرید.....
۱۳	شرح متدهای کارزنده در سرویس دی درگاه پرداخت اینترنتی.....
۱۳	آدرس وب سرویس پرداخت شرکت فن آواکارت:.....
۱۳	MerchantLogin سرویس.....
۱۴	متد خروج (logout).....
۱۵	متد درخواست تسویه فروشنده (merchantSettlementRequest).....
۱۶	متد تایید تراکنش (verifyTransaction).....
۱۷	متد تایید تراکنش امن (secureVerifyTransaction).....
۱۹	متد برگشت تراکنش (reverseTransaction).....
۲۰	سرویس getTransReportResult
۲۲	ضمیمه یک، کدهای خطای برگشتی سرویس ها.....

برای راحتی کاربران و پرهیز از مشکلات ناشی از برخی راه‌حل‌های دشوار که پیاده‌سازی درگاه پرداخت اینترنتی را برای پذیرندگان اینترنتی دشوار می‌سازد، شرکت فن‌آوا کارت از فناوری وب‌سرویس (SOAP) Web Services برای برقراری خدمات پرداخت استفاده نموده است. با به‌کارگیری این فناوری، امکان پیاده‌سازی سرویس‌های عملیات پرداخت برای پذیرندگان اینترنتی با سهولت زیاد فراهم می‌شود و ضمن آن، استقلال این فناوری از بستر پیاده‌سازی و استفاده از استاندارد HTTP برای تبادل ارتباطات و بهره‌گیری از گواهینامه امنیتی SSL معتبر، امکان مجتمع‌سازی امن این خدمات را روی بستر استاندارد امکان‌پذیر می‌کند. در سامانه درگاه پرداخت اینترنتی فن‌آوا کارت، از سوی فن‌آوا کارت متدهایی در اختیار پذیرنده قرار می‌گیرد، که در سایت خود از آن‌ها استفاده نماید.

این نوشته قدم‌های لازم برای ایجاد بستر پرداخت با استفاده از رسید دیجیتالی را در سمت فروشنده برمی‌شمارد. ساختار این نوشته به‌صورت ذیل است.

محدوده

هدف از ارائه این سند، معرفی سرویس‌های به‌هم‌پیوسته ایست که مجموعاً خدمات درگاه پرداخت اینترنتی فن‌آوا کارت از منظر پذیرنده اینترنتی را شکل می‌دهند. محدوده این سند، شرح کلی عملیات لازم برای انجام عملیات پرداخت درخواستی از سوی پذیرنده روی درگاه پرداخت اینترنتی فن‌آوا کارت است و در کنار آن، تمامی متدهای مربوط به این عملیات شرح داده شده‌اند.

پیش‌نیازها

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 می‌کند و آدرس صفحه Domain ارجاع دهنده از وبسایت 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، فن‌آوا کارت اجازه دسترسی به درگاه پرداخت اینترنتی را نخواهد داد. در صورتی که درخواست مزبور از سوی فن‌آوا کارت 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 پورت‌های ۴۴۳ و ۸۰ روی سرور پذیرنده باز هستند و می‌توانند روی این دو پورت اطلاعات را ارسال و دریافت نمایند.

تعاریف

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

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

پذیرنده (Acquirer): موجودیتی واسط میان شبکه عمومی (شامل خریدار و فروشنده) و شبکه بین بانکی (شامل بانک های نگهدارنده 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: سراچه متعلق به بانک است که در آن خریدار شماره کارت و رمز آن را وارد می نماید و انتقال مبلغ خرید به سپرده فروشنده را تأیید می نماید. در صورت موفقیت آمیز بودن انتقال، یک رسید دیجیتالی برای آن انتقال صادر می شود. (درواقع این کار می تواند به عنوان بخشی از وظایف ماشین پذیرنده در نظر گرفته شود) همچنین در این نوشته ترم مای زیر نیز به کار برده خواهند شد:

رسید دیجیتالی (Reference Number): یک سلسله ای از کاراکترها که می تواند تا ۲۴ 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. درواقع مشخصه تراکنش است در سمت فروشنده. این کد می تواند تا ۲۴ 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 می گویند.

