به نام خدا

مستندات API

نرم افزار حسابداری مازی

تهیه شده توسط تیم تولید مازی

شرکت تنها رایانه سوسف خورگام

تاریخچه نسخه :

نسخه	تاریخ	شرح
۱.۰.۰	۱۴۰۵/۰۸/۱۶	اولین نسخه از مستندات API مازی

تغییرات این نسخه :

- **تعاریف پایه :**

- **آدرس پایه :** https://tra.tanharayaneh.com/ -> {baseAddress}

آدرس عملیاتی جهت اتصال به دیتای اصلی

- **آدرس پایه تستی :** https://sandbox.tanharayaneh.com/ -> {testAddress}

آدرس تستی جهت انجام عملیات تستی بدون تاثیر گذاری بر دیتای اصلی

- **قالب کلی پاسخ ها**

در تمام درخواست هایی که ارسال می شود و عملیات چه موفق و چه دارای خطا باشد ، پاسخ بصورت json و با ساختار زیر برگردانده می شود :

- **عملیات موفق** : پاسخ به شکل زیر خواهد بود که مقدار status برابر با true ، مقدار statusCode برابر با کد وضعیت موفق ، مقدار message شامل پیغام عملیات موفق ، مقدار errors برابرnull و مقدار data شامل مقادیر درخواست کاربر می باشد :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": {
    "token": "asdhlaks۹uoerjlxcs...",
    "expireMinute": ۲۰
  }
}
```

- **عملیات دارای خطا** : پاسخ به شکل زیر خواهد بود که مقدار status برابر با false ، مقدار statusCode برابر با کد وضعیت خطا ، مقدار message شامل پیغام عملیات دارای خطا ، مقدار errors برابر با خطای رخ داده که شامل کد خطا و شرح خطا می باشد و مقدار data برابر null می باشد :

```
{
  "status": false,
  "statusCode": ۴۰۰,
  "message": "عملیات دارای خطا",
  "errors": {
    "code": ۱۰۰۰,
    "message": "اطلاعات اعتبار سنجی کلاینت ارسال نشده است"
  },
  "data": null
}
```

۱. Token : جهت ارتباط اولیه با نرم افزار

۱.۱ مرحله اول : دریافت توکن :

√ Endpoint : POST : api/Token

جهت استفاده از endpoint های API ، ابتدا باید کاربر لاگین کرده و یک توکن JWT تولید شود که از طریق token دریافت می شود :

آدرس پایه : {baseAddress}

هدر درخواست :

درخواست باید شامل هدر Authorization با فرمت Basic باشد که دو نقطه(:) مابین clientId و clientSecret قرار گرفته و بصورت base۶۴ کد می شود :

Authorization: Basic base۶۴(clientId:clientSecret)

مثال :

```
string clientId = "۱۲۳۴۵۶۷۸۹"; string clientSecret = "yWrexc&g^m%$g";
byte[] byteArray = Encoding.UTF۸.GetBytes(clientId + ":" + clientSecret);
string strBase۶۴ = Convert.ToBase۶۴String(byteArray);
```

Authorization: Basic strBase۶۴ ==> Basic{۱ space}strBase۶۴

بدنه درخواست (JSON) :

نام فیلد	نوع	اجبار	شرح
userName	string	بله	نام کاربر تعریف شده در نرم افزار مازی
password	string	بله	رمز عبور تعریف شده برای کاربر مازی

نمونه درخواست :

```
curl -X 'POST' \
  'http://tra.tanharayaneh.com/api/Users/token' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -H 'Authorization: Basic {base۶۴{clientId:clientSecret}}' \
  -d '{
  "userName": "{your-user-name}",
  "password": "{your-password}"
}'
```

پاسخ موفق : که درخواست کاربر در data مقدار دهی شده و شامل token و مدت زمان اعتبار توکن به دقیقه را برمی گرداند :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": {
    "token": "asdhlaks۹uoerjlxcs...",
    "expireMinute": ۲۰
  }
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۱.۲ اعتبارسنجی توکن دریافت شده :

✓ Endpoint : GET : api/Token/validateToken

جهت اطمینان از درستی و معتبر بودن توکن دریافت شده استفاده می شود تا نیاز به دریافت توکن مجدد نباشد و باعث افزایش سرعت اجرای دستورات گردد

آدرس پایه : {baseAddress}

هدر درخواست : ندارد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
token	string		توکن دریافت شده از مرحله اول

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/token/validateToken?token=euhgd;lirtu,mfgpor$fgty...

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مشخصات چک به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح

validateStatus	string	وضعیت اعتبارسنجی توکن : • true : توکن معتبر می باشد • false : توکن نامعتبر و نیاز به دریافت توکن جدید می باشد
validateMessage	bool	پیغام اعتبارسنجی توکن

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null,
  "data": {
    "validateStatus": false,
    "validateMessage": "توکن منقضی شده است"
  }
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۱.۳ دریافت کاربران :

√ Endpoint : Get : api/ Token/getUsers

استخراج کاربران تعریف در نرم افزار جهت ورود به سیستم

آدرس پایه : {baseAddress}

هدر درخواست :

درخواست باید شامل هدر Authorization با فرمت Basic باشد که دو نقطه(:) مابین clientId و clientSecret قرار گرفته و بصورت base۶۴ کد می شود :

Authorization: Basic base۶۴(clientId:clientSecret)

مثال :

string clientId = "۱۲۳۴۵۶۷۸۹"; string clientSecret = "yWrexc&g^m%$g";

