



---

نظام پایانه های فروشگاهی و سامانه مودیان

«دستور العمل فنی نحوه اتصال به سامانه مودیان»

(نحوه تبادل اطلاعات میان پایانه فروشگاهی - حافظه مالیاتی با سامانه مودیان)

---

شناسه سند: RC\_TICS.IS\_V01

خرداد ماه ۱۴۰۱

## فهرست مطالب

۱- مقدمه	۵
۲- هدف	۵
۳- دامنه کاربرد	۵
۴- تعاریف	۵
۵- پیکربندی پایانه فروشگاهی-حافظه مالیاتی	۸
۵-۱ دریافت شناسه یکتای حافظه مالیاتی	۸
۵-۲ پیکربندی و ثبت مشخصات	۹
۶- امنیت اطلاعات	۹
۶-۱ توکن	۹
۶-۲ امضا	۱۰
۶-۲-۱ نرمالسازی درخواست	۱۰
۶-۲-۲ فرآیند امضا صورتحساب	۱۱
۶-۳ نحوه رمزگذاری صورتحساب	۱۴
۷- فراخوانی متدهای API جمع آوری اطلاعات سامانه مودیان	۱۵
۷-۱ آدرس API ها	۱۵
۷-۲ ساختار درخواست ها	۱۶
۷-۳ سرویس های جمع آوری و پردازش اطلاعات	۲۰
۷-۳-۱ متد غیرهمگام	۲۰
۷-۳-۱-۱ سرویس درخواست غیرهمگام (ارسال صورتحساب)	۲۰
۷-۳-۱-۲ ساختار بسته صورتحساب	۲۱
۷-۳-۱-۳ پاسخ درخواست غیرهمگام	۲۶
۷-۳-۲ متدهای همگام	۲۷
۷-۳-۲-۱ سرویس درخواست های همگام	۲۷
۷-۳-۲-۲ پاسخ درخواست های همگام	۲۹
۷-۳-۳ متد دریافت توکن	۳۰
۷-۳-۴ متد استعلام اطلاعات حافظه مالیاتی مودی و حد مجاز فروش مودی	۳۰

- ۳۰ ..... ۵-۲-۳-۷ متدهای استعلام بسته‌های ارسالی غیرهمگام
- ۳۱ ..... ۶-۲-۳-۷ متد دریافت اطلاعات سرور
- ۳۱ ..... ۷-۲-۳-۷ متد دریافت لیست کامل شناسه کالا/خدمات و نرخ مالیاتی
- ۳۲ ..... ۸-۲-۳-۷ متد استعلام اطلاعات شماره اقتصادی
- ۳۲ ..... ۸- لیست خطاها
- ۳۲ ..... ۸-۱ مدل داده خطاها
- ۳۳ ..... ۸-۲ لیست خطاهای لایه انتقال
- ۳۴ ..... ۸-۳ لیست خطاهای لایه محتوا
- ۳۸ ..... پیوست ۱
- ۵۵ ..... پیوست ۲

## فهرست شکل‌ها

- شکل ۱. نمودار آماده سازی صورتحساب جهت ارسال ..... ۹
- شکل ۲. روش نرمالسازی درخواست ..... ۱۰
- شکل ۳. نمودار ترتیبی فرآیند ارسال صورتحساب ..... ۲۱

## فهرست جداول

- جدول ۱. روش نرمالسازی JSON ..... ۱۱
- جدول ۲. متدهای همگام و غیرهمگام ..... ۱۶
- جدول ۳. ساختار کلی درخواستها ..... ۱۷
- جدول ۴. توضیحات مربوط به فیلدهای درخواست ..... ۱۷
- جدول ۵. ساختار بسته (PACKET) ..... ۱۸
- جدول ۶. توضیحات فیلدهای بسته (PACKET) ..... ۱۹
- جدول ۷. اقلام صورتحساب ..... ۲۱
- جدول ۸. مدل داده پاسخهای غیرهمگام ..... ۲۶
- جدول ۹. توضیحات فیلدهای خروجی درخواست غیرهمگام ..... ۲۶
- جدول ۱۰. درخواستهای همگام ..... ۲۷
- جدول ۱۱. توضیحات فیلدهای ساختار خروجی درخواست همگام ..... ۲۹
- جدول ۱۲. ساختار خروجی پاسخ خطا ..... ۳۲
- جدول ۱۳. خطاهای لایه انتقال ..... ۳۳
- جدول ۱۴. کدهای خطای دریافتی از هسته مالیاتی یا به عبارتی کدهای خطای لایه محتوا ..... ۳۴

## ۱- مقدمه

در این سند نحوه اتصال به سامانه مودیان، دریافت اطلاعات مورد نیاز جهت پیکربندی پایانه فروشگاهی-حافظه مالیاتی، ارسال اطلاعات صورتحساب الکترونیکی و استعلام وضعیت صورتحساب های ارسالی شرح داده شده است. برای ارسال اطلاعات صورتحساب الکترونیکی از مکانیزم غیر همگام و برای دریافت و استعلام اطلاعات از مکانیزم همگام استفاده شده است. برای هر کدام از این مکانیزم ها، API مستقلی در نظر گرفته شده است. برای مکانیزم غیر همگام از صف استفاده می شود.

در کلیه مراحل استفاده از API، اصول امنیت اطلاعات شامل احراز هویت و سطح دسترسی ارسال کننده، محرمانگی، انکارناپذیری و یکپارچگی رعایت شده است.

## ۲- هدف

این سند با هدف تشریح نحوه اتصال به سامانه مودیان ارائه شده است. در این سند مراحل اتصال به سامانه مودیان شامل دریافت اطلاعات مورد نیاز جهت پیکربندی پایانه فروشگاهی-حافظه مالیاتی، ارسال اطلاعات صورتحساب الکترونیکی و استعلام وضعیت صورتحساب های ارسالی به صورت گام به گام و همراه با جزئیات فنی مورد نیاز برای پیاده سازی آن توسط کلیه ذینفعان ارائه شده است.

## ۳- دامنه کاربرد

ذینفعان این سند شامل:

- اشخاص مشمول (مودیان).
- شرکت های معتمد ارائه کننده خدمات مالیاتی نوع اول

## ۴- تعاریف

- **شناسه یکتای حافظه مالیاتی:** شناسه ای است یکتا دارای مقداری ثابت و منحصر به فرد که به هر حافظه مالیاتی در سطح کشور اختصاص داده می شود، این شناسه از مولفه های تشکیل دهنده شماره

منحصر به فرد مالیاتی می باشد که پس از درخواست مودی در کارپوشه تولید و در اختیار وی قرار خواهد گرفت.

- **شماره منحصر به فرد مالیاتی:** شماره ای است یکتا در سامانه مودیان که به ازای هر صورتحساب تولید شده و به صورت منحصر به فرد به آن صورتحساب تخصیص داده می شود. این شماره دارای بخش های اطلاعاتی خاص بوده که جزئیات آن در سند «قالب شناسه یکتای حافظه مالیاتی و شماره منحصر به فرد مالیاتی» ذکر شده است.

- **صورتحساب الکترونیکی:** صورتحسابی است دارای شماره منحصر به فرد مالیاتی که اطلاعات مندرج در آن، در حافظه مالیاتی فروشنده ذخیره می شود.

مشخصات و اقلام اطلاعاتی صورتحساب الکترونیکی<sup>۱</sup>، متناسب با نوع کسب و کار توسط سازمان تعیین و اعلام شده است. در مواردی که از دستگاه کارتخوان بانکی یا درگاه پرداخت الکترونیکی به عنوان پایانه فروشگاهی استفاده شود، رسید یا گزارش الکترونیکی پرداخت خرید صادره در حکم صورتحساب الکترونیکی است.

- **حافظه مالیاتی:** نوعی حافظه الکترونیکی است که برای ثبت و نگهداری اطلاعات مندرج در صورتحساب های الکترونیکی و انتقال آن به سامانه مودیان مورد استفاده قرار می گیرد. حافظه مالیاتی می تواند به شکل سخت افزاری یا نرم افزاری باشد. حافظه مالیاتی، توسط مودی برای ثبت صورتحساب الکترونیکی مورد استفاده قرار می گیرد. هر حافظه مالیاتی باید دارای شماره شناسه یکتا باشد. شناسه یکتای حافظه مالیاتی توسط سازمان اختصاص داده می شود.

- **پایانه فروشگاهی:** رایانه، دستگاه کارتخوان بانکی، درگاه پرداخت الکترونیکی یا هر وسیله دیگری که امکان اتصال به شبکه های الکترونیکی پرداخت رسمی کشور و سامانه مودیان را داشته و از قابلیت صدور صورتحساب الکترونیکی برخوردار باشد (بند ب ماده ۱ قانون).

- **شرکت های معتمد ارائه کننده خدمات مالیاتی:** اشخاص حقوقی دارای پروانه هستند که حسب ضوابط و دستورالعمل های ابلاغی سازمان، نسبت به ارائه مشاوره و آموزش های لازم به مودیان، نصب

<sup>۱</sup> مطابق سند دستورالعمل صدور صورتحساب الکترونیکی RC\_IITP.IS\_V03

و پشتیبانی تجهیزات مورد نیاز برای ارائه خدمات مالیاتی از قبیل خدمات مربوط به صدور صورتحساب و سایر امور غیرحاکمیتی با سازمان همکاری می کند (بند چ ماده ۱).

- **زیرسامانه جمع آوری و پردازش اطلاعات صورتحساب:** زیرسامانه ای است که داده های ارسالی از پایانه های فروشگاهی-حافظه مالیاتی یا شرکت های معتمد ارائه کننده خدمات مالیاتی نوع اول را از طریق واسطه های نرم افزاری دریافت می کند.

- **حد مجاز فروش:** جمع صورتحساب های الکترونیکی صادره توسط هر مودی در هر دوره مالیاتی نمی تواند بیشتر از سه برابر فروش اظهار شده وی در دوره مشابه سال قبل، که مالیات آن به سازمان پرداخت شده یا ترتیب پرداخت آن داده شده است، باشد. جمع صورتحساب های الکترونیکی صادره شده در هر دوره مالیاتی برای واحدهای جدیدالتاسیس یا واحدهای فاقد سابقه مالیاتی نمی تواند بیش از سه برابر معافیت سالانه موضوع ماده (۱۰۱) قانون مالیات های مستقیم باشد (ماده ۶ قانون پایانه های فروشگاهی و سامانه مودیان).

- **شناسه یکتای ارسال صورتحساب:**<sup>۲</sup> شناسه ای یکتا که هنگام ارسال صورتحساب توسط ارسال کننده به صورتحساب جهت رهگیری آن اختصاص داده می شود. این شناسه با فرمت <sup>۳</sup>uuid به صورت رمز نشده ارسال می گردد.

- **رسید یکتای دریافت صورتحساب:**<sup>۴</sup> هنگام دریافت صورتحساب در زیرسامانه جمع آوری و پردازش اطلاعات، یک شناسه رسید یکتا با فرمت <sup>۳</sup>uuid به هر صورتحساب توسط این زیرسامانه اختصاص داده می شود.

- **لیست صورتحساب ها:** مجموعه ای از صورتحساب های صادر شده توسط پایانه فروشگاهی-حافظه مالیاتی که در قالب یک مجموعه به سامانه مودیان ارسال می شود.

- **کلاینت:** ارسال کننده درخواست به API زیرسامانه جمع آوری و پردازش اطلاعات که می تواند مودی یا شرکت معتمد ارائه کننده خدمات مالیاتی باشد.

- **سرور:** سرور سامانه مودیان

<sup>2</sup> uid

<sup>3</sup> Universal unique identificatier

<sup>4</sup> reference\_number

- **زوج کلید عمومی و خصوصی:** در این سند هرگاه برای انکارناپذیری از کلید عمومی یا خصوصی استفاده می شود منظور کلیدی است که توسط یک مرکز میانی معتبر برای هر یک از ذینفعان سامانه مودیان گواهی شده باشد.
- **مرکز میانی معتبر:** هر مرکز میانی که از شورای سیاست گذاری گواهی الکترونیکی کشور مجوز گرفته باشد (موضوع ماده ۳۲ قانون تجارت الکترونیکی). لیست این مراکز از طریق سایت [www. rca.gov.ir](http://www.rca.gov.ir) بخش مراکز میانی، قابل دسترسی است.

## ۵- پیگردی پایانه فروشگاه-حافظه مالیاتی

### ۵-۱ دریافت شناسه یکتای حافظه مالیاتی

مودی جهت صدور و ارسال صورتحساب الکترونیکی نیاز به دریافت شناسه یکتا حافظه مالیاتی دارد. بنابراین می بایست به بخش عضویت و ثبت نام کارپوشه خود مراجعه نموده و مراحل زیر را طی نماید:

۱. به ازای هر شناسه یکتا حافظه مالیاتی، یکی از سه حالت ارسال اطلاعات صورتحساب را به شرح ذیل انتخاب کند:

- توسط خود مودی (به روش مستقیم)
- ارسال اطلاعات صورتحساب توسط شرکت معتمد ارائه کننده خدمات مالیاتی (به روش غیرمستقیم)
- ارسال اطلاعات صورتحساب توسط شرکت معتمد ارائه کننده خدمات مالیاتی (به روش غیرمستقیم و با استفاده از زیرساخت های اختصاصی)

۲. کلید عمومی (RSA) دریافتی از مراکز میانی معتبر با طول کلید ۲۰۴۸ بیت را بارگذاری نماید.

نکته: در صورتی که ارسال غیرمستقیم باشد و شرکت معتمد ارائه کننده خدمات مالیاتی صدور، رمزگذاری و ارسال صورتحساب را به عهده داشته باشد، بارگذاری کلید عمومی توسط مودی ضرورتی ندارد. در این حالت شرکت معتمد ارائه کننده خدمات مالیاتی باید از طریق کارپوشه خود،



کلید عمومی مربوطه را به سازمان معرفی نماید.

