



مرکز تنظیم مقررات نظام پایانه‌های فروشگاهی و سامانه مودیان

سند

«راهنمای استفاده از SDK جاوا»

شناسه سند:

مهر ماه ۱۴۰۱

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

مقدمه

در این سند راهنمای استفاده از SDK جاوا 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. در لایه انتقال مستقل از اینکه چه نوع داده ای تبادل میشود، وظایف رمزنگاری و امضای بسته را بر عهده دارد. در لایه مفهوم، انواع بسته تعریف شده و بسته ها از طریق لایه انتقال به سامانه مودیان ارسال میشود.

متناظر با این لایه ها دو کتابخانه مستقل طراحی شده است. در صورتی که بسته جدیدی به سامانه اضافه شود، تنها کافی است که کتابخانه لایه محتوا بروزرسانی شود و یا میتوان با استفاده از کتابخانه لایه انتقال بسته های جدید را ارسال کرد. کتابخانه مستقل دیگری با عنوان utils نیز ارائه شده است که در آن توابع کاربردی برای ارسال صورت حساب در آن گنجانده شده است.

شناسه سند	سند «راه‌نمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

فهرست مطالب

۴	۱-۱ شروع سریع	۴
۴	۱-۲ لایه انتقال	۴
۴	۱-۲-۱ پیکربندی لایه انتقال	۴
۸	۲-۲-۱ پیکربندی لایه انتقال با نسخه ۷۱ سامانه مودیان	۸
۸	۳-۲-۱ توابع ارسال بسته	۸
۹	۴-۲-۱ استفاده از توابع ارسال بسته	۹
۱۲	۱-۳ لایه مفهوم	۱۲
۱۲	۱-۳-۱ دریافت اطلاعات سرور	۱۲
۱۳	۲-۳-۱ دریافت توکن دسترسی	۱۳
۱۳	۱-۴ ارسال صورت حساب	۱۳
۱۸	۱-۵ استعلام نتیجه درخواست غیر همگام با استفاده از UID	۱۸
۱۸	۱-۶ استعلام نتیجه درخواست غیر همگام با استفاده از REFERENCE NUMBER	۱۸
۱۹	۱-۷ استعلام نتیجه درخواست غیر همگام بر اساس زمان	۱۹
۱۹	۱-۸ استعلام نتیجه درخواست غیر همگام بر اساس بازه زمان	۱۹
۱۹	۱-۹ گرفتن اطلاعات حافظه	۱۹
۱۹	۱-۱۰ استعلام کد اقتصادی	۱۹
۲۰	۱-۱۱ گرفتن اطلاعات کالا و خدمات	۲۰

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

۱-۱ شروع سریع

برای شروع سریع میتوانیم کتابخانه های اصلی زیر را دانلود کرده و به پروژه اضافه کنید و با استفاده از کد زیر صورت حساب را ارسال کنیم. گرفتن اطلاعات سرور تنها یک بار برای دریافت کلید عمومی سازمان و گرفتن توکن دسترسی در صورت منقضی شدن آن تکرار میشود.

```

ApiConfig apiConfig = new ApiConfig().signatory(new
InMemorySignatory(PRIVATE_KEY, null));

TransferApi transferApi = new ObjectTransferApiImpl(apiConfig);
TaxApi taxApi = new DefaultTaxApiClient(transferApi, CLIENT_ID);

this.taxApi.getServerInformation();

taxApi.requestToken();

InvoiceDto invoiceDto = new InvoiceDto();

taxApi.sendInvoices(Collections.singletonList(invoiceDto));

```

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 ایجاد کلاس TransferApi نیاز است که ابتدا آن را پیکربندی کنیم. برای پیکربندی از کلاس ApiConfig استفاده میکنم. مقادیر پیش فرض پیکربندی به صورت زیر است.