```
byte[] byteArray = Encoding.UTF۸.GetBytes(clientId + ":" + clientSecret);
string strBase۶۴ = Convert.ToBase۶۴String(byteArray);
Authorization: Basic strBase۶۴  ==> Basic{۱ space}strBase۶۴
```

احراز هویت : ندارد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
userKind	int		وضعیت نمایش تمامی چک های دریافتی حتا آنهایی که به اتمام رسیده اند : • ۰ : نمایش کاربران قابل اتصال بصورت دسکتاپ • ۱ : نمایش کاربران قابل اتصال بصورت اپ موبایل • ۲ : نمایش کاربران قابل اتصال بصورت فروشگاه آنلاین **نکته** : در صورت ارسال نشدن ، مقدار پیش فرض آن ۰ می باشد

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/token/getUsers?userKind=۱

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از کاربران به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
userId	int	آی دی یکتای کاربر
userName	string	نام کاربر
branchId	string	آی دی یکتای شعبه مربوط به کاربر

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
```

```
    "userId": ۱,
    "userName": "مدیر سیستم",
    "branchId": ۱,
    ... سایر فیلدها
    },
   {
    "userId": ۲,
    "userName": "کاربر سیستم",
    "branchId": ۲,
    ... سایر فیلدها
    }
  ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۲. Users : جهت عملیات کاربران سیستم

۲.۱ سطح دسترسی کاربر جاری سیستم :

√ Endpoint : Get : api/ Users/getAccessLevel

استخراج سطح دسترسی کاربر جاری در نرم افزار

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ندارد

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/users/getAccessLevel

خروجی : در صورت موفق بودن درخواست ، خروجی شامل سطح دسترسی کاربر به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد که دارای مقدار false (دسترسی ندارد) یا مقدار true (دسترسی دارد) می باشند :

نام فیلد	نوع	شرح
report_cheques	bool	دسترسی کاربر به گزارش های مربوط به چک
report_profitLoss	bool	دسترسی کاربر به گزارش سود و زیان
report_partTrans	bool	دسترسی کاربر به گزارش های معاملات کالا
report_customers	bool	دسترسی کاربر به گزارش اشخاص

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    " report_cheques ": false,
    " report_profitLoss ": false,
    " report_partTrans ": true,
    " report_customers ": true,
    ... سایر فیلدها
    },
  ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۳. Branhces : جهت عملیات شعب

۳.۱ استخراج شعب مربوط به شرکت :

√ Endpoint : Get : api/Branches/getBranches

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ندارد

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/branches/getBranches

خروجی : در صورت موفق بودن درخواست ، خروجی شامل سطح دسترسی کاربر به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد که دارای مقدار false (دسترسی ندارد) یا مقدار true (دسترسی دارد) می باشند :

نام فیلد	نوع	شرح
branchId	int	آی دی یکتای شعبه
title	string	نام شعبه
isActive	bool	فعال بودن شعبه – true:فعال و false:غیر فعال

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    " branchId ": ۱,
    " title ": "شعبه تهران",
    " isActive ": true,
   },
   {
    " branchId ": ۲,
    " title ": "شعبه شیراز",
    " isActive ": true,
   },
   {
    " branchId ": ۳,
```

```
    " title ": "شعبه رشت",
    " isActive ": true,
    },
 ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۳.۲.

۴. General : جهت عملیات عمومی

۴.۱. استخراج مشخصات آخرین سال مالی فعال در نرم افزار :

✓ Endpoint : Get : api/General/getLastYearInfo

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ندارد

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/general/getLastYearInfo

خروجی : در صورت موفق بودن درخواست ، خروجی شامل مشخصات آخرین سال مالی فعال به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
finyearId	string	آی دی یکتای سال مالی
finYear	string	سال مالی مثل ۱۴۰۵
duration	int	دوره مالی از یک سال مثل ۱ یا ۲
beginDate	string	تاریخ شروع سال مالی بدون قالب مثل ۱۴۰۵۰۱۰۱
endDate	string	تاریخ پایان سال مالی بدون قالب مثل ۱۴۰۵۱۲۲۹
isClosed	string	قابل تغییر بودن سال مالی true : قابل تغییر و false : غیر قابل تغییر
isActive	string	فعال بودن سال مالی

		true : فعال و false : غیر فعال

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    " finyearId ": ۱,
    " finYear ": "۱۴۰۵",
    " duration ": ۱,
    " beginDate ": "۱۴۰۵۰۱۰۱",
    " endDate ": "۱۴۰۵۱۲۲۹",
    " isClosed ": true,
    " isActive ": true,
    }
]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۴.۲ واحد پول استفاده شده در نرم افزار حسابداری مازی :

√ Endpoint : Get : api/General/getMoneySymbol

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ندارد

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/general/getMoneySymbol

خروجی : در صورت موفق بودن درخواست ، خروجی شامل مشخصات آخرین سال مالی فعال به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
moneySymbol	string	واحد پول استفاده شده در مازی مثل تومان

