



المستودعات السعودية
وإدارة الفواتير

SA4S Business ERP

دليل المطورين – واجهة SA4S البرمجية

اربط متجرك الإلكتروني أو موقعك أو نظام المحاسبة أو تطبيق التوصيل بنظام SA4S
Business ERP

إصدار الواجهة

v1.0.0

آخر تحديث

2026-10-06

الإصدار

1.1.0

<https://erp.sa4s.com/api/v1>



المحتويات

5	نظرة عامة
5	ماذا يمكنك أن تبني؟
5	مفاهيم أساسية
5	العنوان الأساسي والإصدارات
6	البدء السريع
6	1. أنشي مفتاحاً
6	2. نغذ أول طلب
7	3. عميل قابل لإعادة الاستخدام
12	المصادقة
12	إرسال المفتاح
12	صيغة المفتاح والبيئات
12	تدوير المفتاح (مهلة 24 ساعة)
12	الإلغاء والانتهاة
13	قائمة عناوين IP المسموحة
13	قائمة التحقق الأمنية
14	الصلاحيات
14	قائمة الصلاحيات
15	ما لا تستطيع المفاتيح أبداً
16	الاصطلاحات
16	التقسيم إلى صفحات
17	الترتيب
17	البحث والتصفية
18	التواريخ والمنطقة الزمنية
18	المبالغ والكميات
18	ضريبة القيمة المضافة
19	معرفات الطلبات
20	منع التكرار وإعادة المحاولة بأمان
22	حدود الطلبات
24	الأخطاء

24	جسم الخطأ
25	رموز الأخطاء
28	بيئة التجربة
29	وصفات - المنتجات
29	عرض المنتجات والبحث فيها
30	إنشاء منتج بوحدات وباركودات
33	تعديل منتج
33	البحث عن منتج بالباركود
35	وصفات - العملاء
35	إنشاء عميل
37	البحث عن عميل أو تعديله
38	كشف حساب العميل
40	وصفات - المبيعات
40	فاتورة مبسطة (للأفراد) شاملة الضريبة ومدفوعة بالبطاقة
42	فاتورة ضريبية (بين المنشآت) غير شاملة الضريبة ومدفوعة جزئياً
44	مسودة أولاً، ثم الإصدار عند تطابق الإجمالي
45	إعادة محاولة الفاتورة بأمان
45	تنزيل الفاتورة بصيغة PDF
46	حالة الفوترة الإلكترونية ورمز QR
47	الاسترجاع أو الإرجاع: الإشعار الدائن
48	إلغاء فاتورة
49	وصفات - المدفوعات
49	تسجيل سند قبض من عميل
50	عرض الفواتير غير المسددة
52	الدفع لمورد
54	وصفات - المشتريات
54	إنشاء فاتورة مشتريات
55	إنشاء أمر شراء
57	وصفات - المخزون
57	أرصدة المخزون حسب المستودع
59	حركات المخزون منذ تاريخ
61	وصفات - التقارير



61	قائمة التقارير
62	تشغيل تقرير بصيغة JSON
63	Webhooks
63	الاشتراك
63	صيغة الإرسال
64	قائمة الأحداث
66	التحقق من التوقيع
67	الحماية من إعادة الإرسال
68	إعادة المحاولة وإعادة الإرسال
68	الاختبار محلياً
70	مثال متكامل: طلبات المتجر الإلكتروني — فواتير SA4S
74	مثال متكامل: مزامنة المخزون
77	سجل التغييرات والإصدارات
78	الدعم

نظرة عامة

تتيح واجهة SA4S البرمجية لأنظمتك قراءة بيانات **شركتك** على SA4S وكتابتها: المنتجات والعلماء والموردين وفواتير المبيعات والمشتريات والمدفوعات والمخزون والتقارير. هي واجهة REST تتبادل JSON عبر HTTPS، وتؤمن بمفاتيح API تنشئها وتتحكم بها من داخل النظام، ومعها **Webhooks** موقّعة تُبلغ أنظمتك فور وقوع أي حدث.

الواجهة البرمجية متاحة في الباقيات التي تتضمن ميزة **API** (الأعمال والمؤسسات). تجد المفاتيح وال Webhooks والمرجع التفاعلي في **الإعدادات** ← **التكامل البرمجي (API)**.

ماذا يمكنك أن تبني؟

- **ربط متجر إلكتروني** — تحويل الطلبات المدفوعة إلى فواتير جاهزة لهيئة الزكاة والضريبة والجمارك، ومزامنة كميات المتجر مع المستودع.
- **تطبيقات نقاط البيع والتوصيل** — البحث عن المنتج بالباركود، وإنشاء الفواتير مع الدفعات، وتنزيل الفاتورة بصيغة PDF.
- **المحاسبة وذكاء الأعمال** — سحب الفواتير والمقبوضات وكشوف الحساب والتقارير الجاهزة (المبيعات، المخزون، الأرباح) بصيغة JSON أو ملفات.
- **أدوات المشتريات** — إنشاء أوامر الشراء وفواتير الموردين وتسجيل الدفعات للموردين.

مفاهيم أساسية

المفهوم	ماذا يعني للتكامل
الشركة (المستأجر)	كل مفتاح يتبع شركة واحدة فقط، وكل ما تقرأه أو تكتبه يخص تلك الشركة؛ أما سجلات الشركات الأخرى فلا وجود لها بالنسبة لك (404 NOT_FOUND). لا ترسل معرّف الشركة أبداً.
الفروع والمستودعات	كل مستند يتبع فرعاً ويحزّك المخزون في مستودع. إذا لم ترسل <code>branchId</code> أو <code>warehouseId</code> يُستخدم الفرع الرئيسي ومستودعه الافتراضي. تجد معرّفات المستودعات في صفوف المخزون (<code>warehouse.id</code>) وفي كل مستند.
المنتجات والوحدات	يُحفظ المخزون بـ الوحدة الأساسية للمنتج (مثل القطعة). للوحدات الإضافية (الكرتون = 24 قطعة) معامل <code>factor</code> ؛ وسطر الفاتورة بالكرتون يصرف 24 وحدة أساسية (<code>baseQuantity</code>).
المستندات	تمر الفاتورة بالمرحلة <code>DRAFT → ISSUED → CANCELLED</code> . الإصدار يمنح الرقم المتسلسل بلا فجوات، ويحزّك المخزون، ويرجّل القيود، ويضع الفاتورة الإلكترونية في طاوير الإرسال للهيئة. المستندات المُصدّرة لا تُعدّل أبداً — تُلغى أو يُصدّر لها إشعار دائن.
البيئات	مفاتيح <code>live</code> تعمل فعلياً، ومفاتيح <code>sandbox</code> تقرأ بيانات الشركة نفسها لكنها لا تنشئ إلا مسودات (انظر بيئة التجربة).

العنوان الأساسي والإصدارات

جميع المسارات نسبية إلى العنوان الأساسي:

العنوان الأساسي

`https://erp.sa4s.com/api/v1`

رقم الإصدار جزء من المسار (`/api/v1`). التغييرات داخل **v1 إضافية فقط**: مسارات جديدة، وحقول اختيارية جديدة في الطلب، وحقول جديدة في الاستجابة، ورموز أخطاء جديدة، وأحداث Webhook جديدة. اكتب عميلاً متسامحاً يتجاهل الحقول والأحداث غير المعروفة. أي تغيير غير متوافق سيصدر في `/api/v2` مع إشعار مسبق. وثيقة OpenAPI 3.1 القابلة للقراءة آلياً منشورة على `GET https://erp.sa4s.com/api/v1/developer/openapi.json` (دون مفتاح).



البدء السريع

من الصفر إلى أول طلب ناجح في دقيقتين تقريباً.

1. أنشئ مفتاحاً

1. في النظام افتح الإعدادات ← التكامل البرمجي (API) ← مفاتيح API ← مفتاح API جديد.
2. سمّه باسم النظام الذي سيستخدمه (مثل المتجر الإلكتروني)، واختر **Sandbox** أثناء التطوير، وحدّد الصلاحيات التي تحتاجها — يكفي في هذا المثال `products:view`.
3. انسخ المفتاح السري (`sk_test_...`)؛ فهو يُعرض مرة واحدة فقط، ولا يحفظ SA4S إلا بصمته المشفرة.

Shell

```
# keep the key out of your code: read it from the environment / a secret manager
export SA4S_API_KEY="sk_test_....."
export SA4S_BASE_URL="https://erp.sa4s.com/api/v1"
```

2. نفذ أول طلب

curl

```
curl "https://erp.sa4s.com/api/v1/products?pageSize=2" \
-H "Authorization: Bearer $SA4S_API_KEY"
```

JavaScript

```
// Node.js 18+ (built-in fetch)
const res = await fetch(`${process.env.SA4S_BASE_URL}/products?pageSize=2`, {
  headers: { Authorization: `Bearer ${process.env.SA4S_API_KEY}` },
});
console.log(res.status, res.headers.get('X-RateLimit-Remaining'));
console.log(await res.json());
```

Python

```
import os, requests

res = requests.get(
    f"{os.environ['SA4S_BASE_URL']}/products",
    params={"pageSize": 2},
    headers={"Authorization": f"Bearer {os.environ['SA4S_API_KEY']}"},
    timeout=30,
)
print(res.status_code, res.headers.get("X-RateLimit-Remaining"))
print(res.json())
```

PHP

```
<?php
$ch = curl_init(getenv('SA4S_BASE_URL') . '/products?pageSize=2');
curl_setopt_array($ch, [
    CURLOPT_RETURNTRANSFER => true,
```



```
CURLOPT_HTTPHEADER => ['Authorization: Bearer ' . getenv('SA4S_API_KEY')],
]);
$page = json_decode(curl_exec($ch), true);
echo curl_getinfo($ch, CURLINFO_RESPONSE_CODE), ' ', $page['total'], " products\n";
```

C#

```
using System.Net.Http.Headers;

using var http = new HttpClient();
http.DefaultRequestHeaders.Authorization =
    new AuthenticationHeaderValue("Bearer", Environment.GetEnvironmentVariable("SA4S_API_KEY"));
var res = await http.GetAsync($"{Environment.GetEnvironmentVariable("SA4S_BASE_URL")}/products?pageSize=2");
Console.WriteLine($"{(int)res.StatusCode} {await res.Content.ReadAsStringAsync()}");
```

200 OK (abridged)

```
{
  "items": [
    {
      "id": "51f771bf-cfb5-4814-8e79-55b990be69c9",
      "sku": "SKU-2026-000001",
      "nameAr": "مياه معدنية 330 مل",
      "nameEn": "Mineral water 330 ml",
      "type": "PHYSICAL",
      "sellingPrice": "1.15",
      "vatCategory": "S",
      "isActive": true,
      "stock": { "onHand": "282", "reserved": "0", "available": "282" }
    }
  ],
  "total": 3,
  "page": 1,
  "pageSize": 2
}
```

الاستجابة 200 مع `items` تعني أن المفتاح والصلاحيات والباقة كلها سليمة. أما 401 أو 403 فتخبرك بالضبط بما يجب إصلاحه — انظر الأخطاء.

