آبان ماه ۱۳۹۳

Purchase With Signed Token

شرکت پرداخت نوین

معاونت فنی

واحد نرم افزار

فهرست

ترمينولوژي

سيستم پرداخت الكترونيكي با استفاده از كد تاييد انتقال، از پنج جزء تشكيل يافته است:

- **خريدار**: موجوديتي كه تقاضاي خريد سرويس يا كالا دارد.
- **فروشنده**: موجوديتي كه سرويس يا كالا را در اختيار خريدار قرار مي‌دهد.
- **گردآورنده (پذيرنده) (Acquirer)**: موجوديتي واسط ميان شبكه عمومي (شامل خريدار و فروشنده) و شبكه بين بانكي (شامل بانك‌هاي نگه‌دارنده سپرده‌هاي فروشنده و خريدار).
- **بانك‌هاي عضو شتاب**: موجوديت‌هايي كه سپرده‌هاي خريدار و فروشنده را نزد خود نگه مي‌دارند.
- **سايت صدور رسيد ديجيتالي**:سايتي متعلق به مركز مي‌باشد كه در آن خريدار شماره كارت و رمز آن را وارد مي‌نمايد و انتقال مبلغ خريد به سپرده فروشنده را تاييد مي‌نمايد. در صورت موفقيت‌آميز بودن انتقال، يک رسيد ديجيتالي براي آن انتقال صادر مي‌نمايد. (در واقع این کار می تواند به عنوان بخشی از وظایف ماشین پذیرنده در نظر گرفته شود)

همچنين در اين نوشته ترم‌هاي زير نيز به كار برده خواهند شد:

- **رسيد ديجيتالي (Reference Number)**: یک سلسله‌ي از کاراکترها که می تواند تا ۲۰حرف باشد و مرکز به عنوان رسید پس از انجام یک انتقال به خریدار (و در نهایت به فروشنده) ارائه می‌دهد.
- **شماره رزرو (Reservation Number)**: كدي كه فروشنده براي هر تراكنش خريدار در نظر مي‌گيرد و خريدار مي‌تواند توسط آن كد، خريد خود را پيگيري كند. 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 خريدار (Buyer Deposit Number)**: شماره سپرده‌اي كه خريدار از آن مبلغ خريد را به سپرده فروشنده انتقال داده است.
- **کد فروشنده (Merchant ID)**: کدی است که مرکز برای هر فروشنده اختصاص می دهد.
- **نام کاربری فروشنده (Merchant Username)**: نام کاربری فروشنده که توسط مرکز به هر فروشنده اختصاص داده می شود.
- **سپرده**: در بعضي از ترمينولوژي‌هاي بانكي بدان حساب گويند.
- **تراكنش**: يک عمليات مالي، كه در اين نوشته مصداق آن يک خريد مي‌باشد.

نيازمندي‌هاي امنيتي و بانكي

نيازمندي‌هاي امنيتي

- سایت بانک دارای گواهینامه های معتبر می باشد. پس ارتباط خریدار با بانک و فروشنده با بانک مي 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.
- نیازمندي 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 نمی کند و تمامی این اطلاعات توسط خریدار در سایت بانک وارد می شود.
- نیازمندی امنیتی دیگر این است که فقط فروشندگان مجاز قادر باشند که درخواست تایید یک تراکنش و درخواست برگشت خوردن یک تراکنش را صادر کنند. این امر با قرار دادن یک فایروال جلوی ماشین پذیرنده محقق شده است و آدرس IP فروشنده باید به بانک داده شود، تا اجازه دسترسی برای آن صادر شود.
- برای اطمینان از هویت فروشنده در هنگام درخواست 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)

نيازمندي‌هاي بانكي

فروشنده باید سپرده‌اي را نزد يكي از بانک‌هاي عضو شتاب افتتاح نماید و آن را به عنوان "سپرده فروشنده" به بانک معرفي نماید. بانک نیز به وي یک کد اختصاص خواهد داد که در هر تراکنش فروشنده خود را با ارائه این شماره (که اختصارا به آن کد فروشنده مي‌گوییم) به بانک معرفي مي‌کند.