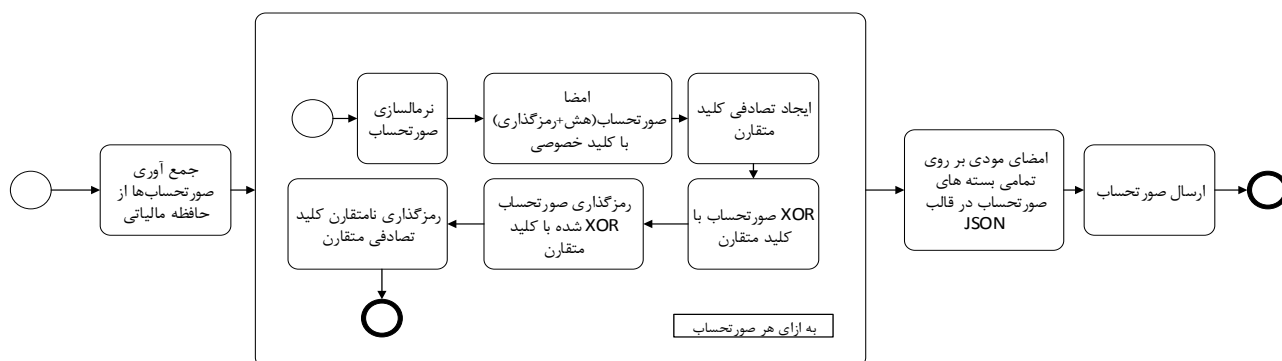
۳. ارتباط شناسه یکتای حافظه مالیاتی درخواستی با کدپستی (های) محل فعالیت تعیین گردد.

## ۵-۲ پیکربندی و ثبت مشخصات

مودی برای راه اندازی پایانه فروشگاهی-حافظه مالیاتی خود باید از طریق فراخوانی متدهای "دریافت اطلاعات سرور" و "دریافت اطلاعات حافظه مالیاتی"، شماره اقتصادی، نام تجاری، ساعت و تاریخ، کلید عمومی سازمان و شناسه یکتای حافظه مالیاتی را از سرور دریافت کند. جزئیات اقلام اطلاعاتی لازم در جدول ۱۰ توضیح داده شده است.

## ۶- امنیت اطلاعات

مکانیزم های امنیتی جهت ارسال صورتحساب مطابق با نمودار ارائه شده در شکل (۱) می باشد.



شکل ۱. نمودار آماده سازی صورتحساب جهت ارسال

## ۶-۱ توکن

با هدف احراز هویت و تعیین سطح دسترسی ارسال کننده اطلاعات، از مکانیزم تخصیص توکن JWT<sup>۵</sup> استفاده شده است. لازم است توکن تخصیص داده شده به مودی یا شرکت معتمد ارائه کننده خدمات مالیاتی نوع اول در Header تمامی درخواست های ارسال شده مودی به API های زیرسامانه جمع آوری و پردازش اطلاعات قرار گیرد.

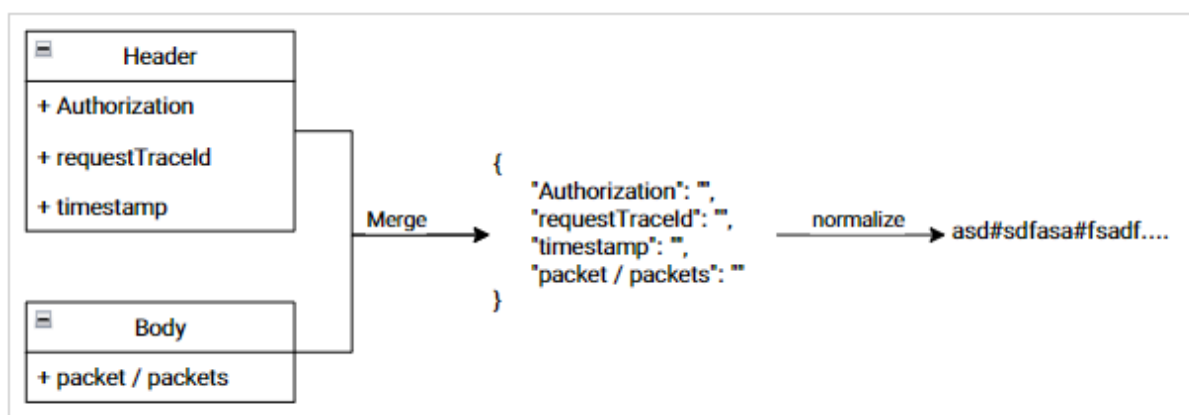
<sup>۵</sup> Json Web Token

## ۶-۲ امضا

با هدف حفظ قابلیت انکارناپذیری، یکپارچگی و احراز هویت، بر روی تمامی درخواست‌ها به API های زیرسامانه جمع آوری و پردازش اطلاعات مکانیزم امضا در نظر گرفته شده است.

## ۶-۲-۱ نرمالسازی درخواست

برای یکسان کردن ساختار تمامی درخواست‌ها به API های زیرسامانه جمع آوری اطلاعات، برای تمامی بسته‌های ارسالی شامل همگام و غیرهمگام امضا بر روی درخواست صورت می‌پذیرد. برای امضا می‌بایست ابتدا Header و Body درخواست ادغام و JSON واحد تولید گردد. ضروری است JSON تولید شده مطابق شکل (۲) به رشته تبدیل، سپس امضا و ارسال شود.



شکل ۲. روش نرمالسازی درخواست

فرآیند تبدیل JSON به رشته، مطابق گام‌های جدول (۱) می‌باشد:

- شی به فرمت کلید-مقدار تبدیل شده به طوری که کلید، عمق جایگاه مقدار را مشخص می‌نماید.
- کلیدها بر اساس حروف الفبا مرتب شوند.
- سپس مقادارها به ترتیب با هم ادغام شوند:
- از کاراکتر # به عنوان جداکننده مقادیر (اقلام اطلاعاتی) استفاده شود.
- در صورتی که کاراکتر # در متن وجود داشته باشد، با ## مشخص می‌گردد.
- در صورتی که مقدار فیلدی در رشته، null یا "" باشد؛ با ### مشخص می‌گردد.
- در آخر آرایه‌ای از روی این رشته ایجاد می‌شود.

## جدول ۱. روش نرمالسازی JSON

گام ۱	{ "k2": "v1", "k4": "v2", "k3": { "k1": "v4", "k5": "v5" } }	"k2": "v1" "k4": "v2" "k3.k1": "v4" "k3.K5": "v5"
گام ۲	"k2": "v1" "k3.k1": "v4" "k3.K5": "v5" "k4": "v2"	
گام ۳	v1#v4#v5#v2	

نکات:

- در صورتی که داخل JSON آرایه وجود داشته باشد، ترتیب عناصر آرایه دست کاری نشده و با همان ترتیب در نظر گرفته می شود.

- در صورتی که ریشه JSON آرایه باشد، داخل فیلد packets قرار گرفته و تبدیل به شی می شود.

روش پیاده سازی نرمالسازی JSON در پیوست های ۱-۱ و ۱-۲ ارائه شده است.

## ۶-۲-۲ فرآیند امضا صورتحساب

برای امضا صورتحساب باید اطلاعات JSON صورتحساب به روشی که برای نرمال سازی بیان شد نرمال

شوند. به عنوان مثال اطلاعات JSON صورتحساب زیر را در نظر بگیرید:

```
{
  "header" :
  {
    "taxid": "AA56CD0E062002F2B4E78",
    "indatim": "1655620821274",
    "indati2m": "1655620821274",
    "inty": 2,
    "inno" : "0002F2B4E7",
    "irtaxid" : null,
    "inp": 1,
  }
}
```

```
"ins" : 1,
"tins" : "32652362589632",
"tob" : 1,
"bid" : null,
"tinb" : null,
"sbc" : null,
"bpc" : null,
"bbc" : null,
"ft" : null,
"bpn" : null,
"scln" : null,
"scc" : null,
"crn" : null,
"billid" : null,
"tprdis" : 1000000,
"tdis" : 0,
"tadis" : 1000000,
"tvam" : 90000,
"todam" : 0,
"tbill" : 1090000,
"setm" : 1,
"cap" : 90000,
"insp" : 0,
"tvop" : 90000,
"dpvb" : null,
"tax17" : null
}
"body" : [
{
"sstid" : 2153265989636,
"sstt" : "پاستیل میوه ای شیبابا",
"mu" : 96,
"am" : 1,
"fee" : 1000000,
"cfee" : null,
"cut" : null,
"exr" : null,
"prdis" : 1000000,
"dis" : 0,
"adis" : 1000000,
"vra" : 0.09,
"vam" : 90000,
"odt" : null,
"odr" : null,
"odam" : null,
```

```

    "olt" : null,
    "olr" : null,
    "olam" : null,
    "consfee" : null,
    "spro" : null,
    "bro" : null,
    "tcpbs" : null,
    "cop" : 90000,
    "vop" : 90000,
    "bsrn" : null,
    "tsstam" : 1090000
  }
],
"payments" : [
  {
    "iinn" : 5646556,
    "acn" : 5656565,
    "trmn" : 54554224,
    "trn" : 544542424,
    "pcn" : "6037-9972-9856-9865",
    "pid" : null,
    "pdt" : 1655620821274,
  }
],
"extension": [
  {
    "key" : null,
    "value" : null
  }
]
]
}

```

❖ تمام اطلاعات پر شده در JSON غیر واقعی و تستی می باشد.

بعد از نرمال سازی JSON صورتحساب مورد نظر، رشته نرمال شده به صورت زیر بدست می آید:

1000000#1#####90000###0###1000000#96#####1000000###2153265  
 989636# پاستیل میوه ای  
 ۱#۱۶۵۵۶۲۰۸۲۱۲۷۴#####۹۰۰۰#####۰,۰۹#۹۰۰۰#۹۰۰۰#۱۰۹۰۰۰####شیبایا  
 ۶۵۵۶۲۰۸۲۱۲۷۴#۰۰۰۲F2B4E7#1#1#0.0#2#####1#1000000###AA56CD0E0620002F2  
 B4E78#1090000#0###32652362589632#1#0.0#1000000#90000#90000#5656565#5646556  
 #6037 - 9972 - 9856 - 9865#1655620821274#null#54554224#544542424

رشته تولید شده در مراحل بالا به وسیله کلید خصوصی (مودی/شرکت معتمد) با الگوریتم RSA2048-

SHA256 هاش و امضا می شود و خروجی آن در فیلد dataSignature در شی packet قرار می گیرد.

❖ کد روش امضا در پیوست شماره ۱-۳ ارائه شده است.

### ۳-۶ نحوه رمز گذاری صورتحساب

برای رمز گذاری صورتحساب می بایست یک کلید متقارن (به روش AES/GCM/NoPadding) با طول ۲۵۶ بیت تولید شود. برای رمز گذاری از طریق AES/GCM نیاز به یک کلید دیگر به نام IV به طول ۱۲۸ بیت می باشد که این کلید به صورت تصادفی تولید می گردد. بعد از تولید کلیدهای مورد نیاز، JSON صورتحساب را ابتدا با کلید متقارن XOR کرده و سپس به روش AES/GCM رمز گذاری می گردد.

جزئیات XOR به این شکل است که باید متنی که می خواهیم XOR کنیم (در اینجا JSON صورتحساب) باید به بلاک های ۲۵۶ بیتی تبدیل شود و هر بلاک با کلید متقارن که ۲۵۶ بیت است XOR شود. با این روش ممکن است که بلاک آخر تعداد بیت کمتر از ۲۵۶ داشته باشد که با همان تعداد از کلید متقارن XOR انجام می شود.

برای صورتحساب تستی بخش قبل کلید متقارن رمز شده و IV به صورت hex در زیر نمایش داده شده است :

AES key : 4fda3c622e966e0839441401bbd3b8f191d4267bf5f19b40812a34b212fd3ed9

IV : 4fda3c622e966e0839441401bbd3b8f191d4267bf5f19b40812a34b212fd3ed9

نمونه صورتحساب رمز شده به شکل زیر است :

```
rvd4iGRTya68VppDwli9cGf+d0Qfo6g9Eu4hsyqQe4vwY17F2dQvXgvK6KwDJR2
RQIdhXcx6SM+UF/XE4CJxMgmjffFMvBBjM34ix2ZFIAr2v2rwZ6XLUPTHWK
HIXROxnA10BagrgOm0Xr4J6Y359ZZpnaNChPabj7s4yE4ZGT+wBHoz+J/yj2xFW
KWmmcuP1b6rWOdWYO2OD6eXI1+faxT2vZ7FSBYyDy+sJP5S6Zllp92baFHX7
Z78VVqVGv+P+iJ+CLljD6j75k19ykD7stVkvHsoMwWYndqNiBIRLBXOjczSkJ/y
L89eiGKcAUvEAGSiXgaeKjEymLaOWXZEdg0IwxVUrgRyX+4kVbWeCUWHSb
jafCdQNRUddFZLWs3bO6MA552gaAh9iXJGmnJNvCbWjpjTvx9NyDbvd3thZhFl
dhX5dkVEi7mWv3iTywRCUN/6QkOxZ4JDG7u3i20vqztuqxq/Kj/51GMOGov2hhF
jd+5c6QckvO7COMX4XVO6azrWGa3VVyZz6dG9vCvVb/BZsVTxGBddQiQKd5
aaQxxIvs+sM7/FibQ4l5lISN+4ax8g420qO5E6hA+mMQ3oECnJtyKJmDK6oR6T5y
//5dByFffMySmhdx+bhS0XxMFvk7ZPUcWFZwDGs6yNKA2AApt58xqGnbuf+N2
86UjPH9CgNogY/G5lvFlPX51qi6q6X/YX165WhD0MbVEL8EEyfk/60rjwLivhEkc
ZWowhyOBnxtHvVmpjocTS/404Oskepra/4jRKiqON3hqwpEJE8V0hanYwp9m32
HtYbiql9RqL3wHycbWtnueFNjptplKi4tczl4NsfD9ezD6TQVgcAT0o14OAnkJTO/
0QnwfTgv8krEy7F/umNYWPswgWrjqSHtGiUHbXLunGP6fwZZOT0vECwXrz9X/
```

aHGx/rm64/orJArZ9Vi2OIdzvGdQbcXsGAKKuAlCvssA4XnXMBvWW9+g4Y+q  
 qsHBTomkvHYrXvLob02FUWL0XsTkab/BXfLlj4HBQUhh0gUk6aX8wetRknFiH  
 Hjb1gjN7Tjc51Jt1GifTZhNg/uHIJB/6/GlwIbxAPXB4YfNiRAU0y0iCiU1Ow8Jl6Tu  
 us3cgGkLvo5uOIF/0vLx9O2BSJHcpA0OBhIjaJ1bdgHSYOr3P3tDY+BVSe33/8oO  
 0DAVBjEDUQKx3F0Ncr6TwxlqewquIHL2zmktSMc2KbIFY12Jx7cG84hh4Ih98Vi  
 Fg+Xad83Ed3G/JiU3uG+7qiaYi8DYI6Icw0rf870qQ/CkopVR44xRtePslZfZjKS0X5  
 39/4qymPDs3+KFYZGntliwc6fRNSLhjoTidCHFRcF9tP2Ttkp6CTsnLsHtQugYZatn  
 rMO+X+GNxihIF9Be9/UJMdJ9vzwLWUn+/0+ducZKJ3mpTw5T8DqYuhzOcpYoR  
 obear9uzYb+2Ky8g==

کد روش رمزگذاری به روش متقارن AES/GCM در پیوست ۱-۴ ارائه شده است.