3. عميل قابل لإعادة الاستخدام

تستخدم وصفات هذا الدليل الدالة المساعدة الصغيرة أدناه (لكل لغة نسختها) لتبقى الأمثلة قصيرة. فهي ترسل المفتاح، وتقرأ JSON، وترفع خطأً محدد النوع يحمل `code` و `details` و `requestId`، وتعيد المحاولة بأمان: دائماً عند 429، وعند أخطاء الشبكة أو 5xx فقط لطلبات GET ولعمليات الإنشاء التي تحمل `clientId` (انظر منع التكرار).

JavaScript

```
// sa4s.mjs - a tiny client for the SA4S Developer API (Node.js 18+, no dependencies)
const BASE = process.env.SA4S_BASE_URL ?? 'https://erp.sa4s.com/api/v1';
const KEY = process.env.SA4S_API_KEY; // sk_live... or sk_test...

export class Sa4sError extends Error {
  constructor(status, body) {
    super(`${body.code ?? status}: ${body.message ?? 'request failed'}`);
    Object.assign(this, { status, code: body.code, details: body.details, requestId: body.requestId });
  }
}

const sleep = (ms) => new Promise((resolve) => setTimeout(resolve, ms));
const backoff = (attempt) => Math.min(30_000, 500 * 2 ** attempt) * (0.5 + Math.random() / 2); // exponential + jitter
```



```

/** sa4s('GET', '/products?pageSize=50') · sa4s('POST', '/customers', { name: '...' }) */
export async function sa4s(method, path, body, { retries = 4 } = {}) {
  // GETs and creates that carry a clientId are safe to repeat; anything else is retried only on 429 (rejected before
  processing)
  const idempotent = method === 'GET' || Boolean(body?.clientId);
  for (let attempt = 0; ; attempt++) {
    let res;
    try {
      res = await fetch(BASE + path, {
        method,
        headers: { Authorization: `Bearer ${KEY}` }, ...(body ? { 'Content-Type': 'application/json' } : {}),
        body: body ? JSON.stringify(body) : undefined,
        signal: AbortSignal.timeout(30_000),
      });
    } catch (err) { // network error or timeout
      if (!idempotent || attempt >= retries) throw err;
      await sleep(backoff(attempt));
      continue;
    }
    if (res.ok) {
      if (res.status === 204) return null;
      return res.headers.get('content-type')?.includes('json') ? res.json() : Buffer.from(await res.arrayBuffer());
    }
    if (attempt < retries && (res.status === 429 || (res.status >= 500 && idempotent))) {
      const after = Number(res.headers.get('Retry-After'));
      await sleep(after > 0 ? after * 1000 : backoff(attempt));
      continue;
    }
    throw new Sa4sError(res.status, await res.json().catch(() => ({})));
  }
}

```

Python

```

# sa4s.py — a tiny client for the SA4S Developer API (pip install requests)
import os, random, time
import requests

BASE = os.environ.get("SA4S_BASE_URL", "https://erp.sa4s.com/api/v1")
session = requests.Session()
session.headers["Authorization"] = f"Bearer {os.environ['SA4S_API_KEY']}"

class Sa4sError(Exception):
    def __init__(self, status, body):
        super().__init__(f"{body.get('code', status)}: {body.get('message', 'request failed')}")
        self.status, self.code = status, body.get("code")
        self.details, self.request_id = body.get("details"), body.get("requestId")

def backoff(attempt):
    return min(30, 0.5 * 2 ** attempt) * random.uniform(0.5, 1) # exponential + jitter

def sa4s(method, path, json=None, params=None, retries=4):
    """sa4s("GET", "/products", params={"pageSize": 50}) · sa4s("POST", "/customers", {"name": "..."})"""
    idempotent = method == "GET" or bool(json and json.get("clientId"))
    for attempt in range(retries + 1):
        try:
            res = session.request(method, BASE + path, json=json, params=params, timeout=30)
        except requests.RequestException:
            if not idempotent or attempt == retries:
                raise
            time.sleep(backoff(attempt))
            continue

```



```

if res.ok:
    if res.status_code == 204:
        return None
    return res.json() if "json" in res.headers.get("Content-Type", "") else res.content
if attempt < retries and (res.status_code == 429 or (res.status_code >= 500 and idempotent)):
    time.sleep(float(res.headers.get("Retry-After") or backoff(attempt)))
    continue
try:
    body = res.json()
except ValueError:
    body = {}
raise Sa4sError(res.status_code, body)

```

PHP

```

<?php
// sa4s.php - a tiny client for the SA4S Developer API (PHP 8.1+, ext-curl, no dependencies)

final class Sa4sError extends RuntimeException
{
    public function __construct(public readonly int $status, public readonly array $body)
    {
        parent::__construct(($body['code'] ?? $status) . ': ' . ($body['message'] ?? 'request failed'));
    }
}

function sa4s_backoff_ms(int $attempt): int
{
    return (int) (min(30000, 500 * 2 ** $attempt) * (0.5 + mt_rand() / mt_getrandmax() / 2));
}

/** sa4s('GET', '/products?pageSize=50') · sa4s('POST', '/customers', ['name' => '...']) */
function sa4s(string $method, string $path, ?array $body = null, int $retries = 4): mixed
{
    $base = getenv('SA4S_BASE_URL') ?: 'https://erp.sa4s.com/api/v1';
    $idempotent = $method === 'GET' || !empty($body['clientId']);
    for ($attempt = 0; ; $attempt++) {
        $headers = [];
        $ch = curl_init($base . $path);
        $opts = [
            CURLOPT_CUSTOMREQUEST => $method,
            CURLOPT_RETURNTRANSFER => true,
            CURLOPT_TIMEOUT => 30,
            CURLOPT_HTTPHEADER => ['Authorization: Bearer ' . getenv('SA4S_API_KEY'), 'Content-Type: application/json'],
            CURLOPT_HEADERFUNCTION => function ($ch, string $line) use (&$headers): int {
                if (str_contains($line, ':')) {
                    [$k, $v] = explode(':', $line, 2);
                    $headers[strtolower(trim($k))] = trim($v);
                }
                return strlen($line);
            },
        ];
        if ($body !== null) {
            $opts[CURLOPT_POSTFIELDS] = json_encode($body, JSON_UNESCAPED_UNICODE | JSON_THROW_ON_ERROR);
        }
        curl_setopt_array($ch, $opts);
        $raw = curl_exec($ch);
        $status = (int) curl_getinfo($ch, CURLINFO_RESPONSE_CODE);
        if ($raw === false) { // network error or timeout
            if (!$idempotent || $attempt >= $retries) {
                throw new RuntimeException('SA4S request failed: ' . curl_error($ch));
            }
            usleep(sa4s_backoff_ms($attempt) * 1000);
            continue;
        }
    }
}

```



```

    }
    if ($status >= 200 && $status < 300) {
        if ($status === 204) return null;
        return str_contains($headers['content-type'] ?? '', 'json') ? json_decode($raw, true) : $raw;
    }
    if ($attempt < $retries && ($status === 429 || ($status >= 500 && $idempotent))) {
        $wait = isset($headers['retry-after']) ? (int) $headers['retry-after'] * 1000 : sa4s_backoff_ms($attempt);
        usleep($wait * 1000);
        continue;
    }
    throw new Sa4sError($status, json_decode($raw, true) ?? []);
}
}

```

C#

```

// Sa4s.cs - a tiny client for the SA4S Developer API (.NET 8+, no dependencies)
using System.Net;
using System.Net.Http.Headers;
using System.Net.Http.Json;
using System.Text.Json;

public sealed class Sa4sException(int status, JsonElement body) : Exception(
    $"{(body.TryGetProperty("code", out var c) ? c.GetString() : status.ToString())}: " +
    $"{(body.TryGetProperty("message", out var m) ? m.GetString() : "request failed")}")
{
    public int Status { get; } = status;
    public JsonElement Body { get; } = body; // code, message, details, requestId
}

public static class Sa4s
{
    public static readonly HttpClient Http = new()
    {
        BaseAddress = new Uri((Environment.GetEnvironmentVariable("SA4S_BASE_URL") ??
"https://erp.sa4s.com/api/v1").TrimEnd('/') + "/"),
        Timeout = TimeSpan.FromSeconds(30),
    };

    static Sa4s() => Http.DefaultRequestHeaders.Authorization =
        new AuthenticationHeaderValue("Bearer", Environment.GetEnvironmentVariable("SA4S_API_KEY"));

    static TimeSpan Backoff(int attempt) =>
        TimeSpan.FromMilliseconds(Math.Min(30_000, 500 * Math.Pow(2, attempt)) * (0.5 + Random.Shared.NextDouble() /
2));

    /// Sa4s.Send(HttpMethod.Get, "products?pageSize=50") · Sa4s.Send(HttpMethod.Post, "customers", new { name = "..." })
    public static async Task<JsonElement> Send(HttpMethod method, string path, object? body = null, int retries = 4)
    {
        bool idempotent = method == HttpMethod.Get
            || (body is not null && JsonSerializer.SerializeToElement(body).TryGetProperty("clientId", out _));
        for (int attempt = 0; ; attempt++)
        {
            using var req = new HttpRequestMessage(method, path.TrimStart('/'));
            if (body is not null) req.Content = JsonContent.Create(body);
            HttpResponseMessage res;
            try { res = await Http.SendAsync(req); }
            catch (Exception e) when ((e is HttpRequestException or TaskCanceledException) && idempotent && attempt <
retries)
            {
                await Task.Delay(Backoff(attempt));
                continue;
            }
            using (res)
            {

```



```
        if (res.IsSuccessStatusCode)
            return res.StatusCode == HttpStatusCode.NoContent ? default : await
res.Content.ReadFromJsonAsync<JsonElement>();
        int status = (int)res.StatusCode;
        if (attempt < retries && (status == 429 || (status >= 500 && idempotent)))
        {
            await Task.Delay(res.Headers.RetryAfter?.Delta ?? Backoff(attempt));
            continue;
        }
        throw new Sa4sException(status, await res.Content.ReadFromJsonAsync<JsonElement>());
    }
}
}
```



المصادقة

إرسال المفتاح

أرسل المفتاح مع كل طلب في أحد الترويسين؛ كلاهما متكافئ، فاستخدم أحدهما فقط.