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    " moneySymbol ": "تومان",
    }
]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۴.۳ استخراج نسخه فعلی نرم افزار حسابداری مازی :

√ Endpoint : Get : api/General/getMaaziVersion

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ندارد

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/general/getMaaziVersion

خروجی : در صورت موفق بودن درخواست ، خروجی شامل مشخصات آخرین سال مالی فعال به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
appVersion	string	نسخه جاری مازی مثل ۶.۰

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    " appVersion ": "۶.۰",
    }
]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۴.۴ استخراج نسخه فعلی API مازی :

√ Endpoint : Get : api/General/getApiVersion

آدرس پایه : {baseAddress}

هدر درخواست : ندارد

احراز هویت : ندارد

پارامتر ورودی : ندارد

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/general/getApiVersion

خروجی : در صورت موفق بودن درخواست ، خروجی شامل مشخصات آخرین سال مالی فعال به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
apiVersion	string	نسخه جاری API مازی مثل ۱.۰.۰

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    " apiVersion ": "۱.۰.۰",
    }
]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۴.۵.

۵. Reports : جهت گزارشگیری از نرم افزار

۵.۱ گزارش چک های دریافتی :

√ Endpoint : Get : api/Reports/v1/rcptCheques

گزارش چک های دریافت شده از مشتریان همراه با آخرین وضعیت هر چک و امکان نمایش چک های در گردش یا تمام چک ها با انواع حالت های مرتب سازی

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
showAll	bool		وضعیت نمایش تمامی چک های دریافتی حتا آنهایی که به

			اتمام رسیده اند : • true : نمایش تمامی چک ها • false : فقط چک های در گردش **نکته** : در صورت ارسال نشدن ، مقدار پیش فرض آن false می باشد
sortType	int		نوع مرتب سازی خروجی چک ها که می تواند شامل یکی از مقادیر زیر باشد : • ۰ = هیچکدام (بدون مرتب سازی) • ۱ = سررسید صعودی + مبلغ صعودی • ۲ = سررسید نزولی + مبلغ نزولی • ۳ = مبلغ صعودی + سررسید صعودی • ۴ = مبلغ نزولی + سررسید نزولی • ۵ = وضعیت چک + سررسید صعودی + مبلغ صعودی • ۶ = وضعیت چک + سررسید نزولی + مبلغ نزولی • ۷ = سریال صعودی • ۸ = سریال نزولی **نکته** : در صورت ارسال نشدن ، مقدار پیش فرض آن ۰ (صفر) می باشد
fromDueDate	string		از تاریخ سررسید : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، در نظر گرفته نمی شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱
toDueDate	string		تا تاریخ سررسید : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، در نظر گرفته نمی شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۳۱
pageSize	int		تعداد رکوردها در هر بار استخراج **نکته** : حداقل تعداد رکورها ۱۰ و حداکثر ۵۰ رکورد در هر بار درخواست استخراج خواهد شد که در صورت مشخص نشدن مقدار آن ، ۱۰ در نظر گرفته می شود
pageNumber	int		صفحه مورد نظر از رکوردها به نسبت عدد مشخص شده در pageSize

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v1/rcptCheques?showAll=false&sortType=0&fromDueDate=1405/01/01&toDueDate=1405/01/31&pageSize=10&pageNumber=1

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مشخصات چک به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
hdrDate	string	تاریخ دریافت چک
Serial	string	سریال چک دریافت شده
itmDate	string	تاریخ سر رسید چک
Amount	decimal	مبلغ چک
DLTitle	string	طرف مقابل عملیات چک
StageDescr	string	آخرین وضعیت چک
BankTitle	string	بانک صادر کننده چک
BankAccountBranch	string	شعبه صادر کننده جک
BankAccountNum	string	شماره حساب مربوطه
fkHdr	int	کلید هدر دریافت
fkItm	int	کلید آیتم دریافت

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    "hdrDate": "۱۴۰۵/۰۱/۰۱",
    "Serial": "۱۲۳۴۵",
    "itmDate": "۱۴۰۵/۰۵/۳۰",
    ... سایر فیلدها
   },
   {
    "hdrDate": "۱۴۰۵/۰۱/۱۲",
    "Serial": "۴۵۶۷۸",
    "itmDate": "۱۴۰۵/۰۷/۰۱",
    ... سایر فیلدها
   }
  ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۲ تاریخچه/کاردکس چک دریافتی :

√ Endpoint : Get : api/Reports/v1/rcptHistory

تاریخچه/کاردکس چک دریافت شده از مشتریان از زمان ثبت را نمایش می دهد

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
fkTrsHdr	int		کلید عملیات دریافتی که چک مربوط به آن است
fkTrsItm	int		کلید چک دریافتی
kind	int		برای تفکیک چک و سفته : • ۰ : چک • ۱ : سفته **نکته** : فعلن تاریخچه چک استخراج می شود

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v1/rcptHistory?fkTrsHdr=۱۲۰&fkTrsItm=۳۵۰&kind=۰

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مشخصات چک و عملیات آن به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
hdrDate	string	تاریخ دریافت
hdrNum	string	شماره دریافت
Serial	string	سریال چک دریافت شده
itmDate	string	تاریخ سر رسید چک
Amount	decimal	مبلغ چک
OtherSideDescr	string	طرف مقابل عملیات چک
CurrentStageDescr	string	وضعیت عملیات چک

StageDate	string	تاریخ عملیات چک
SayadID	string	شناسه صیاد جک

خروجی موفق :