پس از رمزگذاری صورتحساب، از طریق الگوریتم AES/GCM، باید کلید متقارن رمز شده و IV در کنار صورتحساب رمز شده قرار گیرد. برای رمزگذاری کلید متقارن باید از روش نامتقارن RSA-OAEP-SHA256 استفاده شود که برای این منظور از کلید/های عمومی سازمان با طول ۴۰۹۶ بیت اخذ شده از یک مرکز میانی معتبر استفاده می شود (کلید عمومی سازمان با استفاده از متد GET\_SERVER\_INFORMATION به دست می آید).

کد روش رمزگذاری به روش نامتقارن RSA-OAEP-SHA256 در پیوست ۱-۵ ارائه شده است.

## ۷- فراخوانی متدهای API جمع آوری اطلاعات سامانه مودیان

### ۷-۱ آدرس API ها

آدرس API های زیر سامانه جمع آوری و پردازش اطلاعات به صورت زیر است که پیشوند تمام آدرس ها قرار می گیرد.

<https://tp.tax.gov.ir/req/>

آدرس هایی که با tsp شروع می شوند برای شرکت های معتمد ارائه کننده خدمات مالیاتی در نظر گرفته شده و آدرس هایی که با self-tsp شروع می شوند برای مودیانی که قصد دارند خودشان صورتحساب ارسال کنند در نظر گرفته شده است.

دو مکانیزم برای درخواست‌های با اولویت عادی و درخواست‌های با اولویت بالا وجود دارد. آدرس normal-enqueue برای درخواست‌های معمولی و fast-enqueue برای درخواست‌های با اولویت بالا در نظر گرفته شده است.

متدها همراه با آدرس API در جدول (۲) ارائه شده اند.

جدول ۲. متدهای همگام و غیرهمگام

ردیف	اسم متد	آدرس متد
۱	ارسال صورتحساب الکترونیکی	.../api/self-tsp/async/normal-enqueue/ .../api/self-tsp/async/fast-enqueue/ .../api/tsp/async/normal-enqueue/ .../api/tsp/async/fast-enqueue/
۲	متد دریافت توکن	.../api/self-tsp/sync/ GET_TOKEN/ .../api/tsp/sync/ GET_TOKEN/
۳	استعلام اطلاعات حافظه مالیاتی مودی و حد مجاز فروش مودی	.../api/self-tsp/sync/ GET_FISCAL_INFORMATION/ .../api/tsp/sync/ GET_FISCAL_INFORMATION/
۴	استعلام با شناسه یکتای ارسال صورتحساب	.../api/self-tsp/sync/ INQUIRY_BY_UID / .../api/tsp/sync/ INQUIRY_BY_UID /
۵	استعلام با رسید یکتای دریافت صورتحساب	.../api/self-tsp/sync/INQUIRY_BY_REFERENCE_NUMBER / .../api/tsp/sync/INQUIRY_BY_REFERENCE_NUMBER /
۶	دریافت خطاهای بسته‌های ارسالی غیرهمگام با استفاده از زمان	.../api/self-tsp/sync/INQUIRY_BY_TIME/ .../api/tsp/sync/INQUIRY_BY_TIME /
۷	دریافت خطاهای بسته‌های ارسالی غیرهمگام با استفاده از بازه زمانی	.../api/self-tsp/sync/ INQUIRY_BY_TIME_RANGE / .../api/tsp/sync/ INQUIRY_BY_TIME_RANGE /
۸	دریافت اطلاعات سرور	.../api/self-tsp/sync/ GET_SERVER_INFORMATION / .../api/tsp/sync/ GET_SERVER_INFORMATION /
۹	دریافت لیست کامل شناسه کالا/خدمات و نرخ مالیاتی	.../api/self-tsp/sync/ GET_SERVICE_STUFF_LIST / .../api/tsp/sync/ GET_SERVICE_STUFF_LIST /
۱۰	استعلام اطلاعات شماره اقتصادی	.../api/self-tsp/sync/GET_ECONOMIC_CODE_INFORMATION / .../api/tsp/sync/GET_ECONOMIC_CODE_INFORMATION /

## ۷-۲ ساختار درخواست‌ها

در این بخش ساختار درخواست‌ها به API شرح داده شده است. در جدول (۳) ساختار کلی درخواست ارائه شده است.



ساختار کلی سرویس همگام مشابه با سرویس غیر همگام بوده و تفاوت آن در فیلد packet است. بطوریکه در حالت همگام به جای آرایه‌ای از درخواست، فقط یک درخواست می‌توان ارسال نمود (packet). در درخواست غیر همگام می‌توان مجموعه‌ای از بسته‌ها (packets) را ارسال نمود.

جدول ۳. ساختار کلی درخواستها

عنوان	مقدار
متد HTTP	POST
فیلدهای Header	<b>Authorization: "string"</b> <b>requestTraceId: "string"</b> <b>timestamp: "Long"</b>
فیلدهای Body	<pre>{   "packets": Packet[],   "signature": "string",   "signatureKeyId": "string" }</pre>
فیلدهای خروجی	<pre>{   "timestamp": "Long",   "result": AsyncResponse[],   "signature": "string",   "signatureKeyId": "string" }</pre>
خروجی سرویس در صورت رخداد خطای کلی	<pre>{   "timestamp": "Long",   "errors": ErrorResponse[],   "signature": "string",   "signatureKeyId": "string" }</pre>

❖ فیلد signatureKeyId اختیاری بوده و مقدار پیش فرض آن برابر با null خواهد بود.

اطلاعات تکمیلی فیلدها در جدول (۴) آورده شده است:

جدول ۴. توضیحات مربوط به فیلدهای درخواست

نام فیلد	توضیحات
<b>Authorization</b>	توکن با این سرآیند (Header) ارسال می‌شود، در صورتیکه فراخوانی یک متد نیاز به احراز هویت نداشته باشد، این فیلد اختیاری خواهد شد. این سرآیند در API های غیر همگام و برخی از API

نام فیلد	توضیحات
	های همگام اجباری است.
requestTraceId	هر درخواستی باید دارای یک UUID باشد که در سرآیند ارسال می‌شود. از این شناسه برای تشخیص درخواست‌های تکراری استفاده می‌شود.
timestamp	زمان ارسال بسته از کلاینت که در سرآیند ارسال می‌شود. یکی از کاربردهای این فیلد، رد کردن بسته‌های قدیمی است.
packets	لیست بسته‌های ارسالی.
signature	امضای روی درخواست.
signatureKeyId	شناسه کلید عمومی ارسال کننده، برای بررسی امضا. این فیلد اختیاری بوده و مقدار پیش فرض آن برابر با Null خواهد بود.

زمانی که کلاینت پاسخ این سرویس را دریافت می‌کند، الزاماً رسیدگی به بسته‌های اطلاعاتی پایان نیافته است، بلکه تنها در صف رسیدگی قرار گرفته‌اند. تایید رسیدگی و نتیجه اعمال موفق یا ناموفق بسته‌های اطلاعاتی از طریق استعلام به اطلاع کلاینت خواهد رسید.

در صورتی که کلاینت، نتیجه رسیدگی به یک بسته را failed دریافت نماید، لازم است پس از اطمینان از عدم وجود خطاها در بسته ارسالی آن را مجدداً ارسال نماید. در ارسال مجدد باید مقدار فیلد retry برابر true باشد تا سرویس غیرهمگام در جریان ارسال مجدد درخواست باشد.

ساختار بسته‌های ارسالی به سرور و فیلدهای مربوطه به ترتیب مطابق جداول (۵) و (۶) می‌باشد.

جدول ۵. ساختار بسته (packet)

اسم شی	Packet
--------	--------

<pre>{   "uid": "string",   "packetType": "string",   "retry": "Boolean",   "data": "string",   "encryptionKeyId": "string",   "symmetrickey": "string",   "iv": "string",   "fiscalId": "string",   "dataSignature": "string" }</pre>	نوع فیلدها
--	------------

جدول ۶. توضیحات فیلدهای بسته (packet)

نام فیلد	توضیحات
uid	شناسه یکتای ارسال صورتحساب، کدی که در سمت کلاینت هنگام ارسال تولید می شود.
packetType	نوع بسته
retry	نوع فیلد بولین است (مقدار TRUE/FALSE)، مشخص می کند که بسته اولین بار است ارسال شده است یا خیر.
data	داده های داخل درخواست. در صورتی که داده ها رمز شده باشند، آرایه بایت به صورت رشته Base64 ارسال می شود. در صورتی که داده ها کشف باشند، رشته سریال شده JSON قرار داده می شود.
encryptionKeyId	شناسه کلید عمومی سازمان، جهت باز کردن کلید متقارن. در صورتی که داده ها رمز نشده باشند، این فیلد خالی ارسال می شود.
symmetrickey	کلید متقارن رمز شده با کلید رمزنگاری (کلید عمومی سازمان) جهت باز کردن داده های رمز شده. در صورتی که داده ها رمز نشده باشند، این فیلد خالی ارسال می شود.
iv	بردار مقدار اولیه کلید متقارن. در صورتی که داده ها رمز نشده باشند، این فیلد خالی ارسال می شود.
fiscalId	شناسه یکتای حافظه مالیاتی. در صورتی که ارسال کننده شرکت ارائه کننده خدمات مالیاتی باشد، این فیلد اجباری خواهد بود.
dataSignature	امضای صورتحساب.

## ۷-۳ سرویس های جمع آوری و پردازش اطلاعات

درخواست همگام و غیرهمگام دو سرویس اصلی زیر سامانه جمع آوری و پردازش اطلاعات می باشند:

### ۷-۳-۱ متد غیرهمگام

#### ۷-۳-۱-۱ سرویس درخواست غیرهمگام (ارسال صورتحساب)

ارسال صورتحساب به صورت غیرهمگام انجام می شود و جهت ارسال به احراز هویت و امضا دیجیتال صورتحساب و کل لیست ارسالی و همچنین رمزگذاری نیاز است.

برای اینکه ساختار مناسبی برای رهگیری تغییرات بسته ها وجود داشته باشد، نوع بسته (PacketType) در صورتحساب به دو بخش تقسیم بندی می شود.

بخش اول	بخش دوم
INVOICE	شماره نسخه + حرف V

برای مثال اگر صورتحسابی با نسخه بسته 01 داشته باشیم، نوع بسته به صورت زیر خواهد بود:

INVOICE.V01

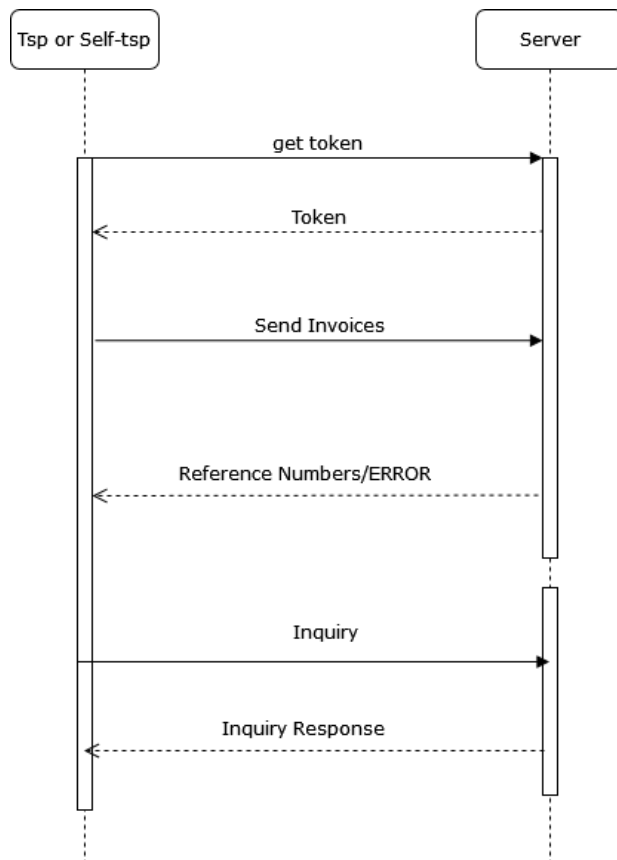
فرآیند ارسال صورتحساب مطابق شکل (۳) و به شرح ذیل می باشد:

۱. مودی یا TSP با ارسال درخواست توکن به سرور جمع آوری فرآیند ارسال صورتحساب را شروع می کند.

۲. توکن دریافت شده در سرآیند درخواست قرار می گیرد و صورتحساب ارسال می شود.

۳. مودی یا TSP رسید یکتای دریافت صورتحساب درخواست خود را از سامانه مودیان دریافت می کند.

۴. مودی یا TSP می تواند به وسیله فراخوانی متدهای استعلام از وضعیت ارسال صورتحساب خود با خبر شود.



شکل ۳. نمودار ترتیبی فرآیند ارسال صورتحساب

در صورتی که وضعیت درخواست به صورت "PENDING" باشد به این معنی است که هنوز درخواست پردازش نشده است.

نمونه CURL درخواست در پیوست ۲-۱ ارائه شده است.

