راهنمای کامل Pishgaman API

مرجع توسعه‌دهندگان به زبان فارسی

راهنمای احراز هویت، کالا و موجودی، واحدها و نسبت‌ها، تفصیلی‌ها، مشترکان و فاکتورها؛ منطبق با نصب‌کننده نسخه 1.0.11. این سند قرارداد درخواست و پاسخ، نمونه‌ها، پیش‌نیازهای سرویس و رسیدگی به خطاها را برای تیم‌های یکپارچه‌سازی توضیح می‌دهد.

مشخصه	مقدار
نسخه مستند	1.0.11
نسخه نصب‌کننده API	1.0.11
تاریخ تدوین	2026-11-17
بستر	ASP.NET Core 5 / JWT / Swagger
پوشش	35 ترکیب متد و مسیر عمومی در کنترلرهای فعلی

فهرست مطالب

عنوان بخش	صفحه
تغییرات این ویرایش	4
معرفی	4
آدرس پایه و Swagger	4
احراز هویت	4
دریافت توکن	4
ارسال توکن	5
قرارداد عمومی پاسخ‌ها	5
فهرست سریع endpointها	6
جزئیات endpointها	7
1. احراز هویت	7
2. کالاها	7
3. تفصیلی‌ها و فروش تفصیلی	13
4. اشتراک‌ها و فروش اشتراک	13
5. فاکتور	14
6. معرفی و ویرایش تفصیلی	17
7. معرفی و ویرایش مشترک	19
8. فاکتورهای فروش مشترک با سربرگ و ردیف‌ها	20
9. به‌روزرسانی کالا از ووکامرس به پیشگامان	21
مدل‌های داده	22
CustomerAccountReportQuery	22
CustomerStatementQuery	22
CustomerBalanceDto	23
CustomerStatementDto	23
CustomerStatementEntryDto	24
CustomerDto	24
CustomerInfoDto	24
CustomerSaleMasterDto	25
CustomerWriteDto	25
FactorDto	26
DetailFactor	26
ProductImageDto	27
ProductImageResultDto	27
ProductsColorsDto	27
ProductsDto	28
ChangedProductDto	29
ChangedProductAcknowledgementDto	29
ProductWithUnitsDto	30

عنوان بخش	صفحه
ProductsGroupDto	30
ProductsSizesColorsDto	31
ProductsSizesDto	31
ProductUnitDto	31
ProductUploadDto	32
ProductUploadResultDto	32
ResultDto	32
ResultDto<T>	32
SubscriberDto	33
SubscriberSaleSumDto	33
SubscriberSaleMasterDto	34
SubscriberSaleDetailDto	34
SubscriberInvoiceDto	35
SubscriberInvoiceMasterDto	35
SubscriberInvoiceDetailDto	36
SubscriberWriteDto	36
Tokens	36
WooProductDto	37
نمونه استفاده با cURL	37
نکات اجرایی	38
گزارش مانده حساب و گردش حساب تفصیلی	38
نکات عمومی	41
دسترسی‌ها و پیش‌نیازهای سرویس	41
راهنمای خطا و استفاده ایمن	42
انتشار و به‌روزرسانی نسخه 1.0.11	42

تغییرات این ویرایش

نسخه 1.0.11: گزارش مانده حساب با کد اختیاری برای همه تفصیلی‌های گروه مجاز و گردش حساب دوره‌ای با کد الزامی؛ تاریخ‌های شمسی اختیاری، مانده قبل از بازه، جمع گردش‌ها، مانده جاری و صفحه‌بندی گردش. دو مجوز مستقل با پیش‌فرض خاموش و کنترل گروه در خود سرویس اعمال می‌شوند.

PioService و حسابداری مقصد باید با گزارش‌های جدید و مبدأ مستقل API سازگار باشند؛ نصب‌کننده فقط API را ارتقا می‌دهد. قرارداد changeToken و تأیید ذخیره ووکامرس، محاسبه مالیات بعد از تخفیف ردیفی و بدون اثر تخفیف انتهای فاکتور، واحدها و نسبت‌ها و اولویت CustomerCode حفظ شده‌اند.

معرفی

این API یک سرویس ASP.NET Core 5 برای اتصال به سرویس عملیاتی Pio و دریافت/ارسال اطلاعات کالا، مشتری، اشتراک، فروش و فاکتور است. بیشتر endpointها با JWT محافظت شده‌اند و پاسخ‌ها با قالب مشترک ResultDto برگردانده می‌شوند.

آدرس پایه و Swagger

آدرس پایه بسته به محل نصب سرویس تعیین می‌شود:

```
http://{host}:{port}/
```

در صورت فعال بودن سرویس، مستند Swagger از مسیر زیر در دسترس است:

```
/swagger
/swagger/v1/swagger.json
```

احراز هویت

برای استفاده از endpointهای محافظت‌شده باید ابتدا توکن JWT دریافت شود و سپس در header درخواست‌ها ارسال شود.

دریافت توکن

به دلیل نحوه تعریف route در UsersController، مسیر عملیاتی احراز هویت به صورت زیر است:

```
POST /api/Users/api/authenticate
Content-Type: application/json
```

بدنه درخواست:

```
{
  "name": "API_USER",
  "password": "API_PASSWORD",
  "clientId": "client1"
}
```

فیلد clientId اختیاری است. اگر ارسال شود، تنظیمات اتصال از بخش Clients:{clientId} خوانده می‌شود؛ در غیر این صورت تنظیمات پیش‌فرض ConnectionSetting استفاده می‌شود.

پاسخ موفق:

```
{
  "token": "eyJhbGciOiJIUzI1NiIs..."
}
```

پاسخ ناموفق:

```
{
  "isSuccess": false,
 "message": "نام کاربری یا کلمه عبور صحیح نمی باشد!",
  "autoNumber": ""
}
```

ارسال توکن

در تمام endpointهای محافظت‌شده، header زیر باید ارسال شود:

```
Authorization: Bearer {token}
```

قرارداد عمومی پاسخ‌ها

بیشتر endpointهای دریافت اطلاعات، پاسخ را در قالب زیر برمی‌گردانند:

```
{
  "isSuccess": true,
 "message": "دریافت با موفقیت انجام شد",
  "data": []
}
```

در خطاهای عملیاتی نیز معمولا کد HTTP همچنان 200 OK است و وضعیت واقعی در isSuccess مشخص می‌شود:

```
{
  "isSuccess": false,
 "message": "شرح خطا",
  "data": null
}
```

endpointهای ثبت/حذف فاکتور از قالب زیر استفاده می‌کنند:

```
{
  "isSuccess": true,
 "message": "ثبت در سرور انجام شد",
  "autoNumber": "12345"
}
```

فهرست سریع endpointها

گروه	متد	مسیر	نیاز به توکن	کاربرد
Auth	POST	/api/Users/api/authenticate	خیر	دریافت JWT
Product	GET	/api/Product	بله	دریافت همه کالاها
Product	GET	/api/Product/{goodCode}/units	بله	واحدهای یک کالا با نسبت به واحد جزء
Product	GET	/api/Product/{GoodCode}/{Publisher}/{Writer}/{Translator}/{Grouppath}/{BaseGrouppath}/{PositiveQuantity}	بله	فیلتر کالا
Product	GET	/api/Product/{FromCode}/{Count}	بله	دریافت صفحه‌ای کالاها
Product	GET	/api/Product/woo/{FromCode}/{Count}	بله	دریافت کالا مناسب WooCommerce با سایز/رنگ
Product	GET	/api/ProductSize	بله	دریافت سایزها
Product	GET	/api/ProductColor	بله	دریافت رنگ‌ها
Product	GET	/api/ProductSizeColor/{OnCurrentAnbar}	بله	دریافت ترکیب کالا/سایز/رنگ
Product	GET	/api/ProductGroup	بله	دریافت گروه‌های کالا
Product	GET	/api/ChangedProduct/{Count}	بله	دریافت کالاهای تغییرکرده
Product	POST	/api/ChangedProduct/ack	بله	تأیید نسخه ذخیره‌شده و حذف امن از صف تغییرات
Product	GET	/api/ProductImage/{FromCode}/{Count}	بله	دریافت تصاویر کالا
Customer	GET	/api/Customer	بله	دریافت تفصیلی‌ها
Customer	GET	/api/Customer/{CustomerID}	بله	دریافت تفصیلی با شناسه
Customer	GET	/api/CustomerSale	بله	دریافت فروش همه تفصیلی‌ها
Customer	GET	/api/CustomerSale/{CustomerID}	بله	دریافت فروش یک تفصیلی
Subscriber	GET	/api/Subscribers	بله	دریافت اشتراک‌ها
Subscriber	GET	/api/Subscribers/{subscriberCode}	بله	دریافت اشتراک با کد
Subscriber	GET	/api/SubscriberSaleSum	بله	جمع فروش اشتراک‌ها
Subscriber	GET	/api/SubscriberSaleMaster	بله	سرجمع/سربرگ فروش اشتراک‌ها
Subscriber	GET	/api/SubscriberSaleDetail	بله	جزئیات فروش اشتراک‌ها
Subscriber	GET	/api/SubscriberSaleDetail/{SubscriberCode}	بله	جزئیات فروش یک اشتراک
Invoice	POST	/api/SubmitInvoice	بله	ثبت سفارش یا فروش قطعی
Invoice	GET	/api/RemoveInvoice/{AutoNumber}/{FinalSale}	بله	حذف سفارش/فاکتور
API	POST	/api/Customer	بله	معرفی تفصیلی
API	PUT	/api/Customer/{code}	بله	ویرایش تفصیلی
API	GET	/api/CustomerInfo	بله	اطلاعات تفصیلی و مانده

گروه	متد	مسیر	نیاز به توکن	کاربرد
API	GET	/api/CustomerBalance	بله	مانده یک یا همه تفصیلی‌های گروه مجاز
API	GET	/api/CustomerStatement	بله	گردش حساب دوره‌ای یک تفصیلی
API	POST	/api/Subscribers	بله	معرفی مشترک
API	PUT	/api/Subscribers/{code}	بله	ویرایش مشترک
API	GET	/api/SubscriberSaleInvoices	بله	فاکتورهای فروش با جزئیات
API	GET	/api/SubscriberSaleInvoices/{subscriberCode}	بله	فاکتورهای یک مشترک
API	POST	/api/Product/update-from-woocommerce	بله	آپلود اطلاعات کالا

جزئیات endpointها

1. احراز هویت

```
POST /api/Users/api/authenticate
```

ورودی:

فیلد	نوع	اجباری	توضیح
name	string	بله	نام کاربری API
password	string	بله	کلمه عبور API
clientId	string	خیر	شناسه مشتری برای انتخاب تنظیمات اتصال چندمشتری

خروجی موفق: Tokens

فیلد	نوع	توضیح
token	string	توکن JWT برای ارسال در header درخواست‌های بعدی

2. کالاها

دریافت همه کالاها

در دریافت عادی، فیلترشده و مرحله‌ای کالاها و مسیر /api/Product/woo/{FromCode}/{Count} سه فیلد زیر نیز ارسال می‌شوند. فیلدهای واحد در مسیر تغییرات کالا ارسال نمی‌شوند؛ این مسیر برای تأیید ذخیره دارای changeToken است.

