



مرکز تنظیم مقررات نظام پایانه‌های فروشگاهی و سامانه مودیان

سند

« راهنمای اتصال به سامانه مودیان از طریق SDK دات نت »

شهریور ماه ۱۴۰۲

شهریور ماه ۱۴۰۲

سند « راهنمای اتصال به سامانه مودیان از طریق SDK دات نت »



مقدمه

در این سند راهنمای استفاده از SDK دات نت جهت سهولت در اتصال به سامانه مودیان شرح داده شده است. API سامانه مودیان از دو لایه تشکیل شده است، لایه انتقال و لایه مفهوم. لایه انتقال مستقل از اینکه چه نوع داده‌ای تبادل می‌شود، وظایف رمزنگاری و امضای بسته را بر عهده دارد. در لایه مفهوم، انواع بسته تعریف شده و بسته‌ها از طریق لایه انتقال به سامانه مودیان ارسال می‌شود.

متناظر با این لایه‌ها دو API مستقل طراحی شده است. در صورتی که بسته جدیدی به سامانه اضافه شود، تنها کافی است که API لایه محتوا بروزرسانی شود و یا می‌توان با استفاده از API لایه انتقال بسته‌های جدید را ارسال کرد.

فهرست مطالب

۴	۱-۱ شروع سریع
۴	۱-۲ لایه انتقال
۴	۱-۲-۱ پیگر بندی لایه انتقال
۴	۲-۲-۱ پیگر بندی با استفاده از کلاس TaxApi Service
۱۰	۳-۲-۱ پیگر بندی با استفاده از Dependency Injection
۱۱	۴-۲-۱ توابع ارسال بسته
۱۲	۱-۳ لایه مفهوم
۱۲	۱-۳-۱ دریافت اطلاعات سرور
۱۲	۲-۳-۱ دریافت توکن دسترسی
۱۳	۱-۴ ارسال صورت حساب
۱۹	۱-۵ استعلام نتیجه درخواست غیر همگام با استفاده از UI D
۱۹	۱-۶ استعلام نتیجه درخواست غیر همگام با استفاده از REFERENCE NUMBER
۱۹	۱-۷ استعلام نتیجه درخواست غیر همگام بر اساس زمان
۲۰	۱-۸ استعلام نتیجه درخواست غیر همگام بر اساس بازه زمان
۲۰	۱-۹ گرفتن اطلاعات حافظه
۲۰	۱-۱۰ استعلام کد اقتصادی
۲۰	۱-۱۱ گرفتن اطلاعات کالا و خدمات

۱-۱ شروع سریع

برای شروع سریع می توان نوگت SDK را دانلود و به پروژه اضافه کرده و با استفاده از کد زیر صورت حساب را ارسال نمود. گرفتن اطلاعات سرور تنها یک بار برای دریافت کلید عمومی سازمان و گرفتن توکن دسترسی در صورت منقضی شدن آن تکرار می شود.

```
TaxApiService.Instance.Init(CLIENT_ID,
new Pkcs8SignatoryConfig(s_pemFilePath, null),
new NormalProperties(ClientType.SELF_TSP),
"https://tp.tax.gov.ir/req/api/");
TaxApiService.Instance.TaxApis.GetServerInformation();
TaxApiService.Instance.TaxApis.RequestToken();

InvoiceDto invoiceDto = new InvoiceDto();
var invoices = new List<InvoiceDto>
{
    invoiceDto
};
TaxApiService.Instance.TaxApis.SendInvoices(invoices, null);
```

۱-۲ لایه انتقال

وظیفه این لایه ارسال بسته به صورت همگام و یا ناهمگام سمت سامانه مودیان است. رمزنگاری و امضای بسته ها نیز توسط این لایه صورت می پذیرد.

۱-۲-۱ پیکربندی لایه انتقال

برای پیکربندی دو راه وجود دارد. راه اول استفاده از کلاس TaxApiService است که در ادامه توضیح داده شده است. راه دوم استفاده از Dependency Injection است.

۲-۲-۱ پیکربندی با استفاده از کلاس TaxApiService

برای ایجاد کلاس TaxApiService نیاز است که ابتدا آن را پیکربندی نمود. این کلاس یک تابع Init با ۲ پیاده سازی دارد که تعدادی ورودی می گیرد. این ورودی ها به صورت زیر هستند.