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

نیازمندی های امنیتی وبانکی

نیازمندی های امنیتی

سایت بانک دارای گواهینامه های معتبر است. 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 این گونه 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.

نیازمندی های بانکی

فروشنده باید سپرده ای را نزدیکی از بانک های 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 میگوئیم) 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 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.
goodReferenceId	خیر	شناسه خرید دارنده کارت	ترکیبی است از عدد و حروف که می تواند حداکثر ۳۰ حرف داشته باشد.
merchantData	خیر	شناسه خرید فروشنده	ترکیبی است از عدد و حروف که می تواند حداکثر ۵۰ 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.
Email	خیر	آدرس پست الکترونیک دارنده کارت است.	فرمت آدرس ایمیل است.
mobileNo	خیر	شماره تلفن همراه دارنده کارت است.	مقادیر عددی با ۱۱ رقم می 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 CVV۲

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.

نام	نوع	Approve	Not approve	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.
TraceNo	String	✓		شماره پیگیری تراکنش انجام شده است.
CustomerRefNum	String	✓		شماره پیگیری مشتری
MID	String	✓	✓	کد فروشنده که بانک به فروشنده اختصاص می دهد.
Language	String	✓	✓	زبان نشان داده شده به کاربر در صفحه خرید است.
CardHashPan	String	✓		Hash شماره کارت خریدار
redirectURL	String	✓		آدرس صفحه دریافت اطلاعات فروشنده است که بعد از خرید انتظار اطلاعات از بانک را دارد.
goodReferenceId	String	✓	✓	شناسه خرید دارنده کارت
CardMaskPan		✓	✓	Mask شماره کارت خریدار
merchantData	String	✓	✓	شناسه خرید فروشنده
transactionAmount		✓	✓	مبلغ تراکنش

فروشنده می تواند 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 تأیید مقدار منتقل شده و همچنین عدم برگشت به صورت سامانه ای مبلغ واریز شده، فروشنده باید توسط وب سرویس ای که در اختیار او قرار داده شده تراکنش خرید را تأیید کند. برای تأیید خرید فروشنده باید مقدار RefNum دریافت شده را ابتدا در پایگاه داده خود ذخیره کند و سپس متد تأیید وب سرویس را صدا بزند. شرح متدهای وب سرویس و نحوه اجرای آنها در ضمیمه ب آورده شده است.

مکزیبیدم: پس از اجرای متد تأیید، 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 حتماً 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 برگ خرید که فروشنده خود در اختیار دارد ۹ 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 برگ خرید فروشنده باشد، فروشنده می تواند پس از اعلام به خریدار، کل سند را برگشت بزند.
- اگر مبلغ پرداختی بیشتر از مبلغ برگ خرید نزد فروشنده باشد، فروشنده می تواند پس از اعلام به خریدار، مابه التفاوت سندها را برگشت بزند.

نکات:

۱. تأکید می شود که یکتا بودن رسید دیجیتالی در سمت فروشنده تعیین و نگهداری می شود و نه در سمت بانک. بانک تنها اعتبار و مبلغ برگشت نخورده رسید دیجیتالی را گزارش می دهد. بدین ترتیب بانک می تواند 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.

۲. در صورتی که جواب تابع تأیید 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 می شود و فروشنده باید از بانک



درخواست کند این زمان به او اعلام شود و مقدار Default (۱۵ دقیقه است). از جانب فروشنده تأیید 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. مسئولیت جلوگیری از Double Spending بر عهده فروشنده است و در صورت ضعفی در پیاده سازی سایت فروشنده، ضرر آن متوجه خود اوست.
۶. امنیت این بخش از سیستم به کمک SSL و ACL تأمین شده است.
۷. مدیریت ریسک این بخش با استفاده از سقف های 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.