فیلد خروجی	نوع	توضیح
unit	int	کد واحد اصلی کالا از ستون goods.unit
price_unit	int	کد واحد قیمت‌گذاری از ستون goods.price_unit
price_unit_ratio	decimal	تعداد واحد اصلی در یک واحد قیمت‌گذاری؛ بر اساس نسبت‌های goods_sub_unit و جهت تبدیل استاندارد dbo.GetQuantity، با دقت decimal

اگر دو واحد برابر باشند، هر سه فیلد همچنان ارسال می‌شوند و نسبت برابر 1 است. اگر واحد اصلی عدد و واحد قیمت‌گذاری کارتن 12تایی باشد، نسبت 12 است؛ در حالت معکوس نسبت 1/12 خواهد بود. برای تبدیل تعداد از واحد قیمت‌گذاری به واحد اصلی،

مقدار را در price_unit_ratio ضرب کنید. این فیلدها کد واحد هستند، نه نام واحد. نرخ و موجودی خروجی با افزودن این فیلدها تغییر یا تبدیل نمی‌شوند. API و سرویس پیشگامان هر دو باید به‌روزرسانی شوند؛ در صورت نبود ستون‌های واحد در پاسخ سرویس قدیمی، API پیام صریح به‌روزرسانی سرویس برمی‌گرداند.

```
GET /api/Product
Authorization: Bearer {token}
```

خروجی: <<ResultDto<List<ProductWithUnitsDto

فیلتر کالا

```
GET /api/Product/{GoodCode}/{Publisher}/{Writer}/{Translator}/{Grouppath}/{BaseGrouppath}/{PositiveQuantity}
```

پارامترها:

پارامتر	نوع	توضیح
GoodCode	string	کد کالا
Publisher	string	ناشر
Writer	string	نویسنده/ویژگی خاص
Translator	string	مترجم/ویژگی دیگر
Grouppath	string	مسیر گروه کالا
BaseGrouppath	string	مسیر گروه پایه
PositiveQuantity	int	فقط کالاهای دارای موجودی مثبت

دریافت صفحه‌ای کالاها

```
GET /api/Product/{FromCode}/{Count}
```

پارامتر	نوع	توضیح
FromCode	string	شروع از کد کالا
Count	int	تعداد رکورد در پاسخ

در صفحه‌بندی معمول کالا، FromCode شامل خود کد نیز می‌شود؛ برای صفحه بعد، بیشترین کد پردازش‌شده را یک واحد افزایش دهید. این رفتار با نشانگر انحصاری تصاویر تفاوت دارد.

دریافت کالاهای WooCommerce

```
GET /api/Product/woo/{FromCode}/{Count}
```

خروجی: <<ResultDto<List<WooProductDto

این endpoint کالاها را همراه با اطلاعات سایز و رنگ در قالب مناسب همگام‌سازی با WooCommerce برمی‌گرداند.

دریافت واحدهای یک کالا با نسبت به واحد جزء

```
GET /api/Product/1001/units
Authorization: Bearer {token}
```

پارامتر مسیر goodCode کد عددی و مثبت کالا است. پاسخ در قالب ResultDto<List<ProductUnitDto>> برمی‌گردد و هر ردیف دقیقا چهار فیلد زیر دارد:

فیلد	نوع	توضیح
good_code	int	کد کالا
price_unit	int	کد واحد جزء محاسبه‌شده برای همان کالا
unit	int	کد واحد این ردیف؛ شامل واحد اصلی و واحدهای فرعی
ratio	decimal	هر یک واحد این ردیف چند واحد جزء است

```
{
  "isSuccess": true,
 "message": "واحدهای کالا با موفقیت دریافت شد.",
  "data": [
    {
      "good_code": 1001,
      "price_unit": 1,
      "unit": 1,
      "ratio": 1
    },
    {
      "good_code": 1001,
      "price_unit": 1,
      "unit": 2,
      "ratio": 6
    },
    {
      "good_code": 1001,
      "price_unit": 1,
      "unit": 4,
      "ratio": 48
    }
  ]
}
```

نمونه فرضی بالا واحد جزء عدد، بسته 6تایی و کارتن 48تایی را نشان می‌دهد. نسبت ذخیره‌شده در جدول واحد فرعی مستقیما صادر نمی‌شود. اگر یک واحد اصلی شامل 48 عدد و 8 بسته باشد، نسبت بسته به عدد از 48 / 8 = 6 محاسبه می‌شود.

واحد جزء از میان واحدهای تعریف‌شده همان کالا انتخاب می‌شود: واحد با بیشترین تعداد در یک واحد اصلی. در برابری نسبت‌ها، اولویت با واحد قیمت‌گذاری تنظیم‌شده، سپس واحد اصلی و سپس کد کوچک‌تر است. price_unit در این مسیر نقش واحد جزء خروجی را دارد و الزاما برابر ستون goods.price_unit نیست؛ معنای فیلدهای مسیرهای قبلی تغییر نکرده است. واحد اصلی همیشه در فهرست وجود دارد؛ کالای تک‌واحدی یک ردیف با نسبت 1 دارد. تعریف تکراری یک واحد با نسبت یکسان ادغام می‌شود، ولی نسبت متناقض، نامشخص یا غیرمثبت خطا می‌دهد و حدس زده نمی‌شود.

دسترسی کالا به رده کالای API اتصال انتخاب‌شده محدود است، مانند دریافت معمول کالاها. کد نامعتبر پاسخ 400، کالای ناموجود یا خارج از محدوده پاسخ 404 و خطای سرویس یا تنظیم نسبت پاسخ 502 دارد. API و سرویس پیشگامان هر دو باید نسخه جدید را اجرا کنند؛ پاسخ سرویس قدیمی به‌جای نتیجه اشتباه، خطای صریح نیاز به به‌روزرسانی سرویس می‌دهد. این مسیر فقط خواندنی است و دیتابیس را تغییر نمی‌دهد.

تفاوت دو خروجی مربوط به واحدها

مسیر	فیلد نسبت	جهت تبدیل
دریافت معمول کالا یا مسیر woo	price_unit_ratio	تعداد واحد اصلی goods.unit در یک واحد قیمت‌گذاری goods.price_unit
دریافت واحدهای یک کالا	ratio	تعداد واحد جزء محاسبه‌شده در یک unit از همان ردیف

این دو نسبت لزوما برابر نیستند. برای کالایی با واحد اصلی کارتن 48تایی و واحد قیمت‌گذاری عدد، خروجی معمول نسبت 1/48 را دارد، ولی ردیف کارتن در مسیر واحدها نسبت 48 دارد. در مسیر واحدها، price_unit همیشه واحد جزء محاسبه‌شده است؛ در دریافت معمول کالا همان واحد قیمت‌گذاری تنظیم‌شده است. کد کالا در خروجی معمول code و در مسیر واحدها good_code است. مسیر تغییرات کالا دارای changeToken و تأیید جداگانه ذخیره است؛ فیلدهای جدید واحد به آن افزوده نشده‌اند.

برای تست در Postman، درخواست Get Product Units And Ratios زیر گروه Product اضافه شده است. ابتدا Authenticate را اجرا کنید؛ سپس متغیر product_units_good_code را به کد مثبت کالای موردنظر تغییر دهید. متغیر good_code قبلی برای فیلتر کالاها مستقل مانده و صفر آن را برای این درخواست استفاده نکنید. آدرس پایه و توکن از base_url و jwt_token خوانده می‌شوند.

دریافت سایزها

```
GET /api/ProductSize
```

خروجی: <<ResultDto<List<ProductsSizesDto

دریافت رنگ‌ها

```
GET /api/ProductColor
```

خروجی: <<ResultDto<List<ProductsColorsDto

دریافت ترکیب کالا، سایز و رنگ

```
GET /api/ProductSizeColor/{OnCurrentAnbar}
```

پارامتر	نوع	توضیح
OnCurrentAnbar	int	تعیین مبنای موجودی انبار جاری

خروجی: <<ResultDto<List<ProductsSizesColorsDto

دریافت گروه‌های کالا

```
GET /api/ProductGroup
```

خروجی: <<ResultDto<List<ProductsGroupDto

دریافت کالاهای تغییرکرده

```
GET /api/ChangedProduct/{Count}
```

پارامتر	نوع	توضیح
Count	int	تعداد کالاهای تغییرکرده؛ فقط 1 تا 100

خروجی: <<ResultDto<List<ChangedProductDto>>

دریافت تغییرات دیگر صف را حذف نمی‌کند. هر کالا علاوه بر مشخصات و MainPrice و Price، فیلد changeToken دارد؛ آن را رشته‌ای غیرقابل تفسیر و وابسته به همان نسخه کالا بدانید. فیلدهای واحد به این مدل افزوده نشده‌اند. نمونه زیر خلاصه پاسخ است؛ مدل کامل در فهرست مدل‌ها آمده است.

```
{
  "isSuccess": true,
  "message": "...",
  "data": [
    {
      "code": 2640,
      "name": "Sample product",
      "mainPrice": 3850000,
      "price": 3750000,
      "quantity": 5,
      "changeToken": "0x0000000000000001"
    }
  ]
}
```

تأیید ذخیره موفق کالاهای تغییریافته

```
POST /api/ChangedProduct/ack
Authorization: Bearer {token}
Content-Type: application/json
```

```
[
  {
    "code": 2640,
    "changeToken": "0x0000000000000001"
  }
]
```

بدنه آرایه‌ای با 1 تا 100 عضو ChangedProductAcknowledgementDto است. code باید مثبت و changeToken باید همان رشته دریافتی با قالب 0x و 16 رقم هگزادسیمال باشد. فقط پس از موفقیت همه ذخیره‌های همان کالا، از جمله تنوع‌ها و دسته‌بندی‌های انتخاب‌شده، تأیید بفرستید. خطای ذخیره یا کالای ردشده را تأیید نکنید. در افزونه این قرارداد برای حالت عادی و سایز و رنگ پیاده شده است.

```
{
  "isSuccess": true,
  "message": "Saved product versions acknowledged.",
  "data": 1
}
```

data تعداد ردیف‌های حذف‌شده از صف است، نه تعداد تغییرات قیمت. نتیجه صفر در تکرار تأیید، یا وقتی نسخه جدیدتری در صف وجود دارد، طبیعی است؛ نسخه جدید باقی می‌ماند و در دریافت بعدی پردازش می‌شود. در قطع ارتباط یا شکست ACK، دوباره دریافت و ذخیره کنید؛ تکرار تأیید بی‌خطر است. این مسیر فقط صف را تغییر می‌دهد و خودش محصول ووکامرس را ذخیره نمی‌کند.