ورودی	نوع ورودی	مقادیر پیش فرض	توضیحات
clientId	String	Null	شناسه کلاینت درخواست دهنده
transferSignatoryConfig	SignatoryConfig /	Null	کانفیگ مربوط به امضای یک رشته در

توضیحات	مقادیر پیش فرض	نوع ورودی	ورودی
<p>لایه انتقال، با توجه به اینکه از چه مکانیزمی برای امضاء استفاده می شود، (توکن نرم-افزاری، سخت افزاری، ...)</p> <p>پیاده سازی های مختلفی برای این کلاس در نظر گرفته شده است. می توانید از پیاده سازی دیگر تابع Init استفاده کنید که ورودی transferSignatoryConfig را ندارد و به جای آن باید کلاس پیاده سازی امضاء به صورت جنریک به تابع Init داده شود.</p> <p>نکته: کلاس پیاده سازی امضاء در لایه انتقال باید حتما اینترفیس ITransferSignatory و در لایه انتقال اینترفیس IContentSignatory را پیاده سازی کند که هر دو اینترفیس شامل دو تابع Sign و GetKeyId است که بایستی پیاده سازی شوند.</p> <p>در SDK پیاده سازی InMemorySignatory و Pkcs8Signatory و Pkcs11Signatory قرار داده شده است. در صورتی که می خواهید از این کلاس های پیاده سازی شده استفاده کنید باید ورودی transferSignatoryConfig را به تابع Init بدهید. در صورتی که بخواهید از PEM File PKCS 8 استفاده کنید، این ورودی باید از جنس Pkcs8SignatoryConfig باشد که در کانستراکتور آن باید آدرس فایل PEM داده شود. در صورتی که بخواهید از توکن سخت افزاری PKCS 11 استفاده کنید، این ورودی باید از جنس Pkcs11SignatoryConfig باشد که در کانستراکتور آن باید id کلید مورد نظر داده شود.</p>		<p>Pkcs8SignatoryConfig / Pkcs11SignatoryConfig</p>	

توضیحات	مقادیر پیش فرض	نوع ورودی	ورودی
<p>این ورودی نوع کلاینت را مشخص می کند. در حال حاضر سامانه از دو نوع Normal و GSB پشتیبانی می کند که برای مودیانی که از طریق GSB صورت حساب ارسال می کنند این ورودی باید از جنس کلاس GSBProperties و سایر مودیان باید از جنس کلاس NormalProperties باشد. کانستراکتور NormalProperties دو ورودی می گیرد:</p> <ul style="list-style-type: none"> • ورودی clientType که نحوه ارسال صورت حساب را که Tsp (شرکت معتمد) یا Self Tsp (خود مودی) است مشخص می کند. • ورودی apiVersion که ورژن سامانه را مشخص می کند که در صورت استفاده از ورژن ۱، این ورودی باید با "v1" پر شود. <p>کانستراکتور GSBProperties سه ورودی می گیرد:</p> <ul style="list-style-type: none"> • ورودی clientType که نوع درخواست را که Tsp یا Self Tsp است مشخص می کند. • ورودی tokenHeaderName که عنوان هدر توکن برای ارسال درخواست ها است که توسط GSB ارائه می شود. • ورودی customHeaders که از نوع Dictionary<string, string> و برای هدرهای مخصوص به GSB (مانند هدر Authorization) است که می توانید آنها را به صورت key و value به این دیکشنری اضافه کنید. 	<p>null</p>	<p>IProperties</p>	<p>properties</p>
<p>آدرس پایه اتصال به سامانه مودیان. این</p>	<p>https://tp.tax.gov.ir/req/api</p>	<p>string</p>	<p>baseUrl</p>

شهریور ماه ۱۴۰۲

سند « راهنمای اتصال به سامانه مودیان از طریق SDK دات نت »