HTTP

```
GET /api/v1/products HTTP/1.1
Host: erp.sa4s.com
Authorization: Bearer sk_live_.....

# or, equivalently
X-API-Key: sk_live_.....
```

الطلب بلا مفتاح يعود بـ `401 UNAUTHORIZED`. استخدم المفاتيح **على الخادم فقط**: لا تضعها في متصفح أو تطبيق جوال أو مستودع عام — فمن يحمل المفتاح السري يتصرف بكل صلاحياته.

صيغة المفتاح والبيئات

البادئة	البيئة	السلوك
<code>sk_live_</code> + 40 حرفاً	live	وصول كامل ضمن الصلاحيات الممنوحة.
<code>sk_test_</code> + 40 حرفاً	sandbox	بيانات الشركة نفسها، قراءة فقط باستثناء إنشاء المسودات (انظر بيئة التجربة).

أول 12 حرفاً (مثل `sk_live_XTPx`) هي بادئة المفتاح التي يعرضها النظام للتمييز بين المفاتيح، والباقي سري. ولكل مفتاح تاريخ انتهاء اختياري، وقائمة عناوين IP مسموحة اختيارية، وحد طلبات في الدقيقة (من 10 إلى 5000، والافتراضي 300).

تدوير المفتاح (مهلة 24 ساعة)

تدوير المفتاح من النظام يمنحك مفتاحاً سرياً جديداً بالاسم والصلاحيات والحدود نفسها، ويظل المفتاح القديم يعمل **24 ساعة** (الحالة قيد التدوير) ثم ينتهي، فتنتشر الجديد دون أي انقطاع:

1. دَوِّر المفتاح وانسخ السر الجديد.
 2. انشره في تكاملك (مدير الأسرار أو متغيرات البيئة).
 3. تأكد من ظهور الطلبات على المفتاح الجديد في مخطط الاستخدام؛ فالقديم يتوقف وحده بعد 24 ساعة.
- يُدَوِّر المفتاح مرة واحدة، وفي المرة التالية دَوِّر المفتاح الجديد. دَوِّر المفاتيح دورياً، وفوراً عند مغادرة أي شخص كان يطلع عليها.

الإلغاء والانهاء

إلغاء المفتاح يوقفه فوراً (`401 API_KEY_REVOKED`) ولا يمكن التراجع عنه — أنشئ مفتاحاً جديداً بدلاً منه. المفتاح الذي تجاوز تاريخ انتهائه يعود بـ `401 API_KEY_EXPIRED`. وعند الانتقال إلى باقة لا تتضمن الواجهة البرمجية تعود جميع المفاتيح بـ `403 FEATURE_NOT_AVAILABLE` إلى أن تتضمنها الباقة من جديد.



قائمة عناوين IP المسموحة

في التكامل بين الخوادم، قيّد المفتاح بعناوين خوادمك: عنوان IPv4 أو IPv6 أو نطاق CIDR في كل سطر (`203.0.113.10`) و `198.51.100.0/24` و `2001:::32`). الطلبات من أي عنوان آخر تعود بـ `401 API_KEY_IP_DENIED` . والقائمة الفارغة تسمح بكل العناوين.

قائمة التحقق الأمنية

- مفتاح مستقل لكل تكامل، بـ **أقل الصلاحيات** اللازمة، ومفاتيح منفصلة للتجربة والتشغيل الفعلي.
- احفظ الأسرار في مدير أسرار أو متغيرات بيئة — لا في الشيفرة ولا في السجلات ولا في الروابط.
- أضف تاريخ انتهاء للوصول المؤقت، وقائمة عناوين IP للخوادم الثابتة.
- راقب *آخر استخدام* ومخطط الاستخدام لكل مفتاح، وألغِ المفاتيح التي لم تعد تُستخدم.
- تحقق من توقيع كل Webhook وارفض الطوايع الزمنية القديمة (انظر الـ [Webhooks](#)).



الصلاحيات

لا يستطيع المفتاح إلا ما تسمح به صلاحياته. كل مسار يتطلب صلاحية واحدة (وقليل منها يتطلب اثنتين)، والطلب دونها يعود بـ 403 .
 PERMISSION_DENIED مع الصلاحية الناقصة في details.missing . تُكتب الصلاحيات بصيغة resource:action .

قائمة الصلاحيات

الإجراءات: **view** قراءة · **create** إنشاء · **edit** تعديل · **delete** حذف · **approve** اعتماد/ترحيل · **cancel** إلغاء · **refund** استرجاع وإشعارات
 داتنة · **print** ملفات **export** · PDF تصدير الملفات (تقارير CSV/XLSX/PDF). مفاتيح التجربة تحتفظ فقط بـ view و create و print و export.

المجموعة	المورد	الإجراءات	ما الذي يتيح
المنتجات	products	view, create, edit, delete, export, print	المنتجات ووحداتها وباركوداتها والبحث بالباركود
المنتجات	categories	view, create, edit, delete, export	قائمة التصنيفات وشجرتها
المنتجات	units	view, create, edit, delete, export	وحدات القياس
العملاء والموردون	customers	view, create, edit, delete, export, print	العملاء وكشوف حساباتهم
العملاء والموردون	suppliers	view, create, edit, delete, export, print	الموردون وكشوف حساباتهم
المبيعات	sales.invoice	view, create, edit, delete, export, approve, cancel, refund, print	فواتير المبيعات وإصدارها وإلغاؤها والإشعارات الداتنة وملف PDF
المبيعات	sales.discount	manage	خصومات الأسطر والمستند في الفواتير (discount و discountPercent و documentDiscount)
المبيعات	sales.return	view, create, approve	مرتجعات المبيعات
المبيعات	sales.delivery	view, print	أذونات التسليم (فواتير التسليم اللاحق والبيع دون توفر المخزون) وتقرير الفواتير غير المسلمة وملف PDF للإذن
المشتريات	purchases.request	view, create, edit, delete, export, approve	طلبات الشراء
المشتريات	purchases.order	view, create, edit, delete, export, approve, cancel, print	أوامر الشراء
المشتريات	purchases.receipt	view, create, cancel	استلام البضاعة
المشتريات	purchases.invoice	view, create, edit, delete, export, approve, cancel, print	فواتير المشتريات؛ و approve يرخلها
المشتريات	purchases.return	view, create, approve	مرتجعات المشتريات
المخزون	inventory	view, create, export	الأرصدة والحركات والنواقص والأرصدة الافتتاحية
المخزون	inventory.cost	view	أسعار الشراء ومتوسط التكلفة وقيمة المخزون
المخزون	inventory.adjust	create, approve	تسويات المخزون

المجموعة	المورد	الإجراءات	ما الذي يتيح
المخزون	inventory.transfer	view, create, approve, cancel	التحويلات بين المستودعات
المخزون	inventory.count	view, create, approve	الجرد
المالية	payments	view, create, edit, delete, export, refund, cancel	مقبوضات العملاء ومدفوعات الموردين
المالية	expenses	view, create, edit, delete, export, approve, cancel	المصروفات
المالية	revenues	view, create, edit, delete, export, cancel	الإيرادات الأخرى
المالية	cash	view	الحسابات النقدية والبنكية
المحاسبة	accounting	view, export	التقارير المبنية على دفتر الأستاذ (الأرباح والخسائر، الميزانية، ميزان المراجعة، أعمار الديون)
التقارير	reports	view, export	محرك التقارير؛ ويحتاج كل تقرير أيضاً صلاحية بياناته

نصيحة

يحتاج المتجر الإلكتروني عادةً إلى `products:view` و `customers:view` و `customers:create` و `sales.invoice:view` و `sales.invoice:create` و `sales.invoice:print` و `sales.invoice:refund` و `inventory:view`. ولا تضاف `inventory.cost:view` إلا إذا احتجت التكاليف فعلاً.

ما لا تستطيعه المفاتيح أبداً

المجالات الإدارية خارج الواجهة البرمجية مهما كانت الصلاحيات: المستخدمون والأدوار، وإعدادات الشركة، والاشتراك والفوترة، والربط مع هيئة الزكاة والضريبة والجمارك، وإدارة المفاتيح وال Webhooks، وسجل التدقيق، والإشعارات، والبحث الشامل، ومسارات أجهزة نقاط البيع. وتعود جميعها بـ `403 API_KEY_NOT_ALLOWED`. كما لا يُمنح للمفاتيح تقرير إقرار ضريبة القيمة المضافة (`vat:view`).



الاصطلاحات

التقسيم إلى صفحات

كل قائمة تعود بالغلاف نفسه، وتقبل `page` (تبدأ من 1) و `pageSize` (من 1 إلى 200، والافتراضي 25):

غلاف القائمة

```
{ "items": [ ... ], "total": 128, "page": 2, "pageSize": 50 }
```

اطلب الصفحة التالية ما دام `page * pageSize < total`. وأي `pageSize` أكبر من 200 يُرفض بـ `400 VALIDATION_ERROR` (تسمح مسارات التقارير بحق 500).