ورودی نامعتبر 400، توکن نامعتبر 401، خطای سرویس در تأیید 502 و نبود پشتیبانی سرویس 503 است. در GET نیز isSuccess و message را بررسی کنید؛ خطای تجاری ممکن است HTTP 200 داشته باشد. برای نسخه جدید، سرویس سازگار پروژه ACC Before 14050613 و اجرای بازسازی دستی دیتابیس از داخل حسابداری برای ستون goods_web_changes_log.change_version لازم است؛ ساخت Schema در اجرای عادی انجام نمی‌شود. فرمان‌های دریافت و تأیید جدید به‌ترتیب 6005 و 6004 هستند تا سرویس قدیمی به‌اشتباه صف را مصرف نکند.

هنگام استقرار ابتدا فراخوانی خودکار را متوقف و پشتیبان تهیه کنید، سپس حسابداری و سرویس سازگار، بازسازی دستی دیتابیس، API و افزونه HJ Connector نسخه 3.1.5 را هماهنگ نصب کنید. لینک اصلی را دوباره از افزونه کپی کنید و حروف Secret را تغییر ندهید. نصب‌کننده API سرویس حسابداری را جایگزین نمی‌کند. مصرف‌کننده‌های قدیمی و مستقل را هم‌زمان روی همین صف اجرا نکنید. لاگ سرویس در WordPressSync/wordpress-sync.log کنار فایل اجرایی است و مقادیر Query از گزارش خطا حذف می‌شوند. وردپرس نتیجه آخرین تلاش را در پنل بروزرسانی و خطا را در لاگ PHP ثبت می‌کند.

دریافت تصاویر کالا

```
GET /api/ProductImage/{FromCode}/{Count}
```

پارامتر	نوع	توضیح
FromCode	string	شروع از کد کالا
Count	int	تعداد رکورد

خروجی: <<ResultDto<List<ProductImageDto

نمونه داده تصویر:

```
{
  "code": 1001,
  "imageBytes": "/9j/4AAQSkZJRgABAQ...",
  "imageHash": "g1R8nqY..."
}
```

imageBytes و imageHash با Base64 برگردانده می‌شوند.

خروجی این مسیر ProductImageResultDto است و علاوه بر isSuccess و message و data، آمار لازم برای پیشرفت مرحله‌ای را دارد. FromCode در نسخه فعلی یک نشانگر انحصاری است: فقط کالاهایی با کد بزرگ‌تر از آن برمی‌گردند. برای شروع از ابتدا مقدار 0 ارسال کنید و برای ادامه، بزرگ‌ترین code دریافت‌شده در مرحله موفق قبلی را بفرستید؛ آن کد را یک واحد افزایش ندهید.

Count تعداد کالاهای دارای تصویر در هر مرحله است، نه الزاما تعداد تصاویر. اگر یک کالا چند تصویر داشته باشد، همه آن تصاویر می‌آیند و تعداد ردیف‌های data می‌تواند از Count بیشتر باشد. برای کنترل حجم پاسخ Count کوچک و مثبت انتخاب کنید؛ مقدار غیرمثبت در سرویس فعلی به معنی بدون محدودیت است.

فیلد آماری	نوع	معنی
totalProducts	int	تعداد کل کالاهای دارای تصویر در محدوده تنظیمات
totalImages	int	تعداد کل ردیف‌های تصویر در همان محدوده
processedProducts	int	تعداد کالاهای تا نشانگر ورودی، قبل از این صفحه
processedImages	int	تعداد تصاویر تا نشانگر ورودی، قبل از این صفحه
pageProducts	int	تعداد کد کالای متمایز دارای تصویر معتبر در پاسخ فعلی
pageImages	int	تعداد تصاویر معتبر در پاسخ فعلی
cursorMode	string	مقدار exclusive برای ادامه بعد از کد آخر

برای نمایش درصد تقریبی عبور از محدوده تصاویر، مجموع processedImages و pageImages را بر totalImages تقسیم کنید؛ اگر مخرج صفر است درصد نسازید. این آمار عبور از داده مبدأ است، نه تأیید ذخیره تصویر در وردپرس. کلاینت باید تعداد ذخیره‌شده، ردشده و خطادار را جداگانه ثبت کند و نشانگر را فقط پس از پردازش موفق مرحله جلو ببرد. تصاویر خالی در API فیلتر می‌شوند؛ آمار مبدأ ممکن است ردیف‌های نامعتبر را نیز بشمارد. با سرویس قدیمی فاقد جدول آمار، مقادیر کل ممکن است صفر باشند؛ درصد دقیق را از صفر حدس نزنید.

3. تفصیلی‌ها و فروش تفصیلی

دریافت تفصیلی‌ها

```
GET /api/Customer
```

خروجی: <<ResultDto<List<CustomerDto>>

دریافت تفصیلی با شناسه

```
GET /api/Customer/{CustomerID}
```

پارامتر	نوع	توضیح
CustomerID	string	شناسه تفصیلی

دریافت فروش همه تفصیلی‌ها

```
GET /api/CustomerSale
```

خروجی: <<ResultDto<List<CustomerSaleMasterDto>>

دریافت فروش یک تفصیلی

```
GET /api/CustomerSale/{CustomerID}
```

4. اشتراک‌ها و فروش اشتراک

دریافت اشتراک‌ها

```
GET /api/Subscribers
```

خروجی: <<ResultDto<List<SubscriberDto>>

دریافت اشتراک با کد

```
GET /api/Subscribers/{subscriberCode}
```

پارامتر	نوع	توضیح
subscriberCode	int	کد اشتراک

دریافت جمع فروش اشتراک‌ها

```
GET /api/SubscriberSaleSum
```

خروجی: <<ResultDto<List<SubscriberSaleSumDto>>

دریافت سربرگ فروش اشتراک‌ها

```
GET /api/SubscriberSaleMaster
```

خروجی: <<ResultDto<List<SubscriberSaleMasterDto

دریافت جزئیات فروش اشتراک‌ها

```
GET /api/SubscriberSaleDetail
GET /api/SubscriberSaleDetail/{SubscriberCode}
```

خروجی: <<ResultDto<List<SubscriberSaleDetailDto

5. فاکتور

ثبت سفارش یا فروش قطعی

```
POST /api/SubmitInvoice
Authorization: Bearer {token}
Content-Type: application/json
```

بدنه درخواست: FactorDto

```
{
  "finalSale": false,
  "fare": 25000,
 "addressInfo": "تهران، خیابان نمونه",
 "dsc2": "توضیحات سفارش",
  "CustomerCode": 1001,
  "customerMobile": "09120000000",
 "customerName": "نام مشتری",
 "customerAddress": "آدرس مشتری",
 "customerInfo": "اطلاعات تکمیلی مشتری",
  "customerBirthDate": "1370/01/01",
  "discunt": 0,
  "paid": false,
  "detail": [
    {
      "code": 1001,
      "quantity": 2,
      "price": 150000,
      "discuntPercent": 0,
      "dsc": "",
      "sizeCode": 0,
      "colorCode": 0,
      "aggregateGoodSizeColorBarcode": "",
      "barcode": "6260000000000"
    }
  ]
}
```

نکته: در مدل فعلی، فیلدهای DetailFactor به صورت field تعریف شده‌اند، نه property. با Newtonsoft.Json این ساختار قابل deserialize است.

رفتار finalSale:

مقدار	نتیجه
false	ثبت سفارش
true	ثبت فروش/تحویل قطعی

پاسخ موفق:

```
{
  "isSuccess": true,
 "message": "ثبت در سرور انجام شد",
  "autoNumber": "12345"
}
```

مالیات ردیف و کنترل مبلغ مالیات سربرگ

در بدنه POST /api/SubmitInvoice، پارامتر اختیاری TaxAmount در سطح سربرگ مبلغ کل مالیات مورد انتظار را دریافت می‌کند. پارامتر اختیاری VatPercent در هر عضو Detail درصد مالیات است، نه مبلغ. درصد از صفر تا صد با حداکثر دو رقم اعشار پذیرفته می‌شود؛ صفر معافیت صریح است و مقدار حذف‌شده یا null از نرخ تعریف‌شده کالا استفاده می‌کند.

```
{
  "CustomerCode": 1001,
  "FinalSale": false,
  "Paid": false,
  "Discunt": 10000,
  "TaxAmount": 9000,
  "Detail": [
    {
      "Code": 2001,
      "Quantity": 1,
      "Price": 100000,
      "DiscuntPercent": 10,
      "VatPercent": 10
    }
  ]
}
```

در مثال، مبلغ پس از تخفیف ردیفی 90000 است؛ مالیات 10 درصد روی همین مبلغ محاسبه می‌شود و برابر 9000 است. تخفیف انتهای فاکتور 10000 فقط از جمع قابل پرداخت کسر می‌شود و مالیات را کاهش نمی‌دهد؛ بنابراین جمع قابل پرداخت بدون کرایه 89000 خواهد بود. سهم تخفیف کلی صرفا برای ثبت حسابداری بین ردیف‌ها تقسیم می‌شود و از پایه مالیات کسر نمی‌شود. کرایه به مبلغ قابل پرداخت افزوده می‌شود و مشمول درصد مالیات ردیف کالا نیست.