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

توضیحات	مقادیر پیش فرض	نوع فیلد	فیلد
<p>کلاس پیاده ساز برای ارسال درخواست های http به صورت پیش فرض از OkHttp برای ارسال استفاده شده است</p>	OkHttpSender	HttpSender	httpSender
<p>برای امضای یک شی، نیاز است که ابتدا آن را نرمال کنیم تا در مقصد به درستی امضا بررسی شود. مقدار پیش فرض ObjectNormalizer است که یک شی را میتواند نرمال کند.</p>	ObjectNormalizer	Normalizer	normalizer
<p>کلاس مربوط به چگونگی رمزنگاری بسته ها جهت ارسال به سامانه مودیان</p>	DefaultEncrypter	Encrypter	encrypter

شناسه سند

مهر ماه ۱۴۰۱

سند «راهنمای استفاده از SDK جاوا»



توضیحات	مقادیر پیش فرض	نوع فیلد	فیلد
<p>کلاس پیاده سازی مربوط به امضای یک رشته، با توجه به اینکه از چه مکانیزمی برای امضا استفاده میشود، (توکن نرم افزاری، سخت افزاری، ...)</p> <p>پیاده سازی های مختلفی برای این کلاس در نظر گرفته شده است.</p> <p>در SDK پیاده سازی InMemorySignatory قرار داده شده است که کلید خصوصی در رم نگهداری میشود.</p>	Null	Signatory	signatory
<p>آدرس پایه اتصال به سامانه مودیان. این آدرس بر اساس اینکه به چه سروری (عملیاتی، آزمایشی) قرار است متصل شوند متفاوت خواهد بود.</p>	https://tp.tax.gov.ir/req/api/self-tsp	String	baseUrl
<p>شماره نسخه API سامانه مودیان</p>	Null	String	apiVersion
<p>الویت ارسال بسته های غیرهمگام.</p> <p>NORMAL برای کلاینت های عمومی با الویت عادی و HIGH برای کلاینت هایی که الویت بالا برای پردازش را دارند.</p>	NORMAL	PriorityLevel	priorityLevel

شناسه سند	<h1>سند «راهنمای استفاده از SDK جاوا»</h1>	
مهر ماه ۱۴۰۱		

نمونه پیکربندی ساده API به صورت زیر است. برای پیکربندی، مشخص کردن کلاس امضا الزامی است. ورودی کلاس امضا کلید خصوصی است که کلید عمومی آن را در کارپوشه آپلود کرده اید.

```
ApiConfig apiConfig = new ApiConfig().signatory(new InMemorySignatory(PRIVATE_KEY, null));
```

مقدار `encrypter` را میتوانیم بعد از اینکه اطلاعات کلید عمومی سازمان را دریافت کرده ایم تنظیم کنیم و یا همان ابتدا هنگام پیکربندی وارد کنیم. نمونه پیکربندی کاملتر به صورت زیر است.

```
ApiConfig apiConfig = new ApiConfig()
    .baseUrl("https://tp.tax.gov.ir/req/api/self-tsp")
    .normalizer(new ObjectNormalizer())
    .encrypter(new DefaultEncrypter(ORG_PUBLIC_KEY, ORG_KEY_ID))
    .signatory(new InMemorySignatory(PRIVATE_KEY, null));
```

پیاده سازی `ObjectTransferApiImpl` نمونه پیاده سازی برای لایه انتقال می باشد. از کد زیر برای ایجاد کلاس لایه انتقال میتوانید استفاده کنید.

```
ApiConfig apiConfig = new ApiConfig()
    .baseUrl("https://tp.tax.gov.ir/req/api/self-tsp")
    .encrypter(new DefaultEncrypter(ORG_PUBLIC_KEY, ORG_KEY_ID))
    .signatory(new InMemorySignatory(PRIVATE_KEY, null));
```

```
TransferApi transferApi = new ObjectTransferApiImpl(apiConfig);
```