JavaScript

```
// walk every page (pageSize ≤ 200); stop when a page comes back short
async function* allProducts(filters = {}) {
  for (let page = 1; ; page++) {
    const q = new URLSearchParams({ ...filters, page: String(page), pageSize: '200', sort: 'createdAt', order: 'asc' });
  });
  const res = await sa4s('GET', `/products?${q}`);
  yield* res.items;
  if (page * res.pageSize >= res.total) return;
}
for await (const p of allProducts({ active: 'true' })) console.log(p.sku);
```

Python

```
def all_products(**filters):
    page = 1
    while True:
        res = sa4s("GET", "/products", params={**filters, "page": page, "pageSize": 200, "sort": "createdAt", "order": "asc"})
        yield from res["items"]
        if page * res["pageSize"] >= res["total"]:
            return
        page += 1

for p in all_products(active="true"):
    print(p["sku"])
```

PHP

```
<?php
function all_products(array $filters = []): Generator
{
    for ($page = 1; ; $page++) {
        $res = sa4s('GET', '/products?' . http_build_query($filters + ['page' => $page, 'pageSize' => 200, 'sort' => 'createdAt', 'order' => 'asc']));
        yield from $res['items'];
        if ($page * $res['pageSize'] >= $res['total']) return;
    }
}
foreach (all_products(['active' => 'true']) as $p) echo $p['sku'], "\n";
```



C#

```
static async IEnumerable<JsonElement> AllProducts()
{
    for (int page = 1; ; page++)
    {
        var res = await Sa4s.Send(HttpMethod.Get, $"products?page={page}&pageSize=200&sort=createdAt&order=asc");
        foreach (var p in res.GetProperty("items").EnumerateArray()) yield return p;
        if (page * res.GetProperty("pageSize").GetInt32() >= res.GetProperty("total").GetInt32()) yield break;
    }
}
await foreach (var p in AllProducts()) Console.WriteLine(p.GetProperty("sku"));
```

الترتيب

يأخذ `sort` اسم العمود، و `order` إما `asc` أو `desc` (الافتراضي `desc`). لكل قائمة أعمدة محددة، وأي قيمة غير معروفة في `sort` تعود بصمت إلى الترتيب الافتراضي (غالباً `createdAt`).

المسار	الأعمدة القابلة للترتيب
GET /products	updatedAt , createdAt , minStock , sellingPrice , nameEn , nameAr , sku (و purchasePrice و cost مع inventory.cost:view)
GET /customers	createdAt , creditLimit , balance , code , name
GET /suppliers	createdAt , creditLimit , balance , code , nameEn , nameAr
GET /sales/invoices	paymentStatus , status , total , number , issueDate , createdAt
GET /purchases/invoices	createdAt , total , status , dueDate , date , number
GET /payments	createdAt , amount , number , date
GET /inventory/balances	warehouse , updatedAt , available , onHand , name , sku (و value و avgCost مع صلاحية التكلفة)
GET /inventory/movements	دائماً حسب الوقت — استخدم order=asc لقراءة الأقدم أولاً

البحث والتصفية

`search` بحث جزئي لا يميز حالة الأحرف في الحقول الطبيعية لكل قائمة:

- المنتجات — الاسم العربي والإنجليزي ورمز الصنف SKU وأي باركود؛
- العملاء — الاسم والاسم الإنجليزي والجوال والرقم الضريبي والرمز والبريد والسجل التجاري، والموردون كذلك؛
- فواتير المبيعات — الرقم والملاحظات واسم العميل أو جواله أو رقمه الضريبي؛
- أرصدة المخزون — رمز الصنف والأسماء.

ولكل قائمة مرشحات دقيقة أيضاً (مثل `GET /products?categoryId=...&active=true&type=SERVICE` و `GET /sales/invoices?` `customerId=...&status=ISSUED&paymentStatus=UNPAID`). القيم المنطقية `true` / `false`. تُرفض معاملات الاستعلام غير المعروفة بـ `400 VALIDATION_ERROR` (مثل `property from should not exist`)، فلا تمر الأخطاء الإملائية بصمت. القائمة الكاملة لكل مسار في تبويب مرجع الواجهة.

التواريخ والمنطقة الزمنية

- الطوابع الزمنية في الاستجابات بصيغة ISO 8601 وبتوقيت UTC: "2026-10-06T17:01:02.289Z".
- مرشحات التاريخ بصيغة YYYY-MM-DD (from و to و dateFrom و dateTo و asOf) أيام تقويمية بتوقيت (UTC+3) Asia/Riyadh: to=2026-10-31 يشمل يوم 31 أكتوبر كاملاً بتوقيت الرياض.
- يقبل GET /sales/invoices أيضاً طابعاً زمنياً كاملاً في from / to (from=2026-10-06T14:00:00Z) — وهذا مفيد للمزامنة التزايدية.
- في جسم الطلب تأخذ الحقول التاريخية مثل dueDate الصيغة YYYY-MM-DD، ويمكن أن يكون issueDate طابعاً زمنياً كاملاً.

المبالغ والكميات

المبالغ والكميات **نصوص عشرية** في الاتجاهين ("115.00" و "2.5"). لا تحولها إلى أعداد عشرية ثنائية للحساب — استخدم نوعاً عشرياً (decimal في #C و Python و bcmath في PHP، وعدد الهللات الصحيح في JavaScript). تعيد الاستجابات القيمة دون أصفار زائدة ("270.25" و "100" و "47.5")، فنسحقها للعرض بنفسك. كل المبالغ بعملة الشركة (الريال السعودي). الكمية في السطر بوحدة السطر، و baseQuantity يعطي الكمية نفسها بالوحدة الأساسية.

ضريبة القيمة المضافة

يحسب SA4S الإجماليات دائماً — ترسل الأسطر ولا ترسل الإجماليات أبداً (أي حقل total غير معروف يُرفض). لكل منتج فئة ضريبية:

الفئة	المعنى	النسبة
S	خاضع للنسبة الأساسية	نسبة الشركة (15%) ما لم يحدد المنتج vatRate
Z	خاضع لنسبة الصفر (مثل الصادرات المؤهلة)	0%
E	معفى (مثل بعض الخدمات المالية)	0%. ويُفصح عنه منفصلاً
O	خارج نطاق الضريبة	0%. ولا يُفصح عنه

يحدد pricesIncludeVat معنى أسعار الوحدات التي ترسلها: true = شاملة الضريبة (المعتاد في البيع للأفراد والمتاجر الإلكترونية)، و false = أسعار صافية (المعتاد بين المنشآت). وإذا لم ترسله يُطبّق إعداد الشركة الافتراضي. تُقَرَّب المبالغ إلى منزلتين عشريتين **لكل سطر**، وإجماليات المستند هي مجموع الأسطر المقربة، فتتطابق الأسطر المطبوعة دائماً مع الإجمالي المطبوع:

- **شاملة الضريبة (true):** إجمالي السطر = الكمية × سعر الوحدة - الخصومات (ومنها حصة السطر من خصم المستند) مقرباً إلى منزلتين — وهو بالضبط ما يدفعه العميل. وتُستخرج الضريبة منه: تقريب (إجمالي السطر × النسبة ÷ (100 + النسبة)) = vatAmount، و vatAmount = إجمالي السطر - taxableAmount. مثال: 40.00 × 2 ← taxableAmount 69.57 + vatAmount 10.43 = lineTotal 80.00.
- **غير شاملة الضريبة (false):** تقريب (الكمية × سعر الوحدة - الخصومات) = taxableAmount، و vatAmount = تقريب (taxableAmount × النسبة ÷ 100). lineTotal = taxableAmount + vatAmount.

تحمل الفاتورة الإلكترونية المرسلة إلى الهيئة هذه المبالغ نفسها على مستوى السطر والمستند (وتبقى ضريبة السطر وضريبة الفئة ضمن هامش التقريب الذي تسمح به قواعد التحقق في فاتورة). قيمة total في الاستجابة هي المرجع — وعليها احسب المبلغ المحصّل أو قارن.

مهم

خصومات الأسطر (discount و discountPercent) و documentDiscount تتطلب منح المفتاح الصلاحية sales.discount:manage (ويمكن منحها لمفاتيح التجربة أيضاً). وبدونها تُرفض بـ 403 DISCOUNT_NOT_ALLOWED، ويمكنك بدلاً من ذلك إرسال unitPrice بعد الخصم.



معرفات الطلبات

كل استجابة تحمل الترويسة `X-Request-Id` ، وكل جسم خطأ يكررها في `requestId` . احفظها في سجلاتك واذكرها عند التواصل مع الدعم. ويمكنك إرسال ترويساتك `X-Request-Id` لربط الطلبات بسجلاتك.

HTTP

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-RateLimit-Limit: 300
X-RateLimit-Remaining: 297
X-Request-Id: 7dfe4b4e-31f0-4721-af65-2553d47bc596
```



منع التكرار وإعادة المحاولة بأمان

الشبكات تتعطل: قد يصل الطلب إلى SA4S بينما تضيع الاستجابة، فتؤدي إعادة POST عادي إلى فاتورة ثانية. لذلك تقبل عمليات الإنشاء التي تحرك الأموال الحقل `clientId` — وهو معرف UUID تختاره أنت:

المسار	الحقل
POST /sales/invoices	clientId
POST /sales/invoices/{id}/credit-note	clientId
POST /payments/receipts	clientId
POST /payments/disbursements	clientId
payments[] داخل الفاتورة	clientId لكل دفعة

إعادة الطلب بـ `clientId` نفسه تعيد السجل الأصلي (201 بالمرء والرقم نفسيهما)، وتضيف استجابة الفاتورة: "idempotentReplay": true. لا يُنشأ شيء مرتين حتى لو تزامنت محاولتان. قواعد عملية:

1. ولّد المعرف قبل المحاولة الأولى واحفظه مع سجلك (الطلب، الحوالة...). والأفضل أن تشتقه من معرفك بـ UUID v5 حتى يبقى المفتاح نفسه حتى لو فُقدت حالتك المحلية (انظر مثال مزامنة الطلبات).
2. أعد المحاولة بالجسم نفسه تماماً؛ فالجسم المختلف مع `clientId` مستخدم يعيد السجل الأول أيضاً.
3. أعد المحاولة عند أخطاء الشبكة وانتهاء المهلة و 429 و 5xx، ولا تعدها عند بقية أخطاء 4xx — أصلح الطلب.
4. بقية عمليات الكتابة (المنتجات، العملاء، الإصدار، الإلغاء) لا تحمل مفتاحاً: قبل إعادتها اقرأ الحالة لتعرف هل نجحت المحاولة الأولى (مثلاً إصدار فاتورة مُصدرة يعود بـ 409 INVOICE_NOT_ISSUABLE، ويمكنك اعتباره نجاحاً).

JavaScript

```
// 1. create the key once, persist it next to the order (DB column, order meta...)
order.sa4sClientId ??= randomUUID();
await saveOrder(order);

// 2. any number of attempts with the SAME body + clientId - at most one invoice is ever created
const invoice = await sa4s('POST', '/sales/invoices', buildInvoice(order)); // the client helper retries
429/5xx/timeouts
if (invoice.idempotentReplay) console.log('already created earlier:', invoice.number);
```

Python

```
order.setdefault("sa4s_client_uuid", str(uuid.uuid4()))
save_order(order)
invoice = sa4s("POST", "/sales/invoices", build_invoice(order))
if invoice.get("idempotentReplay"):
    print("already created earlier:", invoice["number"])
```

PHP