تراکنش پرداخت قبض بروش 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 است.	رشته ای از اعداد باید باشد به طول ۶ تا ۱۳ حرف.
paymentId	بله	شناسه پرداخت قبض است.	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	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 ۵۰ 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 User" و یا "ok" وضعیت انجام تراکنش است. یا می تواند باشد.	✓	✓	String	State
آدرسی که فروشنده به بانک ارسال کرده بود.		✓	String	redirectURL
کدی است که اگر ارسال شده بود برگردانده می شود.	✓	✓	String	resNum
شماره پیگیری تراکنش انجام شده است.		✓	String	RefNum
		✓	String	CustomerRefNum
شناسه قبض دریافت شده از سایت فروشنده است.	✓	✓	String	billId
شناسه پرداخت دریافت شده از سایت فروشنده است.	✓	✓	String	paymentId
کد فروشنده دریافت شده از سایت فروشنده است. بدون خط تیره	✓	✓	String	MID
زبان نشان داده شده به کاربر در صفحه خرید است.	✓	✓	String	language
شماره کارت به صورت Hash شده		✓	String	CardHashPan

• نکته ۱: 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 ۲: 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 ۳: مواردی که از نظر اجباری بودن خیر هستند اگر از سمت فروشنده مقدری 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 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

فروشنده می تواند 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 گزارش گیری 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 تأمین شده است.
۲. مدیریت ریسک این بخش با استفاده از سقف های 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 مبلغ فروش خود را می تواند معرفی نماید.
۹. 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 تأیید کند و پس از آن می تواند سرویس خود را به مشتری ارائه کند.

برگشت تراکنش خرید

فروشنده می تواند امکان لغو خرید را پیاده سازی 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 می تواند 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 می شود 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 می گیرد). وی سپس متد برگشت وب سرویس را با پارامترهای ذیل فرا خواهد خواند. مقدار برگشتی این متد کدی است که نتیجه برگشت را نشان می دهد. فروشنده سپس پایگاه داده خود را به روز می نماید.

(۲) برگشت ناقص (Partial Reverse): فروشنده بخشی از مبلغ خرید را برگشت میزند. برای این کار فروشنده بر اساس Business Rule خود مبلغ برگشتی را محاسبه می کند و با استفاده از پایگاه داده خود، رسید دیجیتالی مربوط به Reservation Number مورد نظر خود را استخراج می نماید. سپس فروشنده باید وضعیت برگشت خوردگی رسید دیجیتالی را چک کند و تنها در صورتی آن را برای برگشت جزئی استفاده نماید که یا پیش تر تأیید شده باشد و یا مبلغ برگشتی به اضافه مبالغ برگشت های 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 می گیرد. در این صورت فروشنده می تواند 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 می دهد. 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 برگشت های جزئی بعدی را مدیریت نماید. ۱

- دوباره تأکید 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 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.
امنیت این بخش از سیستم به کمک `SSL`، `ACL` و `Merchant Username and Password` تأمین شده است.
- شایان ذکر است مدت برگشت خرید تنها در صورت عدم تأیید تراکنش قابل استفاده بوده و اگر تراکنش تأیید شده باشد دیگر امکان برگشت مبلغ وجود نخواهد داشت.

شرح متد های بکار رفته در سرویس دبی درگاه پرداخت اینترنتی

آدرس وب سرویس پرداخت شرکت فن آوا کارت:

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

سرویس Login

متدی است برای ورود فروشنده به سیستم که باید قبل از صدادن دیگر متدها مورداستفاده قرار گیرد. این متد نام کاربری و کلمه عبور فروشنده را دریافت می‌کند و یک رشته به‌عنوان شناسه جلسه کاری کاربر که در سیستم `SESSION_ID` نامیده می‌شود برمی‌گرداند؛ که این رشته باید در دیگر سرویس‌ها به‌عنوان پارامتر ورودی ارسال شود.

نکات:

- صدادن این متد قبل از هر بار صدادن دیگر سرویس‌ها 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 صدادن دیگر سرویس‌ها خطای `WsClientAddressException` دریافت شد، فروشنده باید ابتدا یک‌بار دیگر متد لاگین را فراخوانی کند تا `SESSION_ID` جدید بگیرد و سپس به فراخوانی دیگر متدها بپردازد.

این متد یک پارامتر از نوع `LoginRequest` می‌گیرد که مقادیر آن در زیر تشریح شده‌اند:

نام	نوع	شرح
username	String	نام کاربری فروشنده است که از بانک دریافت کرده است.