```
{
 "status": true,
 "statusCode": ۲۰۰,
 "message": "عملیات موفق",
 "errors": null
 "data": [
  {
   "hdrDate": "۱۴۰۵/۰۱/۰۱",
   "hdrNum": ۴۵,
   "Serial": "۱۲۳۴۵",
   "itmDate": "۱۴۰۵/۰۵/۳۰",
   ... سایر فیلدها
   },
  {
   "hdrDate": "۱۴۰۵/۰۱/۲۲",
   "hdrNum": ۸۵,
   "Serial": "۸۵۴۸۹",
   "itmDate": "۱۴۰۵/۰۶/۲۱",
   ... سایر فیلدها
   }
 ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۳ گزارش چک های پرداختی :

√ Endpoint : Get : api/Reports/v1/payCheques

گزارش چک های پرداخت شده از دسته چک شرکت همراه با آخرین وضعیت هر چک و امکان نمایش چک های در گردش یا تمام چک ها با انواع حالت های مرتب سازی

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
showAll	bool		وضعیت نمایش تمامی چک های دریافتی حتا آنهایی که به اتمام رسیده اند : • true : نمایش تمامی چک ها • false : فقط چک های در گردش **نکته** : در صورت ارسال نشدن ، مقدار پیش فرض آن false می باشد
sortType	int		نوع مرتب سازی خروجی چک ها که می تواند شامل یکی از مقادیر زیر باشد : • ۰ = هیچکدام (بدون مرتب سازی) • ۱ = سررسید صعودی + مبلغ صعودی • ۲ = سررسید نزولی + مبلغ نزولی • ۳ = مبلغ صعودی + سررسید صعودی • ۴ = مبلغ نزولی + سررسید نزولی • ۵ = وضعیت چک + سررسید صعودی + مبلغ صعودی • ۶ = وضعیت چک + سررسید نزولی + مبلغ نزولی • ۷ = سریال صعودی • ۸ = سریال نزولی **نکته** : در صورت ارسال نشدن ، مقدار پیش فرض آن ۰ (صفر) می باشد
fromDueDate	string		از تاریخ سررسید : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، در نظر گرفته نمی شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱
toDueDate	string		تا تاریخ سررسید : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، در نظر گرفته نمی شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۳۱
pageSize	int		تعداد رکوردها در هر بار استخراج **نکته** : حداقل تعداد رکورها ۱۰ و حداکثر ۵۰ رکورد در هر بار درخواست استخراج خواهد شد که در صورت مشخص نشدن مقدار آن ، ۱۰ در نظر گرفته می شود

pageNumber	int		صفحه مورد نظر از رکوردها به نسبت عدد مشخص شده در pageSize

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v1/payCheques?showAll=false&sortType=0&fromDueDate=1405/01/01&toDueDate=1405/01/31&pageSize=10&pageNumber=1

