سازمان امور مالیاتی کشور

نظام پایانه‌های فروشگاهی و سامانه مودیان

«دستورالعمل 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

فهرست شکل‌ها

فهرست جداول

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 صورتحساب‌های 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 غیرهمگام از صف استفاده می‌شود.

در کلیه مراحل استفاده از API، اصول امنیت اطلاعات شامل احراز هویت و سطح دسترسی ارسال‌کننده، 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 صورتحساب‌های ارسالی به صورت گام به گام و همراه با جزئیات فنی مورد نیاز برای پیاده‌سازی آن توسط کلیه ذی‌نفعان ارائه شده است.

۳– دامنه کاربرد

ذینفعان این سند شامل:

- اشخاص مشمول (مودیان).
- شرکت‌های معتمد ارائه‌کننده خدمات مالیاتی نوع اول

۴– تعاریف

- **شناسه یکتای حافظه مالیاتی:** شناسه‌ای است یکتا دارای مقداری ثابت و منحصر به‌فرد که به هر حافظه مالیاتی در سطح کشور اختصاص داده می‌شود، این شناسه از مولفه‌های تشکیل دهنده شماره

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

- **شماره منحصر به‌فرد مالیاتی:** شماره‌ای است یکتا در سامانه مودیان که به ازای هر صورتحساب تولید شده و به‌صورت منحصر به‌فرد به آن صورتحساب تخصیص داده می‌شود. این شماره دارای بخش‌های 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 می‌شود.

مشخصات و اقلام اطلاعاتی صورتحساب الکترونیکی[1]، 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 صورتحساب‌های الکترونیکی و انتقال آن به سامانه مودیان مورد استفاده قرار می‌گیرد. حافظه مالیاتی می‌تواند به شکل سخت‌افزاری یا نرم‌افزاری باشد. حافظه مالیاتی، توسط مودی برای ثبت صورتحساب الکترونیکی مورد استفاده قرار می‌گیرد. هر حافظه مالیاتی باید دارای شماره شناسه یکتا باشد. شناسه یکتای حافظه مالیاتی توسط سازمان اختصاص داده می‌شود.
- **پایانه فروشگاهی:** رایانه، دستگاه کارتخوان بانکی، درگاه پرداخت الکترونیکی یا هر وسیله دیگری که امکان اتصال به شبکه‌های الکترونیکی پرداخت رسمی کشور و سامانه مودیان را داشته و از قابلیت صدور صورتحساب الکترونیکی برخوردار باشد(بند ب ماده ۱ قانون).
- **شرکت‌های معتمد ارائه کننده خدمات مالیاتی:** اشخاص حقوقی دارای پروانه هستند که حسب ضوابط و دستورالعمل‌های ابلاغی سازمان، نسبت به ارائه مشاوره و آموزش‌های لازم به مودیان، نصب

[1] 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:** زیرسامانه‌ای است که داده‌های ارسالی از پایانه‌های فروشگاهی-حافظه مالیاتی یا شرکت‌های معتمد ارائه کننده خدمات مالیاتی نوع اول را از طریق واسط‌های نرم‌افزاری دریافت می‌کند.
- **حد مجاز فروش:** جمع صورتحساب‌های الکترونیکی صادره توسط هر مودی در هر دوره مالیاتی نمی‌تواند 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 صورتحساب‌های الکترونیکی صادره شده در هر دوره مالیاتی برای واحدهای جدیدالتاسیس یا واحدهای فاقد سابقه مالیاتی نمی‌تواند بیش از سه برابر معافیت سالانه موضوع ماده (۱۰۱) قانون مالیات‌های مستقیم باشد(ماده ۶ قانون پایانه‌های فروشگاهی و سامانه مودیان).
- **شناسه یکتای ارسال صورتحساب**[۲]: شناسه‌ای یکتا که هنگام ارسال صورتحساب توسط ارسال کننده به صورتحساب جهت رهگیری آن اختصاص داده می‌شود. این شناسه با فرمت uuid[۳] به صورت رمز نشده ارسال می‌گردد.
- **رسید یکتای دریافت صورتحساب**[۴]: هنگام دریافت صورتحساب در زیرسامانه جمع آوری و پردازش اطلاعات، یک شناسه رسید یکتا با فرمت uuid به هر صورتحساب توسط این زیرسامانه اختصاص داده می‌شود.
- **لیست صورتحساب‌ها:** مجموعه‌ای از صورتحساب‌های صادر شده توسط پایانه فروشگاهی-حافظه مالیاتی که در قالب یک مجموعه به سامانه مودیان ارسال می‌شود.
- **کلاینت:** ارسال کننده درخواست به API زیرسامانه جمع آوری و پردازش اطلاعات که می‌تواند مودی یا شرکت معتمد ارائه کننده خدمات مالیاتی باشد.
- **سرور:** سرور سامانه مودیان

2 uid

3 Universal unique identificatier

4 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 بخش مراکز میانی، قابل دسترسی است.

۵- پیکربندی پایانه فروشگاهی-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 می‌بایست 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 ارائه‌کننده خدمات مالیاتی (به روش غیرمستقیم)
   - ارسال اطلاعات صورتحساب توسط شرکت معتمد ارائه‌کننده خدمات مالیاتی (به روش غیرمستقیم و با استفاده از زیرساخت‌های اختصاصی)
۲. کلید عمومی (RSA) دریافتی از مراکز میانی معتبر با طول کلید ۲۰۴۸ بیت را بارگذاری نماید.

نکته: در صورتی‌که 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.

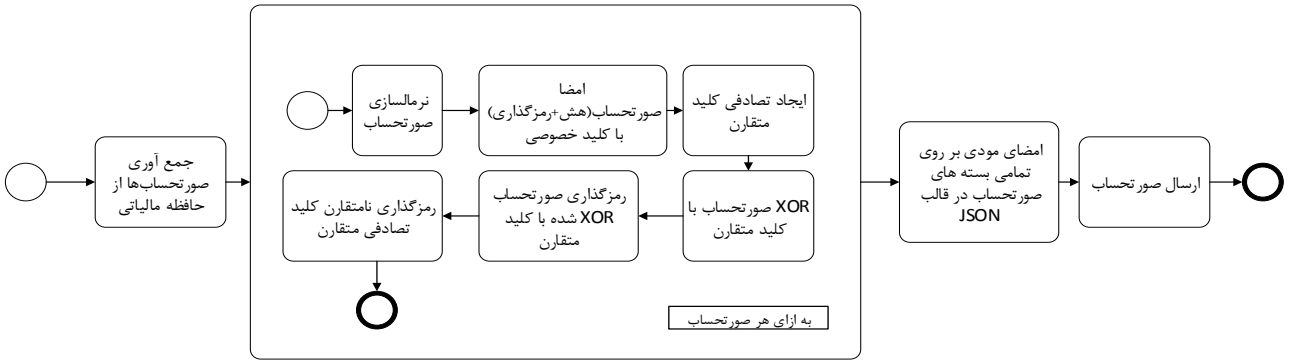
۳. ارتباط شناسه یکتای حافظه مالیاتی درخواستی با کدپستی(های) محل فعالیت تعیین گردد.

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

مکانیزم‌های امنیتی جهت ارسال صورتحساب مطابق با نمودار ارائه شده در شکل(۱) می‌باشند.