<p>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.</p>	<p>String</p>	<p>password</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------	-----------------

مقدار خروجی این متد رشته ای است که باید فروشنده آن را 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.

مخرج (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);
```

عبارت WsContext.SESSION_ID استفاده شود.

هم چنین در WsContext می توان پارامتر زبان را با عنوان "LANGUAGE" قرارداد 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 `HashMap`، `new` کرده و `SESSION_ID` 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 = "2b9e9449-f722-40ac-8c37-500d9e1c3e40";//login response  
WsContext context = new WsContext();  
HashMap<String, String> data = new HashMap<String, String>();  
data.put("SESSION_ID", loginMethodResult);
```

شده باشد.

• **WebServiceRuntimeException**: در صورت بروز خطاهای ناشناخته رخ می دهد.

مدیریت درخواست تسویه فروشنده (merchantSettlementRequest)

متدی است برای تسویه پایانه های فروشنده. این متد زمان تسویه پایانه 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 `SESSION_ID` 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:

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

خطاهای رخ داده در این متد:

- **WsInvalidSessionException**: در صورتی رخ می‌دهد که مقدار SESSION_ID اشتباه باشد یا قبلاً EXPIRE شده باشد.
- **WsDuplicateTransactionException**: در صورتی رخ می‌دهد که درخواست تسویه قبلاً ثبت شده باشد یا زمان تسویه همه پایانه‌های فروشنده رسیده باشد.
- **WebServiceRuntimeException**: در صورت بروز خطاهای ناشناخته رخ می‌دهد.

متد تایید تراکنش (verifyTransaction)

متدی است برای تأیید تراکنش تا فروشنده از مقدار خرید انجام شده باخبر شود و همچنین باعث شود تراکنش به صورت سامانه‌ای برگشت نخورد و مقدار خرید از حساب فروشنده به کارت مشتری واریز نشود.

ورودی این متد دو پارامتر است که اولی از نوع WsContext است و دومی از نوع VerifyRequest.

نحوه پر کردن پارامتر اول مانند متد خروج است. پارامتر دوم یک لیست از نوع String دریافت می‌کند که مقدار شماره پیگیری‌هایی (RefNum) هستند که فروشنده می‌خواهد آن را تأیید کند. اگر بعد از هر خرید یکبار این متد فراخوانی شود، این لیست حاوی یک رکورد بیشتر نیست ولی فروشنده در زمان‌هایی که خریدهای بسیار زیادی در واحد زمان دارد می‌تواند 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 است. مقادیر VerifyResponseResult در جدول زیر تشریح شده‌اند:

نام	نوع	شرح
amount	BigDecimal	مقدار خرید انجام شده که تأیید 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 به صورت دسته‌ای است و توسط این متد می‌توان چند شماره پیگیری را برای تأیید 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 می‌دهد که تراکنش اصلی پیدا نشود یا شماره پیگیری ارسال شده اشتباه باشد.
- TRANSACTION_IS_NOT_VERIFIABLE: این خطا هنگامی رخ می‌دهد که تراکنش قبل از تأیید 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.

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 تأیید تراکنش امن (secureVerifyTransaction)

متدی است برای تأیید تراکنش تا فروشنده از مقدار خرید انجام شده باخبر شود و همچنین باعث شود تراکنش به صورت سامانه‌ای برگشت نخورد و مقدار خرید از حساب فروشنده به کارت مشتری واریز نشود. در تأیید تراکنش امن به غیر از شماره پیگیری، شماره خرید نیز دریافت می‌شود.

ورودی این متد دو پارامتر است که اولی از نوع WsContext است و دومی از نوع SecureVerifyRequest.