بسترسازي سمت فروشنده

انجام تراکنش خرید از طریق توکن امنیتی با امضای دیجیتال

فروشنده خریدار را به نقطه‌اي مي‌رساند که آماده دريافت پول و نهايي کردن خريد مي‌باشد. در اين نقطه فروشنده بايد خريدار را به سايت صدور رسيد ديجيتالي redirect کند. تمامی پارامترهايي که به اين سايت به صورت **POST** مي‌دهد 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.)

- token
- language

نکته: برای بدست آوردن مقدار token فروشنده باید شماره رزرو (resNum)، مبلغ خرید (amount) و redirectUrl را به وب سرویس بانک به نام getPurchaseParamsToSign ارسال کند. خروجی این متد حاوی یک DTO با مقادیر زیر است:

- dataToSign: حاوی رشته ای که شامل اطلاعات خرید است و باید توسط گواهی نامه خود آن را امضای دیجیتال کنید.
- uniqueId: حاوی یک شناسه منحصر به فرد است.

بعد از دریافت دو پارامتر ذکر شده در بالا باید مقدار فیلد dataToSign توسط فروشنده امضای دیجیتال شود. سپس این مقدار باید به متد generateSignedPurchaseToken در وب سرویس بانک فراخوانی ارسال شود. (شرح کامل این متد در بخش شرح Web Method ها آمده است.). خروجی این متد شامل یک توکن امنیتی میباشد که برای انجام خرید باید آن را به سایت بانک ارسال کند.

نشانی ای که فروشنده باید فرم انجام خرید را به آن submit کند باید از مرکز دریافت شود:

https://pna.shaparak.ir/CardServices/tokenController

برای مثال در کد زیر فروشنده اطلاعات خرید مربوط به توکن با شماره 1234567891 2d4b53917ca6f250041b84 را به سایت مرکز ارسال میکند و پس از تکمیل خرید، اطلاعات مربوط به این خرید به سایت فروشنده که از طریق redirectUrl مشخص کرده است، ارسال میشود و فروشنده باید پارامترهاي ارسال شده از سايت مرکز را از بدنه فرم (ارسال شده به صورتPOST) دريافت کند.

```
<html>
<head>
	<title>Token Payment Sample</title>
	<meta charset="UTF-8">
</head>
<body>
<form action="https://localhost:8443/card/tokenController" method="post">
	token:<input name="token" type="text"
value="12345678912d4b53917ca6f250041b84"/><br/>
	language:
	<input name="language" type="radio" checked="checked" value="fa">فارسی
	<input name="language" type="radio" value="en">English
	<br/>
	<input type="submit">
</form>
</body>
</html>
```

پس از ارسال اطلاعات به سایت مرکز، خریدار باید اطلاعات کارت خود را وارد کند:

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

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

سایت مرکز پس از اتمام انتقال وجه، خریدار را دوباره به سایت فروشنده redirect می‌کند (به طوري كه رفت و آمد خريدار از سايت فروشنده به سايت مركز و سپس به سايت فروشنده در همان session سايت فروشنده صورت مي‌گيرد). پس از انجام عمليات خريد، اطلاعات اين خريد براى فروشنده در قالب يک form به صورت POST ارسال ميشود و فروشنده بايد اين پارامترها را از اين فرم دريافت كند. براى مثال پارامترهاى ارسالى از سمت مركز در قالب يک فرم مانند زير ارسال می‌شود:

```
<form action="redirectURL" method="post">
	<hidden name="token" value="token"/>
	<hidden name="redirectURL" value="redirectURL"/>
	<hidden name="MID" value="MID"/>
	<hidden name="ResNum" value="resNum"/>
	<hidden name="RefNum" value="trackingNumber"/>
	<hidden name="CustomerRefNum" value="rrn"/>
	<hidden name="State" value="paymentState"/>
	<hidden name="language" value="language"/>
	<hidden name="CardPanHash" value="cardPanHash"/>
	<submit name="Submit"/>
<form>
```

- token
- redirectURL
- MID
- ResNum
- RefNum
- State
- language
- CardPanHash

فروشنده مي‌تواند بر اساس وضعيت تراكنش، موفقيت‌آميز بودن تراكنش را تشخيص دهد. شرح كامل پارامترها در ضميمه ج آورده شده است.