شکل ۱.نمودار آماده سازی صورتحساب جهت ارسال

۶-۱ توکن

با هدف احراز هویت و تعیین سطح دسترسی ارسال‌کننده اطلاعات، از مکانیزم تخصیص توکنJWT[5] استفاده شده است. لازم است توکن تخصیص داده‌شده به مودی یا شرکت معتمد ارائه‌کننده خدمات مالیاتی نوع اول در Header تمامی درخواست‌های ارسال شده مودی به API های زیرسامانه جمع آوری و پردازش اطلاعات قرار گیرد.

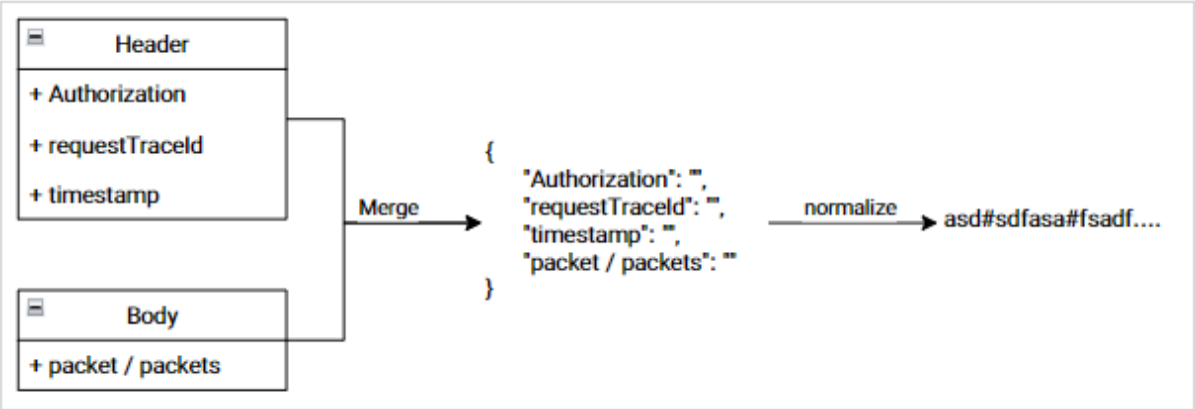
[5] Json Web Token

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 جمع‌آوری اطلاعات، برای تمامی بسته-های 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 مقدارها 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 دست‌کاری نشده و با همان ترتیب در نظر گرفته می‌شود.
- در صورتی که ریشه JSON آرایه باشد، داخل فیلد packets قرارگرفته و تبدیل به شی می‌شود.