نحوه پر کردن پارامتر اول مانند متد خروج است. پارامتر دوم یک لیست از نوع SecureVerifyInfo دریافت می‌کند که این نوع شامل دو مقدار شماره پیگیری (RefNum) و شماره خرید (ResNum) است که فروشنده می‌خواهد آن را تأیید کند. اگر بعد از هر خرید یک بار این متد فراخوانی شود، این لیست حاوی یک رکورد بیشتر نیست ولی فروشنده در زمان‌هایی که خریدهای بسیار زیادی در واحد زمان دارد می‌تواند 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 است. مقادیر SecureVerifyResponseResult در جدول زیر تشریح شده اند:

نام	نوع	شرح
amount	BigDecimal	مقدار خرید انجام شده که تأیید شده است.
refNum	String	مقدار شماره پیگیری که ارسال شده بود.
resNum	String	مقدار شماره خرید که ارسال شده بود.
verificationError	VerificationError	<p>نوع خطای رخ داده برای تراکنشی که مقدار RefNum آن در بالا آمده است. این متد به صورت دسته ای است و توسط این متد می توان چند شماره پیگیری را برای تأیید 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 خطای هنگامی رخ می دهد که تراکنش اصلی پیدا نشود یا شماره پیگیری ارسال شده اشتباه باشد. • TRANSACTION_IS_NOT_VERIFIABLE: این خطای هنگامی رخ می دهد که تراکنش قبل از تأیید شدن توسط سیستم برگشت خورده باشد. • INVALID_SESSION_EXCEPTION: این خطابه 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 می دهد که مقادیر ورودی نامعتبر باشند. • SYSTEM_ERROR: این خطابه 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.

مدرک 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 درخواست آن را 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 آن را 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 باشد.
amount	BigDecimal	مقداری که فروشنده می خواهد به کارت کاربر برگشت بزند. حتماً باید از مقدار خرید کوچک تر 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 به وجود آید و امکان واریز مبلغ به کارت خریدار وجود نداشته باشد و یا خطایی در زیرسیستم 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 می دهد.
- **WsDuplicateTransactionException**: این خطا در صورتی رخ می دهد که تراکنش دیگری قبلاً با شماره پیگیری داده شده انجام شده باشد.
- **WsInvalidCredentialException**: در صورتی که Session فروشنده منقضی شده باشد این خطا رخ می دهد.
- **WebServiceException**: در صورتی که خطایی رخ داده باشد ولی در گروه خطاهای بالا ننگند.

سرویس `getTransReportResult`

از این سرویس جهت گزارش گیری تراکنش های انجام شده استفاده می گردد.

این سرویس کد کاربری را گرفته و بعد از کنترل رمز آن لیست تراکنش های مرتبط با پذیرنده این کاربر را با افزودن فیلترهای ورودی گزارش می نماید.

نکته ۱: مانند سایر سرویس ها در صورتی که result برابر ER_SUCCEED باشد یعنی سرویس با موفقیت اجرا شده است در غیر این صورت شرح خطا.

نکته ۲: SessionID با توجه به تنظیمات مرکز می تواند در هر بار تراکنش نیز انجام گردد ولی بهتر است که یکبار لاگین فراخوانی و مقدار آن در جای امنی در سایت پذیرنده حفظ گردد که در هر بار فراخوانی انجام تراکنش برای دریافت توکن امنیتی، مجدداً لاگین انجام نگیرد.

REST EndPoint= "/getTransReportResult/"

پارامترهای ورودی

فیلترهای ورودی سرویس `getTransReportResult`

عنوان	نوع داده	کاربرد	ملاحظات
param	TransReportParam	اجباری	پارامتر ورودی

ساختار داده ای TransReportParam

عنوان	نوع داده	کاربرد	ملاحظات
WSContext	WSContext	اجباری	اطلاعات لاگین

minAmount	BigDecimal	اختیاری	حداقل مبلغ تراکنش
maxAmount	BigDecimal	اختیاری	حداکثر مبلغ تراکنش
switchTraceNo	String	اختیاری	شماره پیگیری
referenceId	String	اختیاری	شماره رفرنس
reserveNumber	String	اختیاری	شماره برگ خرید
timeMin	Date	اختیاری	حداقل مبلغ
timeMax	Date	اختیاری	حداکثر مبلغ
billId	String	اختیاری	شناسه قبض
payId	String	اختیاری	شناسه پرداخت
billAmount	BigDecimal	اختیاری	مبلغ قبض
offset	Long	اختیاری	افست شروع رکوردهای گزارش
length	Short	اختیاری	تعداد رکوردهای گزارش