پارامتر cardPanHash با استفاده از روش SHA256 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 تاييد، فروشنده بايد نتيجه را بررسی كرده و تصميم گيرد كه خريدار مبلغ مناسب را واريز نموده است يا خير. در صورت درست بودن انتقال، فروشنده رسيد ديجيتالي را بايد در پايگاه داده خود ذخيره كند و پس از آن می تواند سرويس خود را به خريدار ارايه كند. در صورت درست نبودن مبلغ انتقالي، بايد همچنان رسيد ديجيتالي را ذخيره كند و حتما بايد درخواست برگشت آن خريد را به مركز بدهد. (هرچند كه عملا خريدار نبايد بتواند مبلغ نادرستي را انتقال دهد)

با مقايسه مبلغ دريافت شده و مبلغ فاكتوری كه فروشنده خود در اختيار دارد ۳ حالت زير ممكن است به وجود آيد:

- اگر اين دو مبلغ برابر باشند، فروشنده می تواند سرويس خود را ارايه نمايد.
- اگر مبلغ پرداختی كمتر از مبلغ فاكتور فروشنده باشد، فروشنده می تواند پس از اعلام به خريدار، كل سند را برگشت بزند.
- اگر مبلغ پرداختی بيشتر از مبلغ فاكتور نزد فروشنده باشد، فروشنده می تواند پس از اعلام به خريدار، ما به التفاوت سندها را برگشت بزند.

نكات:

۱. تاكيد مي‌شود كه مصرف شدگي رسيد ديجيتالي در سمت فروشنده تعيين و نگه‌داري مي‌شود و نه در سمت مركز. مركز تنها اعتبار و مبلغ برگشت نخورده رسيد ديجيتالي را گزارش مي‌دهد. بدين ترتيب مركز مي‌تواند مشخصات يک رسيد ديجيتالي را چندين بار به فروشنده گزارش دهد بدون مركزه وضعيت مصرف‌شدگي آن تغيير كند. حسن اين روش در اين

است که اگر فروشنده‌اي يک رسيد ديجيتالي را براي اعتبارسنجي به مرکز بدهد و مرکز نيز نتيجه را براي فروشنده ارسال دارد ولي اين جواب به هر دليلي به دست فروشنده نرسد، رسيد ديجيتالي اعتبار خود را از دست نخواهد داد و فروشنده مي تواند دوباره تقاضاي اعتبارسنجي نمايد و در صورت مثبت بودن نتيجه آن را در پايگاه داده خود ذخيره کرده و وضعيت رسيد تراکنش را به مصرف شده تغيير 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 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.

۳. در صورتی که تراکنش به هر دلیلی در مدت زمان مشخصی (این زمان در مرکز تعیین می شود و فروشنده باید از مرکز درخواست کند این زمان به او اعلام شود) از جانب فروشنده تایید نشد، مرکز اقدام به برگشت زدن تراکنش خواهد کرد.

۴. در اين روش پرداخت، امکان اينکه يک رسيد ديجيتالي در دو فروشنده‌ي مختلف استفاده شود، وجود ندارد.

۵. مسوليت جلوگيري از Double Spending بر عهده فروشنده است و در صورت ضعفي در پياده‌سازي سايت فروشنده، ضرر آن متوجه خود اوست.

۶. امنيت اين بخش از سيستم به کمک SSL و ACL تامين شده است.