مطابق روال فعلی سرویس، اعشار جمع مالیات با Math.Truncate حذف می‌شود. اگر TaxAmount ارسال شود، حتی مقدار صفر، مقایسه دقیق با همین جمع انجام می‌شود؛ مغایرت باعث isSuccess=false و پیام شامل مبلغ ارسالی، محاسبه‌شده و اختلاف می‌شود و تراکنش ثبت نمی‌شود. مبلغ منفی یا درصد نامعتبر نیز خطاست. اگر مبلغ و درصد جدید هیچ‌کدام ارسال نشوند، مسیر محاسبه قدیمی حفظ می‌شود. استفاده از این قابلیت نیازمند به‌روزرسانی API و سرویس clsApi است.

محل	نام پارامتر	نوع	معنی و مقدار پیش‌فرض
سربرگ	TaxAmount	decimal?	مبلغ کل مالیات مورد انتظار؛ در حالت حذف یا null تطبیق مبلغ انجام نمی‌شود
هر ردیف Detail	VatPercent	decimal?	درصد مالیات ردیف؛ حذف یا null یعنی نرخ کالا، ولی صفر یعنی معافیت صریح

فرمول محاسبه در حالت ارسال پارامترهای مالیات جدید:

```
line_net = quantity * price * (1 - discount_percent / 100)
line_tax = line_net * vat_percent / 100
invoice_tax = truncate(sum(line_tax))
payable = sum(line_net) - invoice_discount + invoice_tax + freight
```

اگر جمع مبلغ پس از تخفیف ردیف‌ها صفر باشد، تخفیف کلی هم باید صفر باشد و مالیات صفر است. درصد تخفیف باید بین صفر تا صد، تعداد مثبت و نرخ غیرمنفی باشد. تخفیف کلی منفی یا بیشتر از مجموع مبلغ پس از تخفیف ردیف‌ها رد می‌شود. مالیات هر ردیف فقط پس از تخفیف ردیفی محاسبه می‌شود؛ تخفیف انتهای فاکتور، حتی اگر کل مبلغ خالص ردیف‌ها باشد، پایه مالیات را کاهش نمی‌دهد. توزیع سهم تخفیف کلی صرفا برای ثبت حسابداری است. مبلغ TaxAmount دوباره به جمع اضافه نمی‌شود؛ این فیلد فقط کنترل صحت محاسبه است.

برای نمونه درخواست قبلی، اگر TaxAmount را به جای 9000 برابر 8000 ارسال کنید، ثبت انجام نمی‌شود. نمونه خلاصه‌شده پاسخ خطا:

```
{
  "isSuccess": false,
 "message": "1000- اختلاف 9000، محاسبه‌شده 8000، مبلغ ارسالی: مغایرت مالیات.",
  "autoNumber": ""
}
```

پیام سرویس ممکن است پیشوند خطای ذخیره‌سازی هم داشته باشد. این پاسخ در SubmitInvoice با HTTP 200 برمی‌گردد؛ isSuccess=false معیار شکست است. کنترل برای سفارش و فروش قطعی مشترک است. به دلیل حذف اعشار جمع مالیات، مثلا حاصل 99.9 به 99 تبدیل می‌شود؛ برای این مثال TaxAmount باید 99 باشد. مقدار مالیات ارسالی را پس از همین قاعده محاسبه کنید و واحد پولی آن را با Price و Discunt یکسان نگه دارید.

اولویت تعیین تفصیلی فاکتور در نسخه جدید

CustomerCode یک کد عددی اختیاری از نوع int? است و شناسه متنی ID نیست. نام مدل بدنه ثبت فاکتور در کد FactorDto است؛ نام endpoint همچنان SubmitInvoice باقی می‌ماند.

اولویت	شرط	نتیجه
1	CustomerCode بزرگ‌تر از صفر	همان تفصیلی انتخاب می‌شود؛ موبایل و پیش‌فرض جایگزین آن نمی‌شوند
2	کد حذف‌شده، null، خالی یا صفر	موبایل غیرخالی به صورت تطابق دقیق در اطلاعات تفصیلی جست‌وجو می‌شود
3	موبایل خالی یا بدون تطابق	تفصیلی پیش‌فرض بدهکاران وب از تنظیمات انتخاب می‌شود

کد منفی خطاست. کد مثبت ناموجود نیز خطاست و حتی با وجود موبایل معتبر، به موبایل یا پیش‌فرض برنمی‌گردد. اگر یک موبایل به چند تفصیلی مربوط باشد، درخواست با پیام ابهام رد می‌شود و باید CustomerCode صریح ارسال کنید. پیش‌فرض نامشخص یا ناموجود هم خطاست.

تفصیلی انتخاب‌شده در فاکتور و مسیرهای سند حسابداری و تسویه یکسان استفاده می‌شود. تطابق موبایل حتی در صورت غیرفعال بودن تنظیم قدیمی ثبت سند روی تفصیلی بررسی می‌شود. در نبود هویت صریح یا تطابق تفصیلی، منطق قبلی مشترک همچنان ممکن است اجرا شود.

```
{
  "CustomerCode": 1001,
  "CustomerMobile": "09120000000",
  "FinalSale": false,
  "Paid": false,
  "Detail": [
    {
      "Code": 2001,
      "Quantity": 1,
      "Price": 125000
    }
  ]
}
```

در مثال بالا، تفصیلی 1001 اولویت دارد حتی اگر موبایل متعلق به تفصیلی دیگری باشد. برای واگذاری تشخیص به موبایل، CustomerCode را صفر یا null بفرستید یا حذف کنید. برای استفاده از پیش‌فرض، کد و موبایل را خالی بگذارید. در JSON ترجیحا عدد یا null ارسال کنید؛ پذیرش رشته خالی صرفا برای سازگاری است.

تفصیلی پیش‌فرض از گزینه بدهکاران وب در پنل فروش اینترنتی می‌آید. کلید ذخیره آن CutomerManagment_T4CodeBedehkarForosh است. این تنظیم با رده تفصیلی برای معرفی مشتری تفاوت دارد.

آزمون نسخه آخر روی Client1 شامل دوازده سناریوی موفق بوده است: اولویت کد صریح، کد صفر، کد حذف‌شده، null و رشته خالی، نبود هویت، موبایل بدون تطابق، فروش قطعی با کد و با موبایل و با پیش‌فرض، رد کد مثبت ناموجود و رد کد منفی. در فروش قطعی، تفصیلی ردیف بدهکار سند حسابداری نیز با فاکتور تطبیق داده شد. این نتیجه مربوط به محیط آزمایش است و تضمین تنظیم صحیح سایر مجموعه‌ها نیست.

حذف سفارش/فاکتور

```
GET /api/RemoveInvoice/{AutoNumber}/{FinalSale}
```

پارامتر	نوع	توضیح
AutoNumber	string	شماره سند ثبت‌شده
FinalSale	int	وضعیت فروش قطعی/سفارش

پاسخ موفق:

```
{
  "isSuccess": true,
 "message": "حذف در سرور انجام شد",
  "autoNumber": null
}
```

6. معرفی و ویرایش تفصیلی

معرفی تفصیلی جدید

```
POST /api/Customer
Authorization: Bearer {token}
Content-Type: application/json; charset=utf-8
```

کد عددی تفصیلی را سرویس تولید می‌کند؛ در درخواست ایجاد نیازی به ارسال کد نیست. شناسه ID با کد عددی Code متفاوت است. خروجی موفق کد عددی ایجادشده را در data برمی‌گرداند.

فیلد	نوع	الزام	توضیح
ID	string	بله	شناسه غیرخالی مشتری
Name	string	بله	نام غیرخالی تفصیلی
Address	string	خیر	نشانی؛ در صورت حذف خالی است
Tel	string	خیر	تلفن
Mobile	string	خیر	موبایل

```
{
  "ID": "WEB-1001",
  "Name": "Sample customer",
  "Address": "Sample address",
  "Tel": "",
  "Mobile": "09120000000"
}
```

```
{
  "isSuccess": true,
  "message": "...",
  "data": 1001
}
```

رده تفصیلی از تنظیمات پنل فروش اینترنتی با کلید CutomerManagment_TafsiliGroups خوانده می‌شود؛ باید معتبر و موجود باشد. نوع تفصیلی جدید از تنظیم کد نوع اشخاص حقیقی و حقوقی با کلید TYPE_CODE_ASHKHAS می‌آید، نه صفر ثابت. اگر چند کد با کاما جدا شده باشند، نخستین کد انتخاب می‌شود؛ مثلا از 4,5 مقدار 4 استفاده می‌شود. مقدار نامعتبر یا غیرمثبت خطا ایجاد می‌کند. تاریخچه معرفی با هویت API user ثبت می‌شود.

ویرایش تفصیلی موجود

```
PUT /api/Customer/1001
Authorization: Bearer {token}
Content-Type: application/json; charset=utf-8
```

بدنه همان CustomerWriteDto است و ID و Name باید ارسال شوند. کد موجود باید عضو رده تفصیلی تنظیم‌شده باشد. فیلدهای اختیاری حذف‌شده خالی می‌شوند؛ این عملیات PATCH نیست. نوع تفصیلی موجود با این درخواست تغییر نمی‌کند. نتیجه کد ویرایش‌شده را در data می‌دهد و تاریخچه ویرایش ثبت می‌شود.

اطلاعات تفصیلی و مانده حساب

```
GET /api/CustomerInfo
GET /api/CustomerInfo?code=1001
```

این گزارش فقط تفصیلی‌های رده انتخاب‌شده در تنظیمات پنل فروش اینترنتی را همراه مشخصات و مانده حساب برمی‌گرداند. code اختیاری و در صورت ارسال باید مثبت باشد؛ حذف آن یعنی همه تفصیلی‌های همان رده. نام endpoint برابر CustomerInfo است و Balance فقط یکی از ستون‌های خروجی است.