پارامترهای خروجی

getTransReportResult پارامترهای خروجی سرویس

عنوان	نوع داده	ملاحظات
Result	جدول کد پاسخ	کد نتیجه عملیات
transactionList []	TransDetail	لیست تراکنشها

ساختار داده‌ای TransDetail

عنوان	نوع داده	ملاحظات
transType	Integer	نوع تراکنش



switchTraceNo	Integer	کد پیگیری سویچ
decideReqType	String	نوع درخواست
switchRefCode	String	کد رفرنس سویچ
switchAmount	BigDecimal	مبلغ
switchDiscountAmount	BigDecimal	مبلغ تخفیف
decideStatus	String	نوع تعیین تکلیف
decideResCode	Integer	پاسخ تعیین تکلیف
acaptorId	String	کد پذیرنده
reserveNumber	String	شماره برگ خرید
goodsReferenceId	String	شناسه خرید
billId	String	شناسه قبض
payId	String	شناسه پرداخت
billAmount	BigDecimal	مبلغ قبض
<u>billInquiryResponseCode</u>	Integer	کد پاسخ استعلام قبض
billInquiryErrorCode	Integer	شرح پاسخ استعلام قبض
billTypeDesc	String	شرح نوع قبض
merchantGoodsReferenceId	String	شناسه خرید پذیرنده

ضمیمه یک، کدهای برگشتی سرویس ها

erSucceed	سرویس با موفقیت اجرا شد.
erAAS_UseridOrPassIsRequired	کد کاربری و رمز الزامی هست

erAAS_InvalidUseridOrPass	کد کاربری یا رمز صحیح نیست
erAAS_InvalidUserType	نوع کاربر صحیح نیست
erAAS_UserExpired	کاربر منقضی شده است
erAAS_UserNotActive	کاربر غیرفعال هست
erAAS_UserTemporaryInactive	کاربر موقتاً غیرفعال 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 گواهینامه تعریف نشده است
erAAS_InvalidPasswordChars	کاراکترهای غیرمجاز در رمز
erAAS_InvalidSession	شناسه لاگین معتبر نیست
erAAS_InvalidChannelId	کانال معتبر نیست
erAAS_InvalidParam	پارامترها معتبر نیست
erAAS_NotAllowedToService	کاربر مجوز سرویس را ندارد
erAAS_SessionIsExpired	شناسه لاگین معتبر نیست
erAAS_InvalidData	داده‌ها معتبر نیست
erAAS_InvalidSignature	امضاء دیتا درست نیست
erAAS_InvalidToken	توکن معتبر نیست
erAAS_InvalidSourceIp	آدرس آی پی معتبر نیست
erMts_ParamIsNull	پارامترهای ورودی خالی است
erMts_UnknownError	خطای ناشناخته
erMts_InvalidAmount	مبلغ معتبر نیست
erMts_InvalidBillId	شناسه قبض معتبر نیست
erMts_InvalidPayId	شناسه پرداخت معتبر نیست
erMts_InvalidEmailAddLen	طول ایمیل معتبر نیست
erMts_InvalidGoodsReferenceIdLen	طول شناسه خرید معتبر نیست
erMts_InvalidMerchantGoodsReferenceIdLen	طول شناسه خرید پذیرنده معتبر نیست
erMts_InvalidMobileNo	فرمت شماره موبایل معتبر نیست
erMts_InvalidPorductId	طول یا فرمت کد محصول معتبر نیست
erMts_InvalidRedirectUrl	طول یا فرمت آدرس صفحه رجوع معتبر نیست
erMts_InvalidReferenceNum	طول یا فرمت شماره رفرنس معتبر نیست
erMts_InvalidRequestParam	پارامترهای درخواست معتبر نیست
erMts_InvalidReserveNum	طول یا فرمت شماره رزرو معتبر نیست
erMts_InvalidSessionId	شناسه لاگین معتبر نیست
erMts_InvalidSignature	طول یا فرمت امضاء دیتا معتبر نیست
erMts_InvalidTerminal	کد پایانه معتبر نیست
erMts_InvalidToken	توکن معتبر نیست
erMts_InvalidTransType	نوع تراکنش معتبر نیست