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مشخصات چک به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
hdrDate	string	تاریخ پرداخت چک
Serial	string	سریال چک پرداخت شده
itmDate	string	تاریخ سر رسید چک
Amount	decimal	مبلغ چک
DLTitle	string	طرف مقابل عملیات چک
StageDescr	string	آخرین وضعیت چک
BankTitle	string	بانک صادر کننده چک
BankAccountBranch	string	شعبه صادر کننده جک
BankAccountNum	string	شماره حساب مربوطه
fkHdr	int	کلید هدر پرداخت
fkItm	int	کلید آیتم پرداخت

خروجی موفق :

```
{
 "status": true,
 "statusCode": 200,
 "message": "عملیات موفق",
 "errors": null
 "data": [
  {
   "hdrDate": "1405/01/01",
   "Serial": "12345",
   "itmDate": "1405/05/30",
   ... سایر فیلدها
   },
```

```
  {
   "hdrDate": "۱۴۰۵/۰۱/۱۲",
   "Serial": "۴۵۶۷۸",
   "itmDate": "۱۴۰۵/۰۷/۰۱",
   ... سایر فیلدها
   }
 ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۴ تاریخچه/کاردکس چک پرداختی :

√ Endpoint : Get : api/Reports/v۱/payHistory

تاریخچه/کاردکس چک پرداخت شده به مشتریان از زمان ثبت را نمایش می دهد

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
fkTrsHdr	int		کلید عملیات پرداختی که چک مربوط به آن است
fkTrsItm	int		کلید چک پرداختی
kind	int		برای تفکیک چک و سفته : • ۰ : چک • ۱ : سفته **نکته** : فعلن تاریخچه چک استخراج می شود

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v۱/payHistory?fkTrsHdr=۱۲۰&fkTrsItm=۳۵۰&kind=۰

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مشخصات چک و عملیات آن به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
hdrDate	string	تاریخ دریافت
hdrNum	string	شماره دریافت
Serial	string	سریال چک دریافت شده
itmDate	string	تاریخ سر رسید چک
Amount	decimal	مبلغ چک
OtherSideDescr	string	طرف مقابل عملیات چک
CurrentStageDescr	string	وضعیت عملیات چک
StageDate	string	تاریخ عملیات چک
SayadID	string	شناسه صیاد جک

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    "hdrDate": "۱۴۰۵/۰۱/۰۱",
    "hdrNum": ۴۵,
    "Serial": "۱۲۳۴۵",
    "itmDate": "۱۴۰۵/۰۵/۳۰",
    ... سایر فیلدها
    },
   {
    "hdrDate": "۱۴۰۵/۰۱/۲۲",
    "hdrNum": ۸۵,
    "Serial": "۸۵۴۸۹",
    "itmDate": "۱۴۰۵/۰۶/۲۱",
    ... سایر فیلدها
    }
  ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۵. گزارش صورت سود و زیان :

√ Endpoint : Get : api/Reports/v1/profitLoss

گزارش صورت سود و زیان شرکت با امکان گزارش تفکیکی برای هر شعبه برای مجموعه های چند شعبه ای جهت تفکیک سود و زیان هر شعبه

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
finyearId	int		سال مالی : فیلتر خروجی **نکته** : فعلن بدون استفاده و مقدار ۰ (صفر) ارسال شود و بصورت پیش فرض **آخرین سال مالی** لحاظ می گردد
fromDate	string		از تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، از ابتدای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
toDate	string		تا تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، تا انتهای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
branchId	int		ID شعبه : فیلتر خروجی برای یک شعبه مشخص سود و زیان را استخراج می کند **نکته** : در صورت ارسال نشدن ، تجمیع تمام شعبه ها در نظر گرفته می شود که مقدار ۰ (صفر) نیز یعنی تمام شعبه ها

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v1/profitLoss?finyearId=۰&fromDate=۱۴۰۵۰۱۰۱&toDate=۱۴۰۵۰۱۳۱&branched=۰

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی حساب های صورت سود و زیان به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
Descr	string	شرح حساب سود و زیانی
Debit	string	مبلغ بدهکار بصورت قالب بندی شده بر اساس جداکننده مبلغ
Credit	string	مبلغ بستانکار بصورت قالب بندی شده بر اساس جداکننده مبلغ

خروجی موفق :