فیلد خروجی	نوع	توضیح
code	int	کد عددی تفصیلی
id	string	شناسه تفصیلی

فیلد خروجی	نوع	توضیح
name	string	نام
address	string	نشانی
tel	string	تلفن
mobile	string	موبایل
balance	decimal	قدر مطلق مانده حساب
balanceType	string	بدهکار، بستانکار یا تسویه

```
{
  "isSuccess": true,
  "message": "...",
  "data": [
    {
      "code": 1001,
      "id": "WEB-1001",
      "name": "Sample customer",
      "address": "",
      "tel": "",
      "mobile": "09120000000",
      "balance": 125000,
    "balanceType": "بدهکار"
    }
  ]
}
```

7. معرفی و ویرایش مشترک

معرفی مشترک جدید

```
POST /api/Subscribers
Authorization: Bearer {token}
Content-Type: application/json; charset=utf-8
```

فیلد	نوع	الزام	توضیح
Name	string	بله	نام غیرخالی
FamilyName	string	خیر	نام خانوادگی
Address	string	خیر	نشانی
Tel	string	خیر	تلفن
Mobile	string	خیر	موبایل
BirthDate	string	خیر	تاریخ تولد با قالب yyyy/MM/dd یا خالی
MaritalStatus	bool	خیر	وضعیت تأهل؛ پیش‌فرض false

```
{
  "Name": "Sample",
  "FamilyName": "Subscriber",
  "Address": "",
  "Tel": "",
  "Mobile": "09120000000",
  "BirthDate": "1370/01/01",
  "MaritalStatus": false
}
```

کد مشترک در سمت سرویس تولید می‌شود و در data برمی‌گردد. برای ایجاد از POST بدون کد در مسیر استفاده کنید، نه PUT. معرفی و ویرایش با تاریخچه استاندارد و هویت API user ثبت می‌شوند.

ویرایش مشترک موجود

```
PUT /api/Subscribers/321
```

بدنه همان SubscriberWriteDto است. کد مسیر باید مثبت و مربوط به مشترک موجود و فعال باشد. Name لازم است؛ فیلدهای اختیاری را برای حفظ مقدارشان دوباره ارسال کنید. نتیجه موفق کد مشترک را در data برمی‌گرداند. متغیرهایی مانند {{subscriber_code}} در Postman باید قبل از ارسال مقداردهی شوند، وگرنه مسیر عددی تطبیق ندارد و ممکن است 404 دریافت شود.

8. فاکتورهای فروش مشترک با سربرگ و ردیف‌ها

```
GET /api/SubscriberSaleInvoices?subscriberCode=321&autoNumber=12345
GET /api/SubscriberSaleInvoices/321?autoNumber=12345
```

این endpoint فقط فاکتور فروش را از دوره تنظیم‌شده اتصال گزارش می‌کند؛ سفارش‌ها و سایر اسناد در خروجی نیستند. ورودی type وجود ندارد. فیلد type در سربرگ خروجی نوع سند فروش را نشان می‌دهد و مقدار آن 110001 است. خروجی یک فهرست از فاکتورهاست؛ هر عضو دارای master و آرایه details است.

پارامتر	نوع	پیش‌فرض	توضیح
subscriberCode	int	0	در query صفر یعنی همه مشترک‌ها؛ در مسیر باید مثبت باشد
autoNumber	int	0	صفر یعنی همه فاکتورها؛ مثبت یعنی یک شماره فاکتور

پارامتر منفی باعث HTTP 400 می‌شود. دسترسی گزارش سربرگ فروش مشترک و گزارش جزئیات فروش مشترک باید هر دو فعال باشند. اعداد این گزارش، برخلاف برخی گزارش‌های قدیمی، به صورت عددی برمی‌گردند.

```
{
  "isSuccess": true,
  "message": "...",
  "data": [
    {
      "master": {
        "subscriberCode": 321,
        "subscriberName": "Sample Subscriber",
        "periodCode": 1,
        "type": 110001,
        "autoNumber": 12345,
        "date": "1405/07/30",
        "discountPercent": 0,
        "rowDiscountPercent": 0,
        "vatPercent": 0,
        "sum": 250000,
```

```
        "sumDiscount": 0,
        "sumFare": 0,
        "sumAdditions": 0,
        "sumDeductions": 0,
        "sumVat": 0,
        "sumPayable": 250000
      },
      "details": [
        {
          "recordNumber": 1,
          "goodCode": 1001,
          "goodName": "Sample product",
          "quantity": 2,
          "price": 125000,
          "discountPercent": 0,
          "cashDiscountPercent": 0,
          "vatPercent": 0,
          "sum": 250000,
          "sumDiscount": 0,
          "sumFare": 0,
          "sumAdditions": 0,
          "sumDeductions": 0,
          "sumVat": 0,
          "sumPayable": 250000
        }
      ]
    }
  ]
}
```


9. به‌روزرسانی کالا از ووکامرس به پیشگامان

```
POST /api/Product/update-from-woocommerce
Authorization: Bearer {token}
Content-Type: application/json; charset=utf-8
```

این مسیر کالای موجود را با code عددی تطبیق می‌دهد و فقط فیلدهایی را تغییر می‌دهد که پرچم Update مربوط به آن‌ها true باشد. این عملیات ایجاد کالا نیست و بارکد تجمیعی نباید در code قرار گیرد.

فیلد	نوع	الزام	توضیح
Code	int	بله	کد مثبت کالای موجود در پیشگامان
UpdateName / Name	bool / string	شرطی	با فعال بودن پرچم، نام غیرخالی لازم است
UpdatePrice / Price	bool / decimal?	شرطی	نرخ فروش باید بزرگ‌تر از صفر باشد
UpdateMainPrice / MainPrice	bool / decimal?	شرطی	نرخ مصرف‌کننده باید بزرگ‌تر از صفر باشد
UpdateDiscountPercent / DiscountPercent	bool / decimal?	شرطی	درصد تخفیف از صفر تا صد
UpdateImage / ImageBase64	bool / string	شرطی	تصویر Base64 خام؛ حجم رمزگشایی‌شده بیشتر از صفر و حداکثر 15 MiB

```
{
  "Code": 1001,
  "UpdateName": false,
  "UpdatePrice": true,
  "Price": 125000,
  "UpdateMainPrice": true,
  "MainPrice": 150000,
  "UpdateDiscountPercent": true,
  "DiscountPercent": 5,
  "UpdateImage": false
}
```

حداقل یک پرچم به‌روزرسانی باید فعال باشد. تصویر را بدون پیشوند data:image ارسال کنید. خروجی موفق شامل code و پرچم‌های nameUpdated، priceUpdated، mainPriceUpdated، discountPercentUpdated و imageUpdated است. خطای اعتبارسنجی این مسیر ممکن است با HTTP 200 و isSuccess=false برگردد. تغییرات کالا و تصویر در تاریخچه سرویس با هویت API user ثبت می‌شوند.

مدل‌های داده

نام فیلدهای جدول‌ها مطابق JSON است. فیلدهای جدید price_unit، good_code و price_unit_ratio عمدا با زیرخط ارسال می‌شوند. بدنه‌های ورودی مانند FactorDto می‌توانند از نام PascalCase هم استفاده کنند. املای تاریخی discunt و discountPercent باید حفظ شود. علامت ? در نوع داده یعنی مقدار null نیز مجاز است.

CustomerAccountReportQuery

فیلد	نوع	توضیح
customerCode	int?	کد مثبت اختیاری مانده؛ حذف آن یعنی همه تفصیلی‌های گروه مجاز؛ صفر نامعتبر
fromDate	string	تاریخ ابتدا شمسی yyyy/MM/dd؛ اختیاری و شامل روز ابتدا
toDate	string	تاریخ انتها شمسی yyyy/MM/dd؛ اختیاری و شامل روز انتها

CustomerStatementQuery

فیلد	نوع	توضیح
customerCode	int?	کد مثبت تفصیلی؛ در گردش حساب الزامی
fromDate	string	تاریخ ابتدا شمسی yyyy/MM/dd؛ اختیاری و شامل روز ابتدا
toDate	string	تاریخ انتها شمسی yyyy/MM/dd؛ اختیاری و شامل روز انتها
page	int	صفحه گردش حساب؛ پیش‌فرض ۱ و بزرگ‌تر از صفر
pageSize	int	اندازه صفحه گردش؛ پیش‌فرض ۱۰۰ و محدوده ۱ تا ۵۰۰

CustomerBalanceDto

فیلد	نوع	توضیح
customerCode	int	کد عددی تفصیلی
customerName	string	نام مشتری
period	string	دوره مالی اتصال انتخاب‌شده
fromDate	string	تاریخ ابتدا شمسی yyyy/MM/dd؛ اختیاری و شامل روز ابتدا
toDate	string	تاریخ انتها شمسی yyyy/MM/dd؛ اختیاری و شامل روز انتها
openingBalance	decimal	قدر مطلق مانده قبل از بازه
openingBalanceType	string	تشخیص مانده قبل از بازه؛ بدهکار، بستانکار یا تسویه
totalDebit	decimal	جمع بدهکار کل بازه
totalCredit	decimal	جمع بستانکار کل بازه
balance	decimal	قدر مطلق مانده حساب
balanceType	string	بدهکار، بستانکار یا تسویه
totalRows	long	تعداد ردیف‌های حسابداری کل بازه، نه فقط صفحه جاری

CustomerStatementDto

فیلد	نوع	توضیح
customerCode	int	کد عددی تفصیلی
customerName	string	نام مشتری
period	string	دوره مالی اتصال انتخاب‌شده
fromDate	string	تاریخ ابتدا شمسی yyyy/MM/dd؛ اختیاری و شامل روز ابتدا
toDate	string	تاریخ انتها شمسی yyyy/MM/dd؛ اختیاری و شامل روز انتها
openingBalance	decimal	قدر مطلق مانده قبل از بازه
openingBalanceType	string	تشخیص مانده قبل از بازه؛ بدهکار، بستانکار یا تسویه
totalDebit	decimal	جمع بدهکار کل بازه
totalCredit	decimal	جمع بستانکار کل بازه
balance	decimal	قدر مطلق مانده حساب
balanceType	string	بدهکار، بستانکار یا تسویه
totalRows	long	تعداد ردیف‌های حسابداری کل بازه، نه فقط صفحه جاری
page	int	صفحه گردش حساب؛ پیش‌فرض ۱ و بزرگ‌تر از صفر
pageSize	int	اندازه صفحه گردش؛ پیش‌فرض ۱۰۰ و محدوده ۱ تا ۵۰۰
entries	List<CustomerStatementEntryDto>	ردیف‌های صفحه جاری گردش حساب