```
<?php
function uuid4(): string
{
    $b = random_bytes(16);
    $b[6] = chr(ord($b[6]) & 0x0f | 0x40); // version 4
    $b[8] = chr(ord($b[8]) & 0x3f | 0x80); // RFC 4122 variant
    return vsprintf('%s%s-%s-%s-%s-%s', str_split(bin2hex($b), 4));
}
```



```

}

$order['sa4s_client_uuid'] ??= uuid4();
save_order($order);
$invoice = sa4s('POST', '/sales/invoices', build_invoice($order));
if (!empty($invoice['idempotentReplay'])) echo "already created earlier: {$invoice['number']}\n";

```

C#

```

order.Sa4sClientId ??= Guid.NewGuid();
await SaveOrder(order);
var invoice = await Sa4s.Send(HttpMethod.Post, "sales/invoices", BuildInvoice(order));
if (invoice.TryGetProperty("idempotentReplay", out var replay) && replay.GetBoolean())
    Console.WriteLine($"already created earlier: {invoice.GetProperty("number")}");

```

201 Created — replay of the same clientId (abridged)

```

{
  "id": "442ea37d-b9ac-4535-b726-b3ad69832afe",
  "number": "INV-2026-000007",
  "status": "ISSUED",
  "total": "46.9",
  "clientId": "09c3d295-3127-400b-ab9c-819ab51b011c",
  "idempotentReplay": true
}

```



حدود الطلبات

لكل مفتاح حده الخاص (افتراضياً **300 طلب في الدقيقة**، ويمكن ضبطه لكل مفتاح من 10 إلى 5000) ويُقاس على نافذة منزلة مدتها دقيقة. وكل استجابة تخبرك بموقفك:

المعنى	الترويسة
عدد الطلبات المسموح بها في الدقيقة لهذا المفتاح	X-RateLimit-Limit
الطلبات المتبقية في النافذة الحالية	X-RateLimit-Remaining
مع 429 فقط: عدد الثواني قبل المحاولة التالية	Retry-After

عند تجاوز الحد تعود الواجهة بـ `429 RATE_LIMITED` (لم يُعالج الطلب، فإعادته آمنة دائماً). ويطبّق أيضاً حد حماية لكل عنوان IP على كل الطلبات. الوصفة: التزم بـ `Retry-After`، وإلا فتراجع أُشياً مع عشوائية (0.5 ث، 1 ث، 2 ث، 4 ث... بحد أقصى 30 ث)، ووزّع المهام الكبيرة بالتساوي، وفُضّل الـ Webhooks على الاستطلاع المتكرر.

HTTP

```

HTTP/1.1 429 Too Many Requests
X-RateLimit-Limit: 10
X-RateLimit-Remaining: 0
Retry-After: 60
Content-Type: application/json; charset=utf-8

{"statusCode":429,"code":"RATE_LIMITED","message":"Rate limit of 10 requests per minute exceeded","details":{"limit":10,"retryAfterSeconds":60},"requestId":"71b8d797-785b-416a-b56c-9bf23f98011c"}

```

JavaScript

```

// spread a bulk job under the key's Limit instead of hitting 429 (e.g. 300/min → one call every 200 ms)
const LIMIT_PER_MINUTE = 300;
const gap = 60_000 / LIMIT_PER_MINUTE;
for (const sku of skus) {
  const started = Date.now();
  await syncOne(sku);
  await new Promise((r) => setTimeout(r, Math.max(0, gap - (Date.now() - started))));
}

```

Python

```

import time

LIMIT_PER_MINUTE = 300
gap = 60 / LIMIT_PER_MINUTE
for sku in skus:
    started = time.monotonic()
    sync_one(sku)
    time.sleep(max(0, gap - (time.monotonic() - started)))

```

PHP

```

<?php
$gapUs = (int) (60_000_000 / 300);
foreach ($skus as $sku) {
    $t = hrtime(true);

```



```
sync_one($sku);  
usleep(max(0, $gapUs - (int) ((hrtime(true) - $t) / 1000)));  
}
```

C#

```
var gap = TimeSpan.FromMinutes(1) / 300;  
foreach (var sku in skus)  
{  
    var started = DateTime.UtcNow;  
    await SyncOne(sku);  
    var wait = gap - (DateTime.UtcNow - started);  
    if (wait > TimeSpan.Zero) await Task.Delay(wait);  
}
```



الأخطاء

جسم الخطأ

تستخدم الأخطاء رموز HTTP القياسية والجسم نفسه دائماً. اعتمد في منطقك على **code** (ثابت وبأحرف كبيرة) — لا على **message** فهو نص تشخيصي إنجليزي قد يتغير. ويحمل **details** بيانات خاصة بكل رمز مثل الصلاحية الناقصة أو رسائل التحقق.

403 Forbidden

```
{
  "statusCode": 403,
  "code": "PERMISSION_DENIED",
  "message": "You do not have permission to perform this action",
  "details": { "missing": ["products:create"] },
  "requestId": "f20f408d-cc0a-4d0f-9f92-e1d989bfb96e"
}
```

400 Bad Request

```
{
  "statusCode": 400,
  "code": "VALIDATION_ERROR",
  "message": "Validation failed",
  "details": ["property total should not exist"],
  "requestId": "aca1aa3b-cbc9-4d07-912a-121298863808"
}
```

JavaScript

```
import { sa4s, Sa4sError } from './sa4s.mjs';

export async function pushInvoice(order, invoice) {
  try {
    return await sa4s('POST', '/sales/invoices', invoice);
  } catch (e) {
    if (!(e instanceof Sa4sError)) throw e; // network problem after all retries
    switch (e.code) {
      case 'INSUFFICIENT_STOCK': return holdOrder(order, e.details); // business rule: fix data, do not retry
      case 'VALIDATION_ERROR': return logBug(e.details); // your payload is wrong
      case 'PERMISSION_DENIED': return alertAdmin(`grant ${e.details.missing.join(', ')}`);
      case 'API_KEY_REVOKED': case 'API_KEY_EXPIRED': return stopSync('key no longer valid');
      default: throw new Error(`${e.message} (requestId ${e.requestId})`);
    }
  }
}
```

Python

```
from sa4s import sa4s, Sa4sError

try:
    sa4s("POST", "/sales/invoices", invoice)
except Sa4sError as e:
    if e.code == "INSUFFICIENT_STOCK":
        hold_order(order, e.details)
```

```

elif e.code == "PERMISSION_DENIED":
    alert_admin('grant ' + ' '.join(e.details["missing"]))
else:
    raise RuntimeError(f"{e} (requestId {e.request_id})") from e

```

PHP

```

<?php
try {
    sa4s('POST', '/sales/invoices', $invoice);
} catch (Sa4sError $e) {
    match ($e->body['code'] ?? null) {
        'INSUFFICIENT_STOCK' => hold_order($order, $e->body['details']),
        'PERMISSION_DENIED' => alert_admin('grant ' . implode(' ', $e->body['details']['missing'])),
        default => throw new RuntimeException($e->getMessage() . ' (requestId ' . ($e->body['requestId'] ?? '?') . ')',
0, $e),
    };
}

```

C#