۷. مديريت ريسک اين بخش با استفاده از سقف‌هاي برداشت براي موجوديت خريدار و الگوهاي فروش براي فروشنده صورت مي‌گيرد. خريدار با تعيين سقف انتقال سپرده‌هاي خود مي‌تواند ميزان ريسک سپرده‌هاي خود را مديريت نمايد. همچنين فروشنده با معرفي الگوي فروش خود به مرکز، حداکثر مبالغ فروش خود را می تواند معرفي نمايد.

برگشت تراکنش خرید

فروشنده مي‌تواند امکان لغو خريد را پياده‌سازي کند. اين امکان ممکن است به دو صورت مورد نياز واقع شود:

- فروشنده لغو يک خريد را لازم بداند (براي مثال ممکن است فروشنده ديگر کالا يا سرويس مورد نظر خريدار را براي تحويل نداشته باشد).
- خريدار مايل به لغو خريد باشد.

در هر يک از دو حالت بالا، فروشنده مي‌تواند يک خريد را به دو صورت برگشت زند:

۱) برگشت کامل (Full Reverse): سند به صورت کامل برگشت خورده سپرده خريدار به مبلغ خريد بستانکار مي‌شود و سپرده فروشنده به مبلغ خريد بدهکار. براي اين کار فروشنده بايد با استفاده از پايگاه‌داده خود، رسيد ديجيتالي مربوط به Reservation Number مورد نظر خود را استخراج نمايد و سپس با چک کردن وضعيت برگشت‌خوردگي آن مطمئن شود که قبلا برگشت جزئي يا کامل نخورده باشد (البته اين چک براي صرفه‌جويي در ارتباطات شبکه‌اي است وگرنه اين چک در سمت مرکز نيز صورت مي‌گيرد). وي سپس متد برگشت وب سرويس را فرا خواهد خواند. مقدار برگشتي اين متد کدي است که نتيجه برگشت را نشان مي‌دهد. فروشنده سپس پايگاهداده خود را به روز مي‌نمايد. شرح خروجی این متد در ضمیمه ب آورده شده است.

۲) برگشت ناقص (Partial Reverse): فروشنده بخشي از مبلغ خريد را برگشت مي‌زند. براي اين کار فروشنده بر اساس Business Ruleهاي خود مبلغ برگشتي را محاسبه مي‌کند و با استفاده از پايگاهداده خود، رسيد ديجيتالي مربوط به

Reservation Number مورد نظر خود را استخراج مي‌نمايد. سپس فروشنده بايد وضعيت برگشت خوردگي رسيد ديجيتالي را چک کند و تنها در صورتي آن را براي برگشت جزئي استفاده نمايد که يا پيشتر تاييد شده باشد و يا مبلغ برگشتي به اضافه مبالغ برگشت‌هاي جزئي پيشين از کل مبلغ سند بيشتر نباشد (البته اين چک کردن نيز براي صرفه‌جويي در ارتباطات شبکه‌اي است چراکه اين چک در سمت مرکز نيز صورت مي‌گيرد). در اين صورت فروشنده مي‌تواند همان متد برگشت وب سرويس را فراخواند. مقدار برگشتي اين متد فيلدی دارد به نام resultCode. اين کد، کدي است که نتيجه برگشت را نشان مي‌دهد. در اين حالت فروشنده بايد پايگاه‌داده خود را به روز نمايد و وضعيت رسيد ديجيتالي مربوط به آن Reservation Number را برگشت خورده نمايد تا برگشت‌هاي جزئي بعدي را مديريت نمايد.

نکات:

۱) دوباره تاکيد مي‌شود که لزومي ندارد وضعيت برگشت خوردگي رسيد ديجيتالي پيش از صدور درخواست آن، توسط فروشنده چک شود؛ چراکه مرکز قبل از اجراي دستور فروشنده، مبلغ برگشت نخورده رسيد ديجيتالي را محاسبه مي‌کند و در صورت کفايت اين مبلغ، دستور برگشت را اجرا مي‌نمايد.