توضیحات	مقادیر پیش فرض	نوع ورودی	ورودی
آدرس بر اساس اینکه به چه سروری (عملیاتی، آزمایشی) قرار است متصل شوند متفاوت خواهد بود.			
<p>کلاس مربوط به کانفیگ رمزنگاری بسته‌ها جهت ارسال به سامانه مودیان این کلاس دارای سازنده‌ای شامل دو ورودی زیر است:</p> <p>taxOrgPublicKey : کلید عمومی سازمان</p> <p>encryptionKeyId : شناسه کلید رمزنگاری</p> <p>نکته : در صورتی که این مقادیر را ندارید نیازی به پر کردن این ورودی در Init نیست اما باید قبل از فراخوانی دیگر api ها، GetServerInformation() api فراخوانی شود تا این مقادیر به طور خودکار پر شوند.</p>	<p>null</p>	<p>EncryptionConfig</p>	<p>encryptionConfig</p>
<p>کانفیگ مربوط به امضای یک رشته در لایه محتوا، با توجه به اینکه از چه مکانیزمی برای امضاء استفاده می شود، (توکن نرم افزاری، سخت افزاری، ...)</p> <p>در صورتی که مودی بخواهد صورتحساب خود را به وسیله شرکت معتمد ارسال نماید اما با کلید خود صورتحساب را امضا کند، باید این ورودی را (بسته به نوع امضا) پر کند.</p> <p>پیاده سازی های مختلفی برای این کلاس در نظر گرفته شده است. می توانید از پیاده سازی دیگر تابع Init استفاده کنید که ورودی contentSignatoryConfig را ندارد و به جای آن باید کلاس پیاده سازی امضاء به صورت جنریک به تابع Init داده شود.</p> <p>نکته : کلاس پیاده سازی امضا در لایه</p>	<p>Null</p>	<p>SignatoryConfig / Pkcs8SignatoryConfig / Pkcs11SignatoryConfig</p>	<p>contentSignatoryConfig</p>

توضیحات	مقادیر پیش فرض	نوع ورودی	ورودی
<p>انتقال باید حتماً اینترفیس <code>ITransferSignatory</code> و در لایه انتقال اینترفیس <code>IContentSignatory</code> را پیاده سازی کند که هر دو اینترفیس شامل دو تابع <code>Sign</code> و <code>GetKeyId</code> است که بایستی پیاده سازی شوند.</p> <p>در SDK پیاده سازی <code>InMemorySignatory</code> و <code>Pkcs8Signatory</code> و <code>Pkcs11Signatory</code> قرار داده شده است. در صورتی که می خواهید از این کلاس های پیاده سازی شده استفاده کنید باید ورودی <code>contentSignatoryConfig</code> را به تابع <code>Init</code> بدهید. در صورتی که بخواهید از <code>PEM File PKCS 8</code> استفاده کنید، این ورودی باید از جنس <code>Pkcs8SignatoryConfig</code> باشد که در کانستراکتور آن باید آدرس فایل <code>PEM</code> داده شود. در صورتی که بخواهید از توکن سخت افزاری <code>PKCS 11</code> استفاده کنید، این ورودی باید از جنس <code>Pkcs11SignatoryConfig</code> باشد که در کانستراکتور آن باید <code>id</code> کلید مورد نظر داده شود.</p>			

نمونه پیکربندی ساده API به صورت زیر است. در صورتی که کلید خصوصی خود را به صورت `string` دارید، می توانید مانند زیر عمل کنید: (`CLIENT_ID` شناسه کلاینت درخواست دهنده است و `privateKey` کلید خصوصی است که کلید عمومی آن را در کارپوشه آپلود کرده اید. همچنین در هر یک از مثال های زیر در صورتی که نحوه ارسال از طریق شرکت معتمد است به جای `ClientType.SELF_TSP` از `ClientType.TSP` استفاده کنید).

```
TaxApiService.Instance.Init(CLIENT_ID,
new SignatoryConfig(privateKey, null),
new NormalProperties(ClientType.SELF_TSP),
"https://tp.tax.gov.ir/req/api/");
```


در صورتی که کلید خصوصی خود را به صورت یک فایل pem دارید، می توانید مانند زیر عمل کنید:
 CLIENT_ID شناسه کلاینت درخواست دهنده است و s_pemFilePath آدرس فایل pem شامل کلید خصوصی است که کلید عمومی آن را در کارپوشه آپلود کرده اید. دقت کنید که محتویات این فایل حتماً به فرمت زیر باشد که **** همان کلید خصوصی است)

```
-----BEGIN PRIVATE KEY-----
****
-----END PRIVATE KEY-----
```

```
TaxApiService.Instance.Init(CLIENT_ID,
new Pkcs8SignatoryConfig(s_pemFilePath, null),
new NormalProperties(ClientType.SELF_TSP),
"https://tp.tax.gov.ir/req/api/");
```

در صورتی که بخواهید از توکن سخت افزاری استفاده کنید پیکربندی به صورت زیر خواهد بود: (در صورتی که توکن سخت افزاری شما شامل چندین کلید است، keyId آیدی کلید مورد نظر است که باید داده شود. در غیر این صورت می توانید با رشته خالی پر کنید)

```
TaxApiService.Instance.Init(CLIENT_ID,
new Pkcs11SignatoryConfig(keyId),
new NormalProperties(ClientType.SELF_TSP),
"https://tp.tax.gov.ir/req/api/");
```