erMts_InvalidUniqueId	کد یکتا معتبر نیست
erMts_InvalidUserIdOrPass	رمز یا کد کاربری معتبر نیست
erMts_RepeatedBillId	پرداخت قبض تکراری است
erMts_AASError	خطای سرور کنترل کاربران
erMts_SCMEError	خطای سرور مدیریت کانال
erScm_ParamIsNull	پارامترهای ارسالی خالی می‌باشند
erScm_ParamItemIsNull	آیتم‌های پارامترهای ارسالی خالی است
erScm_InvalidAcceptor	پذیرنده معتبر نیست
erScm_InvalidTerminal	پایانه معتبر نیست
erScm_NotBelongTerminalToAcceptor	پایانه متعلق به پذیرنده نیست
erScm_InvalidAmount	مبلغ معتبر نیست
erScm_InvalidLocalDate	زمان محلی معتبر نیست
erScm_InvalidTraceNo	شماره پیگیری معتبر نیست
erScm_InvalidTransType	نوع تراکنش معتبر نیست
erScm_InvalidBillId	شناسه قبض معتبر نیست
erScm_InvalidPayId	شناسه پرداخت معتبر نیست
erScm_InvalidToken	توکن معتبر نیست
erScm_InvalidVocherType	نوع ووچر معتبر نیست
erScm_InvalidReserveNum	فرمت کد رزرو معتبر نیست
erScm_InvalidCardData	اطلاعات کارت معتبر نیست
erScm_InvalidCardDataPan	فرمت شماره کارت معتبر نیست
erScm_InvalidCardDataCvv۲	فرمت سی وی وی کارت معتبر نیست
erScm_InvalidCardDataPin۲	فرمت رمز کارت معتبر نیست
erScm_InvalidCardDataExpireData	فرمت تاریخ انقضاء کارت معتبر نیست
erScm_InvalidSignature	امضاء صحیح نیست
erScm_InvalidUniqueId	کد یکتای داده‌ها معتبر نیست
erScm_InvalidRedirectUrl	فرمت آدرس بازنشانی صحیح نیست
erScm_InvalidDiagKey	کد تراکنش معتبر نیست
erScm_InvalidTransData	دیتای تراکنش معتبر نیست
erScm_InvalidLanguge	زبان معتبر نیست
erScm_NotBindUserIdToAnyAcceptor	کاربری با هیچ پذیرنده‌ای مرتبط نشده است
erScm_BindUserIdToManyTerminal	کاربر با چندین پذیرنده مرتبط هست
erScm_TransactionDonebefore	تراکنش تکراری
erScm_InvalidReferenceNum	کد رفرنس معتبر نیست
erScm_InvalidSessionId	کد لاگین معتبر نیست
erScm_OrgTransNotApproved	تراکنش اصلی موفق نبوده است
erScm_OrgTransVerifiedBefore	است تأییدشده قبلاً اصلی تراکنش
erScm_AmountNotEqualWithOrgTransAmount	مبلغ ارسالی با مبلغ تراکنش برابر نیست
erScm_ResNumNotEqualWithOrgTransResNum	نیست معتبر تراکنش رزرو کد با ارسالی رزرو کد