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

۱-۲-۲ پیکربندی لایه انتقال با نسخه ۷۱ سامانه مودیان

نسخه جدید سامانه مودیان بهینه سازی هایی در بحث امضا صورت گرفته است که به جای استفاده از ObjectNormalizer که بسیار سنگین و پیچیده است از SimpleNormalizer استفاده میکند. کلاس پیاده ساز برای نسخه لایه انتقال SimpleTransferApiImpl است. پیکربندی این نسخه از API به صورت زیر است.

```

ApiConfig apiConfig = new ApiConfig()
    . baseUrl("https://tp.tax.gov.ir/req/api/self-tsp")
    . normalizer(new SimpleNormalizer())
    . apiVersion("v1")
    . encrypter(new DefaultEncrypter(ORG_PUBLIC_KEY, ORG_KEY_ID))
    . signatory(new InMemorySignatory(PRIVATE_KEY, null));

TransferApi transferApi = new SimpleTransferApiImpl(apiConfig);

```

۱-۲-۳ توابع ارسال بسته

دو تابع این لایه ارائه میدهد. تابع ارسال همگام بسته و تابع ارسال ناهمگام بسته ها. تابع sendPacket یک بسته را به صورت همگام ارسال میکند و تابع sendPackets مجموعه ای از بسته ها را به صورت غیرهمگام ارسال میکند.

```

public interface TransferApi {

    AsyncResponseModel sendPackets(List<PacketDto> packets, Map<String, String> headers,
boolean encrypt, boolean sign);

```


شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

```

SyncResponseModel sendPacket(PacketDto packet, Map<String, String> headers, boolean
encrypt, boolean 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. هدرها به شرح زیر است.

عنوان توکن	توضیحات
Authorization	توکن دسترسی به api
requestTraceId	شناسه تصادفی برای جلوگیری از تکرار درخواست

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 بسته های 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 امضا شود مقدار فیلد sign را true قرار میدهیم.

۱-۲-۴ استفاده از توابع ارسال بسته

در این بخش چند نمونه کد برای ارسال بسته سمت سامانه مودیان شرح داده شده است. نمونه کد برای دریافت اطلاعات سرور به صورت زیر است.

```

PacketDto packetDto = new PacketDto();

packetDto.setUid(UUID.randomUUID().toString());

packetDto.setPacketType(PacketType.PACKET_TYPE_GET_SERVER_INFORMATION);

```

شناسه سند	<p>سند «راه‌نمای استفاده از SDK جاوا»</p>	 <p>سازمان امور مالیاتی کشور مُهْمِن</p>
مهر ماه ۱۴۰۱		

```
packetDto.setFiscalId(شناسه حافظه);
```

```
packetDto.setData(null);
```

```
packetDto.setRetry(false);
```

```
SyncResponseModel response = transferApi.sendPacket(packetDto, null, false, false);
```

```
Object responseData = response.getResult().getData();
```

```
ServerInformationModel si = mapper.convertValue(responseData,  
ServerInformationModel.class);
```

```
KeyDto key = si.getPublicKeys().get(0);
```

```
DefaultEncrypter encrypter = new DefaultEncrypter(key.getKey(), key.getId());
```

```
transferApi.getConfig().encrypter(encrypter);
```

با استفاده از اطلاعاتی که از سرور دریافت شده میتوانیم، مقدار کلید عمومی سازمان را بروز کنیم.

```
KeyDto key = si.getPublicKeys().get(0);
```

```
DefaultEncrypter encrypter = new DefaultEncrypter(key.getKey(), key.getId());
```

```
transferApi.getConfig().encrypter(encrypter);
```

ارسال بسته برای دریافت توکن دسترسی به صورت زیر است.

شناسه سند	سند «راه‌نمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

```

GetTokenDto data = new GetTokenDto();

data.setUsername(clientId);

PacketDto packetDto = new PacketDto();

packetDto.setUid(UUID.randomUUID().toString());

packetDto.setPacketType(PacketType.PACKET_TYPE_GET_TOKEN);

packetDto.setFiscalId(شناسه حافظه);

packetDto.setData(data);

packetDto.setRetry(false);

SyncResponseModel response = transferApi.sendPacket(packetDto, null, false, false);

Object responseData = response.getResult().getData();

TokenModel tokenModel= mapper.convertValue(responseData, TokenModel.class);

```

ایجاد بسته، برای ارسال صورت حساب به صورت زیر است.