در صورتی که بخواهید از کلاس امضای دلخواه خود تنها در لایه انتقال استفاده کنید، باید به صورت زیر پیکربندی کنید. CustomTransferSignatory کلاس امضای دلخواه است که حتماً باید اینترفیس ITransferSignatory و IContentSignatory را پیاده سازی کند.

```
TaxApiService.Instance.Init<CustomTransferSignatory>(CLIENT_ID,
new NormalProperties(ClientType.SELF_TSP));
```

در صورتی که بخواهید از کلاس امضای دلخواه خود هم در لایه انتقال و هم در لایه محتوا (برای مثال در صورتی که مودی بخواهد خود بسته را رمزنگاری و از طریق شرکت معتمد ارسال کند) استفاده کنید، باید به صورت زیر پیکربندی کنید. CustomTransferSignatory کلاس امضای دلخواه لایه انتقال است که حتماً باید اینترفیس ITransferSignatory را پیاده سازی کند. CustomContentSignatory کلاس امضای دلخواه لایه محتوا است که حتماً باید اینترفیس IContentSignatory را پیاده سازی کند.

```
TaxApiService.Instance.Init<CustomTransferSignatory,
CustomContentSignatory>(CLIENT_ID,
new NormalProperties(ClientType.SELF_TSP));
```

پس از پیکربندی، می‌توانید از پراپرتی TransferApi کلاس TaxApiService به عنوان کلاس لایه انتقال و پراپرتی TaxApis این کلاس به عنوان کلاس اصلی api ها استفاده کنید. توجه کنید که در صورتی که پیکربندی را انجام نداده باشید، پس از فراخوانی پراپرتی‌های مذکور با خطا مواجه خواهید شد.

۱-۲-۳ پیکربندی با استفاده از Dependency Injection

برای پیکربندی با استفاده از این روش به صورت زیر باید عمل کنید. استفاده از پراپرتی‌ها مشابه روش قبل است.

نکته: برای استفاده از این روش باید حتماً پکیج Microsoft.Extensions.DependencyInjection

نصب شده باشد.

```
serviceCollection.AddTaxApi(
    "https://tp.tax.gov.ir/req/api/",
    CLIENT_ID,
    new NormalProperties(ClientType.SELF_TSP),
    new SignatoryConfig(PRIVATE_KEY, null),
    new EncryptionConfig(ORG_PUBLIC_KEY, ORG_KEY_ID));
```

در صورتی که بخواهید از کلاس امضای دلخواه خود تنها در لایه انتقال استفاده کنید، باید به صورت زیر پیکربندی کنید. CustomTransferSignatory کلاس امضای دلخواه است که حتماً باید اینترفیس‌های ITransferSignatory و IContentSignatory را پیاده سازی کند.

```
serviceCollection.AddTaxApi<CustomTransferSignatory >(
    "https://tp.tax.gov.ir/req/api/",
    CLIENT_ID,
    new NormalProperties(ClientType.SELF_TSP),
    new EncryptionConfig(ORG_PUBLIC_KEY, ORG_KEY_ID));
```

در صورتی که بخواهید از کلاس امضای دلخواه خود هم در لایه انتقال و هم در لایه محتوا (برای مثال در صورتی که مودی بخواهد خود بسته را رمزنگاری و از طریق شرکت معتمد ارسال کند) استفاده کنید، باید به صورت زیر پیکربندی کنید. CustomTransferSignatory کلاس امضای دلخواه لایه انتقال است که حتماً باید اینترفیس ITransferSignatory را پیاده سازی کند. CustomContentSignatory کلاس امضای دلخواه لایه محتوا است که حتماً باید اینترفیس IContentSignatory را پیاده سازی کند.

```
serviceCollection.AddTaxApi<CustomTransferSignatory,
CustomContentSignatory>(
    "https://tp.tax.gov.ir/req/api/",
    CLIENT_ID,
    new NormalProperties(ClientType.SELF_TSP),
```

```
new EncryptionConfig(ORG_PUBLIC_KEY, ORG_KEY_ID);
```

۱-۲-۴ توابع ارسال بسته

دو تابع این لایه ارائه می‌دهد. تابع ارسال همگام بسته و تابع ارسال ناهمگام بسته‌ها. تابع `SendPacketAsync` یک بسته و تابع `SendPacketsAsync` مجموعه‌ای از بسته‌ها را به صورت غیرهمگام ارسال می‌کند.

```
Task<HttpResponse<AsyncResponseModel?>>
SendPacketsAsync<TRequest>(List<PacketDto<TRequest>> packets,
Dictionary<string, string> headers,
bool encrypt,
bool sign);
```

```
Task<HttpResponse<SyncResponseModel<TResponse?>>> SendPacketAsync<TRequest,
TResponse>(PacketDto<TRequest> packet,
Dictionary<string, string> headers,
bool encrypt,
bool sign);
```