۲) دستور برگشت الزاما بايد از سوي فروشنده صادر شود و مرکز براي آن که مطمئن گردد که اين دستور از سوي فروشنده صادر شده است، نام کاربری فروشنده و کلمه عبور وي را چک مي‌کند. در نتيجه اگر سياست فروشنده به گونه‌اي است که خريدار مي‌تواند فرايند برگشت را خود آغاز کند، بايد نام کاربری و کلمه عبور خود را در پياده‌سازي سايت خود، hard code نمايد.

۳) امنيت اين بخش از سيستم به کمک SSL، ACL و Merchant Username and Password تامين شده است.

ضميمه ب: شرح Web Methods

۶ متد (Web Method) در اختيار فروشندگان قرار مي گيرد که شرح پارامترهاي ورودي و خروجي هر يک در زير آمده است. توجه داشته باشيد که نام type مقادير ورودي و برگشتي، نام‌هاي عامي مي‌باشند که در هر زبان برنامه‌نويسي ممکن است تفاوت يابند. همچنين مقدار برگشتي 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 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 بپردازد.

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 مرکز دریافت کرده است.
Password	String	کلمه عبور فروشنده است که از مرکز دریافت کرده است. فروشنده می تواند این مقدار را توسط یک web application که آدرس آن را مرکز 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 مرکز 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.

این متد خروجی ندارد.

توجه:

در صورت موجود نبودن متد "**addData**" ، باید یک 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);
context.setData(data);
```

خطاهای رخ داده در این متد:

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

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);
```