### ۲-۱-۳-۷ ساختار بسته صورتحساب

شناسنامه همه اقلام اطلاعاتی که در انواع و الگوهای صورتحساب وجود دارند به شرح جدول (۷) می باشد:

جدول ۷. اقلام صورتحساب

ردیف	عنوان قلم اطلاعاتی	JSON	محل قرارگیری در صورتحساب	نوع فیلد	طول فیلد	مقادیر مجاز
۱	شماره منحصر به فرد مالیاتی	taxid	سرآمد (Header)	string	۲۲	
۲	تاریخ و زمان صدور صورتحساب (میلادی)	indatim	سرآمد (Header)	Timestamp	۱۴	

ردیف	عنوان قلم اطلاعاتی	JSON	محل قرار گیری در صورت حساب	نوع فیلد	طول فیلد	مقادیر مجاز
۳	تاریخ و زمان ایجاد صورت حساب (میلادی)	Indati2m	سرآمد (Header)	Timestamp	۱۴	
۴	نوع صورت حساب	inty	سرآمد (Header)	int	۱	۱- نوع اول ۲- نوع دوم ۳- نوع سوم
۵	سریال صورت حساب	inno	سرآمد (Header)	string	۱۰	
۶	شماره منحصر به فرد مالیاتی صورت حساب مرجع	irtaxid	سرآمد (Header)	string	۲۲	
۷	الگوی صورت حساب	inp	سرآمد (Header)	int	۱	۱-۲-۳-۴-۵-۶ الگوی ۱: فروش الگوی ۲: فروش ارزی الگوی ۳: صورت حساب طلا، جواهر و پلاتین الگوی ۴: قرارداد پیمانکاری الگوی ۵: قبوض خدماتی الگوی ۶: بلیت هواپیما
۸	موضوع صورت حساب	ins	سرآمد (Header)	int	۱	۱-۲-۳-۴ ۱- اصلی ۲- اصلاحی ۳- ابطالی ۴- برگشت از فروش

ردیف	عنوان قلم اطلاعاتی	JSON	محل قرار گیری در صورتحساب	نوع فیلد	طول فیلد	مقادیر مجاز
۹	شماره اقتصادی فروشنده	tins	سرآمد (Header)	int	۱۰	
۱۰	نوع شخص خریدار	tob	سرآمد (Header)	int	۱	۵-۴-۳-۲-۱ ۱- حقیقی ۲- حقوقی ۳- مشارکت مدنی ۴- اتباع غیر ایرانی ۵- مصرف کننده نهایی
۱۱	شماره/شناسه ملی/شناسه مشارکت مدنی/کد فراگیر خریدار	bid	سرآمد (Header)	int	متغیر	
۱۲	شماره اقتصادی خریدار	tinb	سرآمد (Header)	int	۱۰	
۱۳	کد شعبه فروشنده	sbc	سرآمد (Header)	int	متغیر	
۱۴	کد پستی خریدار	bpc	سرآمد (Header)	int	۱۰	
۱۵	کد شعبه خریدار	bbc	سرآمد (Header)	int	متغیر	
۱۶	نوع پرواز	ft	سرآمد (Header)	int	۱	۲-۱ ۱- داخلی ۲- خارجی
۱۷	شماره گذرنامه خریدار	bpn	سرآمد (Header)	string	متغیر	
۱۸	شماره پروانه گمرکی فروشنده	scln	سرآمد (Header)	int	متغیر	
۱۹	کد گمرک محل اظهار	scc	سرآمد (Header)	int	متغیر	
۲۰	شناسه یکتای ثبت قرارداد فروشنده	crn	سرآمد (Header)	int	متغیر	
۲۱	شماره اشتراک/شناسه بهره بردار قبض	billid	سرآمد (Header)	int	متغیر	
۲۲	مجموع مبلغ قبل از کسر تخفیف	tpdis	سرآمد (Header)	double	متغیر	
۲۳	مجموع تخفیفات	tdis	سرآمد (Header)	double	متغیر	

ردیف	عنوان قلم اطلاعاتی	JSON	محل قرار گیری در صورتحساب	نوع فیلد	طول فیلد	مقادیر مجاز
۲۴	مجموع مبلغ پس از کسر تخفیف	tadis	سرآمد (Header)	double	متغیر	
۲۵	مجموع مالیات بر ارزش افزوده	tvam	سرآمد (Header)	double	متغیر	
۲۶	مجموع سایر مالیات، عوارض و وجوه قانونی	todam	سرآمد (Header)	double	متغیر	
۲۷	مجموع صورتحساب	tbill	سرآمد (Header)	double	متغیر	
۲۸	روش تسویه	setm	سرآمد (Header)	int	۱	۱-۲-۳ ۱- نقد ۲- نسیه ۳- نقد/نسیه
۲۹	مبلغ پرداختی نقدی	cap	سرآمد (Header)	double	متغیر	
۳۰	مبلغ پرداختی نسیه	insp	سرآمد (Header)	double	متغیر	
۳۱	مجموع سهم مالیات بر ارزش افزوده از پرداخت	tvop	سرآمد (Header)	double	متغیر	
۳۲	عدم پرداخت مالیات بر ارزش افزوده خریدار	dpvb	سرآمد (Header)	int	۱	۱- عدم پرداخت ۲- پرداخت
۳۳	مالیات موضوع ماده ۱۷	Tax17	سرآمد (Header)	double	متغیر	
۳۴	شناسه کالا/خدمت	sstid	بدنه (Body)	int	۱۳	
۳۵	شرح کالا/خدمت	sstt	بدنه (Body)	string	متغیر	
۳۶	واحد اندازه گیری	mu	بدنه (Body)	int	۳	از جدول پیوست ۸ دستورالعمل صدور صورتحساب الکترونیکی
۳۷	تعداد/مقدار	am	بدنه (Body)	int	متغیر	
۳۸	مبلغ واحد	fee	بدنه (Body)	double	متغیر	
۳۹	میزان ارز	cfee	بدنه (Body)	double	متغیر	
۴۰	نوع ارز	cut	بدنه (Body)	string	متغیر	
۴۱	نرخ برابری ارز با ریال	exr	بدنه (Body)	double	متغیر	
۴۲	مبلغ قبل از تخفیف	prdis	بدنه (Body)	double	متغیر	
۴۳	مبلغ تخفیف	dis	بدنه (Body)	double	متغیر	
۴۴	مبلغ بعد از تخفیف	adis	بدنه (Body)	double	متغیر	



ردیف	عنوان قلم اطلاعاتی	JSON	محل قرارگیری در صورت حساب	نوع فیلد	طول فیلد	مقادیر مجاز
۴۵	نرخ مالیات بر ارزش افزوده	vra	بدنه (Body)	double	متغیر	
۴۶	مبلغ مالیات بر ارزش افزوده	vam	بدنه (Body)	double	متغیر	
۴۷	موضوع سایر مالیات و عوارض	odt	بدنه (Body)	string	متغیر	
۴۸	نرخ سایر مالیات و عوارض	odr	بدنه (Body)	double	متغیر	
۴۹	مبلغ سایر مالیات و عوارض	odam	بدنه (Body)	double	متغیر	
۵۰	موضوع سایر وجوه قانونی	olt	بدنه (Body)	string	متغیر	
۵۱	نرخ سایر وجوه قانونی	olr	بدنه (Body)	double	متغیر	
۵۲	مبلغ سایر وجوه قانونی	olam	بدنه (Body)	double	متغیر	
۵۳	اجرت ساخت	consfee	بدنه (Body)	double	متغیر	
۵۴	سود فروشنده	spro	بدنه (Body)	double	متغیر	
۵۵	حق العمل	bro	بدنه (Body)	double	متغیر	
۵۶	جمع کل اجرت، حق- العمل و سود	tcpbs	بدنه (Body)	double	متغیر	
۵۷	سهم نقدی از پرداخت	cop	بدنه (Body)	double	متغیر	
۵۸	سهم ارزش افزوده از پرداخت	vop	بدنه (Body)	double	متغیر	
۵۹	شناسه یکتای ثبت قرارداد حق العملکاری	bsrn	بدنه (Body)	int	متغیر	
۶۰	مبلغ کل کالا/خدمت	tsstam	بدنه (Body)	double	متغیر	
۶۱	شماره سوییچ پرداخت	iinn	(Payment اطلاعات پرداخت)	int	متغیر	
۶۲	شماره پذیرنده فروشگاه	acn	(Payment اطلاعات پرداخت)	int	متغیر	
۶۳	شماره پایانه	trmn	(Payment اطلاعات پرداخت)	int	متغیر	
۶۴	شماره پیگیری	trn	(Payment اطلاعات پرداخت)	int	متغیر	

ردیف	عنوان قلم اطلاعاتی	JSON	محل قرارگیری در صورت حساب	نوع فیلد	طول فیلد	مقادیر مجاز
۶۵	شماره کارت پرداخت کننده صورتحساب	pcn	(Payment) اطلاعات پرداخت )	int	متغیر	
۶۶	شماره/شناسه ملی/کد فراگیر اتباع غیر ایرانی پرداخت کننده صورتحساب	pid	(Payment) اطلاعات پرداخت )	int	متغیر	
۶۷	تاریخ و زمان پرداخت صورتحساب	pdt	(Payment) اطلاعات پرداخت )	Timestamp	۱۴	

نکته قابل توجه در ارسال اطلاعات صورتحساب این است که در صورتی که پارامتری فاقد مقدار باشد ، باید به صورت پیشفرض با مقدار null ارسال شود یا کلاً پارامتر ارسال نگردد.

### ۷-۳-۱-۳ پاسخ درخواست غیرهمگام

پس از دریافت درخواست توسط سرور و بررسی مربوط به لایه انتقال پاسخ مناسب مطابق جداول (۸) و (۹) به کلاینت ارائه می شود.

جدول ۸. مدل داده پاسخ های غیرهمگام

اسم شی	AsyncResponse
نوع فیلدها	<pre>{   "uid": "string",   "referenceNumber": "string",   "errorCode": "string",   "errorDetail": "string" }</pre>

جدول ۹. توضیحات فیلدهای خروجی درخواست غیرهمگام

نام فیلد	توضیحات
uid	شناسه یکتای ارسال صورتحساب، شناسه ای که در سمت کلاینت هنگام ارسال صورتحساب تولید می شود.
referenceNumber	رسید یکتای دریافت صورتحساب، در صورتی که بسته با موفقیت توسط سرور دریافت شود، این شناسه به عنوان کد ارجاع برای کلاینت ارسال خواهد شد. به کمک این فیلد، وضعیت پردازش بسته قابل بررسی و پیگیری است. این کد سمت سرور تولید می شود.

نام فیلد	توضیحات
errorCode	کد خطا، در صورتی که دریافت بسته با خطایی مواجه شود، این فیلد پر شده و کد خطا بازگردانی می شود.
errorDetail	جزئیات خطا، در صورتی که دریافت بسته با خطایی مواجه شود، این فیلد پر شده و جزئیات خطا را شرح می دهد.

## ۷-۳-۲ متدهای همگام

### ۷-۳-۲-۱ سرویس درخواست های همگام

در جدول (۱۰) جزئیات ورودی و خروجی بسته های همگام بیان شده است.

لازم به ذکر است در متدهای همگام امضای درخواست ارسالی نیاز است و داده ها به رمزگذاری نیاز ندارند.

جدول ۱۰. درخواست های همگام

نام متد	ورودی داده	خروجی
GET_TOKEN	شناسه یکتای حافظه یا : username شناسه شرکت معتمد	token طول عمر توکن به ثانیه : expiresIn
GET_FISCAL_INFORMATION	-	نام تجاری : nameTrade وضعیت حافظه : fiscalStatus درصد حد مجاز : saleThreshold فروش شماره اقتصادی : economicCode
INQUIRY_BY_UID	{ Uid: [ { uid, fiscalId }, { uid, fiscalId }, { uid, fiscalId } ] }	{ uid, referenceNumber, status, data, packetType, fiscalId }, ... ]

خروجی	ورودی داده	نام متد
	<pre> }, ... }] </pre>	
<pre> [ {   uid,   referenceNumber,   status,   data,   packetType,   fiscalId }, ... ] </pre>	<pre> {   "referenceNumber":   [     referenceNumber_1,     referenceNumber_2,     referenceNumber_3, ...   ] } </pre>	INQUIRY_BY_REFERENCE_NUMBER
<pre> [ {   uid,   referenceNumber,   status,   data,   packetType,   fiscalId }, ... ] </pre>	time	INQUIRY_BY_TIME
<pre> [ {   uid,   referenceNumber,   status,   data,   packetType,   fiscalId }, ... ] </pre>	startDate: تاریخ شروع بازه مورد نظر endDate: تاریخ پایان بازه مورد نظر	INQUIRY_BY_TIME_RANGE
<pre> {   "serverTime": "زمان سرور",   "publicKeys": [{     "key": "کلید عمومی",     "id": "شناسه کلید عمومی",     "algorithm": "الگوریتم کلید",     "purpose": "هدف کلید. ۱"   }],   "رمز گشایی صورت حسابها در جمع آوری" } </pre>	-	GET_SERVER_INFORMATION

خروجی	ورودی داده	نام متد
<pre> ]] } </pre>		
<pre> {   "result" : [     {       "itemId": "شناسه کالا/خدمت",       "tax": "نرخ مالیات بر ارزش افزوده"     },     "و سایر عوارض"   ],   ... }, "pagination": {   "page": "صفحه جاری",   "size": "تعداد در صفحه",   "total": "تعداد کل", } } </pre>	<pre> {   "filters": [     لیست فیلتر بر روی     داده ها   ],   "orderBy": [     لیست مرتب سازی   ],   "page": "صفحه مورد درخواست",   "size": "تعداد در صفحه", } </pre>	GET_SERVICE_STUFF_LIST
<p>nameTrade: نام تجاری</p> <p>taxpayerStatus: وضعیت مودی</p> <p>taxpayerType: نوع شخص</p> <p>postalcodeTaxpayer: کدهای پستی مودی</p> <p>addressTaxpayer: نشانی مودی</p>	economicCode	GET_ECONOMIC_CODE_INFORMATION

## ۷-۳-۲ پاسخ درخواست های همگام

جدول ۱۱. توضیحات فیلدهای ساختار خروجی درخواست همگام

نام فیلد	توضیحات
uid	شناسه یکتای ارسال صورتحساب، شناسه ای که در سمت کلاینت هنگام ارسال صورتحساب تولید می شود.
packetType	نوع بسته پاسخ.
data	پاسخ رمز شده یا کشف بسته.

نام فیلد	توضیحات
<b>encryptionKeyId</b>	شناسه کلید عمومی حافظه مالیاتی، جهت باز کردن کلید متقارن. در صورتی که داده ها رمز نشده باشند، این فیلد خالی ارسال می شود.
<b>symmetricKey</b>	کلید متقارن رمز شده با کلید رمزنگاری (کلید عمومی سازمان) جهت باز کردن داده های رمز شده.
<b>iv</b>	بردار مقدار اولیه کلید.