ورودی دوم این تابع هدرهای درخواست هستند. سه هدر در این لایه در نظر گرفته شده است. در صورتی که هر کدام از این هدرها در ورودی تعریف نشود، به صورت خودکار پر می‌شوند. این هدرها اختیاری هستند و می‌توانید به عنوان ورودی ندهید. هدرها به شرح زیر است.

عنوان توکن	توضیحات
Authorization	توکن دسترسی به api
requestTraceId	شناسه تصادفی برای جلوگیری از تکرار درخواست
timestamp	زمان ارسال بسته به صورت timestamp به میلی ثانیه

قبل از ارسال هر بسته نیاز است که شناسه تصادفی تولید شده و از طریق هدر ارسال شود. با استفاده از این شناسه مانع پردازش دوباره بسته‌های تکراری گرفته خواهد شد. قبل از ارسال بسته نیاز است که زمان تولید بسته را نیز به هدر ضمیمه کنیم تا درخواست‌هایی که زمان ارسال آنها بسیار گذشته مسدود شود.

در صورتی که بخواهیم بسته به صورت رمزنگاری شده ارسال شود، فیلد encrypt را true قرار می دهیم.

در صورتی که بخواهیم محتوای بسته امضاء شود مقدار فیلد sign را true قرار می دهیم.

۱-۳ لایه مفهوم

وظیفه این لایه ایجاد بسته‌ها مطابق بسته‌هایی که سامانه مودیان پشتیبانی می کند است. بسته‌های ایجاد شده به کمک لایه انتقال، به سامانه مودیان ارسال می شود. در این بخش برای هر کدام از بسته‌ها نمونه کد ارائه شده است.

کلاس لایه مفهوم ITaxApi است و پیاده‌سازی DefaultTaxApiClient به عنوان نمونه پیاده‌سازی این لایه در SDK قرار داده شده است.

در این کلاس هر api دو پیاده‌سازی sync و async دارد که بسته به نیاز خود می توانید هر کدام را استفاده کنید.

۱-۳-۱ دریافت اطلاعات سرور

در صورتی که کلید عمومی سازمان برای رمزنگاری در لایه انتقال داده نشده باشد، قبل از استفاده از taxApi نیاز است که یک بار تابع زیر صدا زده شود تا کلید عمومی سازمان در لایه انتقال بارگذاری شود.

```
ServerInformationModel serverInformation =
TaxApiService.Instance.TaxApis.GetServerInformation();
```

۱-۳-۲ دریافت توکن دسترسی

برای گرفتن توکن دسترسی از تابع زیر استفاده می کنیم. با یک بار صدا زدن این api داخل کلاس DefaultTaxApiClient توکن ذخیره می شود و در بقیه درخواست‌ها از آن استفاده می شود. طول عمر توکن در پاسخ بازگردانی می شود و از زمان دریافت آن تا مدت ذکر شده، توکن اعتبار دارد. در صورت منقضی شدن توکن و یا دریافت کد ۴۰۱ نیاز است که توکن دسترسی جدید دریافت شود.

تغییر توکن به صورت برنامه‌ریزی شده نیز فراهم شده است.

```
TokenModel token = TaxApiService.Instance.TaxApis.RequestToken();
```

۱-۴ ارسال صورت حساب

برای ارسال صورت حساب نیاز است که فیلدهای زیر به صورت دقیق پر شوند.