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 "**addData**" ، باید یک 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);
context.setData(data);
```

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 می‌دهد که درخواست تسویه قبلا ثبت شده باشد یا زمان تسویه همه ترمینال‌های فروشنده رسیده باشد.
- **WebServiceRuntimeException**: در صورت بروز خطاهای ناشناخته رخ می‌دهد.

متد دریافت اطلاعات خرید برای امضای دیجیتال (getPurchaseParamsToSign)

برای انجام خرید باید فروشنده پارامترهای زیر را ابتدا به این متد ارسال کند:

نام	نوع	شرح
Context	WsContext	روش پرکردن آن همانند متد خروج (logout) می‌باشد.
resNum	String	شماره کدی که فروشنده کالای خود را شناسایی کرده را نشان می دهد.
Amount	BigDecimal	مبلغ خرید
redirectUrl	String	آدرسی کامل url سایت فروشنده است که باید به صورت HTML Encoded برای مرکز ارسال شود. بعد از انجام عملیات خرید مرکز به این آدرس redirect خواهد کرد.

خروجی این متد شامل یک DTO به نام DataToSignResponse است که شامل مقادیر زیر میباشد:

نام	نوع	شرح
dataToSign	String	شامل اطلاعاتی است که باید توسط فروشنده امضای دیجیتال شود.
uniqueId	String	شماره کدی که فروشنده کالای خود را شناسایی کرده را نشان می دهد.

نکته : مقدار فیلد dataToSign باید توسط فروشنده امضای دیجیتال شود.

متد تولید توکن امنیتی خرید امضا شده (generateSignedPurchaseToken)

بعد از فراخوانی این متد یک توکن امنتی در اختیار فروشنده قرار میگیرد که تمامی اطلاعات خرید از این توکن استخراج خواهد شد.

پارامترهای ورودی این متد و نوع آنها در جدول زیر آمده است:

نام	نوع	شرح
Context	WsContext	روش پرکردن آن همانند متد خروج (logout) می‌باشد.
signature	String	شامل اطلاعات خرید پس از امضای دیجیتال توسط فروشنده
uniqueId	String	شامل یک شماره یکتا میباشد که در خروجی متد getPurchaseParamsToSign آن را دریافت کرده اید.
resNum	String	شماره کدی که فروشنده کالای خود را شناسایی کرده را نشان می دهد.
Amount	BigDecimal	مبلغ خرید
redirectUrl	String	آدرسی کامل url سایت فروشنده است که باید به صورت HTML Encoded برای مرکز ارسال شود. بعد از انجام عملیات خرید مرکز به این آدرس redirect خواهد کرد.

خروجی این متد یک پارامتر است از نوع TokenInfo که شامل دو مقدار زیر می‌باشد:

نام	نوع	شرح
token	String	بلیط امنیتی خرید
expirationDate	String	تاریخ انقضای بلیط امنیتی خرید.

خطاهای رخ داده در این متد:

- WsTransactionNotFoundException: اگر شماره فروشنده نامعتبر باشد. (شماره فروشنده از session دریافت می شود.)

متد تایید تراکنش در خرید با توکن امنیتی (tokenPurchaseVerifyTransaction)

این متد زمانی استفاده می‌شود که فروشنده بخواهد از روش token base برای انجام خرید استفاده کند.
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 و دومی از نوع TokenPurchaseVerificationRequestاست.
نحوه پر کردن پارامتر اول مانند متد خروج است. پارامتر دوم شامل مقادیر زیر است:

نام	نوع	شرح
token	String	بلیط امنیتی خرید
referenceNumber	String	مقدار شماره پیگیری که پس از انجام خرید برای فروشنده توسط مرکز ارسال می‌شود.
amount	String	مقدار خرید

خروجی این متد یک پارامتر است از نوع TokenPurchaseVerificationResponse که تنها شامل یک فیلد است به نام resultTotalAmount از نوع BigDecimal که مبلغ تراکنش خرید را نشان می‌دهد.

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.
- WsClientAddressException: در صورتی که فروشنده از آدرس IP غیر از آدرسی که به مرکز اعلام کرده بخواهد این سرویس را فراخوانی کند.
- WsInvalidTokenException: اگر توکن خرید نامعتبر باشد و یا توکن به فروشنده تعلق نداشته باشد.
- WsPaymentVerificationException: اگر امکان تایید وجود نداشته باشد.