### ۷-۳-۲-۳-۳ متد دریافت توکن

کلاينت‌ها با استفاده از اين متد، توکن JWT دریافت می کنند. با توجه به اینکه در این سرویس امضای درخواست کننده اجباری است، احراز هویت از طریق آن صورت می پذیرد. در صورتی که نام کاربری اعلام شده و امضای آن تطابق داشته باشد، توکن JWT برای کلاينت ارسال می شود. این توکن طول عمر مشخصی دارد، با منقضی شدن آن، کلاينت اقدام به دریافت توکن جدید می نمایند. با استفاده از این توکن در سرآیند درخواست‌ها می توان از این API ها استفاده کرد.

برای ارسال توکن ابتدا عبارت Bearer اضافه شده سپس توکن ارسال می شود. نمونه CURL درخواست در پیوست ۲-۲ ارائه شده است.

### ۷-۳-۲-۴ متد استعلام اطلاعات حافظه مالیاتی مودی و حد مجاز فروش مودی

با استفاده از این متد می توان اطلاعات حافظه مالیاتی مودی را دریافت نمود. این متد نیاز به احراز هویت دارد.

نمونه CURL درخواست در پیوست ۳-۲ ارائه شده است.

### ۷-۳-۲-۵ متد استعلام بسته های ارسالی غیر همگام

برای استعلام وضعیت صورتحساب های ارسال شده می توان از متدهای ذیل استفاده کرد: این متدها به احراز هویت نیاز دارند.

- Inquiry\_by\_uid: کلاينت می تواند با آرایه ای از UID ها، صورتحساب های مورد نظر خود را استعلام نماید. در پاسخ وضعیت آنها باز گردانده می شود.

نمونه CURL درخواست در پیوست ۴-۲-۱ ارائه شده است.

نکته: وجود fiscalId در ورودی درخواست کنار هر uid ضروری می باشد.

- Inquiry\_by\_reference\_number: در این متد، کلاینت می تواند با آرایه ای از reference\_number ها، صورتحساب های مورد نظر خود را استعلام نماید. در پاسخ وضعیت آنها باز گردانده می شود.

نمونه CURL درخواست در پیوست ۲-۴-۲ ارائه شده است.

- Inquiry\_by\_time: با این متد صورتحساب های دارای خطا از یک زمان مشخص تا زمان حال مشخص می شوند.

نمونه CURL درخواست در پیوست ۲-۴-۳ ارائه شده است.

- Inquiry\_by\_time\_range: با این متد صورتحساب های دارای خطا در یک بازه زمانی مشخص می شوند.

نمونه CURL درخواست در پیوست ۲-۴-۴ ارائه شده است.

نکته: فیلد "time"، تاریخ شمسی با فرمت YYYYMMDD است و دقت شود فقط در خروجی این درخواست بسته هایی که وضعیت FAILED دارند برگشت داده می شوند.

### ۷-۳-۲-۶ متد دریافت اطلاعات سرور

این متد برای دریافت اطلاعات عمومی سرور شامل کلیدهای عمومی سازمان، شناسه کلید عمومی که از یک مرکز میانی معتبر اخذ شده است و تاریخ و زمان سرور، مورد استفاده قرار می گیرد. این متد به احراز هویت نیاز ندارد.

نمونه CURL درخواست در پیوست ۲-۵ ارائه شده است.

### ۷-۳-۲-۷ متد دریافت لیست کامل شناسه کالا/خدمات و نرخ مالیاتی

این متد برای دریافت لیست شناسه استاندارد کالاها و خدمات ها که توسط وزارت صمت ابلاغ شده است مورد استفاده قرار می گیرد. با توجه به اینکه حجم نتیجه خروجی این متد ممکن است زیاد باشد، امکان دریافت

اطلاعات به صورت صفحه بندی شده نیز در نظر گرفته شده است. این متد نیاز به احراز هویت ندارد.

نمونه CURL درخواست در پیوست ۲-۶ ارائه شده است.

❖ ارسال فیلد فیلتر و مرتب سازی در ورودی اختیاری است. شماره صفحه از یک شروع میشود. برای

مثال اگر ۱۰ رکورد اول را بخواهیم دریافت کنیم ورودی به صورت زیر خواهد بود:

```
{
  "page": 1,
  "size": 10
}
```

### ۸-۳-۲-۸ متد استعلام اطلاعات شماره اقتصادی

این متد برای استعلام شماره اقتصادی مورد استفاده قرار می گیرد. این متد نیاز به احراز هویت ندارد.

نمونه CURL درخواست در پیوست ۲-۷ ارائه شده است.

## ۸- لیست خطاها

### ۸-۱ مدل داده خطاها

جدول ۱۲. ساختار خروجی پاسخ خطا

ErrorResponse	اسم شی
<pre>{   "errorDetail": "string",   "errorCode": "string" }</pre>	نوع فیلد ها



## ۸-۲ لیست خطاهای لایه انتقال

جدول ۱۳. خطاهای لایه انتقال

کد	خطا	توضیحات
۵۰۰۰	internal.server.error	خطای داخلی سرور
۵۰۰۱	bad.request	خطای general زمانی که بسته مشکل داشته و در سمت سرور خطایی به صورت مشخص وجود نداشته باشد
۵۰۰۲	un.authorized	عدم دارا بودن دسترسی برای ارسال این درخواست
۵۰۰۳	uid.format.is.not.valid	ارسال uid با فرمت اشتباه در packet
۵۰۰۴	invalid.json.structure	ساختار JSON درخواست اشتباه است
۵۰۰۵	duplicate.request.uid	ارسال درخواست با uid تکراری در packet
۵۰۰۶	packet.size.is.too.large	ارسال تعداد packet بیش از حد مجاز (در حال حاضر ۱۰۰ عدد)
۵۰۰۷	not.supported.packet-type	عدم پشتیبانی از نوع packet ارسالی
۵۰۰۸	encryption.key.id.not.valid	مقدار فیلد encryptionKeyId در ارسال رمز شده صورت حساب صحیح نمی باشد
۵۰۰۹	not.match.packet-type.with.request	عدم تطابق packet ارسالی با نوع درخواست sync
۵۰۱۰	request.time.has.passed	گذشت زمان مشخصی از زمان ارسال درخواست و دریافت آن توسط سرور
۵۰۱۱	duplicate.request.trace.id	تکراری بودن requestTraceId در سرآیند درخواست
۵۰۱۲	fiscal.id.not.found	پیدا نشدن اطلاعات حافظه مالیاتی ارسالی
۵۰۱۳	invalid.packet.signature	عدم اعتبار امضای درخواست
۵۰۱۴	invalid.format	فرمت ساختار نادرست است

کد	خطا	توضیحات
۵۰۱۵	invalid.token	توکن نامعتبر

### ۳-۸ لیست خطاهای لایه محتوا

جدول ۱۴. کدهای خطای دریافتی از هسته مالیاتی یا به عبارتی کدهای خطای لایه محتوا

ردیف	خطای واقع شده	شرح خطا
۱	عدم ثبت شماره اقتصادی فروشنده.	Seller economic code is empty
۲	عدم ثبت شماره اقتصادی خریدار در صورتحساب‌های الکترونیکی نوع اول.	Buyer economic code is empty
۳	عدم ثبت تاریخ و زمان صدور صورتحساب (میلادی).	Invoice date time is empty
۴	عدم ثبت تاریخ و زمان پرداخت صورتحساب در صورتحساب‌های الکترونیکی نوع سوم.	Payment date time is empty
۵	عدم ثبت سریال صورتحساب.	Invoice number is empty
۶	عدم ثبت نوع صورتحساب.	Invoice type is empty
۷	عدم ثبت الگوی صورتحساب.	Invoice pattern is empty
۸	عدم ثبت موضوع صورتحساب.	Invoice subject is empty
۹	عدم ثبت شماره منحصر به فرد مالیاتی صورتحساب مرجع در صورتی که موضوع صورتحساب شامل اصلاحی، ابطال و یا برگشت از فروش - یکی از مقادیر ۲، ۳ و ۴ باشد.	Reference tax-id is empty
۱۰	عدم ثبت شناسه کالا/خدمت.	Service-stuff-id is empty
۱۱	عدم ثبت مبلغ واحد.	Fee is empty
۱۲	عدم ثبت میزان ارز در صورتحساب‌های با الگوی فروش ارزی.	Currency-fee is empty
۱۳	عدم ثبت نرخ مالیات بر ارزش افزوده.	Vat rate is empty
۱۴	عدم ثبت تعداد/مقدار.	Amount is empty
۱۵	عدم ثبت شناسه یکتای ثبت قرارداد فروشنده در صورتحساب‌های الکترونیکی با الگوی قرارداد پیمانکاری.	Contract registration number is empty
۱۶	عدم ثبت شماره پروانه گمرکی فروشنده در صورتحساب‌های الکترونیکی با الگوی فروش ارزی.	Seller customs license is empty

ردیف	خطای واقع شده	شرح خطا
۱۷	عدم ثبت کد گمرک محل اظهار فروشنده در صورتحساب- های الکترونیکی با الگوی فروش ارزی.	Seller customs code is empty
۱۸	عدم ثبت نوع شخص خریدار.	Buyer type is empty
۱۹	عدم ثبت نوع پرواز در صورتحساب های الکترونیکی با الگوی بلیط هواپیما.	Flight type is empty
۲۰	عدم ثبت نوع ارز در صورتحساب های الکترونیکی با الگوی فروش ارزی.	Currency type is empty
۲۱	عدم ثبت نرخ برابری ارز با ریال در صورتحساب های الکترونیکی با الگوی فروش ارزی.	Exchange rate is empty
۲۲	عدم ثبت شماره اشتراک /شناسه قبض در صورتحساب های الکترونیکی با الگوی قبوض خدماتی.	Billing identification is empty
۲۳	عدم ثبت مبلغ قبل از تخفیف در صورتحساب های الکترونیکی با الگوی اول تا پنجم.	Pre-discount amount is empty
۲۴	عدم ثبت مبلغ تخفیف در صورتحساب های الکترونیکی با الگوی اول تا پنجم.	Discount amount is empty
۲۵	عدم ثبت مبلغ بعد از تخفیف در صورتحساب های الکترونیکی با الگوی اول تا پنجم (فروش-فروش ارزی- صورتحساب طلا، جواهر، پلاتین- قرارداد پیمانکاری-قبوض خدماتی).	After discount amount is empty
۲۶	عدم ثبت مبلغ مالیات بر ارزش افزوده.	Vat amount is empty
۲۷	عدم ثبت سهم مالیات بر ارزش افزوده از پرداخت در صورتحساب های الکترونیکی با الگوی اول تا چهارم (فروش-فروش ارزی- صورتحساب طلا، جواهر، پلاتین- قرارداد پیمانکاری).	Vat of payment is empty
۲۸	عدم ثبت روش تسویه در صورتحساب های الکترونیکی با الگوی اول تا چهارم (فروش-فروش ارزی- صورتحساب طلا، جواهر، پلاتین- قرارداد پیمانکاری).	Settlement method is empty
۲۹	عدم ثبت مبلغ کل کالا/خدمت.	Total service-stuff amount is empty

ردیف	خطای واقع شده	شرح خطا
۳۰	عدم ثبت مجموع مبلغ کل قبل از کسر تخفیف (به جز صورتحساب‌های الکترونیکی با الگوی بلیط هواپیما و صورتحساب‌های نوع سوم).	Total Pre-discount amount is empty
۳۱	عدم ثبت مجموع تخفیفات (به جز صورتحساب‌های الکترونیکی با الگوی بلیط هواپیما و صورتحساب‌های نوع سوم).	Total Discount amount is empty
۳۲	عدم ثبت مجموع مبلغ کل پس از کسر تخفیف (به جز صورتحساب‌های الکترونیکی با الگوی بلیط هواپیما و صورتحساب‌های نوع سوم).	Total After discount amount is empty
۳۳	عدم ثبت مجموع مالیات بر ارزش افزوده.	Total Vat amount is empty
۳۴	عدم ثبت مجموع سایر مالیات، عوارض و وجوه قانونی.	Total other-duty amount is empty
۳۵	عدم ثبت مجموع صورتحساب.	Total bill is empty
۳۶	عدم ثبت مجموع سهم مالیات بر ارزش افزوده از پرداخت.	Total Vat of payment is empty
۳۷	عدم رعایت قالب فایل متنی ارسالی.	JSON file is invalid
۳۸	شماره منحصر به فرد مالیاتی نامعتبر	Invalid tax-id
۳۹	سریال صورتحساب ارسالی نامعتبر	Invalid invoice number
۴۰	شماره منحصر به فرد مالیاتی صورتحساب مرجع نامعتبر	Invalid reference tax-id
۴۱	عدم رعایت مهلت زمانی ابلاغی صدور صورتحساب اصلاحی، ابطالی و برگشت از فروش	Invalid invoice date time
۴۲	تاریخ و زمان صدور صورتحساب نامعتبر (وارد کردن زمان آینده)	Invalid invoice date time
۴۳	نوع صورتحساب نامعتبر	Invalid invoice type
۴۴	الگوی صورتحساب نامعتبر	Invalid invoice pattern
۴۵	شماره اقتصادی فروشنده نامعتبر	Invalid seller economic code
۴۶	شماره اقتصادی خریدار نامعتبر	Invalid buyer economic code
۴۷	عدم تکمیل فیلدهای ضروری مرتبط با الگو	Essential field is empty
۴۸	شناسه یکتای ثبت قرارداد فروشنده نامعتبر	Invalid Contract registration number
۴۹	شناسه کالا/خدمت نامعتبر	Invalid Service-stuff-id
۵۰	واحد اندازه گیری نامعتبر (کدهای آن در جدول واحد وجود نداشته باشد)	Invalid measurement unit

