مرکز تنظیم مقررات نظام پایانه‌های فروشگاهی و سامانه مودیان

سند

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

دی ماه ۱۴۰۲

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.

آخرین تغییرات انجام شده در سند

ردیف	عنوان تغییرات	بخش مرتبط
۱	افزودن فیلد Cui (عیار) به فیلدهای صورتحساب	قسمت ۱.۴

سازمان امور مالیاتی کشور مهیجن	سند « راهنمای اتصال به سامانه مودیان از طریق SDK دات نت »	دی‌ماه ۱۴۰۲

فهرست مطالب

۱-۱ شروع سریع

برای شروع سریع می‌توان 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 می‌شود.

```
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 بسته‌ها نیز توسط این لایه صورت می‌پذیرد.

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.

۱-۲-۲ 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 ورودی‌ها 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

ورودی	نوع ورودی	مقادیر پیش فرض	توضیحات
	Pkcs8SignatoryConfig / Pkcs11SignatoryConfig		لایه انتقال، با توجه به اینکه از چه مکانیزمی برای امضاء استفاده می‌شود، (توکن نرم-افزاری، سخت‌افزاری، ...) پیاده سازی‌های مختلفی برای این کلاس در نظر گرفته شده است. می‌توانید از پیاده-سازی دیگر تابع Init استفاده کنید که ورودی transferSignatoryConfig را ندارد و به جای آن باید کلاس پیاده‌سازی امضاء به صورت جنریک به تابع Init داده شود. نکته : کلاس پیاده‌سازی 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 پیاده‌سازی شوند. در SDK پیاده‌سازی InMemorySignatory و Pkcs8Signatory و Pkcs11Signatory قرار داده شده است. در صورتی که می-خواهید از این کلاس‌های پیاده سازی شده استفاده کنید باید ورودی transferSignatoryConfig را به تابع Init بدهید. در صورتی که بخواهید از PEM File PKCS 8 است 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.

ورودی	نوع ورودی	مقادیر پیش فرض	توضیحات
properties	IProperties	null	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 صورتحساب 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: • ورودی clientType که نحوه ارسال صورتحساب 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. کانستراکتور GSBProperties سه ورودی می گیرد: • ورودی clientType که نوع درخواست را که Tsp یا Self Tsp است مشخص می کند. • ورودی tokenHeaderName که عنوان هدر توکن برای ارسال درخواست‌ها است که توسط GSB ارائه می‌شود. • ورودی customHeaders که از نوع Dictionary<string, string> و برای هدرهای مخصوص به GSB (مانند هدر Authorization) است که می‌توانید آنها را به صورت key و value به این دیکشنری اضافه کنید.
baseUrl	string	https://tp.tax.gov.ir/req/api	آدرس پایه اتصال به سامانه مودیان. این

ورودی	نوع ورودی	مقادیر پیش فرض	توضیحات
			آدرس بر اساس اینکه به چه سروری (عملیاتی، آزمایشی) قرار است متصل شوند متفاوت خواهد بود.
encryptionConfig	EncryptionConfig	null	کلاس مربوط به کانفیگ رمزنگاری بسته‌ها جهت ارسال به سامانه مودیان این کلاس دارای سازنده‌ای شامل دو ورودی زیر است: taxOrgPublicKey : کلید عمومی سازمان encryptionKeyId : شناسه کلید رمزنگاری 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.
contentSignatoryConfig	SignatoryConfig / Pkcs8SignatoryConfig / Pkcs11SignatoryConfig	Null	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، ...) در صورتی که مودی بخواهد صورتحساب خود را به وسیله شرکت معتمد ارسال نماید اما با کلیدِ خود صورتحساب را امضا کند، باید این ورودی را (بسته به نوع امضا) پر کند. 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. نکته : کلاس پیاده سازی امضا در لایه

ورودی	نوع ورودی	مقادیر پیش فرض	توضیحات
			انتقال باید حتماً اینترفیس ITransferSignatory و در لایه انتقال اینترفیس IContentSignatory را پیاده‌سازی کند که هر دو اینترفیس شامل دو تابع Sign و GetKeyId است که بایستی پیاده‌سازی شوند. در SDK پیاده‌سازی InMemorySignatory و Pkcs8Signatory و Pkcs11Signatory قرار داده شده است. در صورتی که می‌خواهید از این کلاس‌های پیاده سازی شده استفاده کنید باید ورودی contentSignatoryConfig را به تابع Init بدهید. در صورتی که بخواهید از PEM File PKCS 8 است 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.

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 کرده‌اید. 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، می‌توانید 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 کرده‌اید. دقت کنید که محتویات این فایل حتما به فرمت زیر باشدکه **** 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 پیاده‌سازی 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 حتما باید اینترفیس ITransferSignatory را پیاده سازی کند. CustomContentSignatory کلاس امضای دلخواه لایه محتوا است که حتما باید اینترفیس IContentSignatory را پیاده‌سازی 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 درصورتی که پیکربندی را انجام نداده باشید، پس از فراخوانی پراپرتی‌های مذکور با خطا مواجه خواهید شد.

۳-۲-۱ 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 حتما باید اینترفیس ITransferSignatory را پیاده سازی کند. CustomContentSignatory کلاس امضای دلخواه لایه محتوا است که حتما باید اینترفیس IContentSignatory را پیاده‌سازی 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));
```