CustomerStatementEntryDto

فیلد	نوع	توضیح
rowNumber	long	شماره ردیف در کل بازه گزارش
date	string	تاریخ سند در قالب سامانه مبدا
documentNumber	long	شماره سند حسابداری
documentRowNumber	int	شماره ردیف سند حسابداری
accountCode	long	کد معین
description	string	شرح ردیف حسابداری
debit	decimal	مبلغ بدهکار ردیف
credit	decimal	مبلغ بستانکار ردیف
balance	decimal	قدر مطلق مانده جاری با احتساب مانده قبل از بازه و صفحات قبلی
balanceType	string	بدهکار، بستانکار یا تسویه

CustomerDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
name	string	نام
address	string	نشانی
mobile	string	شماره موبایل
tel	string	تلفن
id	string	شناسه متنی تفصیلی

CustomerInfoDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
id	string	شناسه متنی تفصیلی
name	string	نام
address	string	نشانی
tel	string	تلفن
mobile	string	شماره موبایل
balance	decimal	قدر مطلق مانده حساب
balanceType	string	بدهکار، بستانکار یا تسویه

CustomerSaleMasterDto

فیلد	نوع	توضیح
customerID	string	شناسه متنی مشتری
customerCode	int	کد عددی تفصیلی
customerName	string	نام مشتری
date	string	تاریخ سند در قالب سامانه مبدا
autoNumber	string	شماره سند
sum	string	جمع ناخالص
sumDiscount	string	جمع تخفیف
sumFare	string	جمع کرایه
sumAdditions	string	جمع اضافات
sumDeductions	string	جمع کسورات
sumVat	string	جمع مالیات
sumPayable	string	جمع قابل پرداخت

CustomerWriteDto

فیلد	نوع	توضیح
id	string	شناسه متنی تفصیلی
name	string	نام
address	string	نشانی
tel	string	تلفن
mobile	string	شماره موبایل

FactorDto

فیلد	نوع	توضیح
finalSale	bool	فروش قطعی true؛ سفارش false
fare	decimal	کرایه
addressInfo	string	اطلاعات نشانی سفارش
dsc2	string	شرح فاکتور
customerCode	int?	کد تفصیلی اختیاری؛ کد مثبت بر موبایل اولویت دارد
customerMobile	string	موبایل مشتری
customerName	string	نام مشتری
customerAddress	string	نشانی مشتری
customerInfo	string	اطلاعات تکمیلی مشتری
customerBirthDate	string	تاریخ تولد مشتری
discunt	decimal	تخفیف سطح فاکتور با املای تاریخی
taxAmount	decimal?	مبلغ کل مالیات مورد انتظار برای کنترل مغایرت؛ اختیاری
paid	bool?	وضعیت پرداخت؛ برای آزمایش false ارسال کنید
detail	List<DetailFactor>	فهرست ردیف‌های ورودی فاکتور

DetailFactor

فیلد	نوع	توضیح
code	int	کد عددی رکورد
quantity	decimal	تعداد یا موجودی
price	decimal	نرخ فروش
discuntPercent	decimal	درصد تخفیف با املای تاریخی
vatPercent	decimal?	درصد مالیات ردیف؛ صفر معاف، null یا حذف یعنی نرخ کالا
dsc	string	شرح
sizeCode	int	کد سایز
colorCode	int	کد رنگ
aggregateGoodSizeColorBarcode	string	بارکد تجمیعی؛ مستقل از code عددی کالا
barcode	string	بارکد کالا

ProductImageDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
imageBytes	string	تصویر با کدگذاری Base64
imageHash	string	هش تصویر با کدگذاری Base64

ProductImageResultDto

فیلد	نوع	توضیح
isSuccess	bool	نتیجه عملیات تجاری
message	string	پیام عملیات
data	List<ProductImageDto>	داده خروجی
totalProducts	int	کل کالاهای دارای تصویر در محدوده
totalImages	int	کل ردیف‌های تصاویر مبدأ
processedProducts	int	کالاهای تا نشانگر ورودی، پیش از صفحه جاری
processedImages	int	تصاویر تا نشانگر ورودی، پیش از صفحه جاری
pageProducts	int	کالاهای متمایز با تصویر معتبر در صفحه جاری
pageImages	int	تعداد تصاویر معتبر در پاسخ جاری
cursorMode	string	روش ادامه نشانگر؛ exclusive

ProductsColorsDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
name	string	نام
dsc	string	شرح
htmlCode	string	کد HTML رنگ

ProductsDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
name	string	نام
barcode	string	بارکد کالا
price	decimal	نرخ فروش
mainPrice	decimal	نرخ اصلی یا مصرف‌کننده
writer	string	نویسنده یا ویژگی ویژه
translator	string	مترجم یا ویژگی دیگر
publisher	string	ناشر
defineDate	string	تاریخ معرفی
groupPath	string	مسیر گروه
groupName	string	نام گروه
dsc	string	شرح
discuntPercent	decimal	درصد تخفیف با املای تاریخی
quantity	decimal	تعداد یا موجودی
quantityInAllStorage	decimal	موجودی کل انبارها

ChangedProductDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
name	string	نام
barcode	string	بارکد کالا
price	decimal	نرخ فروش
mainPrice	decimal	نرخ اصلی یا مصرف‌کننده
writer	string	نویسنده یا ویژگی ویژه
translator	string	مترجم یا ویژگی دیگر
publisher	string	ناشر
defineDate	string	تاریخ معرفی
groupPath	string	مسیر گروه
groupName	string	نام گروه
dsc	string	شرح
discuntPercent	decimal	درصد تخفیف با املای تاریخی
quantity	decimal	تعداد یا موجودی
quantityInAllStorage	decimal	موجودی کل انبارها
changeToken	string	شناسه نسخه تغییر؛ بدون تغییر پس از ذخیره موفق برای ACK ارسال شود

ChangedProductAcknowledgementDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
changeToken	string	شناسه نسخه تغییر؛ بدون تغییر پس از ذخیره موفق برای ACK ارسال شود

ProductWithUnitsDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
name	string	نام
barcode	string	بارکد کالا
price	decimal	نرخ فروش
mainPrice	decimal	نرخ اصلی یا مصرف‌کننده
writer	string	نویسنده یا ویژگی ویژه
translator	string	مترجم یا ویژگی دیگر
publisher	string	ناشر
defineDate	string	تاریخ معرفی
groupPath	string	مسیر گروه
groupName	string	نام گروه
dsc	string	شرح
discuntPercent	decimal	درصد تخفیف با املای تاریخی
quantity	decimal	تعداد یا موجودی
quantityInAllStorage	decimal	موجودی کل انبارها
unit	int	کد واحد اصلی کالا
price_unit	int	کد واحد قیمت‌گذاری تنظیم‌شده کالا
price_unit_ratio	decimal	تعداد واحد اصلی در یک واحد قیمت‌گذاری

ProductsGroupDto

فیلد	نوع	توضیح
pathCode	string	کد مسیر گروه
pathName	string	نام مسیر گروه
pathDsc	string	شرح گروه

ProductsSizesColorsDto

فیلد	نوع	توضیح
goodCode	int	کد کالا
sizeCode	int	کد سایز
colorCode	int	کد رنگ
aggregateGoodSizeColorBarcode	string	بارکد تجمیعی؛ مستقل از code عددی کالا
price	decimal	نرخ فروش
quantity	decimal	تعداد یا موجودی

ProductsSizesDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
name	string	نام
dsc	string	شرح

ProductUnitDto

فیلد	نوع	توضیح
good_code	int	کد کالا
price_unit	int	کد واحد جزء محاسبه‌شده برای کالا
unit	int	کد واحد این ردیف؛ واحد اصلی یا فرعی
ratio	decimal	تعداد واحد جزء در یک واحد این ردیف

ProductUploadDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
updateName	bool	پرچم انتخاب تغییر نام
name	string	نام
updatePrice	bool	پرچم انتخاب تغییر نرخ فروش
price	decimal?	نرخ فروش
updateMainPrice	bool	پرچم انتخاب تغییر نرخ اصلی یا مصرف‌کننده
mainPrice	decimal?	نرخ اصلی یا مصرف‌کننده
updateDiscountPercent	bool	پرچم انتخاب تغییر درصد تخفیف
discountPercent	decimal?	درصد تخفیف
updateImage	bool	پرچم انتخاب تغییر Image
imageBase64	string	تصویر Base64 خام بدون پیشوند data

ProductUploadResultDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
nameUpdated	bool	وضعیت اعمال تغییر نام
priceUpdated	bool	وضعیت اعمال تغییر نرخ فروش
mainPriceUpdated	bool	وضعیت اعمال تغییر نرخ اصلی یا مصرف‌کننده
discountPercentUpdated	bool	وضعیت اعمال تغییر درصد تخفیف
imageUpdated	bool	وضعیت اعمال تغییر Image

ResultDto

فیلد	نوع	توضیح
isSuccess	bool	نتیجه عملیات تجاری
message	string	پیام عملیات
autoNumber	string	شماره سند

ResultDto<T>

فیلد	نوع	توضیح
isSuccess	bool	نتیجه عملیات تجاری
message	string	پیام عملیات
data	T	داده خروجی

SubscriberDto

فیلد	نوع	توضیح
code	int	کد عددی رکورد
name	string	نام
familyName	string	نام خانوادگی
address	string	نشانی
mobile	string	شماره موبایل
tel	string	تلفن
gender	string	جنسیت
birthDate	string	تاریخ تولد
maritalStatus	bool	وضعیت تأهل

SubscriberSaleSumDto

فیلد	نوع	توضیح
subscriberCode	int	کد مشترک
subscriberName	string	نام مشترک
count	string	تعداد
sum	string	جمع ناخالص
sumDiscount	string	جمع تخفیف
sumFare	string	جمع کرایه
sumAdditions	string	جمع اضافات
sumDeductions	string	جمع کسورات
sumVat	string	جمع مالیات
sumPayable	string	جمع قابل پرداخت

SubscriberSaleMasterDto

فیلد	نوع	توضیح
subscriberCode	int	کد مشترک
subscriberName	string	نام مشترک
date	string	تاریخ سند در قالب سامانه مبدا
autoNumber	string	شماره سند
sum	string	جمع ناخالص
sumDiscount	string	جمع تخفیف
sumFare	string	جمع کرایه
sumAdditions	string	جمع اضافات
sumDeductions	string	جمع کسورات
sumVat	string	جمع مالیات
sumPayable	string	جمع قابل پرداخت