```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
    {
      "descr": "          فروش ناخالص کالا",
      "debit": "",
      "credit": "۱۲۳,۴۵۶,۷۸۹"
    },
    {
      "descr": "كسر مي شود :",
      "debit": "",
      "credit": ""
    },
    {
      "descr": "برگشت از فروش کالا",
      "debit": "(۱,۲۳۴,۵۶۷,)",
      "credit": ""
    },
    ... سایر موارد
  ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۶ گزارش معاملات – تجمیعی روزانه :

√ Endpoint : Get : api/Reports/v1/transDaily

گزارش تجمیعی معاملات کالا به ازای هر روز یک ردیف که شامل سر جمع مبلغ و تعداد معامله برای فروش ، برگشت از فروش ، خرید و برگشت از خرید می باشد

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
finyearId	int		سال مالی : فیلتر خروجی **نکته** : فعلن بدون استفاده و مقدار ۰ (صفر) ارسال شود و بصورت پیش فرض **آخرین سال مالی** لحاظ می گردد
fromDate	string		از تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، از ابتدای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
toDate	string		تا تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، تا انتهای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
branchId	int		ID شعبه : فیلتر خروجی برای یک شعبه مشخص سود و زیان را استخراج می کند **نکته** : در صورت ارسال نشدن ، تجمیع تمام شعبه ها در نظر گرفته می شود که مقدار ۰ (صفر) نیز یعنی تمام شعبه ها
pageSize	int		تعداد رکوردها در هر بار استخراج **نکته** : حداقل تعداد رکورها ۱۰ و حداکثر ۵۰ رکورد در هر بار درخواست استخراج خواهد شد که در صورت مشخص نشدن مقدار آن ، ۱۰ در نظر گرفته می شود

pageNumber	int		صفحه مورد نظر از رکوردها به نسبت عدد مشخص شده در pageSize

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v1/transDaily?finyearId=0&fromDate=14050101&toDate=14050131&branched=0

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مقادیر معاملات، در هر روز یک ردیف به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
Year	Int	سال مالی مثال ۱۴۰۵
Date	Int	تاریخ معامله مثال ۱۴۰۵۰۱۳۱
Amount_Sale	decimal	مبلغ فروش کالا
Qty_Sale	decimal	تعداد فروش کالا
Amount_SaleReturn	decimal	مبلغ برگشت از فروش کالا
Qty_SaleReturn	decimal	تعداد برگشت از فروش کالا
Amount_Buy	decimal	مبلغ خرید کالا
Qty_Buy	decimal	تعداد خرید کالا
Amount_BuyReturn	decimal	مبلغ برگشت از خرید کالا
Qty_BuyReturn	decimal	تعداد برگشت از خرید کالا

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
    {
      "Year": ۱۴۰۵,
      "Date": ۱۴۰۵۰۱۰۱,
      "Amount_Sale": ۱۲۴۵۶۰,
      "Qty_Sale": ۱۲,
      ... سایر فیلدها
    },
    {
```

```
    "Year": ۱۴۰۵,
    "Date": ۱۴۰۵۰۱۰۲,
    "Amount_Sale": ۴۵۶۷۸۰,
    "Qty_Sale": ۸,
    ... سایر فیلدها
  },
  {
    "Year": ۱۴۰۵,
    "Date": ۱۴۰۵۰۱۰۳,
    "Amount_Sale": ۶۷۸۹۹۰,
    "Qty_Sale": ۱۱,
    ... سایر فیلدها
  },
  ... سایر موارد
 ]
}
```


و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۷ گزارش معاملات – تجمیعی ماهیانه :

√ Endpoint : Get : api/Reports/v۱/transMonthly

گزارش تجمیعی معاملات کالا به ازای هر ماه یک ردیف که شامل سر جمع مبلغ و تعداد معامله برای فروش ، برگشت از فروش ، خرید و برگشت از خرید می باشد

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
finyearId	int		سال مالی : فیلتر خروجی **نکته** : فعلن بدون استفاده و مقدار ۰ (صفر) ارسال شود و بصورت پیش فرض **آخرین سال مالی** لحاظ می گردد
fromDate	string		از تاریخ : فیلتر خروجی

			نکته : در صورت ارسال نشدن ، از ابتدای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
toDate	string		تا تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، تا انتهای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
branchId	int		ID شعبه : فیلتر خروجی برای یک شعبه مشخص سود و زیان را استخراج می کند **نکته** : در صورت ارسال نشدن ، تجمیع تمام شعبه ها در نظر گرفته می شود که مقدار ۰ (صفر) نیز یعنی تمام شعبه ها

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v1/transMonthly?finyearId=۰&fromDate=۱۴۰۵۰۱۰۱&toDate=۱۴۰۵۰۱۳۱&branched=۰

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مقادیر معاملات، در هر ماه یک ردیف به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
Year	Int	سال مالی مثال ۱۴۰۵
Month	Int	شماره ترتیبی ماه مثال ۱
MonthTitle	string	عنوان ماه به فارسی مثال فروردین
Amount_Sale	decimal	مبلغ فروش کالا
Qty_Sale	decimal	تعداد فروش کالا
Amount_SaleReturn	decimal	مبلغ برگشت از فروش کالا
Qty_SaleReturn	decimal	تعداد برگشت از فروش کالا
Amount_Buy	decimal	مبلغ خرید کالا
Qty_Buy	decimal	تعداد خرید کالا
Amount_BuyReturn	decimal	مبلغ برگشت از خرید کالا
Qty_BuyReturn	decimal	تعداد برگشت از خرید کالا

خروجی موفق :