```

AsyncResponseModel sendInvoices(List<InvoiceDto> invoices) {

    ArrayList<PacketDto> packets = new ArrayList<>();

    for (InvoiceDto invoiceDto : invoices) {

        PacketDto packetDto = new PacketDto();

        packetDto.setUid(UUID.randomUUID().toString());

```

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

```

packetDto.setPacketType("INVOICE.V01");

packetDto.setFiscalId(شناسه حافظه);

packetDto.setData(invoiceDto);

packetDto.setRetry(false);

packets.add(packetDto);
}

```

```

HashMap<String, String> headers = new HashMap<>();

headers.put(TransferConstants.AUTHORIZATION_HEADER, "Bearer " + token.getToken());

return transferApi.sendPackets(packets, headers, true, true);
}

```

۱-۳ 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 TaxApi است و پیاده‌سازی DefaultTaxApiClient به عنوان نمونه پیاده‌سازی این لایه در SDK قرار داده شده است. در سازنده این کلاس، شی مربوط به لایه انتقال و شناسه حافظه وارد میشود.

```
TaxApi taxApi = new DefaultTaxApiClient(transferApi, شناسه حافظه);
```

۱-۳-۱ دریافت اطلاعات سرور

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

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

```
ServerInformationModel serverInformation = this.taxApi.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 DefaultTaxApiClient توکن 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 برنامه ریزی شده نیز فراهم شده است.

```
TokenModel token = this.taxApi.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
۱	شماره منحصر به فرد مالیاتی	header	taxid
۲	تاریخ و زمان صدور صورت حساب (میلادی)	header	indatim
۳	تاریخ و زمان ایجاد صورت حساب (میلادی)	header	Indati2m
۴	نوع صورت حساب	header	inty
۵	سریال صورت حساب	header	inno
۶	شماره منحصر به فرد مالیاتی صورت حساب مرجع	header	irtaxid
۷	الگوی صورت حساب	header	inp
۸	موضوع صورت حساب	header	ins
۹	شماره اقتصادی فروشنده	header	tins
۱۰	نوع شخص خریدار	header	tob

شناسه سند

مهر ماه ۱۴۰۱

سند «راهنمای استفاده از SDK جاوا»



ردیف	عنوان قلم اطلاعاتی	جایگاه	فیلد
۱۱	شماره/شناسه ملی/شناسه مشارکت مدنی/کد فراگیر خریدار	header	bid
۱۲	شماره اقتصادی خریدار	header	tinb
۱۳	کد شعبه فروشنده	header	sbc
۱۴	کد پستی خریدار	header	bpc
۱۵	کد شعبه خریدار	header	bbc
۱۶	نوع پرواز	header	ft
۱۷	شماره گذرنامه خریدار	header	bpn
۱۸	شماره پروانه گمرکی فروشنده	header	scln
۱۹	کد گمرک محل اظهار	header	scc
۲۰	شناسه یکتای ثبت قرارداد فروشنده	header	crn
۲۱	شماره اشتراک/شناسه قبض بهره بردار	header	billid
۲۲	مجموع مبلغ قبل از کسر تخفیف	header	tprdis
۲۳	مجموع تخفیفات	header	tdis
۲۴	مجموع مبلغ پس از کسر تخفیف	header	tadis
۲۵	مجموع مالیات بر ارزش افزوده	header	tvam
۲۶	مجموع سایر مالیات، عوارض و وجوه قانونی	header	todam
۲۷	مجموع صورتحساب	header	tbill
۲۸	روش تسویه	header	setm
۲۹	مبلغ پرداختی نقدی	header	cap
۳۰	مبلغ پرداختی نسیه	header	insp

ردیف	عنوان قلم اطلاعاتی	جایگاه	فیلد
۳۱	مجموع سهم مالیات بر ارزش افزوده از پرداخت	header	tvop