SubscriberSaleDetailDto

فیلد	نوع	توضیح
subscriberCode	int	کد مشترک
subscriberName	string	نام مشترک
goodCode	string	کد کالا
goodName	string	نام کالا
quantity	string	تعداد یا موجودی
price	string	نرخ فروش
sum	string	جمع ناخالص
sumDiscount	string	جمع تخفیف
sumFare	string	جمع کرایه
sumAdditions	string	جمع اضافات
sumDeductions	string	جمع کسورات
sumVat	string	جمع مالیات
sumPayable	string	جمع قابل پرداخت

SubscriberInvoiceDto

فیلد	نوع	توضیح
master	SubscriberInvoiceMasterDto	سربرگ فاکتور
details	List<SubscriberInvoiceDetailDto>	فهرست ردیف‌های فاکتور

SubscriberInvoiceMasterDto

فیلد	نوع	توضیح
subscriberCode	int	کد مشترک
subscriberName	string	نام مشترک
periodCode	int	شناسه دوره اتصال
type	int	نوع سند در خروجی؛ گزارش جدید فقط فروش 110001
autoNumber	int	شماره سند
date	string	تاریخ سند در قالب سامانه مبدا
discountPercent	decimal	درصد تخفیف
rowDiscountPercent	decimal	درصد تخفیف ردیف در سربرگ
vatPercent	decimal	درصد مالیات
sum	decimal	جمع ناخالص
sumDiscount	decimal	جمع تخفیف
sumFare	decimal	جمع کرایه
sumAdditions	decimal	جمع اضافات
sumDeductions	decimal	جمع کسورات
sumVat	decimal	جمع مالیات
sumPayable	decimal	جمع قابل پرداخت

SubscriberInvoiceDetailDto

فیلد	نوع	توضیح
recordNumber	int	شماره ردیف
goodCode	int	کد کالا
goodName	string	نام کالا
quantity	decimal	تعداد یا موجودی
price	decimal	نرخ فروش
discountPercent	decimal	درصد تخفیف
cashDiscountPercent	decimal	درصد تخفیف نقدی
vatPercent	decimal	درصد مالیات
sum	decimal	جمع ناخالص
sumDiscount	decimal	جمع تخفیف
sumFare	decimal	جمع کرایه
sumAdditions	decimal	جمع اضافات
sumDeductions	decimal	جمع کسورات
sumVat	decimal	جمع مالیات
sumPayable	decimal	جمع قابل پرداخت

SubscriberWriteDto

فیلد	نوع	توضیح
name	string	نام
familyName	string	نام خانوادگی
address	string	نشانی
tel	string	تلفن
mobile	string	شماره موبایل
birthDate	string	تاریخ تولد
maritalStatus	bool	وضعیت تأهل

Tokens

فیلد	نوع	توضیح
token	string	توکن احراز هویت

WooProductDto

فیلد	نوع	توضیح
rowType	string	نوع ردیف
rowTypeCode	int	کد نوع ردیف
productCode	int	کد کالای پایه
uniqueCode	string	کد یکتای ردیف یا گونه
barcode	string	بارکد کالا
name	string	نام
price	decimal	نرخ فروش
mainPrice	decimal	نرخ اصلی یا مصرف‌کننده
discuntPercent	decimal	درصد تخفیف با املای تاریخی
quantity	decimal	تعداد یا موجودی
quantityInAllStorage	decimal	موجودی کل انبارها
unit	int	کد واحد اصلی کالا
price_unit	int	کد واحد قیمت‌گذاری تنظیم‌شده کالا
price_unit_ratio	decimal	تعداد واحد اصلی در یک واحد قیمت‌گذاری
dsc	string	شرح
writer	string	نویسنده یا ویژگی ویژه
translator	string	مترجم یا ویژگی دیگر
publisher	string	ناشر
defineDate	string	تاریخ معرفی
groupPath	string	مسیر گروه
groupName	string	نام گروه
sizeCode	int?	کد سایز
sizeName	string	نام سایز
colorCode	int?	کد رنگ
colorName	string	نام رنگ

نمونه استفاده با cURL

دریافت توکن:

```
curl -X POST "http://localhost:5000/api/Users/api/authenticate" \
  -H "Content-Type: application/json" \
  -d "{\"name\":\"API_USER\",\"password\":\"API_PASSWORD\",\"clientId\":\"client1\"}"
```

دریافت کالاها:

```
curl "http://localhost:5000/api/Product/0/50" \
  -H "Authorization: Bearer {token}"
```

ثبت فاکتور:

```
curl -X POST "http://localhost:5000/api/SubmitInvoice" \
  -H "Authorization: Bearer {token}" \
  -H "Content-Type: application/json" \
  -d @factor.json
```

نکات اجرایی

گزارش مانده حساب و گردش حساب تفصیلی

دو مسیر زیر با توکن Bearer فراخوانی می‌شوند. در گردش حساب دوره‌ای، customerCode الزامی و بزرگ‌تر از صفر است. در گزارش مانده حساب، کد اختیاری است؛ با حذف آن همه تفصیلی‌های عضو رده تنظیمات API، حتی تفصیلی‌های بدون گردش، گزارش می‌شوند. ارسال صریح کد صفر یا منفی معتبر نیست:

```
GET /api/CustomerBalance
GET /api/CustomerBalance?fromDate=1405/01/01&toDate=1405/08/30
GET /api/CustomerBalance?customerCode=100&fromDate=1405/01/01&toDate=1405/08/30
GET /api/CustomerStatement?customerCode=100&fromDate=1405/01/01&toDate=1405/08/30&page=1&pageSize=100
```

- fromDate و toDate اختیاری، شمسی و با قالب دقیق yyyy/MM/dd هستند. هر دو روز ابتدا و انتها در گزارش محاسبه می‌شوند. تاریخ نامعتبر یا ابتدا بعد از انتها خطاست.

- بدون تاریخ، گزارش کل دوره مالی تنظیم‌شده در اتصال یا Client توکن ارائه می‌شود؛ این گزارش چند دوره مالی را ادغام نمی‌کند. با فقط یک تاریخ نیز می‌توان گزارش گرفت.

- در مدیریت پنل فروش اینترنتی ← تنظیمات ← دسترسی‌های API، دو تیک مستقل «گزارش مانده حساب تفصیلی» و «گزارش گردش حساب دوره‌ای تفصیلی» وجود دارند. هر دو پیش‌فرض خاموش‌اند؛ با دکمه ذخیره همان صفحه ذخیره می‌شوند.

- شناسه تنظیم دسترسی‌ها به‌ترتیب CutomerManagment_ApiCustomerBalance و CutomerManagment_ApiCustomerStatement است. نبود تنظیم، صفر و مقدار نامعتبر اجازه گزارش نمی‌دهند.

- تفصیلی باید عضو رده تفصیلی انتخاب‌شده در همان تنظیمات، یعنی CutomerManagment_TafsiliGroups باشد. رده از ورودی API دریافت نمی‌شود. تفصیلی خارج از رده یا ناموجود، بدون افشای اطلاعات، رد می‌شود.

- در گزارش مانده بدون کد، data آرایه‌ای از گزارش‌های تفصیلی است؛ رده معتبر بدون عضو، آرایه خالی [] می‌دهد. رده تنظیم‌نشده یا نامعتبر خطای 403 دارد. با ارسال کد مثبت، پاسخ تک‌تفصیلی قبلی به‌شکل یک شیء حفظ شده است. خروجی گردش حساب همیشه تک‌تفصیلی و نیازمند کد است.

- مبنا همان گردش اسناد حسابداری در گزارش استاندارد صورت‌حساب است؛ سفارش یا فروش فاقد سند حسابداری، گردش حساب محسوب نمی‌شود. مبلغ‌ها در واحد پول ثبت‌شده حسابداری و بدون تبدیل ریال/تومان هستند.

- گردش حساب صفحه‌بندی دارد: page پیش‌فرض ۱ و pageSize پیش‌فرض ۱۰۰، حداقل ۱ و حداکثر ۵۰۰. جمع‌ها و totalRows برای کل بازه‌اند، نه فقط صفحه جاری. مانده هر ردیف، مانده قبل از بازه و گردش صفحات پیشین را هم لحاظ می‌کند.

- صفحه‌های گزارش وضعیت جاری اسناد را می‌خوانند؛ اگر بین دو درخواست سندی اضافه یا ویرایش شود، برای مقایسه قطعی، گزارش را مجدد از صفحه اول بگیرید.

فیلدهای مشترک گزارش تک‌تفصیلی یا هر عضو آرایه مانده:

فیلد	شرح
customerCode, customerName	کد و نام تفصیلی درخواست‌شده
period	دوره مالی اتصال انتخاب‌شده
fromDate, toDate	بازه درخواست؛ در صورت حذف، null
openingBalance, openingBalanceType	مبلغ مطلق مانده قبل از بازه و تشخیص بدهکار/بستانکار/تسویه
totalDebit, totalCredit	جمع بدهکار و بستانکار بازه
balance, balanceType	مبلغ مطلق مانده پایان بازه، با احتساب مانده قبل از بازه، و تشخیص آن
totalRows	تعداد ردیف‌های حسابداری در بازه

خروجی گردش علاوه بر فیلدهای بالا، pageSize، page و entries دارد. هر ردیف شامل documentNumber، date، rowNumber، accountCode، documentRowNumber (کد معین)، balance، credit، debit، description و balanceType است. مبالغ و شمارنده‌ها عددی هستند.

نمونه خروجی مانده:

```
{
  "isSuccess": true,
 "message": "مانده حساب دریافت شد.",
  "data": {
    "customerCode": 100,
  "customerName": "مشتری نمونه",
    "period": "1405",
    "fromDate": "1405/01/01",
    "toDate": "1405/08/30",
    "openingBalance": 100000,
  "openingBalanceType": "بدهکار",
    "totalDebit": 200000,
    "totalCredit": 50000,
    "balance": 250000,
  "balanceType": "بدهکار",
    "totalRows": 2
  }
}
```

خطاها: 400 برای ورودی نامعتبر یا ارسال‌نشدن کد تفصیلی در گردش حساب؛ 401 برای توکن نامعتبر/ارسال‌نشده؛ 403 برای دسترسی خاموش، رده نامعتبر یا تفصیلی غیرمجاز؛ 502 برای خطای سرویس یا پاسخ ناقص/نامعتبر. نصب API به‌تنهایی کافی نیست؛ سرویس پیشگامان و حسابداری سازگار هم باید به‌روزرسانی شوند. این قابلیت تغییر ساختار پایگاه داده نیاز ندارد.