ردیف	عنوان قلم اطلاعاتی	جایگاه	فیلد	نوع
۱	شماره منحصر به فرد مالیاتی	header	Taxid	String
۲	تاریخ و زمان صدور صورت حساب (میلادی)	header	Indatim	Long
۳	تاریخ و زمان ایجاد صورت حساب (میلادی)	header	Indati2m	Long
۴	نوع صورت حساب	header	Inty	Int
۵	سریال صورت حساب	header	Inno	String
۶	شماره منحصر به فرد مالیاتی صورت حساب مرجع	header	Irtaxid	String
۷	الگوی صورت حساب	header	Inp	Int
۸	موضوع صورت حساب	header	Ins	Int
۹	شماره اقتصادی فروشنده	header	Tins	String
۱۰	نوع شخص خریدار	header	Tob	Int
۱۱	شماره/شناسه ملی/شناسه مشارکت مدنی/کد فراگیر خریدار	header	Bid	String
۱۲	شماره اقتصادی خریدار	header	Tinb	String
۱۳	کد شعبه فروشنده	header	Sbc	String
۱۴	کد پستی خریدار	header	Bpc	String
۱۵	کد شعبه خریدار	header	Bbc	String
۱۶	نوع پرواز	header	Ft	Int
۱۷	شماره گذرنامه خریدار	header	Bpn	String
۱۸	شماره پروانه گمرکی فروشنده	header	Scln	String
۱۹	کد گمرک محل اظهار	header	Sec	String
۲۰	شناسه یکنای ثبت قرارداد فروشنده	header	Crn	String

ردیف	عنوان قلم اطلاعاتی	جایگاه	فیلد	نوع
۲۱	شماره اشتراک / شناسه قبض بهره بردار	header	Billid	String
۲۲	مجموع مبلغ قبل از کسر تخفیف	header	Tprdis	Decimal
۲۳	مجموع تخفیفات	header	Tdis	Decimal
۲۴	مجموع مبلغ پس از کسر تخفیف	header	Tadis	Decimal
۲۵	مجموع مالیات بر ارزش افزوده	header	Tvam	Decimal
۲۶	مجموع سایر مالیات، عوارض و وجوه قانونی	header	Todam	Decimal
۲۷	مجموع صورتحساب	header	Tbill	Decimal
۲۸	روش تسویه	header	Setm	Int
۲۹	مبلغ پرداختی نقدی	header	Cap	Decimal
۳۰	مبلغ پرداختی نسیه	header	Insp	Decimal
۳۱	مجموع سهم مالیات بر ارزش افزوده از پرداخت	header	Tvop	Decimal
۳۲	مالیات موضوع ماده ۱۷	header	Tax17	Decimal
۳۳	شناسه کالا/خدمت	body	Sstid	String
۳۴	شرح کالا/خدمت	body	Sstt	String
۳۵	واحد اندازه گیری	body	Mu	String
۳۶	تعداد/مقدار	body	Am	Double
۳۷	مبلغ واحد	body	Fee	Decimal
۳۸	میزان ارز	body	Cfee	Decimal
۳۹	نوع ارز	body	Cut	String
۴۰	نرخ برابری ارز با ریال	body	Exr	Decimal
۴۱	مبلغ قبل از تخفیف	body	Prdis	Decimal
۴۲	مبلغ تخفیف	body	Dis	Decimal
۴۳	مبلغ بعد از تخفیف	body	Adis	Decimal

شهریور ماه ۱۴۰۲

سند « راهنمای اتصال به سامانه مودیان از طریق SDK دات نت »



ردیف	عنوان قلم اطلاعاتی	جایگاه	فیلد	نوع
۴۴	نرخ مالیات بر ارزش افزوده	body	Vra	Decimal
۴۵	مبلغ مالیات بر ارزش افزوده	body	Vam	Decimal
۴۶	موضوع سایر مالیات و عوارض	body	Odt	String
۴۷	نرخ سایر مالیات و عوارض	body	Odr	Decimal
۴۸	مبلغ سایر مالیات و عوارض	body	Odam	Decimal
۴۹	موضوع سایر وجوه قانونی	body	Olt	String
۵۰	نرخ سایر وجوه قانونی	body	Olr	Decimal
۵۱	مبلغ سایر وجوه قانونی	body	Olam	Decimal
۵۲	اجرت ساخت	body	Consfee	Decimal
۵۳	سود فروشنده	body	Spro	Decimal
۵۴	حق العمل	body	Bros	Decimal
۵۵	جمع کل اجرت، حق العمل و سود	body	Tcpbs	Decimal
۵۶	سهم نقدی از پرداخت	body	Cop	Decimal
۵۷	سهم ارزش افزوده از پرداخت	body	Vop	Decimal
۵۸	شناسه یکتای ثبت قرارداد حق العملکاری	body	Bsrn	String
۵۹	مبلغ کل کالا/خدمت	body	Tsstam	Decimal
۶۰	شماره سویچ پرداخت	payment	Iinn	String
۶۱	شماره پذیرنده فروشگاه	payment	Acn	String
۶۲	شماره پایانه	payment	Trmn	String
۶۳	شماره پیگیری	payment	Trn	String
۶۴	شماره کارت پرداخت کننده صورتحساب	payment	Pcn	String
۶۵	شماره/شناسه ملی/کد فراگیر اتباع غیر ایرانی پرداخت کننده صورتحساب	payment	Pid	String
۶۶	تاریخ و زمان پرداخت صورتحساب	payment	Pdt	Long