۴-۲-۱ 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 می کند.

```
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 می توانید به عنوان ورودی ندهید. هدر ها 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 می‌دهیم.

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

۳–۱ لایه مفهوم

وظیفه این لایه ایجاد بسته‌ها مطابق بسته‌هایی که سامانه مودیان پشتیبانی می‌کند است. بسته‌های ایجاد شده به کمک لایه انتقال، به سامانه مودیان ارسال می‌شود. 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 می‌توانید 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 می‌کنیم. با یک بار صدا زدن این api داخل کلاس DefaultTaxApiClientتوکن ذخیره می‌شود و در بقیه درخواست‌ها از آن استفاده می‌شود. طول عمر توکن در پاسخ بازگردانی می‌شود 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();
```

۴-۱ ارسال صورت حساب

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

کد	عنوان قلم اطلاعاتی	جایگاه	فیلد	نوع
۱	شماره منحصر به فرد مالیاتی	header	Taxid	String
۲	تاریخ و زمان صدور صورتحساب(میلادی)	header	Indatim	Long
۳	تاریخ و زمان ایجاد صورتحساب(میلادی)	header	Indati2m	Long
۴	نوع صورتحساب	header	Inty	Int
۵	سریال صورتحساب	header	Inno	String
۶	شماره منحصر به فرد مالیاتی صورتحساب مرجع	header	Irtaxid	String
۷	الگوی صورتحساب	header	Inp	Int
۸	موضوع صورتحساب	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	Scc	String
۲۰	شناسه یکتای ثبت قرارداد فروشنده	header	Crn	String

کد	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

کد	عنوان قلم اطلاعاتی	جایگاه	فیلد	نوع
۴۴	نرخ مالیات بر ارزش‌افزوده	body	Vra	Decimal
۴۵	مبلغ مالیات بر ارزش افزوده	body	Vam	Decimal
۴۶	موضوع سایرمالیات و عوارض	body	Odt	String
۴۷	نرخ سایرمالیات و عوارض	body	Odr	Decimal
۴۸	مبلغ سایرمالیات 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	حق‌العمل	body	Bros	Decimal
۵۵	جمع کل اجرت، حق‌العمل 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 سوییچ پرداخت	payment	Iinn	String
۶۱	شماره پذیرنده فروشگاهی	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

کد	عنوان قلم اطلاعاتی	جایگاه	فیلد	نوع
۶۷	شماره کوتاژ اظهارنامه گمرکی	header	Cdcn	String
۶۸	تاریخ کوتاژ 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
۷۷-۷۹	این شماره‌ها برای الگوهایی است که در آینده مورد استفاده قرار خواهد گرفت			
۸۰	تفاوت نرخ خرید و فروش ارز/کارمزد فروش ارز	body	Pspd	Decimal
۸۱-۹۸	این شماره‌ها برای الگوهایی است که در آینده مورد استفاده قرار خواهد گرفت			
۹۹	عیار	body	Cui	Decimal

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

نکته : فیلد شماره ۸۰ (Pspd) مخصوص الگوی فروش ارز می‌باشد.

نکته: فیلد شماره ۹۹ (Cui) مخصوص الگوی «طلا جواهر و پلاتین» می‌باشد که در این نسخه از SDK اضافه شده است.

برای ارسال صورت حساب از TaxApiService.Instance.TaxApis.SendInvoices استفاده می‌شود. 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 می‌شود 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,
      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 : می‌توان 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 می‌توانید از کلاس TaxApiService.Instance.TaxIdGenerator استفاده کنید. کد زیر نمونه تولید شماره مالیاتی را نمایش می‌دهد.

```
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 (می‌توانید 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
{
    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 غیرهمگام 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;
```