```

try
{
    await Sa4s.Send(HttpMethod.Post, "sales/invoices", invoice);
}
catch (Sa4sException e) when (e.Body.GetProperty("code").GetString() == "INSUFFICIENT_STOCK")
{
    await HoldOrder(order, e.Body.GetProperty("details"));
}

```

رموز الأخطاء

الرموز التي قد يتلقاها عميل يستخدم مفتاح API. وتأتي رموز منطق الأعمال مع 400 أو 404 أو 409.

الرمز	الحالة	المعنى	ماذا تفعل
UNAUTHORIZED	401	لم يُرسل مفتاح (ولا جلسة)	أرسل Authorization: Bearer sk_... أو X-API-Key
API_KEY_INVALID	401	مفتاح بصيغة خاطئة أو غير معروف أو لا تطابق بيئته بادئته	تأكد من نسخ المفتاح كاملاً
API_KEY_REVOKED	401	المفتاح ملغى	أنشئ مفتاحاً جديداً
API_KEY_EXPIRED	401	تجاوز تاريخ انتهائه (ومنه المفتاح القديم بعد 24 ساعة من التدوير)	استخدم المفتاح الجديد
API_KEY_IP_DENIED	401	عنوان المتصل ليس في القائمة المسموحة	أضف عنوان الخادم أو نطاقه
FEATURE_NOT_AVAILABLE	403	الباقة لا تتضمن الواجهة البرمجية (أو الوحدة، مثل التقارير)	رقّ الباقة
COMPANY_SUSPENDED	403	حساب الشركة موقوف	تواصل مع مالك الحساب
API_KEY_NOT_ALLOWED	403	المسار ليس جزءاً من الواجهة البرمجية	استخدم المسارات الموثقة فقط
PERMISSION_DENIED	403	صلاحية ناقصة (details.missing)	عدّل المفتاح وأضف الصلاحية

الرمز	الحالة	المعنى	ماذا تفعل
SANDBOX_READ_ONLY	403	مفتاح تجربة حاول تعديل البيانات	أنشئ مسودات فقط أو استخدم مفتاحاً فعلياً
DISCOUNT_NOT_ALLOWED	403	أُرسلت حقول خصم بمفتاح لا يملك <code>sales.discount:manage</code>	امنح الصلاحية أو أرسل <code>unitPrice</code> بعد الخصم
SUBSCRIPTION_EXPIRED	402	انتهى الاشتراك — الكتابة محظورة والقراءة مستمرة	جُدّد الاشتراك
PLAN_LIMIT_REACHED	402	استُنفدت حصة في الباقة (مثل عدد الفواتير الشهري؛ <code>details.key</code> و <code>details.limit</code>)	رُقِّ الباقة أو انتظر الفترة التالية
RATE_LIMITED	429	طلبات كثيرة جداً	انتظر <code>Retry-After</code> ثانية
VALIDATION_ERROR	400	الجسم أو الاستعلام غير صالح، والحقول غير المعروفة مرفوضة (<code>details</code>) يسرد الرسائل)	أصلح الطلب
BAD_REQUEST	400	معامل مسار غير صالح (مثل معرف ليس (UUID	أرسل معرّفاً صحيحاً
NOT_FOUND	404	لا يوجد سجل بهذا المعرف في شركتك	تحقق من المعرف
DUPLICATE	409	قيمة فريدة موجودة مسبقاً (<code>details.fields</code>)	استخدم قيمة أخرى أو حدّث السجل الموجود
IN_USE / CONFLICT	409	تغيّر السجل في الأثناء / مرتبط ببيانات أخرى	أعد القراءة ثم المحاولة، أو اتركه
BARCODE_TAKEN , SKU_TAKEN , VAT_NUMBER_TAKEN , CODE_TAKEN	409	القيمة تخص سجلاً آخر (يذكره <code>details</code>)	حدّث ذلك السجل بدلاً من ذلك
BARCODE_NOT_FOUND	404	لم يجد <code>GET /products/lookup</code> أي منتج	اعتبره منتجاً غير معروف
COST_PERMISSION_REQUIRED	403	أُرسل <code>purchasePrice</code> دون <code>inventory.cost:view</code>	أضف الصلاحية أو احذف الحقول
PRODUCT_NOT_FOUND , UNIT_NOT_FOUND , PRODUCT_INACTIVE	400/409	السطر يشير إلى منتج غير موجود أو غير نشط أو وحدة ليست له	أصلح السطر
CUSTOMER_REQUIRED , CUSTOMER_NOT_FOUND , CUSTOMER_INACTIVE	400/409	الفاتورة الضريبية تحتاج عميلاً صالحاً ونشطاً	أرسل <code>customerId</code>
INSUFFICIENT_STOCK	409	المخزون غير كافٍ (<code>details.available</code>) و (<code>details.requested</code>)	زوّد المخزون أو غيّر الكمية
CREDIT_LIMIT_EXCEEDED	409	البيع يتجاوز حد ائتمان العميل	حصّل الدفعة أو ارفع الحد
PAYMENT_REQUIRES_ISSUED	400	أُرسلت دفعات مع مسودة	أرسل <code>status: "ISSUED"</code> أو ادفع عند الإصدار
OVERPAYMENT	409	الدفعات أكبر من إجمالي الفاتورة أو رصيدها المفتوح	ادفع بحد أقصى <code>total</code>

الرمز	الحالة	المعنى	ماذا تفعل
INVOICE_NOT_ISSUABLE	409	الفاتورة مُصدّرة أو ملغاة مسبقاً	اعتبرها منجزة (إصدار غير مكرر)
INVOICE_HAS_PAYMENTS INVOICE_HAS_NOTES	409	الإلغاء ممنوع لوجود دفعات أو إشعارات	أصدر إشعاراً دائماً بدلاً منه
ZATCA_REPORTED_USE_CREDIT_NOTE	409	الفاتورة أُرسلت إلى الهيئة مسبقاً	أصدر إشعاراً دائماً
ALLOCATION_EXCEEDS_PAYMENT	400	مجموع التوزيعات أكبر من مبلغ الدفعة	صحّح مبالغ التوزيع
INVOICE_NOT_PAYABLE INVOICE_PARTY_MISMATCH	409	توزيع على فاتورة غير مُصدّرة أو تخص طرفاً آخر	وزّع على فواتير الطرف نفسه المفتوحة
INSUFFICIENT_FUNDS	409	رصيد الحساب النقدي/البنكي الدافع غير كافٍ	اختر accountId آخر
INVOICE_NOT_DRAFT DUPLICATE_SUPPLIER_INVOICE	409	فاتورة المشتريات مرحلة مسبقاً / رقم المورد مستخدم	اقرأ المستند الموجود
REPORT_NOT_FOUND	404	اسم تقرير غير معروف (details.available يسرد الأسماء)	استخدم اسماً من GET /reports
INTERNAL_ERROR	500	عطل غير متوقع لدينا	أعد المحاولة بتراجع تدريجي وأبلغ عن requestId

409 Conflict

```
{
  "statusCode": 409,
  "code": "INSUFFICIENT_STOCK",
  "message": "Insufficient stock for Sukkari dates 1 kg",
  "details": { "productId": "f24f96e1-7e8a-4ba6-8fc3-c1f235a16f41", "warehouseId": "6f46787b-84c8-453c-a722-7ed26ae45313", "available": "45", "requested": "100000" },
  "requestId": "d3601ec7-6c3c-4c8d-ade8-50d7bdcfaf06"
}
```



بيئة التجربة

تتيح مفاتيح التجربة (sk_test_...) بناء التكامل واختباره على بياناتك الحقيقية دون أي أثر جانبي:

- يعمل كل طلب GET ضمن صلاحيات القراءة والطباعة والتصدير (بما فيها ملفات PDF والتقارير)؛
 - يُسمح بـ POST فقط لإنشاء مسودات من: فواتير المبيعات، وطلبات الشراء وأوامرها واستلامها وفواتيرها ومرجعاتها، وتسويات المخزون وتحويلاته وجرده — مع عدم إرسال status أو إرساله "DRAFT"، ودون post: true؛
 - وكل ما عدا ذلك — إنشاء العملاء أو المنتجات، والإصدار، والترحيل، والدفعات، والإلغاء، والتعديل والحذف — يعود بـ 403 ؛SANDBOX_READ_ONLY
 - تُحذف تلقائياً من مفاتيح التجربة كل الصلاحيات عدا create و view و print و export.
- المسودات المنشأة بمفتاح تجربة مسودات عادية تظهر في النظام ويمكن حذفها منه. وحين يعمل التكامل أنشئ مفتاحاً فعلياً بالصلاحيات نفسها واستبدل المفتاح السري.

403 Forbidden — a sandbox key tried to issue an invoice

```
{
  "statusCode": 403,
  "code": "SANDBOX_READ_ONLY",
  "message": "Sandbox keys can only read data and create DRAFT documents",
  "details": {
    "allowed": ["/sales/invoices", "/purchases/requests", "/purchases/orders", "/purchases/receipts",
"/purchases/invoices", "/purchases/returns", "/inventory/adjustments", "/inventory/transfers", "/inventory/counts"]
  },
  "requestId": "22bb9db1-0f29-48d7-bad4-fb18a5330f7a"
}
```



وصفات — المنتجات

تعرض كل وصفة الطلب بـ curl وبالعَميل في لغتك (انظر عميل قابل لإعادة الاستخدام)، ثم استجابة حقيقية من الواجهة (الاستجابات الطويلة مختصرة، والحقول غير المعروضة تُعاد كذلك).

عرض المنتجات والبحث فيها

GET /products — الصلاحية products:view. المرشحات: search، و categoryId (يشمل التصنيفات الفرعية)، و brandId، و active، و type (PHYSICAL و SERVICE و COMPOSITE و BUNDLE)، و lowStock، و warehouseId (لحصر أرقام stock) في مستودع واحد).

curl

```
curl "https://erp.sa4s.com/api/v1/products?page=1&pageSize=20&search=water&sort=nameEn&order=asc&active=true" \
-H "Authorization: Bearer $SA4S_API_KEY"
```

JavaScript

```
import { sa4s } from './sa4s.mjs';

const q = new URLSearchParams({ page: '1', pageSize: '20', search: 'water', sort: 'nameEn', order: 'asc', active: 'true' });
const { items, total } = await sa4s('GET', `/products?${q}`);
console.log(`${items.length} of ${total}`, items.map((p) => `${p.sku} ${p.nameEn} ${p.sellingPrice}`));
```

Python

```
from sa4s import sa4s

page = sa4s("GET", "/products", params={"page": 1, "pageSize": 20, "search": "water", "sort": "nameEn", "order": "asc", "active": "true"})
for p in page["items"]:
    print(p["sku"], p["nameEn"], p["sellingPrice"], p["stock"]["available"])
```

PHP

```
<?php
require 'sa4s.php';

$q = http_build_query(['page' => 1, 'pageSize' => 20, 'search' => 'water', 'sort' => 'nameEn', 'order' => 'asc', 'active' => 'true']);
$page = sa4s('GET', "/products?$q");
foreach ($page['items'] as $p) {
    echo "{$p['sku']} {$p['nameEn']} {$p['sellingPrice']}\n";
}
```

C#

```
var page = await Sa4s.Send(HttpMethod.Get, "products?page=1&pageSize=20&search=water&sort=nameEn&order=asc&active=true");
foreach (var p in page.GetProperty("items").EnumerateArray())
    Console.WriteLine($"{p.GetProperty("sku")} {p.GetProperty("nameEn")} {p.GetProperty("sellingPrice")}");
```



200 OK (abridged)

```
{
  "items": [
    {
      "id": "51f771bf-cfb5-4814-8e79-55b990be69c9",
      "sku": "SKU-2026-000001",
      "nameAr": "مياه معدنية 330 مل",
      "nameEn": "Mineral water 330 ml",
      "type": "PHYSICAL",
      "categoryId": "c33a89cb-5d9d-4112-b1cc-4dfec304c7a",
      "baseUnitId": "0757ef16-15fa-4b58-996d-ebb3d8bb5fcb",
      "sellingPrice": "1.15",
      "purchasePrice": "0.6",
      "cost": "0.6",
      "vatCategory": "S",
      "vatRate": null,
      "minStock": "48",
      "isActive": true,
      "trackInventory": true,
      "category": { "id": "c33a89cb-5d9d-4112-b1cc-4dfec304c7a", "nameAr": "مشروبات", "nameEn": "Beverages" },
      "baseUnit": { "id": "0757ef16-15fa-4b58-996d-ebb3d8bb5fcb", "code": "PCS", "nameAr": "قطعة", "nameEn": "Piece" },
      "units": [
        { "unitId": "0757ef16-15fa-4b58-996d-ebb3d8bb5fcb", "factor": "1", "isDefaultSale": true, "sellingPrice": null,
        "unit": { "code": "PCS" } },
        { "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4", "factor": "24", "isDefaultPurchase": true, "sellingPrice":
        "25", "unit": { "code": "BOX" } }
      ],
      "barcodes": [
        { "code": "6281000616970", "symbology": "EAN13", "unitId": null, "isPrimary": true },
        { "code": "6281010616977", "symbology": "EAN13", "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4", "isPrimary":
        false }
      ],
      "stock": { "onHand": "282", "reserved": "0", "available": "282" },
      "createdAt": "2026-10-06T17:01:01.711Z",
      "updatedAt": "2026-10-06T17:01:01.711Z"
    }
  ],
  "total": 1,
  "page": 1,
  "pageSize": 20
}
```

إنشاء منتج بوحدات وباركودات

POST /products — الصلاحية products:create (ويحتاج purchasePrice أيضاً inventory.cost:view). الحقل الإلزامي الوحيد nameAr: إذ يأخذ nameEn قيمته افتراضياً، وتكون الوحدة الأساسية PCS، ويُرقم رمز الصنف تلقائياً (SKU-2026-000001) إذا لم ترسله. يضيف units وحدات بديلة بمعاملها factor، ويمكن ربط الباركود بوحدة (باركود الكرتون يبيع الكرتون كاملاً)، ويُكتشف نوع الباركود (EAN-13 و EAN-8 و Code 128 و QR) من قيمته.

curl