۳۲	عدم پرداخت مالیات بر ارزش افزوده خریدار	header	dvpb
۳۳	مالیات موضوع ماده ۱۷	header	Tax17
۳۴	شناسه کالا/خدمت	body	sstid
۳۵	شرح کالا/خدمت	body	ssst
۳۶	واحد اندازه گیری	body	mu
۳۷	تعداد/مقدار	body	am
۳۸	مبلغ واحد	body	fee
۳۹	میزان ارز	body	cfee
۴۰	نوع ارز	body	cut
۴۱	نرخ برابری ارز با ریال	body	exr
۴۲	مبلغ قبل از تخفیف	body	prdis
۴۳	مبلغ تخفیف	body	dis
۴۴	مبلغ بعد از تخفیف	body	adis
۴۵	نرخ مالیات بر ارزش افزوده	body	vra
۴۶	مبلغ مالیات بر ارزش افزوده	body	vam
۴۷	موضوع سایر مالیات و عوارض	body	odt
۴۸	نرخ سایر مالیات و عوارض	body	odr
۴۹	مبلغ سایر مالیات و عوارض	body	odam
۵۰	موضوع سایر وجوه قانونی	body	olt

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

ردیف	عنوان قلم اطلاعاتی	جایگاه	فیلد
۵۱	نرخ سایر وجوه قانونی	body	olr
۵۲	مبلغ سایر وجوه قانونی	body	olam
۵۳	اجرت ساخت	body	consfee
۵۴	سود فروشنده	body	spro
۵۵	حق العمل	body	bros
۵۶	جمع کل اجرت، حق العمل و سود	body	tcpbs
۵۷	سهم نقدی از پرداخت	body	cop
۵۸	سهم ارزش افزوده از پرداخت	body	vop
۵۹	شناسه یکتای ثبت قرارداد حق العملکاری	body	bsrn
۶۰	مبلغ کل کالا/خدمت	body	tsstam
۶۱	شماره سویچ پرداخت	payment	iinn
۶۲	شماره پذیرنده فروشگاه	payment	acn
۶۳	شماره پایانه	payment	trmn
۶۴	شماره پیگیری	payment	trn
۶۵	شماره کارت پرداخت کننده صورتحساب	payment	pcn
۶۶	شماره/شناسه ملی/کد فراگیر اتباع غیر ایرانی پرداخت کننده صورتحساب	payment	pid
۶۷	تاریخ و زمان پرداخت صورتحساب	payment	pdt

شماره مالیاتی از سه بخش تشکیل میشود. بخش اول شناسه حافظه، بخش دوم تاریخ ایجاد صورت حساب و بخش سوم سریال صورت حساب است. برای سهولت تولید شماره مالیاتی میتوانید از کتابخانه Utils استفاده کنید. کد زیر نمونه تولید شماره مالیاتی را نمایش میدهد.

شناسه سند	سند «راه‌نمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

```
String taxId = TaxUtils.generateTaxId("A1116H", 10001, invoiceCreatedDate);
```

پارامتر اول شناسه حافظه، پارامتر دوم سریال صورت حساب که عددی حداکثر با طول ۱۲ رقم است و پارامتر آخر زمان صدور صورت حساب با فرمت Instant جاوا است.

نمونه کد ارسال صورت حساب به صورت زیر است.

```
//Generate Random Serial number
Random random = new Random();
long randomSerialDecimal = random.nextInt(999999999);
Instant invoiceCreatedDate = Instant.now();
String taxId = TaxUtils.generateTaxId("A1116H", randomSerialDecimal, invoiceCreatedDate);
```

```
InvoiceHeaderDto header = new InvoiceHeaderDto();
header.setIndatim(invoiceCreatedDate.toEpochMilli());
header.setIndati2m(invoiceCreatedDate.toEpochMilli());
header.setTaxid(taxId);
header.setInty(1);
header.setInp(1);
header.setInno(randomSerialDecimal);
header.setIns(1);
header.setTins("5555555555");
header.setTprdis(BigDecimal.valueOf(1000_000));
header.setTdis(BigDecimal.ZERO);
header.setTvam(BigDecimal.ZERO);
header.setTodam(BigDecimal.ZERO);
header.setTbill(BigDecimal.valueOf(1000_000));
header.setSetm(1);
header.setCap(BigDecimal.valueOf(1000_000));
header.setInsp(BigDecimal.valueOf(1000_000));
header.setTvop(BigDecimal.ZERO);
header.setTax17(BigDecimal.ZERO);
```