روش پیاده‌سازی نرمالسازی JSON در پیوست‌های 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 نرمال‌سازی بیان شد نرمال شوند. به عنوان مثال اطلاعات JSON صورتحساب زیر را در نظر بگیرید:

```
{
  "header" :
  {
   "taxid": "AA56CD0E0620002F2B4E78",
   "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,
      "bros" : 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 می‌باشد.

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
شیبابا###۱۰۹۰۰۰۰#۹۰۰۰۰#۰,۰۹#۹۰۰۰۰###############۹۰۰۰۰######۱۶۵۵۶۲۰۸۲۱۲۷۴#۱
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 هش و امضا می‌شود و خروجی آن در فیلد dataSignature در شی packet قرارمی گیرد.

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

۳-۶ نحوه رمزگذاری صورتحساب

برای رمزگذاری صورتحساب می‌بایست یک کلید متقارن (به روش AES/GCM/NoPADDING) با طول ۲۵۶ بیت تولید شود. برای رمزگذاری از طریق AES/GCM نیاز به یک کلید دیگر به نام IV به طول ۱۲۸ بیت می‌باشدکه 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 رمزگذاری می گردد.

جزئیات XOR به این شکل است که باید متنی که می‌خواهیم 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
RQIdhXcx6SMe+UF/XE4CJxMgmjfFMvBBjM34ix2ZFIaR2v2rwZ6XLUPTHWK
HlXROxnA10BagrgOm0Xr4J6Y359ZZpnaNChPabj7s4yE4ZGT+wBHoz+J/yj2xFW
KWmmcuP1b6rWOdWYO2OD6eXI1+faxT2vZ7FSBYyDy+sjP5S6Zllp92baFHX7
Z78VVqVGv+P+iJ+CLljD6j75k19ykD7stVkvHsoMwWYndqNiBIRLBXOjczSkJ/y
L89eiGKcAUvEAGSiXgaeKjEymLaOWXZEdg0IwxVUrgRyX+4kVbWeCUWHSb
jafCdQNRUddFZLWs3bO6MA552gaAh9iXJGmnJNvCbWjpjTvx9NyDbvd3thZhFl
dhX5dkVEi7mWv3iTywRCUN/6QkOxZ4JDG7u3i20vqztuqxq/Kj/51GMOGov2hhF
jd+5c6QCkvO7COMX4XVO6azrWGa3VVyZz6dG9vCvVb/BZsVTxGBddQiQKd5
aaQxxIvs+sM7/FIbQ4l5llSN+4ax8g420qO5E6hA+mMQ3oECnJtyKJmDK6oR6T5y
//5dByFffMySmhdx+bhS0XxMFvk7ZPUcWFZwDGs6yNKa2AApt58xqGnbuf+N2
86UjPH9CgNogY/G5lvFlPX51qi6q6X/YX165WhD0MbVEL8EEyfk/60rjwLivhEkc
ZWowhyOBnxtnHvVmpjocTS/404Oskepra/4jRKiqON3hqwpEJE8V0hanYwp9m32
HtYbiql9RqL3wHycbWtnueFNjptplKi4tczl4NsfD9ezD6TQVgcAT0o14OAnklJTO/
0QnwfTgv8krEy7F/umNYWPswgWrjqSHtGiUHbXLunGP6fwZZOT0vECwXrz9X/

aHGX/rm64/orJArZ9Vi2OIdzvGdQbcXsGAKKuAlCvssA4XnXMBvWW9+g4Y+q
qsHBTomkvHYrXvLob02FUWL0XsTkap/BXfLIj4HBQUhh0gUk6aX8wetRknFiH
HjblgjN7Tjc51Jt1GifTZHnG/uHIJB/6/GlwIbxAPXB4YfNiRAU0y0iCiU1Ow8Jl6Tu
us3cgGkLvo5uOlF/0vLx9O2BSJHcpA0OBhIjaJ1bdgHSYOr3P3tDY+BVSe33/8oO
0DAVBjEDUQKx3F0Ncr6TwxlqewquIHL2zmktSMc2KbIFY12Jx7cG84hh4Ih98Vi
Fg+Xad83Ed3G/JiU3uG+7qiaYi8DYI6Icw0rf870qQ/CkopVR44xRtePslZfZjKS0X5
39/4qymPDs3+KFYzGntIiwc6fRNSLhjoTidCHFRcF9tP2Ttkp6CTsnLsHtQugYZatn
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 رمزشده قرار گیرد. برای رمزگذاری کلید متقارن باید از روش نامتـــــقارن RSA-OAEP-SHA256 استفاده شودکه برای این منظور از کلید/های عمومی سازمان با طول ۴۰۹۶ بیت اخذ شده از یک مرکز میانی معتبر استفاده می‌شود (کلید عمومی سازمان با استفاده از متد GET_SERVER_INFORMATION به دست می‌آید).

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

۷- فراخوانی متدهای API جمع‌آوری اطلاعات سامانه مودیان

۷-۱ آدرس API ها

آدرس APIهای زیرسامانه جمع‌آوری و پردازش اطلاعات به‌صورت زیر است که پیشوند تمام آدرس‌ها قرار می‌گیرد.

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

آدرس‌هایی که با tsp شروع می‌شوند برای شرکت‌های معتمد ارائه کننده خدمات مالیاتی در نظر گرفته شده و آدرس‌هایی که با self-tsp شروع می‌شوند 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 /

۲-۷ 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 غیرهمگام 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 غیرهمگام می‌توان مجموعه‌ای از بسته‌ها (packets) را ارسال نمود.

جدول ۳.ساختار کلی درخواستها

عنوان	مقدار
متد HTTP	POST
فیلدهای Header	**Authorization: "string"** **requestTraceId: "string"** **timestamp: "Long"**

فیلدهای Body	{
	"packets": **Packet[]**,
	"signature": **"string"**,
	"signatureKeyId": **"string"**
	}

فیلدهای خروجی	{
	"timestamp": **"Long"**,
	"result": **AsyncResponse[]**,
	"signature" : **"string"**,
	"signatureKeyId" : **"string"**
	}

خروجی سرویس در صورت رخداد خطای کلی	{
	"timestamp": **"Long"**,
	"errors": **ErrorResponse[]**,
	"signature" : **"string"**,
	"signatureKeyId" : **"string"**
	}

- 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
Authorization	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 غیرهمگام 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

نوع فیلدها	

	{
	"uid": "string",
	"packetType": "string",
	"retry": "Boolean",
	"data": "string",
	"encryptionKeyId": "string",
	" symmetrickey ": "string",
	"iv": "string",
	"fiscalId": "string",
	"dataSignature": “string”
	}

جدول ۶. توضیحات فیلدهای بسته(packet)

نام فیلد	توضیحات
uid	شناسه یکتای ارسال صورتحساب، کدی که در سمت کلاینت هنگام ارسال تولید می‌شود.
packetType	نوع بسته
retry	نوع فیلد بولین است (مقدار TRUE/FALSE)، مشخص می‌کندکه بسته اولین بار است ارسال شده است یا خیر.
data	داده‌های داخل درخواست. در صورتی که داده‌ها رمز شده باشند، آرایه بایت به صورت رشته Base64 ارسال می‌شود. در صورتی که داده‌ها کشف باشند، رشته سریال شده JSON قرار داده می‌شود.
encryptionKeyId	شناسه کلید عمومی سازمان، جهت باز کردن کلید متقارن. در صورتی که داده‌ها رمز نشده باشند، این فیلد خالی ارسال می‌شود.
symmetrickey	کلید متقارن رمز شده با کلید رمزنگاری (کلید عمومی سازمان) جهت باز کردن داده‌های رمز شده. در صورتی که داده ها رمز نشده باشند، این فیلد خالی ارسال می‌شود.
iv	بردار مقدار اولیه کلید متقارن. در صورتی که داده ها رمز نشده باشند، این فیلد خالی ارسال می‌شود.
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 سرویس‌های جمع آوری و پردازش اطلاعات

درخواست همگام و غیرهمگام دو سرویس اصلی زیر سامانه جمع‌آوری و پردازش اطلاعات می‌باشند:

۷-۳-۱ متد غیرهمگام

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

ارسال صورتحساب به صورت غیرهمگام انجام می‌شود 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 بسته‌ها وجود داشته باشد، نوع بسته (PacketType) در صورتحساب به دو بخش تقسیم بندی می‌شود.

بخش دوم	بخش اول
شماره نسخه + حرف V	INVOICE

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

INVOICE.V01

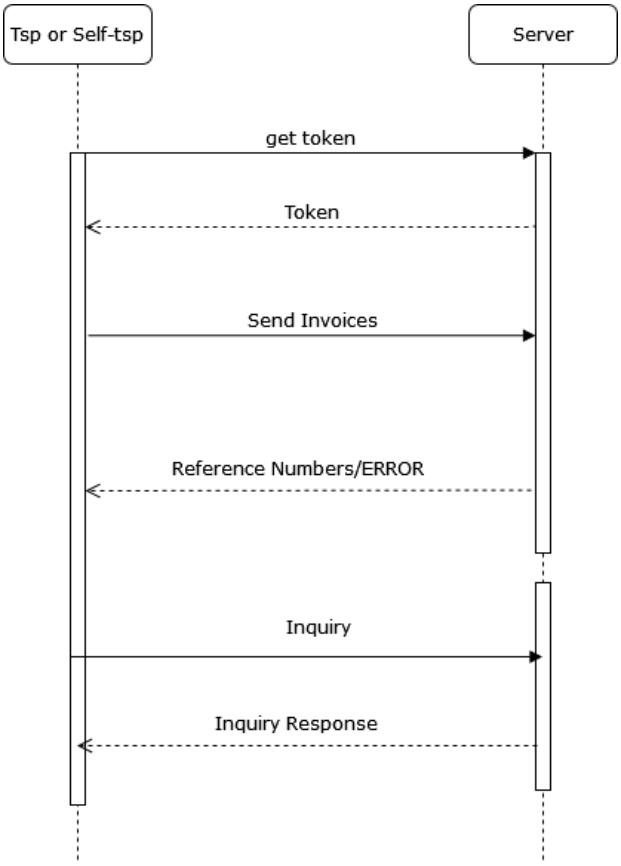
فرآیند ارسال صورتحساب مطابق شکل(۳) و به شرح ذیل می‌باشد:

۱. مودی یا TSP با ارسال درخواست توکن به سرور جمع‌آوری فرآیند ارسال صورتحساب را شروع می‌کند.

۲. توکن دریافت شده در سرآیند درخواست قرار می‌گیرد و صورتحساب ارسال می‌شود.

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

۴. مودی یا TSP می‌تواند 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.

۷-۳-۱-۲ ساختار بسته صورتحساب

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

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 قرارگیری در صورتحساب	نوع فیلد	طول فیلد	مقادیر مجاز
۳	تاریخ و زمان ایجاد صورتحساب(میلادی)	Indati2m	سرآمد (Header)	Timestamp	۱۴	
۴	نوع صورتحساب	inty	سرآمد (Header)	int	۱	۱ – ۲ - ۳ ۱- نوع اول ۲- نوع دوم ۳- نوع سوم
۵	سریال صورتحساب	inno	سرآمد (Header)	string	۱۰	
۶	شماره منحصر به فرد مالیاتی صورتحساب مرجع	irtaxid	سرآمد (Header)	string	۲۲	
۷	الگوی صورتحساب	inp	سرآمد (Header)	int	۱	۱-۲-۳-۴-۵-۶ الگوی۱: فروش الگوی۲: فروش ارزی الگوی۳: صورتحساب طلا، جواهر و پلاتین قرارداد :الگوی۴ پیمانکاری الگوی۵: قبوض خدماتی الگوی۶: بلیت هواپیما
۸	موضوع صورتحساب	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 قرارگیری 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	۱- ۲ – ۳- ۴ - ۵ 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	۱-۲ 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	tprdis	سرآمد (Header)	double	متغیر	
۲۳	مجموع تخفیفات	tdis	سرآمد (Header)	double	متغیر	

ردیف	عنوان قلم اطلاعاتی	JSON	محل قرارگیری 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	۱-۲ 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 اندازه‌گیری	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 صورتحساب	نوع فیلد	طول فیلد	مقادیر مجاز
۴۵	نرخ مالیات بر ارزش‌افزوده	vra	بدنه (Body)	double	متغیر	
۴۶	مبلغ مالیات بر ارزش افزوده	vam	بدنه (Body)	double	متغیر	
۴۷	موضوع سایرمالیات و عوارض	odt	بدنه (Body)	string	متغیر	
۴۸	نرخ سایرمالیات و عوارض	odr	بدنه (Body)	double	متغیر	
۴۹	مبلغ سایرمالیات 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	حق‌العمل	bros	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 فروشگاهی	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 صورتحساب	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 کلا 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 می‌شود.

جدول ۸. مدل داده پاسخ‌های غیرهمگام

اسم شی	AsyncResponse
نوع فیلدها	{ "uid": "string", "referenceNumber": "string", "errorCode": "string", "errorDetail": "string" }

جدول ۹.توضیحات فیلدهای خروجی درخواست غیرهمگام

نام فیلد	توضیحات
uid	شناسه یکتای ارسال صورتحساب، شناسه‌ای که در سمت کلاینت هنگام ارسال صورتحساب تولید می‌شود.
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 می‌شود.

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

۷-۳-۲ متدهای همگام

۷-۳-۲-۱ سرویس درخواست‌های همگام

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

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

جدول ۱۰.درخواست‌های همگام

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

نام متد	ورودی داده	خروجی
	} , ...]}	

INQUIRY_BY_
REFERENCE_NUMBER
	{
	"referenceNumber":
	[
	referenceNumber_1,
	referenceNumber_2,
	referenceNumber_3, ...
	]
	}
		[
		{ uid,
		referenceNumber,
		status,
		data,
		packetType,
		fiscalId
		}, ...
		]

INQUIRY_BY_TIME	time	[{ uid, referenceNumber, status, data, packetType, fiscalId }, ...]
INQUIRY_BY_TIME_RANGE	startDate: تاریخ شروع بازه مورد نظر, endDate: تاریخ پایان بازه مورد نظر	[{ uid, referenceNumber, status, data, packetType, fiscalId }, ...]

GET_SERVER_INFORMATION	-	{
		"serverTime": "زمان سرور",
		publicKeys" : [{
		"key": "کلید عمومی",
		"id": "شناسه کلید عمومی",
		"algorithm" : "الگوریتم کلید",
		"purpose" : "هدف کلید. ۱
		رمزگشایی صورتحسابها در جمع آوری"

نام متد	ورودی داده	خروجی
		}] }

GET_SERVICE_STUFF_LIST	{
	"filters": [
	لیست فیلتر بر روی داده ها
	],
	"orderBy": [
	لیست مرتب سازی
	],
	"page": صفحه مورد درخواست,
	"size": تعداد در صفحه
	}
		{
		"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",
		}
		}

GET_ECONOMIC_CODE_ INFORMATION	economicCode	nameTrade: نام تجاری, taxpayerStatus: وضعیت مودی, taxpayerType: نوع شخص, postalcodeTaxpayer: کدهای پستی مودی addressTaxpayer: نشانی مودی

۷-۳-۲-۲ پاسخ درخواست‌های همگام

جدول ۱۱.توضیحات فیلدهای ساختار خروجی درخواست همگام

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

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

۷-۳-۲-۳ متد دریافت توکن

کلاینت‌ها با استفاده از این متد، توکن JWT دریافت می‌کنند. با توجه به اینکه در این سرویس امضای درخواست کننده اجباری است، احراز هویت از طریق آن صورت می‌پذیرد. در صورتی که نام کاربری اعلام شده و امضای آن تطابق داشته باشد، توکن JWT برای کلاینت ارسال می‌شود. این توکن طول عمر مشخصی دارد، با منقضی شدن آن، کلاینت 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 می‌شود. نمونه CURL درخواست در پیوست ۲-۲ ارائه شده است.

۷-۳-۲-٤ متد استعلام اطلاعات حافظه مالیاتی مودی و حد مجاز فروش مودی

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

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

۷-۳-۲-۵ متد استعلام بسته‌های ارسالی غیرهمگام

برای استعلام وضعیت صورتحساب‌های ارسال شده می‌توان از متدهای ذیل استفاده کرد: این متدها به احراز هویت نیاز دارند.

- Inquiry_by_uid: کلاینت می تواند با آرایه ای از UID ها، صورتحساب‌های مورد نظر خود را استعلام نماید. در پاسخ وضعیت آنها باز گردانده می‌شود.

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

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

- Inquiry_by_reference_number: در این متد، کلاینت می‌تواند با آرایه‌ای از reference_number ها، صورتحساب‌های مورد نظر خود را استعلام نماید. در پاسخ وضعیت آنها باز گردانده می‌شود.

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

- Inquiry_by_time: با این متد صورتحساب‌های دارای خطا از یک زمان مشخص تا زمان حال مشخص می‌شوند.

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

- Inquiry_by_time_range: با این متد صورتحساب‌های دارای خطا در یک بازه زمانی مشخص می‌شوند.

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 بسته‌هایی که وضعیت FAILED دارند برگشت داده می‌شوند.

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 می‌گیرد. 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 خدمت‌ها که توسط وزارت صمت ابلاغ شده است مورد استفاده قرار می‌گیرد. با توجه به اینکه حجم نتیجه خروجی این متد ممکن است زیاد باشد، امکان دریافت

اطلاعات به صورت صفحه‌بندی 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
}
```