- WsTransactionNotFoundException: اگر شماره فروشنده نامعتبر باشد. (شماره فروشنده از session دریافت می شود.)
- WsInvalidAmountException: اگر مبلغ تایید با مبلغ تراکنش اصلی برابر نباشد.

متد تایید تراکنش (tokenPurchaseVerifyTransaction)

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 TokenPurchaseVerificationRequest.

نحوه پر کردن پارامتر اول مانند متد خروج است. پارامتر دوم از نوع TokenPurchaseVerificationRequest میباشد که شرح مقادیر آن در جدول زیر آمده است:

نام	نوع	شرح
token	String	مقدار توکن امنیتی امضا شده که قبلا از طریق وب سرویس دریافت شده
referenceNumber	String	مقدار شماره پیگیری که ارسال شده بود.
amount	BigDecimal	مبلغ خرید انجام شده

خروجی این متد یک پارامتر است از نوع TokenPurchaseVerificationResponse که حاوی یک فیلد به نام resultTotalAmount میباشد که کل مبلغ خرید انجام شده را نشان میدهد.

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 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 مرکز تماس بگیرید.
- **WsClientAddressException**: در صورتی که فروشنده از آدرس IP غیر از آدرسی که به مرکز 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.
- **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 نگنجد.

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 offset،lengthوonlyReversedکه 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.
timeMin	Date	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+0 0:0 0
timeMax	Date	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+0 0:0 0
onlyReversed	boolean	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.
refNum	String	شماره پیگیری تراکنش انجام شده است. در صورت تنظیم این مقدار و صحیح بودن آن تنها یک تراکنش برگردانده خواهد شد.
customerRefNum	String	شماره پیگیری مخصوص مشتری برای تراکنش انجام شده است. در صورت تنظیم این مقدار و صحیح بودن آن تنها یک تراکنش برگردانده خواهد شد.
resNum	String	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 برروی BILL_PAYMENT تنظیم شده است این مقدار باید خالی (null) باشد.
billTypes	Set<BillType>	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: UNKNOWN، WATER، ELECTRICITY، GAS، IMMOBILE_PHONE (تلفن ثابت)، MOBILE_PHONE (تلفن همراه)، MUNICIPALITY_DUE (قبض شهرداری با کد ۶)، MUNICIPALITY_7 (قبض شهردای با کد ۷)، TAX (امور مالیاتی) و CUSTOM در صورتی که transactionType برروی PURCHASE تنظیم شده است این مقدار باید خالی (null) باشد.
billId	String	شناسه قبض می باشد. در صورتی که transactionType برروی PURCHASE تنظیم شده است این مقدار باید خالی (null) باشد.
paymentId	String	شناسه پرداخت قبض می باشد. در صورتی که transactionType برروی PURCHASE تنظیم شده است این مقدار باید خالی (null) باشد.
orderField	OrderField	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.
orderType	OrderType	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.
Offset	long	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.
Length	short	تعداد رکورد های برگشتی را مشخص می کند. در صورت تعیین این مقدار در خروجی تنها همین تعداد تراکنش وجود دارد.

در خروجی این متد دو پارامتر وجود دارد. اولی از نوع List<ReportResponseResult> 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+00:00
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 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.
billId	String	شناسه قبض را نشان می دهد. در صورتی که تراکنش از نوع خرید کالا باشد این مقدار خالی (null) است.
paymentId	String	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.

WebServiceRuntimeException: در صورتی که خطایی ناشناخته رخ دهد.