```
{
```

```
"status": true,
"statusCode": ۲۰۰,
"message": "عملیات موفق",
"errors": null
"data": [
  {
    "Year": ۱۴۰۵,
    "Month": ۱,
    "MonthTitle": "فروردین",
    "Amount_Sale": ۱۲۴۵۶۷۸۹۰,
    "Qty_Sale": ۱۲۳,
    ... سایر فیلدها
  },
  {
    "Year": ۱۴۰۵,
    "Month": ۲,
    "MonthTitle": "اردیبهشت",
    "Amount_Sale": ۴۵۶۷۸۹۱۲۰,
    "Qty_Sale": ۸۹,
    ... سایر فیلدها
  },
  {
    "Year": ۱۴۰۵,
    "Month": ۳,
    "MonthTitle": "خرداد",
    "Amount_Sale": ۶۷۸۹۹۸۴۰۰,
    "Qty_Sale": ۱۱۰,
    ... سایر فیلدها
  },
  ... سایر موارد
]
}
```


و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۸ گزارش معاملات – تجمیعی فصلی :

√ Endpoint : Get : api/Reports/v1/transSeason

گزارش تجمیعی معاملات کالا به ازای هر فصل یک ردیف که شامل سر جمع مبلغ و تعداد معامله برای فروش ، برگشت از فروش ، خرید و برگشت از خرید می باشد

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
finyearId	int		سال مالی : فیلتر خروجی **نکته** : فعلن بدون استفاده و مقدار ۰ (صفر) ارسال شود و بصورت پیش فرض **آخرین سال مالی** لحاظ می گردد
fromDate	string		از تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، از ابتدای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
toDate	string		تا تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، تا انتهای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
branchId	int		ID شعبه : فیلتر خروجی برای یک شعبه مشخص سود و زیان را استخراج می کند **نکته** : در صورت ارسال نشدن ، تجمیع تمام شعبه ها در نظر گرفته می شود که مقدار ۰ (صفر) نیز یعنی تمام شعبه ها

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v1/transSeason?finyearId=۰&fromDate=۱۴۰۵۰۱۰۱&toDate=۱۴۰۵۰۱۳۱&branched=۰

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مقادیر معاملات، در هر فصل یک ردیف به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
Year	Int	سال مالی مثال ۱۴۰۵
Season	Int	شماره ترتیبی فصل مثال ۱
SeasonTitle	string	عنوان فصل به فارسی مثال بهار
Amount_Sale	decimal	مبلغ فروش کالا
Qty_Sale	decimal	تعداد فروش کالا
Amount_SaleReturn	decimal	مبلغ برگشت از فروش کالا
Qty_SaleReturn	decimal	تعداد برگشت از فروش کالا
Amount_Buy	decimal	مبلغ خرید کالا
Qty_Buy	decimal	تعداد خرید کالا
Amount_BuyReturn	decimal	مبلغ برگشت از خرید کالا
Qty_BuyReturn	decimal	تعداد برگشت از خرید کالا

خروجی موفق :


```
{
 "status": true,
 "statusCode": ۲۰۰,
 "message": "عملیات موفق",
 "errors": null
 "data": [
  {
   "Year": ۱۴۰۵,
   "Season": ۱,
   "SeasonTitle": "بهار",
   "Amount_Sale": ۱۲۴۵۶۷۸۹۰,
   "Qty_Sale": ۱۲۳,
   ... سایر فیلدها
  },
  {
   "Year": ۱۴۰۵,
   "Month": ۲,
   "MonthTitle": "تابستان",
   "Amount_Sale": ۴۵۶۷۸۹۱۲۰,
   "Qty_Sale": ۸۹,
   ... سایر فیلدها
  },
  ... سایر موارد
```