ردیف	خطای واقع شده	شرح خطا
۵۱	نوع ارز نامعتبر (فقط در الگوی فروش ارزی صورتحساب نوع اول و دوم)	Invalid currency type
۵۲	خطا در محاسبه محدوده مجاز ارقام	Error in digit ranges
۵۳	روش تسویه نامعتبر (فقط در صورتحساب الگوهای ۱، ۲، ۳، ۴ و ۶ نوع اول)	Invalid Settlement method
۵۵	موضوع صورتحساب نامعتبر	Invalid invoice subject
۵۶	خطای نوع مقدار وارد شده مغایر با نوع فیلد	Invalid Data type
۵۷	تکراری بودن فیلد «شماره منحصر به فرد مالیاتی صورتحساب»	Duplicate tax id
۵۸	عدم تطابق فیلد «نوع شخص خریدار» با اطلاعات خریدار در سامانه	Mismatch buyer info
۵۹	فیلد «شماره اقتصادی فروشنده» با شناسه حافظه مالیاتی، مغایرت دارد.	Mismatch seller economic code and fiscal Id
۶۰	مقدار فیلد شناسه حافظه مالیاتی (fiscal Id) در صورتحساب با شناسه حافظه مالیاتی موجود در فیلد شماره منحصر به فرد مالیاتی (taxId) تطابق ندارند	Tax id and fiscal Id does not match
۶۱	شماره اقتصادی فروشنده، موجود در اقلام صورتحساب (tins) و شناسه حافظه مالیاتی موجود در فیلد شماره منحصر به فرد مالیاتی taxId تطابق ندارند	Seller Economic code and fiscal Id does not match

## پیوست ۱

## ۱-۱ کد فرمالسازی JSON به زبان java

```
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;

import java.text.Collator;
import java.util.*;

public class CryptoUtils {

    private final static ObjectMapper mapper = new ObjectMapper();

    public static byte[] hexStringToByteArray(String s) {
        int len = s.length();
        byte[] data = new byte[len / 2];
        for (int i = 0; i < len; i += 2) {
            data[i / 2] = (byte) ((Character.digit(s.charAt(i), 16) << 4)
                + Character.digit(s.charAt(i + 1), 16));
        }
        return data;
    }

    public static String normalJson(Object object, Map<String, Object> header) {

        if (object == null && header == null)
            return null;
    }
```

```
Map<String, Object> map = null;

if (object != null) {
    if (object instanceof String) {
        try {
            object = mapper.readValue((String) object, Object.class);
        } catch (JsonProcessingException e) {
            throw new RuntimeException(e.getMessage());
        }
    }

    if (object instanceof Collection) {
        PacketsWrapper packetsWrapper = new PacketsWrapper((Collection) object);
        map = mapper.convertValue(packetsWrapper, Map.class);
    } else {
        map = mapper.convertValue(object, Map.class);
    }
}

if (map == null && header != null) {
    map = header;
}

if (map != null && header != null) {
    for (Map.Entry<String, Object> entry : header.entrySet()) {
        map.put(entry.getKey(), entry.getValue());
    }
}

Map<String, Object> result = new HashMap<>();
```

```
flatMap(result, null, map);

StringBuilder sb = new StringBuilder();
List<String> keys = new ArrayList<>(result.keySet());
Collections.sort(keys, Collator.getInstance(Locale.ENGLISH));
for (String key : keys) {
    String textValue;
    Object value = result.get(key);
    if (value != null) {
        textValue = value.toString();
        if (textValue == null || textValue.equals("")) {
            textValue = "#";
        } else {
            textValue = textValue.replaceAll("#", "##");
        }
    } else {
        textValue = "#";
    }
    sb.append(textValue).append('#');
}
return sb.deleteCharAt(sb.length() - 1).toString();
}

private static String getKey(String rootKey, String myKey) {
    if (rootKey != null) {
        return rootKey + "." + myKey;
    } else {
        return myKey;
    }
}
```



```
}  
  
private static void flatMap(Map<String, Object> result, String rootKey,  
Object input) {  
    if (input instanceof Collection) {  
        Collection list = (Collection) input;  
        int i = 0;  
        for (Object e : list) {  
            String key = getKey(rootKey, "E" + i++);  
            flatMap(result, key, e);  
        }  
    } else if (input instanceof Map) {  
        Map<String, Object> map = (Map) input;  
        for (Map.Entry<String, Object> entry : map.entrySet()) {  
            flatMap(result, getKey(rootKey, entry.getKey()), entry.getValue());  
        }  
    } else {  
        result.put(rootKey, input);  
    }  
}  
  
private static class PacketsWrapper {  
    private Collection packets;  
    public PacketsWrapper() {  
    }  
    public PacketsWrapper(Collection packets) {  
        this.packets = packets;  
    }  
    public Collection getPackets() {  
        return packets;  
    }  
}
```

```
public void setPackets(Collection packets) {
    this.packets = packets;
}
}
}
```

◀ در کلاس CryptoUtils در کد بالا می توانید از متد استاتیک normalJson استفاده کنید.

## ۲-۱ کد DotNet برای نورمالسازی JSON

```
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Org.BouncyCastle.Crypto;
using Org.BouncyCastle.Crypto.Engines;
using Org.BouncyCastle.Crypto.Modes;
using Org.BouncyCastle.Crypto.Parameters;
using Org.BouncyCastle.OpenSsl;
using Org.BouncyCastle.Security;
using System.Security.Cryptography;
using System.IO;
using tax_collect_data_sdk_dotnet;

namespace ir.tax.gov.sdk.util
{
    public class CryptoUtils
```

```

{
    public static byte[] StringToByteArray(string hex)
    {
        return Enumerable.Range(0, hex.Length)
            .Where(x => x % 2 == 0)
            .Select(x => Convert.ToByte(hex.Substring(x, 2), 16))
            .ToArray();
    }

    public static string NormalJson(object obj, Dictionary<string, string>
header)
    {
        if (obj == null && header == null)
            throw new AccessViolationException();

        Dictionary<string, object> map = null;
        if (obj != null)
        {
            if (obj.GetType() == typeof(string))
            {
                if (obj.ToString().Trim().StartsWith("[")
                {
                    obj = ToList<object>((string)obj);
                }
                else
                {
                    obj = JsonConvert.DeserializeObject<object>((string)obj);
                }
            }

            if (obj.GetType().IsGenericType &&
obj.GetType().GetGenericTypeDefinition() == typeof(List<>))
            {

```

```
    PacketsWrapper packetsWrapper = new PacketsWrapper(obj);  
    map = ToDictionary<object>(packetsWrapper);  
}  
else  
{  
    map = ToDictionary<object>(obj);  
}  
}  
if (map == null && header != null)  
{  
    map = new Dictionary<string, object>();  
    foreach (var headerElem in header)  
        map.Add(headerElem.Key, headerElem.Value.ToString());  
}  
if (map != null && header != null)  
{  
    foreach (var headerElem in header)  
        map.Add(headerElem.Key, headerElem.Value);  
}  
Dictionary<string, object> result = new Dictionary<string, object>();  
result =  
JsonHelper.DeserializeAndFlatten(JsonConvert.SerializeObject(map));  
StringBuilder sb = new StringBuilder();  
HashSet<string> keysSet = new HashSet<string>(result.Keys);  
if (keysSet == null || !keysSet.Any())  
{  
    return null;  
}  
var keys = keysSet.OrderBy(x => x).ToList();
```

```
foreach (var key in keys)
{
    string textValue;
    object value;
    if (result.TryGetValue(key, out value))
    {
        if (value != null)
        {
            if (value.Equals(true) || value.Equals(false) ||
value.ToString().Equals("False") || value.ToString().Equals("True"))
            {
                textValue = value.ToString().ToLower();
            }
            else
            {
                textValue = value.ToString();
            }
            if (textValue == null || textValue.Equals(""))
            {
                textValue = "#";
            }
            else
            {
                textValue = textValue.Replace("#", "##");
            }
        }
        else
        {
            textValue = "#";
        }
    }
}
```

```
    }  
    }  
    else  
    {  
        textValue = "#";  
    }  
    sb.Append(textValue).Append('#');  
}  
return sb.Remove(sb.Length - 1, 1).ToString();  
}  
private static string getKey(string rootKey, string myKey)  
{  
    if (rootKey != null)  
    {  
        return rootKey + "." + myKey;  
    }  
    else  
    {  
        return myKey;  
    }  
}  
  
public static Dictionary<string, TValue> ToDictionary<TValue>(object obj)  
{  
    var json = JsonConvert.SerializeObject(obj);  
    var dictionary = JsonConvert.DeserializeObject<Dictionary<string,  
TValue>>(json);  
    return dictionary;  
}
```

```
public static List<Dictionary<string, object>> ToList<TValue>(string obj)
{
    var dictionary = JsonConvert.DeserializeObject<List<Dictionary<string,
object>>>(obj);
    return dictionary;
}
}
```

◀ کد بالا از کلاس دیگری به نام JSONHelper استفاده می کند که در ادامه آورده شده:

```
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;

namespace ir.tax.gov.sdk.util
{
    public class JsonHelper
    {
        public static Dictionary<string, object> DeserializeAndFlatten(string json)
        {
            Dictionary<string, object> dict = new Dictionary<string, object>();

            JToken token = JToken.ReadFrom(new JsonTextReader(new
StringReader(json)));

            FillDictionaryFromJToken(dict, token, "");

            return dict;
        }
    }
}
```

```
private static void FillDictionaryFromJToken(Dictionary<string, object>
dict, JToken token, string prefix)
{
    switch (token.Type)
    {
        case JTokenType.Object:
            foreach (JProperty prop in token.Children<JProperty>())
            {
                FillDictionaryFromJToken(dict, prop.Value, Join(prefix, prop.Name));
            }
            break;
        case JTokenType.Array:
            int index = 0;
            foreach (JToken value in token.Children())
            {
                FillDictionaryFromJToken(dict, value, Join(prefix, index.ToString()));
                index++;
            }
            break;
        default:
            dict.Add(prefix, ((JValue)token).Value);
            break;
    }
}

private static string Join(string prefix, string name)
{
    return (string.IsNullOrEmpty(prefix) ? name : prefix + "." + name);
}
```



}

در کلاس CryptoUtils می توانید از متد NormalJson برای نرمالسازی JSON مدنظر استفاده کنید.

### ۳-۱ روش امضا رشته (string)

۱-۳-۱ کد جاوا

```
public static String getSignedText(String text, String algorithm, PrivateKey
privateKey) throws UnsupportedOperationException, NoSuchAlgorithmException,
InvalidKeyException, SignatureException {
    byte[] data = text.getBytes("UTF8");

    Signature sig = Signature.getInstance(algorithm == null ? "SHA256WITHRSA"
: algorithm);
    sig.initSign(privateKey);
    sig.update(data);
    byte[] signatureBytes = sig.sign();
    return Base64.getEncoder().encodeToString(signatureBytes);
}
```

۲-۳-۱ کد C#

```
public static string SignData(String stringToBeSigned, string
privateKey)
{
    var pem = "-----BEGIN PRIVATE KEY-----\n" + privateKey + "\n-----
END PRIVATE KEY-----";    // Add header and footer
```

```
PemReader pr = new PemReader(new StringReader(pem));
AsymmetricKeyParameter privateKeyParams =
(AsymmetricKeyParameter)pr.ReadObject();
RSAParameters rsaParams =
DotNetUtilities.ToRSAParameters((RsaPrivateCrtKeyParameters)privateKeyParams)
;

RSACryptoServiceProvider csp = new RSACryptoServiceProvider();//
cspParams);
csp.ImportParameters((RSAParameters)rsaParams);

var dataBytes = Encoding.UTF8.GetBytes(stringToBeSigned);
return Convert.ToBase64String(csp.SignData(dataBytes,
HashAlgorithmName.SHA256, RSASignaturePadding.Pkcs1));
}
```

## ۱-۴ کد روش رمزگذاری به روش متقارن AES/ GCM

۱-۴-۱ کد جاوا:

◀ متد نحوه ایجاد کلید متقارن Random:

```
public static SecretKey getAESKey(int keysize) throws
NoSuchAlgorithmException {
    KeyGenerator keyGen = KeyGenerator.getInstance("AES");
    keyGen.init(keysize, SecureRandom.getInstanceStrong());
    return keyGen.generateKey();
}
```

◀ متد ایجاد یک مقدار Random مورد استفاده در رمزنگاری متقارن (IV)

```
public static byte[] getRandomNonce(int byteSize) {
    byte[] nonce = new byte[byteSize];
    new SecureRandom().nextBytes(nonce);
    return nonce;
}
```

◀ نحوه رمزنگاری با استفاده از SecretKey و IV

```
public static byte[] encrypt(byte[] pText, SecretKey secret, byte[] iv)
throws Exception {
    Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding");
    cipher.init(Cipher.ENCRYPT_MODE, secret, new GCMParameterSpec(128, iv));
    return cipher.doFinal(pText);
}
```

◀ کد نحوه XOR کردن دو آرایه

```
public static byte[] xor(byte[] a, byte[] b) {
    int aLen = a.length;
    int bLen = b.length;
    int min = 0;
```

```

int size = aLen > bLen ? aLen : bLen;
byte[] c = new byte[size];

if (size == aLen) {
    min = bLen;
    System.arraycopy(a, min, c, min, size - min);
} else {
    min = aLen;
    System.arraycopy(b, min, c, min, size - min);
}

for (int i = 0; i < min; i++) {
    c[i] = (byte) (a[i] ^ b[i]);
}

return c;
}

```

#### ۱-۴-۲ کد C#

در C# می‌توانید با `byte[]` به صورت کاملاً تصادفی دو کلید IV و Secret Key را ایجاد کنید و از روش زیر متن خود را encrypt کنید:

```

public static string AesEncrypt(byte[] payload, byte[] key, byte[] iv)
{
    var cipher = new GcmBlockCipher(new AesEngine());

    byte[] baPayload = new byte[0];

    cipher.Init(true, new AeadParameters(new KeyParameter(key), 128, iv,
    baPayload));

    var cipherBytes = new byte[cipher.GetOutputSize(payload.Length)];

    int len = cipher.ProcessBytes(payload, 0, payload.Length, cipherBytes, 0);
}

```

```

cipher.DoFinal(cipherBytes, len);
return Convert.ToBase64String(cipherBytes);
}

public static byte[] Xor(byte[] left, byte[] right)
{ /*from w w w . j a v a 2 s . c o m*/
    byte[] val = new byte[left.Length];
    for (int i = 0; i < left.Length; i++)
        val[i] = (byte)(left[i] ^ right[i]);
    return val;
}

```

## ۵- کد روش رمزگذاری به روش نامتقارن RSA-OAEP-SHA256

۱-۵-۱ کد جاوا

```

public static String encrypt(String text, PublicKey publicKey) throws
NoSuchPaddingException, NoSuchAlgorithmException, BadPaddingException,
IllegalBlockSizeException, InvalidKeyException {
    Cipher encryptCipher = Cipher.getInstance("RSA/ECB/OAEPWITHSHA-
256ANDMGF1PADDING");
    encryptCipher.init(Cipher.ENCRYPT_MODE, publicKey);

    byte[] secretMessageBytes = text.getBytes(StandardCharsets.UTF_8);
    byte[] encryptedMessageBytes = encryptCipher.doFinal(secretMessageBytes);
    return Base64.getEncoder().encodeToString(encryptedMessageBytes);
}