۷-۳-۲-۸ متد استعلام اطلاعات شماره اقتصادی

این متد برای استعلام شماره اقتصادی مورد استفاده قرار می‌گیرد. 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

اسم شی	ErrorResponse

نوع فیلد ها	{
	"errorDetail": "string",
	"errorCode": "string"
	}

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 صورتحساب 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

۸-۳ 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 صورتی‌که موضوع صورتحساب شامل اصلاحی، ابطالی و یا برگشت از فروش -یکی از مقادیر ۲، ۳ و یا 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 صورتحساب‌های الکترونیکی با الگوی بلیط هواپیما.	Flight type is empty
۲۰	عدم ثبت نوع ارز در صورتحساب‌های الکترونیکی با الگوی فروش ارزی.	Currency type is empty
۲۱	عدم ثبت نرخ برابری ارز با ریال در صورتحساب‌های الکترونیکی با الگوی فروش ارزی.	Exchange rate is empty
۲۲	عدم ثبت شماره اشتراک /شناسه قبض در صورتحساب‌های الکترونیکی با الگوی قبوض خدماتی.	Billing identification is empty
۲۳	عدم ثبت مبلغ قبل از تخفیف در صورتحساب‌های الکترونیکی با الگوی اول تا پنجم.	Pre-discount amount is empty
۲۴	عدم ثبت مبلغ تخفیف در صورتحساب‌های الکترونیکی با الگوی اول تا پنجم.	Discount amount is empty
۲۵	عدم ثبت مبلغ بعد از تخفیف در صورتحساب‌های 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 صورتحساب‌های 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 صورتحساب‌های 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 اندازه‌گیری 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 (fiscalId) 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

پیوست۱

۱-۱ کد نرمالسازی 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.

۱-۲ کد DotNet برای نرمالسازی 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.