```
 ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۹ گزارش اشخاص :

√ Endpoint : Get : api/Reports/v1/customers

گزارش اشخاص بصورت سرجمع تراکنش با قابلیت حالت های مختلف نمایش اشخاص و انواع مرتب سازی

آدرس پایه : {baseAddress}

احراز هویت : درخواست باید شامل هدر زیر باشد :

Authorization: Bearer <JWT Token>

توکن تولید شده در مرحله اول در هدر درخواست قرار می گیرد

پارامتر ورودی : ارسال پارامترها بصورت Query در URL

نام پارامتر	نوع	اجبار	شرح
finyearId	int		سال مالی : فیلتر خروجی **نکته** : فعلن بدون استفاده و مقدار ۰ (صفر) ارسال شود و بصورت پیش فرض **آخرین سال مالی** لحاظ می گردد
fromDate	string		از تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، از ابتدای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
toDate	string		تا تاریخ : فیلتر خروجی **نکته** : در صورت ارسال نشدن ، تا انتهای سال مالی در نظر گرفته می شود **نکته** : قالب ارسالی باید به شکل yyyy/MM/dd یا yyyyMMdd باشد. مثال مقدار ارسالی ۱۴۰۵/۰۱/۰۱ یا ۱۴۰۵۰۱۰۱
remainKind	int		نوع فیلتر بر اساس گردش مالی شخص که شامل یکی از مقادیر زیر است : • ۰ = اشخاص دارای گردش حساب

			• ۱ = اشخاص دارای مانده حساب • ۲ = اشخاص دارای مانده بدهکار • ۳ = اشخاص دارای مانده بستانکار • ۴ = اشخاص بدون مانده حساب • ۵ = اشخاص بدون گردش حساب • ۶ = تمام اشخاص
sortType	int		نوع مرتب سازی خروجی که شامل یکی از مقادیر زیر است : • ۰ = هیچکدام • ۱ = مانده بدهکار نزولی + مانده بستانکار نزولی • ۲ = مانده بدهکار صعودی + مانده بستانکار صعودی • ۳ = گردش بدهکار نزولی + گردش بستانکار نزولی • ۴ = گردش بدهکار صعودی + گردش بستانکار صعودی
customerName	int		نام شخص : فیلتر خروجی اشخاصی که شامل این مقدار باشند، استخراج می شوند **نکته** : در صورت ارسال نشدن ، همه اشخاص استخراج می شوند که مقدار خالی نیز یعنی تمام اشخاص
pageSize	int		تعداد رکوردها در هر بار استخراج **نکته** : حداقل تعداد رکورها ۱۰ و حداکثر ۵۰ رکورد در هر بار درخواست استخراج خواهد شد که در صورت مشخص نشدن مقدار آن ، ۱۰ در نظر گرفته می شود
pageNumber	int		صفحه مورد نظر از رکوردها به نسبت عدد مشخص شده در pageSize

بدنه درخواست (JSON) : ندارد

مثال :

{baseAddress}api/Reports/v1/customers?finyearId=0&fromDate=1405/01/01&toDate=1405/01/31&remainKind=1&sortType=0&customerName=&pageSize=10&pageNumber=1

خروجی : در صورت موفق بودن درخواست ، خروجی شامل لیستی از مشخصات چک به شرح زیر می باشد که بصورت json در فیلد data در قالب خروجی قرار می گیرد :

نام فیلد	نوع	شرح
DLTitle	string	نام شخص / شرکت
SumDebit	decimal	گردش بدهکار شخص
SumCredit	decimal	گردش بستانکار شخص
Remain	decimal	مانده شخص
Det	string	ماهیت مانده :

		• بد = بدهکار • بس = بستانکار

خروجی موفق :

```
{
  "status": true,
  "statusCode": ۲۰۰,
  "message": "عملیات موفق",
  "errors": null
  "data": [
   {
    "DLTitle": "آقای مشتری اول",
    "SumDebit": ۱۲۰۰۰۰,
    "SumCredit": ۱۱۰۰۰۰,
    "Remain": ۱۰۰۰۰,
    "Det": "بد",
   },
   {
    "DLTitle": "خانم مشتری دوم",
    "SumDebit": ۲۵۰۰۰۰,
    "SumCredit": ۳۸۰۰۰۰,
    "Remain": ۱۳۰۰۰۰,
    "Det": "بس",
   }
  ]
}
```

و در صورت خطا ، مشخصات آن در قالب کلی پاسخ برگشتی و در errors مقدار دهی می شود که شامل شرح و کد خطا می باشد

۵.۱۰.