```
# the ids of PCS, BOX ... (every company starts with a standard set of units)
curl "https://erp.sa4s.com/api/v1/units?pageSize=50" -H "Authorization: Bearer $SA4S_API_KEY"
```

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/products" \
-H "Authorization: Bearer $SA4S_API_KEY" \
-H "Content-Type: application/json" \
-d '{
```



```

"nameAr": "مياه معدنية 330 مل",
"nameEn": "Mineral water 330 ml",
"categoryId": "c33a89cb-5d9d-4112-b1cc-4dfebc304c7a",
"baseUnitId": "0757ef16-15fa-4b58-996d-ebb3d8bb5fcb",
"sellingPrice": "1.15",
"purchasePrice": "0.60",
"vatCategory": "S",
"minStock": "48",
"units": [
  { "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4", "factor": "24", "sellingPrice": "25.00", "isDefaultPurchase":
true }
],
"barcodes": [
  { "code": "6281000616970", "isPrimary": true },
  { "code": "6281010616977", "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4" }
]
}'

```

JavaScript

```

const units = await sa4s('GET', '/units?pageSize=50');
const pcs = units.items.find((u) => u.code === 'PCS');
const box = units.items.find((u) => u.code === 'BOX');

const product = await sa4s('POST', '/products', {
  nameAr: 'مياه معدنية 330 مل',
  nameEn: 'Mineral water 330 ml',
  baseUnitId: pcs.id,
  sellingPrice: '1.15',           // per base unit (piece)
  purchasePrice: '0.60',         // needs the inventory.cost:view scope
  vatCategory: 'S',
  minStock: '48',
  units: [{ unitId: box.id, factor: '24', sellingPrice: '25.00', isDefaultPurchase: true }], // 1 box = 24 pieces
  barcodes: [{ code: '6281000616970', isPrimary: true }, { code: '6281010616977', unitId: box.id }],
});
console.log(product.id, product.sku); // SKU is numbered automatically when omitted

```

Python

```

units = {u["code"]: u["id"] for u in sa4s("GET", "/units", params={"pageSize": 50})["items"]}

product = sa4s("POST", "/products", {
    "nameAr": "مياه معدنية 330 مل",
    "nameEn": "Mineral water 330 ml",
    "baseUnitId": units["PCS"],
    "sellingPrice": "1.15",
    "purchasePrice": "0.60",
    "vatCategory": "S",
    "minStock": "48",
    "units": [{"unitId": units["BOX"], "factor": "24", "sellingPrice": "25.00", "isDefaultPurchase": True}],
    "barcodes": [{"code": "6281000616970", "isPrimary": True}, {"code": "6281010616977", "unitId": units["BOX"]}],
})
print(product["id"], product["sku"])

```

PHP

```

<?php
$units = array_column(sa4s('GET', '/units?pageSize=50')['items'], 'id', 'code');

$product = sa4s('POST', '/products', [
    'nameAr' => 'مياه معدنية 330 مل',
    'nameEn' => 'Mineral water 330 ml',
    'baseUnitId' => $units['PCS'],
    'sellingPrice' => '1.15',

```



```
'purchasePrice' => '0.60',
'vatCategory' => 'S',
'minStock' => '48',
'units' => [[ 'unitId' => $units['BOX'], 'factor' => '24', 'sellingPrice' => '25.00', 'isDefaultPurchase' => true]],
'barcodes' => [[ 'code' => '6281000616970', 'isPrimary' => true], [ 'code' => '6281010616977', 'unitId' =>
$units['BOX'] ]],
]);
echo $product['id'], ' ', $product['sku'], "\n";
```

C#

```
var product = await Sa4s.Send(HttpMethod.Post, "products", new
{
    nameAr = "مياه معدنية 330 مل",
    nameEn = "Mineral water 330 ml",
    baseUnitId = pcsUnitId,
    sellingPrice = "1.15",
    purchasePrice = "0.60",
    vatCategory = "S",
    minStock = "48",
    units = new[] { new { unitId = boxUnitId, factor = "24", sellingPrice = "25.00", isDefaultPurchase = true } },
    barcodes = new object[] { new { code = "6281000616970", isPrimary = true }, new { code = "6281010616977", unitId =
boxUnitId } },
});
Console.WriteLine(product.GetProperty("sku"));
```

201 Created (abridged)

```
{
  "id": "51f771bf-cfb5-4814-8e79-55b990be69c9",
  "sku": "SKU-2026-000001",
  "nameAr": "مياه معدنية 330 مل",
  "nameEn": "Mineral water 330 ml",
  "type": "PHYSICAL",
  "baseUnitId": "0757ef16-15fa-4b58-996d-ebb3d8bb5fcb",
  "purchasePrice": "0.6",
  "sellingPrice": "1.15",
  "cost": "0",
  "vatCategory": "S",
  "minStock": "48",
  "isActive": true,
  "trackInventory": true,
  "units": [
    { "unitId": "0757ef16-15fa-4b58-996d-ebb3d8bb5fcb", "factor": "1", "isDefaultSale": true, "unit": { "code": "PCS",
"nameEn": "Piece" } },
    { "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4", "factor": "24", "isDefaultPurchase": true, "sellingPrice":
"25", "unit": { "code": "BOX", "nameEn": "Box" } }
  ],
  "barcodes": [
    { "id": "5e5043c6-49e1-4eac-b89f-8f147cb20f6f", "code": "6281000616970", "symbology": "EAN13", "unitId": null,
"isPrimary": true },
    { "id": "7f6457f4-1571-4fe4-aca7-3b8927691c48", "code": "6281010616977", "symbology": "EAN13", "unitId": "b82dbc97-
8f7b-43c0-8b73-8b4aed1ebdb4", "isPrimary": false }
  ],
  "stock": { "onHand": "0", "reserved": "0", "available": "0" },
  "stockByWarehouse": []
}
```

إعادة استخدام باركود أو رمز صنف تفشل بتعارض يذكر صاحبه الحالي:

**409 Conflict**

```
{
  "statusCode": 409,
  "code": "BARCODE_TAKEN",
  "message": "This barcode is already assigned to another product",
  "details": { "code": "628100616970", "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9", "sku": "SKU-2026-000001" },
  "requestId": "5ecd4f0d-256e-4818-afef-23dd38d3bc1d"
}
```

تعديل منتج

— `products:edit` الصلاحية. أرسل ما يتغير فقط. وإن أرسلت `units` أو `barcodes` فالقائمة تستبدل الحالية؛ ولإضافة باركود واحد استخدم `POST /products/{id}/barcodes`.

curl

```
# change the price only – fields you leave out are not touched
curl -X PATCH "https://erp.sa4s.com/api/v1/products/51f771bf-cfb5-4814-8e79-55b990be69c9" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{ "sellingPrice": "1.25" }'

# add one more barcode without resending the List
curl -X POST "https://erp.sa4s.com/api/v1/products/51f771bf-cfb5-4814-8e79-55b990be69c9/barcodes" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{ "code": "6281020618633" }'
```

JavaScript

```
await sa4s('PATCH', `/products/${productId}`, { sellingPrice: '1.25' });
await sa4s('POST', `/products/${productId}/barcodes`, { code: '6281020618633' });
```

Python

```
sa4s("PATCH", f"/products/{product_id}", {"sellingPrice": "1.25"})
sa4s("POST", f"/products/{product_id}/barcodes", {"code": "6281020618633"})
```

PHP

```
<?php
sa4s('PATCH', "/products/$productId", ['sellingPrice' => '1.25']);
sa4s('POST', "/products/$productId/barcodes", ['code' => '6281020618633']);
```

البحث عن منتج بالباركود

— `products:view` الصلاحية. يطابق الباركود أولاً ثم رمز الصنف، ويخبرك بالوحدة التي يبيعها الرمز (باركود الكرتون يعيد وحدة الكرتون ومعاملها) وبالمخزون، ويمكن حصره في `warehouseId`.

curl

```
curl "https://erp.sa4s.com/api/v1/products/lookup?code=6281010616977&warehouseId=1ae3935b-6fae-4c7f-803a-bf405dbc0ea0" \
-H "Authorization: Bearer $SA4S_API_KEY"
```

**JavaScript**

```
try {
  const hit = await sa4s('GET', `/products/lookup?code=${encodeURIComponent(scanned)}`);
  console.log(hit.matchedBy, hit.product.nameEn, hit.unit?.code, hit.unit?.factor, hit.stock.available);
} catch (e) {
  if (e.code === 'BARCODE_NOT_FOUND') console.log('unknown code'); else throw e;
}
```

Python

```
from sa4s import sa4s, Sa4sError

try:
    hit = sa4s("GET", "/products/lookup", params={"code": scanned})
    print(hit["matchedBy"], hit["product"]["nameEn"], hit["unit"]["code"], hit["stock"]["available"])
except Sa4sError as e:
    if e.code != "BARCODE_NOT_FOUND":
        raise
```

PHP

```
<?php
try {
    $hit = sa4s('GET', '/products/lookup?code=' . urlencode($scanned));
    echo $hit['matchedBy'], ' ', $hit['product']['nameEn'], ' ', $hit['unit']['code'], "\n";
} catch (Sa4sError $e) {
    if (($e->body['code'] ?? '') !== 'BARCODE_NOT_FOUND') throw $e;
}
```

200 OK (abridged)

```
{
  "matchedBy": "barcode",
  "product": { "id": "51f771bf-cfb5-4814-8e79-55b990be69c9", "sku": "SKU-2026-000001", "nameEn": "Mineral water 330 ml", "sellingPrice": "1.15", "vatCategory": "S" },
  "barcode": { "id": "7f6457f4-1571-4fe4-aca7-3b8927691c48", "code": "6281010616977", "symbology": "EAN13", "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4", "isPrimary": false },
  "unit": { "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4", "code": "BOX", "nameAr": "علبة", "nameEn": "Box", "factor": "24", "sellingPrice": "25", "purchasePrice": "14.4" },
  "stock": { "onHand": "282", "reserved": "0", "available": "282" }
}
```

404 Not Found

```
{
  "statusCode": 404,
  "code": "BARCODE_NOT_FOUND",
  "message": "No product matches this code",
  "details": { "code": "000000000000" },
  "requestId": "d7c8deca-f7ac-4f0f-9d28-effe98b2a121"
}
```



وصفات — العملاء

إنشاء عميل

POST /customers — الصلاحية customers:create . الحقل الإلزامي الوحيد name . العميل الذي له vatNumber (15 رقماً يبدأ وينتهي بالرقم 3) تصدر له فواتير ضريبية (بين المنشآت) افتراضياً، وتظهر حقول العنوان الوطني في فواتيره الضريبية. ويُرقَّم code تلقائياً (-2026-CUS-000001).