۱-۳ روش امضا رشته (string)

۱-۳-۱ کد جاوا

```
public static String getSignedText(String text, String algorithm, PrivateKey
privateKey) throws UnsupportedEncodingException, 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);

 }
```

۱-۳-۲ 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));
        }
```

۱-٤ کد روش رمزگذاری به روش متقارن AES/ GCM

۱-٤-۱ 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;
 }
```

۱-٤-۲ کد C#

در C# می‌توانید با []byte به صورت کاملا 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. ja 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;
                RSAParameters rsaParameters = new RSAParameters();
                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";
            }
        }
```

پیوست ۲

۲-۱ دریافت صورتحساب

- نمونه CURL:

```
curl --location --request POST '
https://tp.tax.gov.ir/req/api/tsp/async/normal-enqueue' \
--header 'requestTraceId: 346edbfa-cf5f-11ec-9d64-0242ac120002' \
--header 'timestamp: 1645619844431' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJjbGllbnRUeXBlIjoiTUVNT1JZIiwicGVybWlzc2lvbnMiOlsic2Vs
Zi10c3AuYXN5bmMuZmFzdC1lbnF1ZXVlIiwic2VsZi10c3Auc3luYyIsInNlbGYtdHNwLmFzeW5jL
m5vcm1hbC1lbnF1ZXVlIl0sInBhY2tldFR5cGVzIjpbImludm9pY2UuZHJhZnQudjAwIiwiZ2V0X2
Zpc2NhbF9pbmZvcm1hdGlvbiIsImlucXVpcnlfYnlfdGltZV9yYW5nZSIsImlucXVpcnlfYnlfdGN
uIiwiaW5xdWlyeV9ieV90aW1lIiwiZ2V0X3Rva2VuIiwiZ2V0X3NlcnZpY2Vfc3R1ZmZfbGlzdCIs
ImdldF9lY29ub21pY19jb2RlX2luZm9ybWF0aW9uIiwiZ2V0X2Zpc2NhbF9mdWxsX2luZm9ybWF0a
W9uIiwiaW5xdWlyeV9ieV9yZWZlcmVuY2VfbnVtYmVyIiwiZ2V0X3NlcnZlcl9pbmZvcm1hdGlvbi
JdLCJpc3MiOiJUQVggT3JnYW5pemF0aW9uIiwiaWQiOiJ0ZXN0LXNlbGZ0Y3AtbWVtb3J5LWlkLTE
iLCJleHAiOjE2NDU3MjkzNzUsInRheHBheWVySWQiOiJ0ZXN0LXRheHBheWVyLWlkLTEifQ.z0rQZ
TXjdgLN7kiQ22-
h095HUOex3hK3tSg2RvASis02lmS9xYOha3kBDVf8rN6eqbrHrWuc7u3kvsqq34fcCw' \
--header 'Content-Type: application/json' \
--data-raw '{
 "packets":[{
  "uid": "346edbfa-cf5f-11ec-9d64-0242ac120002",
  "packetType": "INVOICE.V01",
  "retry": false,
  "data":
"Mt2l/b9FrZtxzM8+SGdNxVSCwHfPioMlX026yx6n9cUbwHaZSL68JX6bhV4MKhyiEyntIx0VTn7C
m0heshfLtCOdE0SaJxClte3qQw5NSzW85t8VIAYnd9rqOufapmSGUa5Vs+YNhX2QAEQGHYMcVS8uK
0S++zIthLhm58DDK+v5iH+R240nvgVT79b/bH2O43ig/DwkT/0hXvW/JbEvJmrhf4QjMxz9ceEyTC
en5fzYdaHdWd5/Y8Ci+5QAE9YrqKDoPhJtiS6UjrY2/hS+iQmk/FnO5pNC8xNT+R4D9oiDwgZmOz3
KV6MF1494ELT7oFQborCgzxXjjzOYlRB6bgWxgvTvs8b8YzoXKsabIUSACoFHm0HxpyGFm9k5VcXY
cV9yTKFS/RpVSRfWYz5IT4qUQ0hikj6HfvrscKn5CFYVQcPpYx8Y7nWHPeY+PW/g6tJ+xZZ8d2r6Q
KdvMXU6GZjb2Kvqf8FpVzeDNx065/CaYte1An5H+dimVz0XIQ7UE7pgLZcsv2fy2EC8IpFvOAnuaL
eKvcSluBpGj6vx1Pa5dKSlFucTFQG8aslv5SX76bgO15gCpMglVHIAdn3wDqvp3OceHv/BS9xd2Dw
kg35eGztMpj3hEHV1DHAMVF26L5A9vkC9R2efTjX6rQhJwlP/kGTVhkneV32HcAF562LNTqodxcVN
+s/NDGAbFHeND6LVOnEnwCSmuW40PokRl9wCuBnxGaxQ9xQqVowgo2optTgeGK2EVnNwxWgdlVXp2
2d9AVrzTogh3pxXNmyYO4JUihUkN6Vgmxlnc1cdDmtGEC51hNsDuFcibBLh/Q4MD4MchkHZBxqHJd
JMF/yt7QZPWiEvBcvJRJe36tBzbd6sNed1KycDM3kAVEQPJgBH6QjsmbgXDJ+F4f2eW/GD5z0V72+
oWLvugHQJf90NxBB9j0wQZBj20tBjoww5o6kfIEeu7h8k26F7PW4f5AbC+6ler4xKD7mDxDDmI0qe
bBXetgCOUdcWhIZ4MvAOMmRsZsYlZ1Noz4eFfLonL3U78C33e2EYICYsFstSdTBYFB9NWSoM6655l
```