ردیف	عنوان قلم اطلاعاتی	جایگاه	فیلد	نوع
۶۷	شماره کوتاه اظهارنامه گمرکی	header	Cdcn	String
۶۸	تاریخ کوتاه اظهارنامه گمرکی	header	Cdcd	Int
۶۹	مجموع وزن خالص	header	Tonw	Decimal
۷۰	مجموع ارزش ریالی	header	Torv	Decimal
۷۱	مجموع ارزش ارزی	header	Tocv	Decimal
۷۲	وزن خالص	body	Nw	Decimal
۷۳	ارزش ریالی کالا	body	Ssrv	Decimal
۷۴	ارزش ارزی کالا	body	Sscv	Decimal
۷۵	روش پرداخت	payment	Pmt	Int
۷۶	مبلغ پرداختی	payment	Pv	Long
۷۷	تفاوت نرخ خرید و فروش ارز/کارمزد فروش ارز	body	Pspd	Decimal

نکته: فیلدهای ۶۷ تا ۷۴ مخصوص الگوی صورتحساب صادرات هستند.

نکته: فیلد شماره ۷۷ (Pspd) مخصوص الگوی فروش ارز می باشد و در این نسخه SDK اضافه شده

است.

برای ارسال صورت حساب از `TaxApiService.Instance.TaxApis.SendInvoices` استفاده می شود. در صورتی که بخواهید علاوه بر صورتحساب، شناسه حافظه و `uid` مربوط به صورتحساب را نیز ارسال کنید (برای مثال شرکت معتمد ممکن است بخواهد به این صورت بفرستد) از `TaxApiService.Instance.TaxApis.SendTspInvoices` استفاده می شود که به صورت زیر است. `MEMORY_ID` شناسه حافظه مالیاتی و `invoice` صورتحساب ارسالی است.

```
var invoiceDtoWrappers = new List<InvoiceDtoWrapper>()
{
    new InvoiceDtoWrapper()
    {
        FiscalId = MEMORY_ID,
        Invoice = invoice,
    }
}
```


شهریور ماه ۱۴۰۲

سند « راهنمای اتصال به سامانه مودیان از طریق SDK دات نت »



```
    Uid = uid
  }
};
TaxApiService.Instance.TaxApis.SendTspInvoices(invoiceDtoWrappers, null);
```

برای ارسال صورت حساب باید مانند زیر عمل کنید. توجه کنید که invoice1 و invoice2 و ... هر یک صورت حساب جداگانه هستند که باید پر شوند. همچنین هر کدام از جنس InvoiceDto هستند. نکته: می توان به جای invoice1 و invoice2 و ... ، یک صورت حساب را ارسال کرد.

```
var invoices = new List<InvoiceDto>{
    invoice1,
    invoice2,
    invoice3,
    ...
};
TaxApiService.Instance.TaxApis.SendInvoices(invoices, null);
```

شماره مالیاتی از سه بخش شامل شناسه حافظه، تاریخ ایجاد صورت حساب و سریال صورت حساب تشکیل شده است. برای سهولت تولید شماره مالیاتی می توانید از کلاس TaxApiService.Instance.TaxIdGenerator استفاده کنید. کد زیر نمونه تولید شماره مالیاتی را نمایش می دهد.

```
string taxId =
TaxApiService.Instance.TaxIdGenerator.GenerateTaxId("A1119R", 10001,
invoiceCreatedDate);
```

پارامتر اول شناسه حافظه، پارامتر دوم سریال صورت حساب که عددی حداکثر با طول ۱۲ رقم است (می توانید به سند دستورالعمل صدور صورتحساب الکترونیکی مراجعه نمایید) و پارامتر آخر زمان صدور صورت حساب با فرمت DateTime زبان C# است.