erScm_OrgTransReversedBefore	است شده برگشت قبلاً تراکنش
erScm_InvalidTransKey	نیست معتبر تراکنش کد
erScm_CompleteTransBefore	تراکنش قبلاً تکمیل شده است
erScm_QueueStoreError	تکلیف تعیین جهت صف خطای
erScm_DecideTransNotFound	تراکنش تعیین تکلیف پیدا نشد
erScm_InvalidParam	نیست معتبر ارسالی پارامترهای
erScm_NotApproveFromSwitch	پاسخ موفق از مرکز تراکنش دریافت نگردید
erScm_TimeoutFromSwitch	نگردید دریافت تراکنش مرکز از پاسخی
erScm_NotConnectSwitch	عدم برقراری ارتباط با مرکز تراکنش
erScm_ProcessException	تراکنش پردازش خطای
erScm_InvalidUseridOrPass	کد یا رمز کاربر صحیح نیست
erScm_TimeMismatch	تراکنش زمان هماهنگی عدم
erScm_WaitTimeForTransIsFinished	فرصت انجام تراکنش به پایان رسیده است
erScm_TransRepeatCountIsLimited	است مجاز بیش از حد تراکنش تکرار مجاز تعداد
erScm_TransactionCompletebefore	تراکنش قبلاً تعیین تکلیف شده است
erScm_AcceptorHaveManyTerminal	است اینترنتی پایانه چندین دارای پذیرنده
erScm_InvalidAcceptorType	پذیرنده از نوع اینترنتی نیست
erScm_InquiryNotApprove	نیود موفقیت آمیز استعلام جواب
erScm_InvalidPorductId	فرمت کد محصول معتبر نیست
erScm_InvalidRequestParam	داده های درخواستی معتبر نیست
erScm_InvalidMobileNo	فرمت کد موبایل معتبر نیست
erScm_BillPaymentIsRepeated	پرداخت قبض تکراری است
erScm_ReverseNotAllowed	برگشت تراکنش مجاز نیست
erScm_TransactionNotPermitted	تراکنش معتبر نیست
erScm_InvalidParamLength	طول پارامتر معتبر نیست
erScm_InvalidGoodsReferenceIdLen	فرمت شناسه خرید معتبر نیست
erScm_InvalidMerchantGoodsReferenceIdLen	فرمت شناسه خرید پذیرنده معتبر نیست
erScm_InvalidEmailAddLen	فرمت ایمیل معتبر نیست
erScm_InvalidSourceIP	آی پی مجاز نیست
erScm_InvalidBillList	لیست قبوض خالی هست
erScm_RepeatedBillId	شناسه قبض تکراری است
erScm_AcceptorWebInfoNotDefined	پارامترهای اینترنتی پذیرنده تعریف نشده است
erScm_Repeat	تکرار باید انجام گیرد
erScm_InvalidChannelId	کد کانال معتبر نیست
erScm_InvalidUserId	کد کاربر معتبر نیست
erScm_InvalidAcceptorStatus	پذیرنده فعال نیست
erScm_InvalidTerminalStatus	پایانه فعال نیست
erScm_UseridOrPassIsRequired	کد کاربر و رمز اجباری است
erScm_InvalidUserType	نوع کاربر معتبر نیست
erScm_UserExpired	کاربر منقضی شده است



erScm_UserNotActive	کاربر غیرفعال هست
erScm_UserTemporaryInActive	کاربر موقتاً غیرفعال هست
erScm_UserSessionGenerateError	خطا در صدور مجوز لاگین
erScm_UserPassMinLengthError	خطا در حداقل طول رمز
erScm_UserPassMaxLengthError	خطا در حداکثر طول رمز
erScm_InvalidPasswordChars	کاراکترهای غیرمجاز در رمز
erScm_InvalidSession	مجوز لاگین معتبر نیست
erScm_NotAllowedToService	کاربر مجوز انجام سرویس را ندارد
erScm_SessionIsExpired	مجوز لاگین منقضی شده است
erScm_InvalidData	دیتا معتبر نیست
erScm_InvalidUser	کاربر معتبر نیست
erScm_InvalidUserCertificatePass	خطای رمز گواهینامه
erScm_InvalidUserSourceIp	آی پی مجاز نیست
erScm_AASError	خطای سرور
erScm_SCMEError	خطای سرور
erScm_UnknownError	خطای ناشناخته
erScm_MerchantNotHaveOrManySomeProduct	پذیرنده یا دارای این محصول نیست یا چند محصول مشابه دارد
erScm_VerifyUserNotEqualWithTransUser	کد کاربر تأییدکننده با کد کاربر درخواست کننده تراکنش متفاوت هست
erScm_TransAmountNotEqualWithSrvAmnt	مبلغ تراکنش با مبلغ سرویس فروش و وچر متفاوت هست
erScm_DBException	خطای ناشناخته بانک اطلاعاتی