```
InvoiceBodyDto body = new InvoiceBodyDto();
body.setSstid("1111111111");
body.setSstt("شیر کم چرب پاستوریزه");
body.setMu(23);
body.setAm(2D);
body.setFee(BigDecimal.valueOf(500_000));
body.setPrdis(BigDecimal.valueOf(500_000));
body.setDis(BigDecimal.ZERO);
```

شناسه سند	سند «راهنامه‌ی استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

```
body.setAdis(BigDecimal.valueOf(500_000));
body.setVra(BigDecimal.ZERO);
body.setVam(BigDecimal.ZERO);
body.setTsstam(BigDecimal.valueOf(1000_000));
```

```
PaymentDto payment = new PaymentDto();
payment.setLinn("1131244211");
payment.setAcn("2131244212");
payment.setTrmn("3131244213");
payment.setTrn("4131244214");
```

```
InvoiceDto invoiceDto = new InvoiceDto();
invoiceDto.setBody(Collections.singletonList(body));
invoiceDto.setHeader(header);
invoiceDto.setPayments(Collections.singletonList(payment));
```

```
AsyncResponseModel responseModel =
taxApi.sendInvoices(Collections.singletonList(invoiceDto));
PacketResponse packetResponse = responseModel.getResult().get(0);
this.uid = packetResponse.getUid();
this.referenceNumber = packetResponse.getReferenceNumber();
```

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

خروجی ارسال بسته های غیر همگام UID و referenceNumber است. فیلد uid سمت کلاینت و فیلد referenceNumber سمت سرور تولید میشود. با استفاده از این کد ها میتوانیم از وضعیت بسته ارسال شده با خبر شویم. نمونه کد استعلام به صورت زیر است.

```
UidAndFiscalId uidAndFiscalId = new UidAndFiscalId();
uidAndFiscalId.setUid(this.uid);
uidAndFiscalId.setFiscalId(CLIENT_ID);
List<InquiryResultModel> inquiryResultModels =
this.taxApi.inquiryByUidAndFiscalId(Collections.singletonList(uidAndFiscalId));
```

۱-۶ استعلام نتیجه درخواست غیر همگام با استفاده از referenceNumber

با استفاده از این کد ها میتوانیم از وضعیت بسته ارسال شده با خبر شویم. نمونه کد استعلام به صورت زیر است.

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

```
List<InquiryResultModel> inquiryResultModels =
this.taxApi.inquiryByReferenceId(Collections.singletonList(this.referenceNumber));
```

۱-۷ استعلام نتیجه درخواست غیر همگام بر اساس زمان

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

```
List<InquiryResultModel> inquiryResultModels = this.taxApi.inquiryByTime("14010101");
```

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

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

```
List<InquiryResultModel> inquiryResultModels = this.taxApi.inquiryByTimeRange("14010101",
"14020101");
```

۱-۹ گرفتن اطلاعات حافظه

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

```
FiscalInformationModel fiscalInformation = taxApi.getFiscalInformation(CLIENT_ID);
```

۱-۱۰ استعلام کد اقتصادی

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

```
EconomicCodeModel economicCodeInformation =
taxApi.getEconomicCodeInformation("۵۵۵۵۵۵۵۵۵۵۵۵")
```

شناسه سند	سند «راهنمای استفاده از SDK جاوا»	
مهر ماه ۱۴۰۱		

۱-۱۱ گرفتن اطلاعات کالا و خدمات

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

```
SearchDto searchDto = new SearchDto();

searchDto.setPage(1);
searchDto.setSize(10);
SearchResultModel<ServiceStuffModel> serviceStuffList =
taxApi.getServiceStuffList(searchDto);
```