```
oa6QQGZPyadlAKQJMZtR1UKR/lVKgHaKfyqqQGGdmgOjm1HUY+5Lm4J9l1Z8ETqs0sS383T06BNiF
HTlX+ApFn8UbM3gKORiA1e5F8v4I5e7dqTb+et2Ivxee5TEkNXZXJoMGXFrQE/bho0W8A9kGi/ye1
udMcoxczmYSxWXXSdSbnlTHstcT7Fa2G31c1w0LDdVsq20yCK9vGCMqSNu9PMrefUb/3jP/a71d15
ecSNlQDbGdZ03ASkVQ8b356cuW6yuqECbvq1EeI1JlbB9T3wthKRvTcVRlTYvrLzf8ebdjAmysc+h
rgnQ1FuJnwxoX6wXfnw+2ph1KMiaPrLH6p+bnm7+AUiaGu/LuSdBJPzYNQOEIrkJS5DrJi5KHn9jD
4b3gCQ97B8qspBspvd3mYLLVHQGU+qa4GP5EdoiU76vFzAHSHu+TOTDmKs9NQ7jUwuFlUov3+Fx/M
LhUYkWyyDogbwGiyauK5wdgwhUi4AmBR8SwNMG/TxGIdB7xIsMydNOJOctKDJzTK0odvd4d7VkosN
wVzhQw6C9okLJmBblnJbJQtOQZN5MIzXG6MfGkbgJ6g/zFPRKJf7+X9u9Vt83uUCx3ftrmdff/9uN
+5hbVep2gyxmMmiq4Lw2uKg8vyRSzDe7iACkfQWXO2Vstb1U3ZIT2iCzVNGyEYgsLLQdVO8yIjn8P
fIuGsytMAqQx9iH2Q32rZqhB8vGsJQwBznKywv/u5wYRjlLVEN1dgGoqfXgjFdbdhjBzXLgbbLQ8+
FVi0GfCY88mRRKtkmzLnGJtwTenaNCsfrDDg7BeO/SCzuqij2P0r6WOV2X7zCQWKCThTWUWRu0IM0
nCdRIoshg2U2E/IQ+AIt4oQWjIFOQSy2cY4GUuVPXgwiFNvKV3xkYE3YyfeglPhJSKjlWnth9+V17
/C2IBGxXXVQxoLLgheq6IMLArZ6N0bvKX4y0Z2PE0COfpwYxBxEop6SdV4hFy3mZ1C5Wg4FaccfoG
EJRpfCuYT59cIMF1xquwuMRyRBXrfGGzOdZe/1I4l/snweEqbXG2bXM/24/WOhdD3rR0X0+K+wmso
bkoBUyTamhZDI8uuaHTYx1zQ/SJ9BgaV6C70s+K7jiUjXJgc0yinsmsk0B3yoq5tCqpIfcyZ5hbHw
R7Vqc+2TgtygmQXMjg+Tu3O1j3J/Xqse6UoFy+RTM0hEmfMyGMmXQeX4X2IZ5fa98hglPdsdrvm4J
mDvfjDnLCOpLDNeHqvEhKj8p9Kmxwdox6lV058UeWwZA5ZH7O/Krk+NdFzua1g=",
  "encryptionKeyId": "31cf3e11-c468-4b14-90a4-d7c8c7e72fd5",
  "symmetricKey":
"cBAkPRW8JJiouZM0zLgiqIgtaqjUcWlNr2CoZj6uH4ESMIfJjTgR7kdOj5msAlblf5vcN21AYtv3
3+UHNipM0bAq30QIf9TnCQgNY+fbalFHQl5Em7/VE+5WYEr+N5uPnRHX/JK2j/A2/jiwwYmi9+Gey
FK2QQYGLUYrfPUw+ifUG1veJldPAocwkDlRxZVXp+hV1L2fKxSwDUCVF5Uvr5jX8ACwmMa8t5qCco
w/xWXumod1BfE9sPOUonOZ1hVINscJYrN4y7MG0Re88F/WRyL4grznetSSKOsAGTrGVflKZ1Mm3ng
DF+o+8+t1WtVzF5Hj0zv6ODAiX3ikS03efMbFtGGQ5WygoEhhWLgt9/CqjkEm7qHE5T0iQ+pE+f1o
TNXfh4eGH/+j9zeI//uLxcHSH/TnTmUlGxlareNXakumtcrKKulNAw2Mm7F+XBLKXXMfW3iJrTPd3
/hpycPTVpu/6SO/7JMKUA44+irWw5AaXl7j4EDWF72AvVDJCPlh2Qw7KrW7u4HFeT3Qll5L2vuP/E
TWvFVgBitP5DbdB9nFoSb8NwKZn6AqGgYalOVz+8Y8SHgEBueix3V4e5PIm31WTbViqgnANP4TVft
T3ZL3b3yBkjvQpvbc2CtstXgG05C16FzWX16UiR+IYZ7ahOolaTwmZr+GJA87ZLjKaK4=",
  "iv": "dee90162e26db3f2246f052f38168ed0", "dataSignature" :
"VSvyWrpmT1TwI8I1vUUe9wF0EGNFMZ/2dPgM1tjO0vr8agPDHoTOiL5J3XxqvFCauCOfaFs6iVpw
dccwAtXfsFJ4wNQF3Mqr1I9un1v5AEePPeXAUp46CKl4Ny5sm3WGDTdnezx1+4FBCvO5uP07RkeDw
ASyYOySFdWl9d6fHes7KNkaLW72NetbTaZWytQtHx7ib+7cJyG+54eLAY3L6EIuoNiRSIlC1Pn53Q
8AT90uRV8UPSCcpiZhg/YINQXrunF2YH/LbTSqckwFsdukqg6BGZJbaqPwMfvm2WdPZLNTEW/BgNm
2SmSNwxqivQb1AY2mcfNgO6dkyDCRVWTUeA==",
  "fiscalId": "AA56CD"
 }],
 "signature":
"FohMakKq0WRMPQD0nStssK2iP0bbFojw7w+NVUoWjTdyRFNeKBQeLP9FpL3r5u6EliSKUiO3vulI
vBzEbAaFBuEHSbmyr3qOTN0CpuORdzqHDT/oNk/evvpOoEFt3LvowfLBoxNLDhAGrzD1K56jlRco9
3U4iplJgKzFmm9hlmByzTfDEe9FLX/22Gy9regeYtTg9sc12URu7csVttUr8YjcqFxuR82l4m0tZd
Rf6jR709YZmskarPXavPsDr8buz7dzOHI0hIeokOvToyG3J1bAVMl0SdWREq1Ix0czJ5jLnMiI8N3
1s8rvnypKzo5yWDOjXECTrLfRr+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":
"IiIdkclswu3Krc8ZM7MQvEy7ZWzJmBPSl1CQrI0dhLGdRPRrmomVo+UkbdzRyuth9G4EnbgOjnjz
5WJcfO8MuBVouASTMfv/OCOhAkxTudQtWzUO0d6BU/YiRT5alNwdey0dMsn3T083luLv9iG/lKKz9
ewUem0RwBYOnehD6rJFXHirGDfJPHBOTSHCqHL1vQe0JLZAQwaTTieEE8zNWXwNr53BS2KxRKX8+M
leoUl8LWUn6wZS/zs3auOKSRSO5pgJVq6zZCadd5D7vlhrw1KB/XfO4pv8GexAx2dbRMiGG5eumQG
BcLo1RvJW2mZsGu+dQRm/NwnIpN7CP5qlkg=="

}'
```


- نمونه پاسخ درخواست:


```
{
    "signature": null,
    "signatureKeyId": null,
    "timestamp": 1655185988494,
    "result": {
        "uid": null,
        "packetType": "TOKEN_RESULT",
        "data": {
            "token":
"eyJhbGciOiJIUzUxMiJ9.eyJjbGllbnRUeXBlIjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3L
TBlZjQtNDRmNC05YWZkLWI5MjdlNDM1ZmI2NyIsInBlcm1pc3Npb25zIjpbInRzcC5hc3luYy5m
YXN0LWVucXVldWUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldWUiLCJ0c3Auc3luYyJdLCJwYWN
rZXRUeXBlcyI6WyJHRVRfRUNPTk9NSUNfQ09ERV9JTkZPUk1BVElPTiIsIklOUVVJUllfQllfVE
lNRSIsIkdFVF9TRVJWRVJfSU5GT1JNQVRJT04iLCJJTlFVSVJZX0JZX1VJRCIsIklOUVVJUllfQ
llfUkVGRVJFTkNFX05VTUJFUiIsIkdFVF9UT0tFTiIsIkdFVF9GSVNDQUxfSU5GT1JNQVRJT04i
LCJJTlZPSUNFLlYwMSIsIklOUVVJUllfQllfVElNRV9SQU5HRSIsIkdFVF9TRVJWSUNFX1NUVUZ
GX0xJU1QiXSwiaXNzIjoiVEFYIE9yZ2FuaXphdGlvbiIsImlkIjoidGVzdC10c3AtaWQtMSIsIm
V4cCI6MTY1NTIwMDM4OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0V
lX7oqgVITzy5S3ksehSHSbdrfwiaYxDT1mMn541TV8zP-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.eyJjbGllbnRUeXBlIjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3LT
BlZjQtNDRmNC05YWZkLWI5MjdlNDM1ZmI2NyIsInBlcm1pc3Npb25zIjpbInRzcC5hc3luYy5mY
XN0LWVucXVldWUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldWUiLCJ0c3Auc3luYyJdLCJwYWNr
ZXRUeXBlcyI6WyJHRVRfRUNPTk9NSUNfQ09ERV9JTkZPUk1BVElPTiIsIklOUVVJUllfQllfVEl
NRSIsIkdFVF9TRVJWRVJfSU5GT1JNQVRJT04iLCJJTlFVSVJZX0JZX1VJRCIsIklOUVVJUllfQl
lfUkVGRVJFTkNFX05VTUJFUiIsIkdFVF9UT0tFTiIsIkdFVF9GSVNDQUxfSU5GT1JNQVRJT04iL
CJJTlZPSUNFLlYwMSIsIklOUVVJUllfQllfVElNRV9SQU5HRSIsIkdFVF9TRVJWSUNFX1NUVUZG
X0xJU1QiXSwiaXNzIjoiVEFYIE9yZ2FuaXphdGlvbiIsImlkIjoidGVzdC10c3AtaWQtMSIsImV
4cCI6MTY1NTIwMDM4OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0Vl
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+zYwHyJ0xZCntfGw59tuZ9jUyjPLFWJq7j8KOZ1VkFrGuSaxFXukndE0NCf3jKHFGBUqu
Fe9jGCktGBRz6b+oBycCZJtH4BcwiyTkrg+nLfduBEc6bhRA48ZiKBBNAM/8RRSRj8tC0AoStj7
90gdw10SkS1yYZ4t0KRfxVl2a21wGWSF5ahsyt3LQptloKUjZTrHt6UtVU2WvIPnFkb4p6P/pw2
```