```

۱-۵-۲ کد C#

```

public static string EncryptData(String stringToBeEncrypted, string
publicKey)
{
    try
    {

```

```
//var pem = "-----BEGIN PUBLIC KEY-----\n" + publicKey + "\n-  
-----END PUBLIC KEY-----"; // Add header and footer  
AsymmetricKeyParameter asymmetricKeyParameter =  
PublicKeyFactory.CreateKey(Convert.FromBase64String(publicKey));  
RsaKeyParameters rsaKeyParameters =  
(RsaKeyParameters)asymmetricKeyParameter;  
RSAPParameters rsaParameters = new RSAPParameters();  
rsaParameters.Modulus =  
rsaKeyParameters.Modulus.ToByteArrayUnsigned();  
rsaParameters.Exponent =  
rsaKeyParameters.Exponent.ToByteArrayUnsigned();  
RSACng rsa = new RSACng();  
rsa.ImportParameters(rsaParameters);  
  
string base64 =  
Convert.ToBase64String(rsa.Encrypt(Encoding.UTF8.GetBytes(stringToBeEncrypted  
) , RSAEncryptionPadding.OaepSHA256));  
  
if (base64.Length % 4 == 3 )  
{  
    base64 += "=";  
} else if (base64.Length % 4 == 2)  
{  
    base64 += "==";  
}  
return base64;  
}  
catch (Exception e)  
{  
    return "error";  
}  
}
```

## یوست ۲

## ۲-۱ دریافت صورتحساب

- نمونه CURL :

[illegible]

```

oa6QQGZPyad1AKQJmZtR1UKR/1VKgHaKfyqqQGdmgOjm1HUy+5Lm4J911Z8ETqs0sS383T06BNiF
HT1X+ApFn8UbM3gKORiA1e5F8v4I5e7dqTb+et2Ivxee5TEkNXZXJoMGXFrQE/bho0W8A9kGi/ye1
udMcoxcmYsXwXXSdSbn1THstcT7Fa2G31c1w0LDdVsQ20yCK9vGCMqSNU9PMrefUb/3jP/a71d15
ecSN1QDbGdZ03ASkVQ8b356cuW6yuqECbvq1EeI1J1bB9T3wthKRvTcVR1TYvrLzf8ebdjAmysc+h
rgnQ1FuJnwxoX6wXfnw+2ph1KMiaPrLH6p+bnm7+AUiaGu/LuSdBJPzYNQOEIrkJ5KHn9jD
4b3gCQ97B8qspBspvd3mYLLVHQGU+qa4GP5EdoiU76vFzAHSHu+TOTDmKs9NQ7jUwuFlUov3+Fx/M
LhUYkWyDogbwGiyauK5wdgwhUi4AmBR8SwNMG/TxGI dB7xIsMydNOJOctKDJzTK0odvd4d7VkosN
wVzhQw6C9okLJmBblnJbJQtOQZN5MIzXG6MfGkgbJ6g/zFPRKJf7+X9u9Vt83uUCx3ftrmdff/9uN
+5hbVep2gyxmMmiq4Lw2uKg8vyRSzDe7iAckfQWXO2Vstb1U3ZIT2iCzVNGYEGysLLQdVO8yIjn8P
fIuGsytMAQx9iH2Q32rZqhB8vGsJQwBznKywv/u5wYRj1LVEN1dgGoqfXgJfDbdhjBzXLgbbLQ8+
FVi0GfCY88mRRKtkmzLnGJtwTenaNCsfrDDg7Be0/SCzuqij2P0r6W0V2X7zCQWKCThTWUWRu0IM0
nCdRIoshg2U2E/IQ+AI t4oQWjIF0QSy2cY4GUuVPXgwiFNVKv3xkYE3Yyfeg1PhJSKj1Wnth9+V17
/C2IBGxXXVQxoLLgheq6IMLARz6N0bvKX4y0Z2PE0C0fpwYxBxEop6SdV4hFy3mZ1C5Wg4FaccfoG
EJRpfCuYT59cIMF1xquwuMRyRBXrfGGzOdZe/1I4l/snweEqbXG2bXM/24/W0hdD3rR0X0+K+wmso
bkoBUyTamhZDI8uuaHTYx1zQ/SJ9BgaV6C70s+K7jiUjXJgc0yinsmsk0B3yoq5tCqpIfcyZ5hbHw
R7Vqc+2TgtygmQXMjg+Tu301j3J/Xqse6UoFy+RTM0hEmfMyGMmXQeX4X2IZ5fa98hg1Pdsvrm4J
mDvfjDnLCOPLDNeHqvEhKj8p9Kmxwdox61V058UeWwZA5ZH70/Krk+NdFzua1g=",
"encryptionKeyId": "31cf3e11-c468-4b14-90a4-d7c8c7e72fd5",
"symmetricKey":
"cbAKPRW8JJiouZM0zLgiqIgtajUcW1Nr2CoZj6uH4ESMI fJjTgR7kd0j5msAlb1f5vcN21AYtv3
3+UHNipM0bAq30QIF9TnCQgNY+fba1FHQ15Em7/VE+5WYEr+N5uPnRHX/JK2j/A2/jiwwYmi9+Gey
FK2QQYGLUYrFPuW+ifUG1veJldPAocwkD1RxZVXp+hV1L2fKxSwDUCVF5Uvr5jX8ACwmMa8t5qCco
w/xWxumod1BfE9sPOUonOZ1hVINscJYrN4y7MG0Re88F/WRyL4grznetSSK0sAGTrGVf1KZ1Mm3ng
DF+o+8+t1WtVzF5Hj0zv60DAiX3ikS03efMbFtGGQ5WygoEhhWLgt9/CqjkEm7qHE5T0iQ+pE+f1o
TNXfh4eGH/+j9zeI//uLxcHSH/TnTmU1GxlareNXakumtcrKKu1NAw2Mm7F+XBLKXXMfW3iJrTPd3
/hpycPTVpu/6SO/7JMKUA44+irWw5AaX17j4EDWF72AvVDJCP1h2Qw7Krw7u4HFeT3Q115L2vuP/E
TWvFVGbBitP5DbdB9nFoSb8NwKZn6AqGgYaLOVz+8Y8SHgEBueix3V4e5PIm31WTbViqgnANP4TVft
T3ZL3b3yBkjbvQpvc2CtstXgG05C16FzWX16UiR+IYZ7ahOolaTwmZr+GJA87ZLjKaK4=",
"iv": "dee90162e26db3f2246f052f38168ed0", "dataSignature":
"VSvyWrpmT1TwI8I1vUUE9wF0EGNFMZ/2dPgM1tj00vr8agPDHoT0iL5J3XxqvFCauCOfaFs6iVpw
dccwAtXfsFJ4wNQF3Mqr1I9un1v5AEePPEXAU46CK14Ny5sm3WGDtdnezx1+4FBCv05uP07RkeDw
ASyYOySfdW19d6fHes7KNkaLW72NetbTaZWyTQtHx7ib+7cJyG+54eLAY3L6EiUoNiRSI1C1Pn53Q
8AT90uRV8UPSCcpizhg/YINQXrunF2YH/LbTSqckwFsdukg6BGZJbaqPwMfvm2WdPZLNTEW/BgNm
2SmSNwxqivQb1AY2mcfNg06dkyDCRVWTUeA==",
"fiscalId": "AA56CD"
}],
"signature":
"FohMakKq0WRMPQD0nStssK2iP0bbFoJw7w+NVUoWjTdyRFNeKBQeLP9FpL3r5u6EliSKUi03vu1I
vBzEbAaFBuEHSbmyr3q0TN0CpuORDzqHDT/oNk/evvpOoEFt3LvowFLBoxNLDhAGrzd1K56j1Rco9
3U4ip1JgKzFmm9h1mByzTfDEe9FLX/22Gy9regeYtTg9sc12URu7csVttUr8YjCqF XuR8214m0tZd
Rf6jR709YZmskarPXavPsDr8buz7dz0HI0hIeok0vToyG3J1bAVM10SdWREq1Ix0czJ5jLnMiI8N3
1s8rvnypKzo5yWD0jXECTrLfRr+8MgK42Sw=="
}'

```

• نمونه پاسخ درخواست بالا :



```
{
  "signature": "",
  "signatureKeyId": "",
  "timestamp": 1652605524,
  "result": [
    {
      "uid": "8a00f17a-bd35-46bc-ae52-3f61fab868c2",
      "referenceNumber": "967072eb-203e-428e-b9bb-6d2efdb9d356",
      "errorCode": null,
      "errorDetail": null
    }
  ]
}
```

• نمونه پاسخ درخواست در صورت رخداد خطا:

```
{
  "signature": "",
  "signatureKeyId": "",
  "timestamp": 1652605524,
  "result": [
    {
      "uid": "8a00f17a-bd35-46bc-ae52-3f61fab868c2",
      "referenceNumber": null,
      "errorCode": "5008",
      "errorDetail": "invalid.request.uid"
    }
  ]
}
```

## ۲-۲ متد دریافت توکن

• نمونه CURL:

```
curl --location --request POST
'https://tp.tax.gov.ir/req/api/tsp/sync/GET_TOKEN' \
--header 'requestTraceId: 1655185848687' \
--header 'timestamp: 1655185848687' \
--header 'Content-Type: application/json' \
--data-raw '{
  "time": 1,
  "packet": {
    "uid": null,
```



```
"packetType": "GET_TOKEN",
"retry": false,
"data": {
  "username": "test-tsp-id-1"
},
"encryptionKeyId": "",
"symmetricKey": "",
"iv": "",
"fiscalId": "",
"dataSignature": ""
},
"signature":
"IiIdkc1swu3Krc8ZM7MQvEy7ZWzJmBPS11CQrI0dhLGdRPRrmomVo+UkbdzRyuth9G4Enbg0jnjz
5WJcf08MuBVouASTmfV/OC0hAkxTudQtWzU00d6BU/YiRT5a1Nwdey0dMsn3T0831uLv9iG/1KKz9
ewUem0RwBYOnehD6rJFXHirGDfJPHBOTSHCqHL1vQe0JLZAQwaTTieEE8zNWxwNr53BS2KxRKX8+M
leoU18LWUn6wZS/zs3au0KSRs05pgJVq6zZCadd5D7vlhrw1KB/Xf04pv8GexAx2dbRMiGG5eumQG
BcLo1RvJW2mZsGu+dQRm/NwnIpN7CP5q1kg=="
}
```

- نمونه پاسخ درخواست:

```
{
  "signature": null,
  "signatureKeyId": null,
  "timestamp": 1655185988494,
  "result": {
    "uid": null,
    "packetType": "TOKEN_RESULT",
    "data": {
      "token":
        "eyJhbGciOiJIUzUxMiJ9.eyJjbGllbnRUeXBBIjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3LWBlZjQ0tNDRmNC05YWZkLWI5MjdldNDM1ZmI2NyIsInB1cm1pc3Npb25zIjpibInRzcC5hc3luYy5mYXN0LWVucXVldWUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldWUiLCJ0c3Auc3luYyJdLCJwYWNrZXRUeXB1cyI6WyJHRVRFRUNPTk9NSUNFQ09ERV9JTkdPuk1BVElPTiIsIk1OUVVJU1lfQ1lfVElNRSI6IkdFVF9TRVJWRVJfSU5GT1JNQVRJT04iLCJJTlFVSUVjZX0jZX1VJRCIsIk1OUVVJU1lfQ1lfQ1fuKGRVJFTkNFUX05VTUJFUiiSiIkdFVF9UT0tFTiIsIkdfVF9GSVNDAQ1fSU5GT1JNQVRJT04iLCJJTlZPSUNFLlYwMSIsIk1OUVVJU1lfQ1lfVElNRV9SQU5HRSIsIkdfVF9TRVJWSUNFX1NUVUZGX0xJU1QiXSwiaXNzIjoiveFEFYIE9yZ2FuXphdGlubiIsImlkIjoidGVzdC10c3AtawQtMSIsImV4cCI6MTY1NTIiwMDM0OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0VlX7oqgVITzy5S3kseSHSbdrfwiaYxDt1mMn541TV8zP-HowhBs8u1fjs_S81kiZON4FA",
```

```

    "expiresIn": 1655200388492
  },
  "encryptionKeyId": null,
  "symmetricKey": null,
  "iv": null
}
}

```

## ۲-۳ استعلام اطلاعات حافظه مالیاتی مودی و حد مجاز فروش مودی

- نمونه CURL :

```

curl --location --request POST
'https://tp.tax.gov.ir/req/api/tsp/sync/GET_FISCAL_INFORMATION' \
--header 'requestTraceId: 1655188530459' \
--header 'timestamp: 1655188530459' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJjbG1bnRUeXB1IjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3LT
BlZjQtNDRmNC05YWZkLWI5Mjd1NDM1ZmI2NyIsInBlcm1pc3Npb25zIjpbInRzcC5hc3luYy5mY
XN0LWVucXVldWUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldWUiLCJ0c3Auc3luYyJdLCJwYWNr
ZXRUeXB1cyI6WyJHRVRFRUNPTk9NSUNfQ09ERV9JTkZPUk1BVElPTiIsIk1OUVVJU1lfQ1lfVE1
NRSIsIkdfVF9TRVJWRVJfSU5GT1JNQVRJT04iLCJ0c3Auc3luYyJdLCJwYWNrZXRUeXB1cyI6WyJ
HRSIsIkdfVF9TRVJWSUNFX1NUVUZG
X0xJU1QiXSwiaXNzIjoiVEFYIE9yZ2FuaXphdGlvbiIsImklIjoidGVzdC10c3AtaWQtMSIsImV
4cCI6MTY1NTIwMDM0CwiY3JlYXR1RGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0V1
X7oqgVITzy5S3ksehSHSbdrfwiaYxDt1mMn541TV8zP-HowhBs8u1fjs_S81kiZON4FA' \
--header 'Content-Type: application/json' \
--data-raw '{
  "time": 1,
  "packet": {
    "uid": null,
    "packetType": "GET_FISCAL_INFORMATION",
    "retry": false,
    "data": null,
    "encryptionKeyId": "",
    "symmetricKey": "",
    "iv": "",
    "fiscalId": "AA56CD",
    "dataSignature": ""
  },
  "signature":
"WjL0C+zYwHyJ0xZCnftGw59tuZ9jUyJPLFWJq7j8KOZ1VkJFrGuSaxFXukndE0NCf3jKHFGBUqu
Fe9jGCKtGBRz6b+oBycZJtH4BcwiyTkrng+nLfduBEc6bhRA48ZiKBBNAM/8RRSRj8tC0AoStj7
90gdw10SkS1yYZ4t0KRfxV12a21wGWSF5ahsy3LQptloKUjZTrHt6UtVU2WwIPnFkb4p6P/pw2