نمونه کد ارسال صورت حساب به صورت زیر است.

```
//Generate Random Serial number
var random = new Random();
long randomSerialDecimal = random.Next(999999999);
var now = new DateTimeOffset(DateTime.Now).ToUnixTimeMilliseconds();
var taxId = TaxApiService.Instance.TaxIdGenerator.GenerateTaxId("A1119R",
randomSerialDecimal, DateTime.Now);
var header = new InvoiceHeaderDto
{
```

شهریور ماه ۱۴۰۲

سند « راهنمای اتصال به سامانه مودیان از طریق SDK دات نت »



```
Inty = 1,
Inp = 1,
Inno= string.Format("{0:X}", randomSerialDecimal).PadLeft(10,'0') ,
Tins = "5555555555",
Tprdis = 1000_000,
Tdis = 0,
Tvam = 0,
Todam = 0,
Tbill = 1000_000,
Setm = 1,
Cap = 1000_000,
Insp = 1000_000,
Tvop = 0,
Tax17 = 0,
Indatim = now,
Indati2m = now,
Taxid = taxId
};
var body = new InvoiceBodyDto
{
    Sstid = "1111111111",
    Sstt = "پاستوریزه چرب کم شیر",
    Mu = "23",
    Am = 2,
    Fee = 500_000,
    Prdis = 500_000,
    Dis = 0,
    Adis = 500_000,
    Vra = 0,
    Vam = 0,
    Tsstam = 1000_000
};
var payment = new PaymentDto
{
    Iinn = "1131244211",
    Acn = "2131244212",
    Trmn = "3131244213",
    Trn = "4131244214"
};
var invoices = new List<InvoiceDto>
{
    new()
    {
        Body = new() {body},
        Header = header,
        Payments = new() {payment}
    }
};

var responseModel = TaxApiService.Instance.TaxApis.SendInvoices(invoices,
null);
var packetResponse = responseModel.Body.Result.First();
var uid = packetResponse.Uid;
var referenceNumber = packetResponse.ReferenceNumber;
```

۱-۵ استعلام نتیجه درخواست غیر همگام با استفاده از UID

خروجی ارسال بسته‌های غیر همگام UID و referenceNumber است. فیلد uid سمت کلاینت و فیلد referenceNumber سمت سرور تولید می‌شود. در صورتی که بخواهید به وسیله uid نتیجه درخواست خود را استعلام کرده و از وضعیت بسته ارسال شده با خبر شوید باید به صورت زیر عمل کنید. (CLIENT_ID شناسه کلاینت است)

```
var uidAndFiscalId = new UidAndFiscalId(uid, CLIENT_ID);
var inquiryResultModels =
TaxApiService.Instance.TaxApis.InquiryByUidAndFiscalId(new() {uidAndFiscalId
});
```

۱-۶ استعلام نتیجه درخواست غیر همگام با استفاده از Reference Number

در صورتی که بخواهید به وسیله referenceNumber نتیجه درخواست خود را استعلام کرده و از وضعیت بسته ارسال شده با خبر شوید باید به صورت زیر عمل کنید.

```
var inquiryResultModels =
TaxApiService.Instance.TaxApis.InquiryByReferenceId(new() { referenceNumber
});
```

۱-۷ استعلام نتیجه درخواست غیر همگام بر اساس زمان

نتیجه درخواست‌ها از یک زمان بزرگتر را می‌توانیم با استفاده از این تابع با خبر شویم. ورودی این تابع، تاریخ به صورت شمسی است.

```
var inquiryResultModels =
TaxApiService.Instance.TaxApis.InquiryByTime("14010101");
```

۱-۸ استعلام نتیجه درخواست غیر همگام بر اساس بازه زمان

نتیجه درخواست‌ها از در یک بازه زمانی را می‌توانیم با استفاده از این تابع با خبر شویم. ورودی این تابع، تاریخ به صورت شمسی است.

```
var inquiryResultModels =
TaxApiService.Instance.TaxApis.InquiryByTimeRange("14010101", "14020101");
```

۱-۹ گرفتن اطلاعات حافظه

ورودی این تابع شناسه حافظه بوده و خروجی آن اطلاعات حافظه مربوطه است.

```
var fiscalInformation =
TaxApiService.Instance.TaxApis.GetFiscalInformation(CLIENT_ID);
```

۱-۱۰ استعلام کد اقتصادی

ورودی این تابع کد اقتصادی بوده و خروجی آن اطلاعات کد اقتصادی است.

```
var economicCodeInformation =
TaxApiService.Instance.TaxApis.GetEconomicCodeInformation("5555555555");
```

۱-۱۱ گرفتن اطلاعات کالا و خدمات

ورودی این تابع فیلترها و صفحه بندی‌ها جهت گرفتن اطلاعات کالا و خدمات است. نمونه کد زیر برای بارگذاری اطلاعات ۱۰ کالا است.

```
var searchDto = new SearchDto(page:1, size:10);
var serviceStuffList =
TaxApiService.Instance.TaxApis.GetServiceStuffList(searchDto).Result;
```