نمونه مانده گروهی با بازه تاریخ؛ تفصیلی بدون گردش نیز در آرایه می‌ماند:

```
{
  "isSuccess": true,
 "message": "مانده حساب تفصیلی‌های مجاز دریافت شد.",
  "data": [
    {
      "customerCode": 100,
   "customerName": "مشتری نمونه",
      "period": "1405",
      "fromDate": "1405/01/01",
      "toDate": "1405/01/31",
      "openingBalance": 50,
   "openingBalanceType": "بدهکار",
      "totalDebit": 100,
      "totalCredit": 40,
```

```
      "balance": 110,
    "balanceType": "بدهکار",
      "totalRows": 2
    },
    {
      "customerCode": 101,
    "customerName": "مشتری بدون گردش",
      "period": "1405",
      "fromDate": "1405/01/01",
      "toDate": "1405/01/31",
      "openingBalance": 0,
    "openingBalanceType": "تسویه",
      "totalDebit": 0,
      "totalCredit": 0,
      "balance": 0,
    "balanceType": "تسویه",
      "totalRows": 0
    }
  ]
}
```


نمونه گردش حساب همان مشتری در صفحه دوم با اندازه صفحه یک؛ جمع‌ها برای کل بازه‌اند و مانده ردیف، گردش صفحه قبل را نیز لحاظ کرده است:


```
{
  "isSuccess": true,
 "message": "گردش حساب دریافت شد.",
  "data": {
    "customerCode": 100,
  "customerName": "مشتری نمونه",
    "period": "1405",
    "fromDate": "1405/01/01",
    "toDate": "1405/01/31",
    "openingBalance": 50,
  "openingBalanceType": "بدهکار",
    "totalDebit": 100,
    "totalCredit": 40,
    "balance": 110,
  "balanceType": "بدهکار",
    "totalRows": 2,
    "page": 2,
    "pageSize": 1,
    "entries": [
      {
        "rowNumber": 2,
        "date": "1405/01/20",
        "documentNumber": 12,
        "documentRowNumber": 1,
        "accountCode": 101,
     "description": "دریافت وجه",
        "debit": 0,
        "credit": 40,
        "balance": 110,
     "balanceType": "بدهکار"
      }
    ]
  }
}
```


در Postman، پوشه Customer Account Reports سه نمونه مانده گروهی، مانده یک تفصیلی و گردش دوره‌ای دارد.
account_customer_code را برای دو درخواست دارای کد تنظیم کنید. متغیرهای account_from_date و account_to_date شمسی

هستند و در صورت عدم نیاز می‌توانید پارامترهای تاریخ را حذف کنید. عنوان پیام ممکن است فارسی باشد؛ معیار موفقیت، کد HTTP همراه با isSuccess و ساختار data است.

نکات عمومی

- همه مسیرها نسبت به آدرس پایه سرویس هستند.

- بیشتر خطاهای تجاری با OK 200 و isSuccess=false برگردانده می‌شوند؛ بنابراین کلاینت نباید فقط به HTTP status تکیه کند.

- توکن JWT در کد فعلی با اعتبار 600 دقیقه تولید می‌شود.

- در حالت چندمشتری، claim با نام client داخل توکن ذخیره می‌شود و برای انتخاب Clients:{clientId} استفاده می‌گردد.

- نام برخی فیلدها با غلط املایی تاریخی در مدل‌ها باقی مانده‌اند، مثل discuntPercent و discunt. برای سازگاری با API فعلی، همین نام‌ها باید در کلاینت استفاده شوند.

دسترسی‌ها و پیش‌نیازهای سرویس

دسترسی‌های معرفی، ویرایش و گزارش مشتری و مشترک در مدیریت پنل فروش اینترنتی، تب تنظیمات، بخش دسترسی‌های API قرار دارند؛ در پورتال یا افزونه وردپرس تعریف نمی‌شوند. مقدار صفر یا تنظیم ذخیره‌نشده به معنی دسترسی غیرفعال است. پس از فعال کردن گزینه مربوط، تنظیمات را ذخیره کنید. توکن معتبر به تنهایی این مجوزها را فعال نمی‌کند.

عملیات	دسترسی مورد نیاز
POST /api/Customer	معرفی تفصیلی
PUT /api/Customer/{code}	ویرایش تفصیلی
GET /api/Customer و دریافت یک شناسه	دسترسی گزارش مرتبط با تفصیلی
GET /api/CustomerInfo	اطلاعات تفصیلی و مانده حساب
GET /api/CustomerBalance	گزارش مانده حساب؛ دسترسی مستقل ApiCustomerBalance
GET /api/CustomerStatement	گردش حساب دوره‌ای؛ دسترسی مستقل ApiCustomerStatement
GET /api/CustomerSale و گزارش یک مشتری	دسترسی فروش تفصیلی مربوط
POST /api/Subscribers	معرفی مشترک
PUT /api/Subscribers/{code}	ویرایش مشترک
GET /api/Subscribers و دریافت یک کد	گزارش مشترک
GET /api/SubscriberSaleSum	جمع فروش مشترک
GET /api/SubscriberSaleMaster	سربرگ فروش مشترک
GET /api/SubscriberSaleDetail	جزئیات فروش مشترک
GET /api/SubscriberSaleInvoices	هر دو دسترسی سربرگ و جزئیات فروش مشترک

نسخه سرویس عملیاتی پیشگامان باید با فرمان‌های نسخه API هماهنگ باشد. نصب‌کننده API به تنهایی جایگزین PioService نرم‌افزار حسابداری نمی‌شود. برای استفاده از اولویت تفصیلی جدید، سرویس اصلاح‌شده نیز باید در مقصد به‌روزرسانی شود؛ صرفا نصب API جدید روی سرویس قدیمی کافی نیست.

راهنمای خطا و استفاده ایمن

وضعیت	معنی	اقدام
200	پاسخ HTTP موفق؛ عملیات تجاری ممکن است ناموفق باشد	isSuccess و message را بررسی کنید
400	ورودی نامعتبر در مسیرهای دارای اعتبارسنجی HTTP	نوع، کد و بدنه را اصلاح کنید
401	توکن نامعتبر، حذف‌شده یا منقضی	دوباره احراز هویت کنید
403	مجوز API غیرفعال	دسترسی مربوط در پنل فروش اینترنتی را فعال و ذخیره کنید
404	مسیر یا کالا پیدا نشد	آدرس، کد کالا و محدوده دسترسی کالا را بررسی کنید
502	خطای سرویس پشت API یا دریافت پاسخ	message و نسخه و اتصال PioService را بررسی کنید

برای ثبت فاکتور کلید جلوگیری از تکرار تضمین‌شده وجود ندارد؛ پس از قطع ارتباط یا پاسخ نامطمئن، پیش از ارسال مجدد وضعیت ثبت را در نرم‌افزار بررسی کنید. RemoveInvoice با GET داده را حذف می‌کند؛ آن را در پیش‌نمایش لینک، خزنده یا آزمایش خودکار بدون قصد حذف اجرا نکنید. نمونه‌های این سند فرضی هستند و اجرای POST و PUT داده واقعی را تغییر می‌دهد.

در نصب API، خالی گذاشتن پورت به معنی پورت 80 است؛ انتخاب پورت باید با دسترسی شبکه و میزبان سازگار باشد. باز شدن Swagger در رایانه کاربر، دسترسی هاست وردپرس به همان IP و پورت را اثبات نمی‌کند. خطای cURL 28 مربوط به پایان مهلت ارتباط است؛ وجود BOM را از روی این خطا نمی‌توان نتیجه گرفت.

بدنه JSON را با UTF-8 و هدر application/json ارسال کنید. اطلاعات محرمانه، رمز و توکن واقعی را در مستند یا گزارش عمومی قرار ندهید. برای محیط عملیاتی HTTPS و محدودسازی دسترسی شبکه را در نظر بگیرید. نسخه فعلی پروژه .NET 5 است؛ هشدارهای موجود درباره پایان پشتیبانی چارچوب و وابستگی‌ها با این انتشار مستندات رفع نشده‌اند.

انتشار و به‌روزرسانی نسخه 1.0.11

نصب‌کننده این انتشار PioWebApi-Setup-1.0.11.exe و مستندهای آن Pishgaman_API_Reference_FA_1.0.11.pdf و Pishgaman_API_Reference_EN_1.0.11.pdf هستند. مانده گروهی و گردش دوره‌ای، بدون حذف قابلیت‌های قبلی، در این نسخه قرار دارند.

PioService مقصد باید فرمان‌های 6006 و 6007 و حسابداری باید دو تیک دسترسی را پشتیبانی کند؛ نصب‌کننده فقط API را ارتقا می‌دهد. فروش و سفارش جدید در سرویس سازگار با مبدأ API برابر ۸ ثبت می‌شوند و سوابق قبلی بازنویسی نمی‌شوند. گزارش‌های جدید تغییر ساختار دیتابیس نیاز ندارند.

پیش از استقرار پشتیبان بگیرید و تنظیمات اتصال مقصد را کنترل کنید؛ appsettings موجود در ارتقای درجا حفظ می‌شود. پس از نصب هماهنگ، مانده گروهی، تفصیلی مجاز و غیرمجاز و صفحه دوم گردش را بررسی کنید. پورت خالی برابر ۸۰ است و .NET 5 Hosting Bundle در بسته قرار دارد.

آزمون گزارش‌های این انتشار شامل ۱۳۳ بررسی سرویس و SQL روی جدول‌های موقت، ۵۳ بررسی قرارداد API و ۳۱ بررسی HTTP و Swagger بوده است. هیچ اطلاعات یا تنظیمات واقعی مشتری در این تست‌ها تغییر نکرده است؛ نصب واقعی و پذیرش نهایی در محیط مقصد انجام نشده‌اند. SQL با نگارش سازگار با 2008 نوشته شده، ولی اجرای تست روی SQL Server 2016 بوده و اجرای واقعی موتور 2008 ادعا نمی‌شود.

ارتقای قرارداد قبلی ACK به سرویس و حسابداری سازگار، بازسازی دستی change_version و افزونه 3.1.5 نیاز دارد؛ گزارش‌های جدید بازسازی نمی‌خواهند. بسته سرویس جدا ارائه نمی‌شود.