```
YjA8YfcHCdn/WqrCFvA4YC+09cvlqwRdxk8tEfxzyiczStRrd5Jq8UVZDBjXJIwel7GTeSWI9+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.eyJjbGllbnRUeXBlIjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3LT
BlZjQtNDRmNC05YWZkLWI5MjdlNDM1ZmI2NyIsInBlcm1pc3Npb25zIjpbInRzcC5hc3luYy5mY
XN0LWVucXVldWUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldWUiLCJ0c3Auc3luYyJdLCJwYWNr
ZXRUeXBlcyI6WyJHRVRfRUNPTk9NSUNfQ09ERV9JTkZPUk1BVElPTiIsIklOUVVJUllfQllfVEl
NRSIsIkdFVF9TRVJWRVJfSU5GT1JNQVRJT04iLCJJTlFVSVJZX0JZX1VJRCIsIklOUVVJUllfQl
lfUkVGRVJFTkNFX05VTUJFUiIsIkdFVF9UT0tFTiIsIkdFVF9GSVNDQUxfSU5GT1JNQVRJT04iL
CJJTlZPSUNFLlYwMSIsIklOUVVJUllfQllfVElNRV9SQU5HRSIsIkdFVF9TRVJWSUNFX1NUVUZG
```

```
X0xJU1QiXSwiaXNzIjoiVEFYIE9yZ2FuaXphdGlvbiIsImlkIjoidGVzdC10c3AtaWQtMSIsImV
4cCI6MTY1NTIwMDM4OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0V1
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+iAJEmgkl8jc1yp73SrNHz0RHaFbsgXICN5F52zV8HPfa3YI
cddBi3ZjVNVs9RA+HsEHolavOecPLo9AOIGS7y+BtvV1FBKLTizbflpejprrSvN78ledJQmEsjg
MMiVZA1zexQ8HU73/kf3fIyaviXXVyVcwM7DFqiyVdVV4IX3klJYVzyamUO8PDl9zTcwQMxwiSd
QiQ/plZtx0wzP2UTyLmy47jskdJVjpDseM1iSEt6fy+18R98iqvBFiKi5njo5ONr+yTHQusTfwF
X6I2uhYNML6/aC8EByFbTjVuJXyIOrM7CqnP77SbavA=="
}'
```

۲-۴-۲ استعلام با رسید یکتای دریافت صورتحساب (Inquiry_by_reference_number) :

- نمونه ارسال درخواست CURL :

```
curl --location --request POST
''https://tp.tax.gov.ir/req/api/tsp/sync/INQUIRY_BY_REFERENCE_NUMBER' \
--header 'requestTraceId: 1655186644886' \
--header 'timestamp: 1655186644886' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJjbGllbnRUeXBlIjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3LTBl
ZjQtNDRmNC05YWZkLWI5MjdlNDM1ZmI2NyIsInBlcm1pc3Npb25zIjpbInRzcC5hc3luYy5mYXN0L
WVucXVldWUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldWUiLCJ0c3Auc3luYyJdLCJwYWNrZXRUeX
BlcyI6WyJHRVRfRUNPTk9NSUNfQ09ERV9JTkZPUk1BVElPTiIsIklOUVVJU1lfQ1lfVElNRSIsIkd
FVF9TRVJWRVJfSU5GT1JNQVRJT04iLCJJTlFVSVJZX0JZX1VJRCIsIklOUVVJU1lfQ1lfUkVGRVJF
```

```
TkNFX05VTUJFUiIsIkdFVF9UT0tFTiIsIkdFVF9GSVNDQUxfSU5GT1JNQVRJT04iLCJJTlZPSUNFL
lYwMSIsIklOUVVJUllfQllfVElNRV9SQU5HRSIsIkdFVF9TRVJWSUNFX1NUVUZGX0xJU1QiXSwiaX
NzIjoiVEFYIE9yZ2FuaXphdGlvbiIsImlkIjoidGVzdC10c3AtaWQtMSIsImV4cCI6MTY1NTIwMDM
4OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0VlX7oqgVITzy5S3kse
HSbdrfwiaYxDT1mMn541TV8zP-HowhBs8u1fjs_S81kiZON4FA' \
--header 'Content-Type: application/json' \
--data-raw '{
    "time": 1,
    "packet": {
        "uid": null,
       "packetType": "INQUIRY_BY_REFERENCE_NUMBER",
        "retry": false,
        "data": {
         "referenceNumber":
            ["04af9b37-07e4-4019-88c2-efeaa844552f","99fab38b-b32f-49cb-89d8-
71e52cae2d91","6c2983cb-0fe4-4da4-b93f-2e30e9652631"]
        },
        "encryptionKeyId": "",
        "symmetricKey": "",
        "iv": "",
        "fiscalId": "AA56CD",
        "dataSignature":""
    },
    "signature":
"MMy9CYojEIPEYg1I0gNbahYLJfST7VEEYA7bXr7/ZTHjvYMEoUeFQU9EcaN2sIhx/833abKw7FD0
cHeICnEBCWngc0WowS1ZScfWlviW8hrRQlAAZjaZcPw0JUXLgkM2rdUDPTBy8ENUbOGmGvDajcOOF
NbO3GAtRD97szK8Na56qSDsnG7uDpqxS2r+xI0JtUKxJFJyL1i57u2wZbwUCw2uN7Ocd0XDD08oLZ
I2CQxvwu/TV+vzaFOPqlTYtLk54H82ShFtXtaqbHnt7RTtX70aOs0P16SYxDaMfrr0EuV0p3FI5lF
wz3TZzYF0OAVaSExv95UT5fRXKSOZDWhzPw=="
}'
```

۲-٤-۳ دریافت خطاهای بسته‌های ارسالی غیرهمگام با استفاده از زمان (INQUIRY_BY_TIME):

- نمونه CURL:

```
curl --location --request POST
''https://tp.tax.gov.ir/req/api/tsp/sync/INQUIRY_BY_TIME' \
--header 'requestTraceId: 1655186330848' \
--header 'timestamp: 1655186330848' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJjbGllbnRUeXBlIjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3LTBl
ZjQtNDRmNC05YWZkLWI5MjdlNDM1ZmI2NyIsInBlcm1pc3Npb25zIjpbInRzcC5hc3luYy5mYXN0L
WVucXVldWUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldWUiLCJ0c3Auc3luYyJdLCJwYWNrZXRUeX
BlcyI6WyJHRVRfRUNPTk9NSUNfQ09ERV9JTkZPUk1BVElPTiIsIklOUVVJUllfQllfVElNRSIsIkd
FVF9TRVJWRVJfSU5GT1JNQVRJT04iLCJJTlFVSVJZX0JZX1VJRCIsIklOUVVJUllfQllfUkVGRVJF
```

```
TkNFX05VTUJFUiIsIkdFVF9UT0tFTiIsIkdFVF9GSVNDQUxfSU5GT1JNQVRJT04iLCJJTlZPSUNFL
lYwMSIsIklOUVVJUllfQllfVElNRV9SQU5HRSIsIkdFVF9TRVJWSUNFX1NUVUZGX0xJU1QiXSwiaX
NzIjoiVEFYIE9yZ2FuaXphdGlvbiIsImlkIjoidGVzdC10c3AtaWQtMSIsImV4cCI6MTY1NTIwMDM
4OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0VlX7oqgVITzy5S3kksehS
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":
"elHDykQjZBzgM9NIMSEVR+dYfDcAbowWF0OvEzI31WRuBwMVmmmYkXg6b4q6oIj+78Cgwd1O6Jo4
5zuaH1eNSxVc67xr7hagpnyvnVQYt2ot3Y5Rk+MJ0BkyfE06qv45yzV2kwe5CkfCAH2ccvFnnkEGd
ol5CoaAyJk51hWNTPpGmOdXAwoEoUE2gjDRZqtRsgzMOmum5ep7GoPQ/8WildWNgxuVRLaWIqNmZs
EuBDVLgJsY7xFMGczFCyZtlNwZwV1FVTDZVmn8XNDClu7eOIRX/krnF932EAxWiWG5zidyr/geXyz
gpmFevQQbe1OeedCuLcFVT+GrNGfJUuzakw=="
}'
```

۴-۴-۲ دریافت خطاهای بسته‌های ارسالی غیرهمگام با استفاده از بازه زمانی(INQUIRY_BY_TIME_RANGE)

- نمونه CURL:

```
curl --location --request POST
''https://tp.tax.gov.ir/req/api/tsp/sync/INQUIRY_BY_TIME_RANGE' \
--header 'requestTraceId: 1655187263981' \
--header 'timestamp: 1655187263981' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJjbGllbnRUeXBlIjoiVFNQIiwidG9rZW5JZCI6IjZlODlhZDQ3LTBl
ZjQtNDRmNC05YWZkLWI5MjdlNDM1ZmI2NyIsInBlcm1pc3Npb25zIjpbInRzcC5hc3luYy5mYXN0L
WVucXVldWUiLCJ0c3AuYXN5bmMubm9ybWFsLWVucXVldWUiLCJ0c3Auc3luYyJdLCJwYWNrZXRUeX
BlcyI6WyJHRVRfRUNPTk9NSUNfQ09ERV9JTkZPUk1BVElPTiIsIklOUVVJUllfQllfVElNRSIsIkd
FVF9TRVJWRVJfSU5GT1JNQVRJT04iLCJJTlFVSVJZX0JZX1VJRCIsIklOUVVJUllfQllfUkVGRVJF
TkNFX05VTUJFUiIsIkdFVF9UT0tFTiIsIkdFVF9GSVNDQUxfSU5GT1JNQVRJT04iLCJJTlZPSUNFL
lYwMSIsIklOUVVJUllfQllfVElNRV9SQU5HRSIsIkdFVF9TRVJWSUNFX1NUVUZGX0xJU1QiXSwiaX
NzIjoiVEFYIE9yZ2FuaXphdGlvbiIsImlkIjoidGVzdC10c3AtaWQtMSIsImV4cCI6MTY1NTIwMDM
4OCwiY3JlYXRlRGF0ZSI6MTY1NTE4NTk4ODQ5Mn0.SF06IpHsdSEKHDC0VlX7oqgVITzy5S3ksehS
HSbdrfwiaYxDT1mMn541TV8zP-HowhBs8u1fjs_S81kiZON4FA' \
--header 'Content-Type: application/json' \
--data-raw '{
    "time": 1,
    "packet": {
        "uid": null,
        "packetType": "INQUIRY_BY_TIME_RANGE",
        "retry": false,
        "data": {
           "startDate" :
                14010321,
           "endDate" :
                14010324
        },
        "encryptionKeyId": "",
        "symmetricKey": "",
        "iv": "",
        "fiscalId": "",
        "dataSignature":""
    },
    "signature":
"I/xb+7NttCGMd2+vw3qOWrVlFTKRHgcXDB+TBn0kTLrLg8bFLvT+Q7XSUNfDAXi5wo+Sn5sDGpJ1
jDF+nb66y3XnBP31ssbwkMTMemlNG1r6MWUednt6SYg2WNqqaDCsHUmfldfijVX7N/KUlmgnuVcMx
oLEo8xf7o6UawWdYsOMrqbw2NWBZkrlYaFQQO9QdIzA5LxBjguumSW10D9fziFcyP+itTK9l/FUzZ
j2uQCx/C/gzcWbZu/MdxMd1dZOy8MjWgCVccTe9oKPcVSKE8sWggiz1Vp6dClauekOeBbhjlFQh1E
sOyxN10FVqbj2PIAP/im58LaKqe09GBbc4A=="
}'
```

۲-۴-۵ نمونه پاسخ درخواست های استعلام:

```
{
```

```
    "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+Wm1lhzjjhAGfSUOuDvOKRoUEQwP8oUcXRmYzcvCUgcfoRT5iz7HbovqH+bIeJwT4rmLmFcbf
Pke+E3DLUxOtIZifEXrKXWgSVPkRnhMgym6UiAtnzwA1rmKstJoWpk9Nv34CYgTk8DKQN5jQJqb9L
/Ng0zOEEtI3zA424tsd9zv/kP4/SaSnbbnj0evqsZ29X6aBypvnTnwH9t3gbWM4I9eAVQhPYClawH
Tqvdaz/O/feqfm06QBFnCgL+CBdjLs30xQSLsPICjnlV1jMzoTZnAabWP6FRzzj6C2sxw9a/WwlXr
Kn3gldZ7Ctv6Jso72cEeCeUI1tzHMDJPU3Qy12RQzaXujpMhCz1DVa47RvqiumpTNyK9HfFIdhgou
pFkxT14XLDl65S55MF6HuQvo/RHSbBJ93FQ+2/x/Q2MNGB3BXOjNwM2pj3ojbDv3pj9CHzvaYQUYM
1yOcFmIJqJ72uvVf9Jx9iTObaNNF6pl52ADmh85GTAH1hz+4pR/E9IAXUIl/YiUneYu0G4tiDY4ZX
```

```
ykYNknNfhSgxmn/gPHT+7kL31nyxgjiEEhK0B0vagWvdRCNJSNGWpLtlq4FlCWTAnPI5ctiFgq925
e+sySjNaORCoHraBXNEwyiHT2hu5ZipIW2cCAwEAAQ==",
                    "id": "6a2bcd88-a871-4245-a393-2843eafe6e02",
                    "algorithm": "RSA",
                    "purpose": 1
                }
            ]
        },
        "encryptionKeyId": null,
        "symmetricKey": null,
        "iv": null
    }
}
```


۶-۲ 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
    }
}
```