```



```
YjA8YfcHCdn/WqrCFvA4YC+09cv1qwRdxk8tEfxyziczStRrd5Jq8UVZDBjXJIwel7GTeSWI9+h
bCqcpAeHhK8cbUi7grdVkBgXucBlVE6Lfam+/SqBD0A=="
}'
```

- نمونه پاسخ سرویس بالا:

```
{
  "signature": null,
  "signatureKeyId": null,
  "timestamp": 1655188711025,
  "result": {
    "uid": null,
    "packetType": "FISCAL_INFORMATION",
    "data": {
      "nameTrade": "AA56CD",
      "fiscalStatus": "ACTIVE",
      "saleThreshold": 50.0,
      "economicCode": "12345678911234"
    },
    "encryptionKeyId": null,
    "symmetricKey": null,
    "iv": null
  }
}
```

## ۲-۴ دریافت پاسخ بسته‌های ارسالی غیرهمگام (استعلام)

۲-۴-۱ استعلام از طریق UID (Inquiry\_by\_uid):

○ نمونه CURL :

```
curl --location --request POST \
'https://tp.tax.gov.ir/req/api/tsp/sync/INQUIRY_BY_UID' \
--header 'requestTraceId: 1655186523487' \
--header 'timestamp: 1655186523487' \
--header 'Authorization: Bearer eyJhbGciOiJIUzUxMiJ9.eyJjbGllbnRUeXBBIjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3LTBlZjQtNDRmNC05YWZkLWI5Mjd1NDM1ZmI2NyIsInBlcm1pc3Npb25zIjpjbWVudC5hc3luYy5mYXN0LWVucXVldwUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldwUiLCJ0c3Auc3luYyJdLCJwYWNrZXRUeXBlc3I6WyJHRVRfRUNPTk9NSUNfQ09ERV9JTkJkPUk1BEVEPTiIsIk1OUVVJU1lfQ1lfVE1NRSI5IkdfVFV9TRVJRVRJfSU5GT1JNQVRJT04iLCJJTlFVSVMjZDQ3ZX0JZXRJVCIsIk1OUVVJU1lfQ1lfVE1fFUKVGRVJFTkNF05VTUJFU1IsIkdfVFV9UT0tFTIIsIkdfVFV9GSVNDDQxfSU5GT1JNQVRJT04iLCJJTlZPSUNFLlYwMSIsIk1OUVVJU1lfQ1lfVE1NRV9SOU5HRSIsIkdfVFV9TRVJSUNFX1NUVUZG
```

```
X0xJU0iXswiaXNZjoiVEFYIE9yZ2FuaXphdGlvbiIsImlkIjoidGVzdC10c3AtaWQtMSIsImV
4cCI6MTY1NTIwMDM0OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0Vl
X7oqgVITzy5S3ksehSHSbdrfwiaYxDt1mMn541TV8zP-HowhBs8u1fjs_S81kiZON4FA' \
--header 'Content-Type: application/json' \
--data-raw '{
    "time": 1,
    "packet": {
        "uid": null,
        "packetType": "INQUIRY_BY_UID",
        "retry": false,
        "data": [
            {"uid":"3b33584c-331e-4fbc-97ea-8d294a2009db","fiscalId":
"AA56CD"}
, {"uid":"8d595073-6349-423e-847f-83ae8a126394","fiscalId": "AA56CD"}
, {"uid":"c0b286fd-b30f-49d7-97ca-640e91652f6f","fiscalId": "AA56CD"}
],
        "encryptionKeyId": "",
        "symmetricKey": "",
        "iv": "",
        "fiscalId": "",
        "dataSignature":""
    },
    "signature":
"Vny+goEh3eDH1HCijyvD7LLvUi+iAJEmgk18jc1yp73SrNHZ0RHaFbsgXICN5F52zV8HPfa3YI
cddBi3ZjVNVs9RA+HsEHolavOecPLo9A0IGS7y+BtvV1FBKLTizbflpejprrrSvN78ledJQmEsjg
MMiVZA1zexQ8HU73/kf3fIyaviXXVyVcwM7DFqiyVdVV4IX3klJYVzyamU08PD19zTcwQMxwiSd
QiQ/pLztX0wzP2UTyLmy47jskdJVjpDseM1iSEt6fy+18R98iqvBFiKi5njo50Nr+yTHQuStfWf
X6I2uhYNML6/aC8EBYFbTjVuJXyIOrM7CqnP77SbavA=="
}'
```

۲-۴-۲ استعمال با رسید بکتابی دریافت صورت حساب (Inquiry\_by\_reference\_number):

- نمونه ارسال درخواست CURL:

[illegible]



```
TkNFX05VTUJFUiiIsIkdFVF9UT0tFTiIsIkdFVF9GSVNDQUxfSU5GT1JNQVRJT04iLCJTTlZPSUNFL
lYwMSIsIk1OUVVJU1lfQ1lfVE1NRV9SQU5HRSIIsIkdFVF9TRVJWSUNFX1NUVUZGX0xJU1QiXSwiaX
NzIjoiveFEYIE9yZ2FuaXphdGlvbiIsImlkIjoidGVzdC10c3AtaWQtMSIsImV4cCI6MTY1NTIwMDM
4OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0V1X7oqgVITzy5S3ksehS
HSbdrfwiaYxDT1mMn541TV8zP-HowhBs8u1fjs_S81kiZON4FA' \
--header 'Content-Type: application/json' \
--data-raw '{
  "time": 1,
  "packet": {
    "uid": null,
    "packetType": "INQUIRY_BY_TIME",
    "retry": false,
    "data": {
      "time" :
      14010321
    },
    "encryptionKeyId": "",
    "symmetricKey": "",
    "iv": "",
    "fiscalId": "",
    "dataSignature":""
  },
  "signature":
  "e1HDykQjZBzgM9NIMSEVR+dYfDcAbowWF00vEzI31WRuBwMVmmmYkXg6b4q6oIj+78Cgwd106Jo4
5zuaH1eNSxVc67xr7hagpnyvnVQYt2ot3Y5Rk+MJ0BkyfE06qv45yzV2kwe5CkfCAH2ccvFnnkEGd
o15CoaAyJk51hWNTppGmOdXAwoEoUE2gjDRZqtRsgzMOmum5ep7GoPQ/8WldwNgxuVRLaWiQNmZs
EuBDVLgJsY7xFMGczFCyZt1NwZwV1FVTDZVmn8XNDC1u7e0IRX/krnF932EAxWiWG5zidyr/geXyz
gpmFevQQbe10eedCuLcFVT+GrNGfJUuzakw=="
}'
```

۴-۴-۲ دریافت خطاهای بسته‌های ارسالی غیرهمگام با استفاده از بازه زمانی (INQUIRY\_BY\_TIME\_RANGE)

• نمونه CURL:

۲-۴-۵ نمونه پاسخ درخواست های استعلام:

صفحه ۶۴؛ ۶۹



```

"uid": null,
"packetType": "INQUIRY_RESULT",
"data": [
  {
    "referenceNumber": "967072eb-203e-428e-b9bb-6d2efdb9d356",
    "uid": "8a00f17a-bd35-46bc-ae52-3f61fab868c2",
    "status": "SUCCESS",
    "data": {
      "confirmationReferenceId": "d4c0e7e6-d42e-11ec-9d64-0242ac120002",
      "taxResult": "SUCCESS"
    },
    "packetType": "RECEIVE_INVOICE_CONFIRM",
    "fiscalId": "SAU5BC"
  },
],
"encryptionKeyId": null,
"symmetricKey": null,
"iv": null
}

```

نمونه پاسخ درخواست های استعلام در صورت رخ دادن خطا:

```

{
  "uid": null,
  "packetType": "INQUIRY_RESULT",
  "data": [
    {
      "referenceNumber": "6df7b185-5254-4f58-b3e6-61169b5d5929",
      "uid": "3b33584c-331e-4fbc-97ea-8d294a2009db",
      "status": "FAILED",
      "data": {
        "confirmationReferenceId": null,
        "taxResult": "memory-id.is.null"
      },
      "packetType": "ERROR",
      "fiscalId": null
    }
  ],
  "encryptionKeyId": null,
  "symmetricKey": null,
  "iv": null
}

```

## ۲-۵- دریافت اطلاعات سرور (GET\_SERVER\_INFORMATION)

- نمونه CURL:

```
curl --location --request POST
'https://tp.tax.gov.ir/req/api/tsp/sync/GET_SERVER_INFORMATION' \
--header 'requestTraceId: 1654938179880' \
--header 'timestamp: 1654938179880' \
--header 'Content-Type: application/json' \
--data-raw '{
  "time": 1,
  "packet": {
    "uid": null,
    "packetType": "GET_SERVER_INFORMATION",
    "retry": false,
    "data": null,
    "encryptionKeyId": "",
    "symmetricKey": "",
    "iv": "",
    "fiscalId": "",
    "dataSignature": ""
  }
}'
```

- نمونه پاسخ درخواست:

```
{
  "signature": null,
  "signatureKeyId": null,
  "timestamp": 1655184692934,
  "result": {
    "uid": null,
    "packetType": "SERVER_INFORMATION",
    "data": {
      "serverTime": 1655184692934,
      "publicKeys": [
        {
          "key":
"MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAXdzREOEfk3vBQogDPGTMqdDQ7t0oDhuK
MZkA+Wm1lhzzjhAGfSU0uDvOKRoUEQWP8oUcXRmYzcvCUGcfoRT5iz7HbovqH+bIeJwT4rmLmFcbf
Pke+E3DLUXOtIZifEXrKXWgSVPkRnhMgym6UiAtnzwA1rmKstJoWpk9Nv34CYgTk8DKQN5jQJqb9L
/Ng0zOEEtI3zA424tsd9zv/kP4/SaSnbbnj0evqsZ29X6aBypvnTnwH9t3gbWM4I9eAVQhPYC1awH
TqvdaZ/O/feqfm06QBFnCGl+CBdjLs30xQSLsPICjn1V1jMzoTZnAabWP6FRzzj6C2sxw9a/Ww1Xr
Kn3gldZ7Ctv6Jso72cEeCeUI1tzHMDJPU3Qy12RQzaXuJpMhCz1Dva47RvqiumpTnyK9HFFIdhgou
pFkxT14XLD16S55MF6HuQvo/RHSbBJ93FQ+2/x/Q2MNGB3BX0jNwM2pj3ojbDv3pj9CHzvaYQUYM
1yOcFmIJqJ72uvVf9Jx9iT0baNNF6p152Admh85GTAH1hz+4pR/E9IAXUI1/YiUneYu0G4tiDY4ZX
```

```

ykYNknNfhSgxmn/gPHT+7kL31nyxgjiEEhK0B0vagWvdRCNJSNGWpLt1q4F1CWTAnPI5ctiFgq925
e+sySjNaORCoHraBXNEwyiHT2hu5ZipIW2cCAwEAAQ==",
      "id": "6a2bcd88-a871-4245-a393-2843eafe6e02",
      "algorithm": "RSA",
      "purpose": 1
    }
  ]
},
"encryptionKeyId": null,
"symmetricKey": null,
"iv": null
}
}

```

۲-۶ دریافت لیست کامل شناسه کالا/خدمات و نرخ مالیاتی (GET\_SERVICE\_STUFF\_LIST)

• نمونه CURL:

```

curl --location --request POST 'https://tp.tax.gov.ir/req/api/tsp/sync/
GET_SERVICE_STUFF_LIST \
--header 'requestTraceId: 1654521784527' \
--header 'timestamp: 1654521784527' \
--header 'Content-Type: application/json' \
--data-raw '{
  "time": 1,
  "packet": {
    "uid": null,
    "packetType": "GET_SERVICE_STUFF_LIST",
    "retry": false,
    "data": {
      "page" : 1,
"size" : 10
    },
    "encryptionKeyId": "",
    "symmetricKey": "",
    "iv": "",
    "fiscalId": "",
    "dataSignature": ""
  }
}'

```

• نمونه پاسخ:

```

{
  "signature": null,

```

```

"signatureKeyId": null,
"timestamp": 1655190910633,
"result": {
  "uid": null,
  "packetType": "SERVICE_STUFF_LIST",
  "data": {
    "result": [
      {
        "tax": 9,
        "itemId": "1"
      }
    ],
    "pagination": {
      "size": 10,
      "page": 1,
      "total": 1
    }
  },
  "encryptionKeyId": null,
  "symmetricKey": null,
  "iv": null
}
}

```

۷-۲ استعلام اطلاعات شماره اقتصادی (GET\_ECONOMIC\_CODE\_INFORMATION) :

نمونه CURL:

```

curl --location --request POST
'https://tp.tax.gov.ir/req/api/tsp/sync/GET_ECONOMIC_CODE_INFORMATION' \
--header 'requestTraceId: 1654521302712' \
--header 'timestamp: 1654521302712' \
--header 'Content-Type: application/json' \
--data-raw '{
  "time": 1,
  "packet": {
    "uid": null,
    "packetType": "GET_ECONOMIC_CODE_INFORMATION",
    "retry": false,
    "data": {
      "economicCode": "12345678911234"
    }
  },
}
'

```

```

    "encryptionKeyId": "",
    "symmetricKey": "",
    "iv": "",
    "fiscalId": "",
    "dataSignature": ""
  }
}'

```

• نمونه پاسخ:

```

{
  "signature": null,
  "signatureKeyId": null,
  "timestamp": 1655191021181,
  "result": {
    "uid": null,
    "packetType": "ECONOMIC_CODE_INFORMATION",
    "data": {
      "nameTrade": "پیشخوان دولت",
      "taxpayerStatus": "فعال",
      "taxpayerType": "حقوقی",
      "postalcodeTaxpayer": "",
      "addressTaxpayer": "تهران"
    },
    "encryptionKeyId": null,
    "symmetricKey": null,
    "iv": null
  }
}

```