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 تراکنش 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 List<DetailReportResponseResult> 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+00:00
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 نشان می دهد.
mainTransactionId	Long	کد تراکنش اصلی (خرید یا پرداخت قبض) را نشان می دهد.
reverseType	ReverseType	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.

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.

WebServiceRuntimeException: در صورتی که خطایی ناشناخته رخ دهد.

ضمیمه ج: پارامترهاي تبادلي ميان سايت فروشنده و سايت صدور رسيد ديجيتالي

تراکنش خرید اینترنتی با توکن امنیتی امضا شده

پارامترهايي كه سايت فروشنده به سايت صدور رسيد ديجيتالي براي تراكنش خريد بايد به صورت POST ارائه كند عبارت است از:

نام	نوع	اجباری	شرح
Token	String	بله	توکن امنیتی خرید امضا شده
language	String	خیر	زبان نشان داده شده به کاربر در صفحه خرید است. می تواند مقادیر fa برای پارسی و en برای انگلیسی داشته باشد.

پارامترهايي كه سايت صدور رسيد ديجيتالي به سايت فروشنده به صورت **POST** برمي‌گرداند در زير تشريح شده اند. توجه كنيد كه اين پارامترها را از url query string **دريافت نكنيد** و اين پارامترها در بدنه POST وجود دارند.

نام	نوع	شرح
Token	String	توکن امنیتی خرید که فروشنده برای انجام خرید به مرکز ارسال کرده بود.
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 مرکز ارسال کند.
refNum	String	شماره پیگیری تراکنش انجام شده است.
MID	String	کد فروشنده که مرکز به فروشنده اختصاص می دهد.
language	String	زبان نشان داده شده به کاربر در صفحه خرید است.
cardPanHash	String	شماره کارت خریدار، با روش SHA2 5 6 Hash شده و برای فروشنده ارسال می شود.

- **نكته۱**: در صورتي كه مركز شماره پيگيری تهي به فروشنده برگرداند، به معناي اين است كه مشكلي در انتقال توسط خريدار بوجود آمده است.
- **نكته۲**: در صورتي كه تراكنش با موفقيت انجام شده باشد، State برابر با ok خواهد بود و فروشنده می تواند سرويس خود را به كاربر ارايه كند.

- **نکته۴**: 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.
- **نکته۵:** 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 می‌شود. 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 مرکز صادر کننده نسبت به تغییر آن اقدام نمود.

لینک های مورد نیاز:

آدرس فرم نمونه انجام تراکنش خرید

https://pna.shaparak.ir/CardServices/***tokenController***

آدرس Web Service Provider

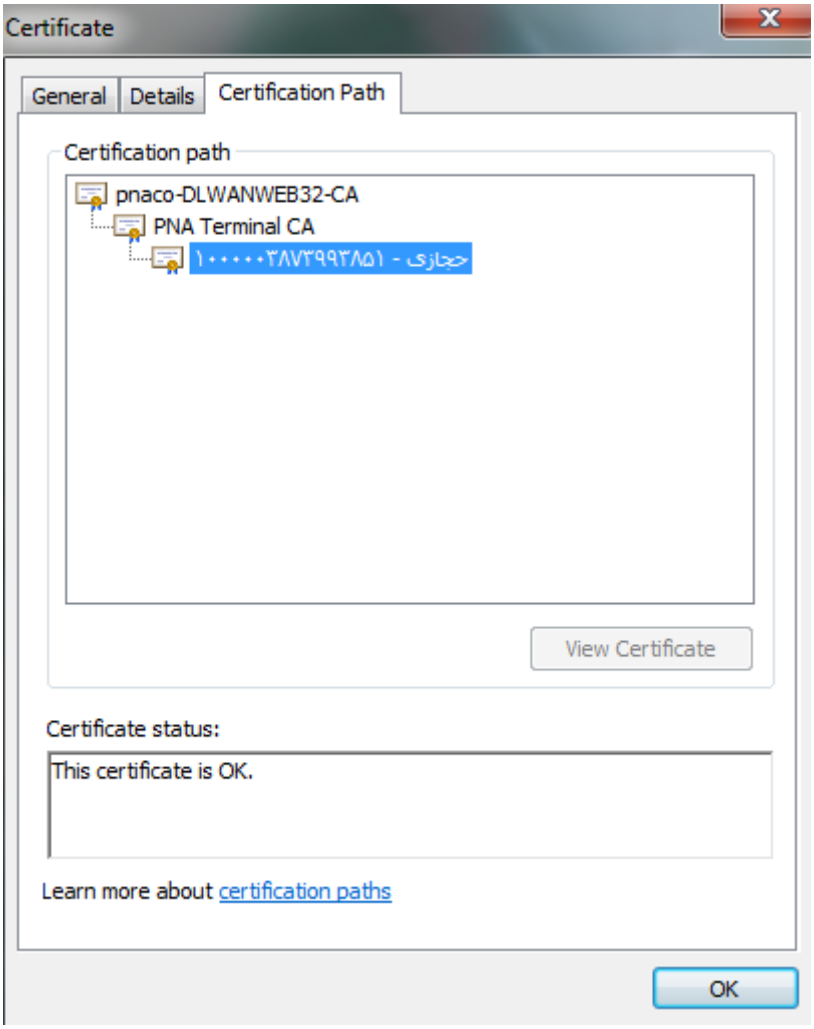
https://pna.shaparak.ir/ref-payment/jax/merchantAuth?wsdl

راهنمای نصب certificate ها :

طریقه استفاده از certificate جهت انجام امضای دیجیتالی در نمونه کد ارسالی آمده است، ابتدا برای تاییدcertificat chain باید گواهی نامه ریشه (pnaRoot) و گواهی نامه میانی (pnaIntermidiate) در قسمت certificate storage سیستم عامل وارد گردد.
به طور مثال روی سرور ویندوزی در بخش Control Panel ، گزینه Internet Options را انتخاب می کنیم، در گزینهcontent ، روی گزینه certificates کلیک می کنیم :

در این بخش در گزینه Trusted Root Certifications Authorities ، گواهی نامه ریشه را وارد می کنیم ، سپس در گزینه Intermediate Certification Authorities ، گواهی نامه میانی را وارد می کنیم .
سپس هر دو را روی سیستم عامل نصب می کنیم.

در صورت انجام درست فرآیند اضافه کردن certificate های ریشه و میانی ، درگواهی نامه مخصوص کاربر chain ایجاد شده قابل مشاهده است.



سپس باید فایل با پسوند p12 . را در برنامه خود لود کنیم – مسیر فایل ، نام گواهی نامه و رمز عبور آن را همانند نمونه کد ارسالی در تابع GetSignerCert از کلاس Sign ، قرار می دهیم ، تا روال امضای دیجیتالی با certificate درست انجام گردد.