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/customers" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{
  "type": "COMPANY",
  "name": "شركة الأفق للتجارة",
  "nameEn": "Horizon Trading Co",
  "vatNumber": "310123456700003",
  "crNumber": "1010123456",
  "phone": "+966500000001",
  "email": "ap@horizon.example",
  "city": "Riyadh", "district": "Al Olaya", "street": "King Fahd Rd",
  "buildingNumber": "1234", "postalCode": "12211", "additionalNumber": "5678",
  "creditLimit": "50000.00",
  "paymentTermsDays": 30
}'
```

JavaScript

```
const customer = await sa4s('POST', '/customers', {
  type: 'COMPANY', // INDIVIDUAL (default) | COMPANY | GOVERNMENT
  name: 'شركة الأفق للتجارة', // primary (Arabic) name – required
  nameEn: 'Horizon Trading Co',
  vatNumber: '310123456700003', // 15 digits, starts and ends with 3 → invoices default to TAX (B2B)
  crNumber: '1010123456',
  phone: '+966500000001',
  email: 'ap@horizon.example',
  city: 'Riyadh', district: 'Al Olaya', street: 'King Fahd Rd', buildingNumber: '1234', postalCode: '12211',
  additionalNumber: '5678',
  creditLimit: '50000.00',
  paymentTermsDays: 30,
});
console.log(customer.code); // CUS-2026-000001
```

Python

```
customer = sa4s("POST", "/customers", {
  "type": "COMPANY",
  "name": "شركة الأفق للتجارة",
  "nameEn": "Horizon Trading Co",
  "vatNumber": "310123456700003",
  "phone": "+966500000001",
  "email": "ap@horizon.example",
  "city": "Riyadh", "district": "Al Olaya", "street": "King Fahd Rd",
  "buildingNumber": "1234", "postalCode": "12211", "additionalNumber": "5678",
  "creditLimit": "50000.00",
  "paymentTermsDays": 30,
})
print(customer["id"], customer["code"])
```



PHP

```
<?php
$customer = sa4s('POST', '/customers', [
    'type' => 'COMPANY',
    'name' => 'شركة الأفق للتجارة',
    'nameEn' => 'Horizon Trading Co',
    'vatNumber' => '310123456700003',
    'phone' => '+966500000001',
    'email' => 'ap@horizon.example',
    'city' => 'Riyadh', 'district' => 'Al Olaya', 'street' => 'King Fahd Rd',
    'buildingNumber' => '1234', 'postalCode' => '12211', 'additionalNumber' => '5678',
    'creditLimit' => '50000.00',
    'paymentTermsDays' => 30,
]);
echo $customer['code'], "\n";
```

C#

```
var customer = await Sa4s.Send(HttpMethod.Post, "customers", new
{
    type = "COMPANY",
    name = "شركة الأفق للتجارة",
    nameEn = "Horizon Trading Co",
    vatNumber = "310123456700003",
    phone = "+966500000001",
    city = "Riyadh", district = "Al Olaya", street = "King Fahd Rd", buildingNumber = "1234", postalCode = "12211",
    creditLimit = "50000.00",
    paymentTermsDays = 30,
});
Console.WriteLine(customer.GetProperty("code"));
```

201 Created

```
{
  "id": "fab57752-483f-4b1a-8d80-8c0401809ffe",
  "code": "CUS-2026-000001",
  "type": "COMPANY",
  "name": "شركة الأفق للتجارة",
  "nameEn": "Horizon Trading Co",
  "vatNumber": "310123456700003",
  "crNumber": "1010123456",
  "phone": "+966500000001",
  "email": "ap@horizon.example",
  "country": "SA",
  "city": "Riyadh",
  "district": "Al Olaya",
  "street": "King Fahd Rd",
  "buildingNumber": "1234",
  "postalCode": "12211",
  "additionalNumber": "5678",
  "creditLimit": "50000",
  "paymentTermsDays": 30,
  "openingBalance": "0",
  "balance": "0",
  "isWalkIn": false,
  "isActive": true,
  "notes": null,
  "createdAt": "2026-10-06T17:01:01.916Z",
  "updatedAt": "2026-10-06T17:01:01.916Z",
```



```

"deletedAt": null,
"availableCredit": "50000",
"overCreditLimit": false
}

```

409 Conflict

```

{
  "statusCode": 409,
  "code": "VAT_NUMBER_TAKEN",
  "message": "Another record already uses this VAT number",
  "details": { "vatNumber": "310123456700003" },
  "requestId": "175ebe07-cc59-447f-a30e-2106c84d503f"
}

```

البحث عن عميل أو تعديله

المتغيرة فقط). ونمط شائع في المتاجر: ابحث بالحوال أو الرقم الضريبي، وأنشئ العميل إن لم يوجد:

curl

```

# search matches name, English name, phone, VAT number, code, e-mail and CR number
curl "https://erp.sa4s.com/api/v1/customers?search=%2B966551234567&pageSize=5" \
-H "Authorization: Bearer $SA4S_API_KEY"

```

JavaScript

```

// find a store customer by phone (or VAT number), create them when missing
async function upsertCustomer({ name, phone, email, vatNumber }) {
  const key = vatNumber ?? phone;
  const found = await sa4s('GET', `/customers?search=${encodeURIComponent(key)}&pageSize=5`);
  const match = found.items.find((c) => (vatNumber ? c.vatNumber === vatNumber : c.phone === phone));
  if (match) return sa4s('PATCH', `/customers/${match.id}`, { email }); // keep contact data fresh
  return sa4s('POST', '/customers', { name, phone, email, vatNumber, type: vatNumber ? 'COMPANY' : 'INDIVIDUAL' });
}

```

Python

```

def upsert_customer(name, phone, email=None, vat_number=None):
    key = vat_number or phone
    found = sa4s("GET", "/customers", params={"search": key, "pageSize": 5})["items"]
    match = next((c for c in found if (c["vatNumber"] == vat_number if vat_number else c["phone"] == phone)), None)
    if match:
        return sa4s("PATCH", f"/customers/{match['id']}", {"email": email})
    body = {"name": name, "phone": phone, "email": email, "type": "COMPANY" if vat_number else "INDIVIDUAL"}
    if vat_number:
        body["vatNumber"] = vat_number
    return sa4s("POST", "/customers", body)

```

PHP

```

<?php
function upsert_customer(string $name, string $phone, ?string $email = null, ?string $vat = null): array
{
    $found = sa4s('GET', '/customers?' . http_build_query(['search' => $vat ?? $phone, 'pageSize' => 5]))['items'];
    foreach ($found as $c) {
        if ($vat ? $c['vatNumber'] === $vat : $c['phone'] === $phone) {
            return sa4s('PATCH', "/customers/${c['id']}", ['email' => $email]);
        }
    }
}

```



```
return sa4s('POST', '/customers', array_filter(['name' => $name, 'phone' => $phone, 'email' => $email, 'vatNumber'
=> $vat,
    'type' => $vat ? 'COMPANY' : 'INDIVIDUAL']));
}
```

curl

```
curl -X PATCH "https://erp.sa4s.com/api/v1/customers/b4382d00-46be-436d-af45-c3dcd8b0222c" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{ "phone": "+966551234568", "city": "Jeddah" }'
```

JavaScript

```
await sa4s('PATCH', `/customers/${customerId}`, { phone: '+966551234568', city: 'Jeddah' });
```

Python

```
sa4s("PATCH", f"/customers/{customer_id}", {"phone": "+966551234568", "city": "Jeddah"})
```

PHP

```
<?php
sa4s('PATCH', "/customers/$customerId", ['phone' => '+966551234568', 'city' => 'Jeddah']);
```

كشف حساب العميل

التراكمي، وأعمار البنود المفتوحة. — `GET /customers/{id}/statement?from=...&to=...` — الصلاحية `customers:view` : الرصيد الافتتاحي، وكل فاتورة ودفعة مع الرصيد

curl

```
curl "https://erp.sa4s.com/api/v1/customers/fab57752-483f-4b1a-8d80-8c0401809ffe/statement?from=2026-10-01&to=2026-10-31" \
-H "Authorization: Bearer $SA4S_API_KEY"
```

JavaScript

```
const st = await sa4s('GET', `/customers/${customerId}/statement?from=2026-10-01&to=2026-10-31`);
console.log(st.openingBalance, st.closingBalance, st.outstanding.aging);
```

Python

```
st = sa4s("GET", f"/customers/{customer_id}/statement", params={"from": "2026-10-01", "to": "2026-10-31"})
```

PHP

```
<?php
$st = sa4s('GET', "/customers/$customerId/statement?from=2026-10-01&to=2026-10-31");
```

200 OK (abridged)

```
{
  "kind": "CUSTOMER",
  "party": { "id": "fab57752-483f-4b1a-8d80-8c0401809ffe", "code": "CUS-2026-000001", "name": "شركة الأفق للتجارة",
  "vatNumber": "310123456700003" },
  "currency": "SAR",
  "from": "2026-10-01",
  "to": "2026-10-31",
```



```

"openingBalance": "0.00",
"lines": [
  { "date": "2026-10-06T17:01:02.289Z", "type": "INVOICE", "refType": "SALES_INVOICE", "number": "INV-2026-000002",
    "debit": "270.25", "credit": "0.00", "balance": "270.25" },
  { "date": "2026-10-06T17:01:02.330Z", "type": "PAYMENT", "refType": "PAYMENT", "number": "RCV-2026-000002",
    "description": "BANK_TRANSFER · TRX-55120", "debit": "0.00", "credit": "100.00", "balance": "170.25" },
  { "date": "2026-10-06T17:01:02.393Z", "type": "INVOICE", "refType": "SALES_INVOICE", "number": "INV-2026-000003",
    "debit": "230.00", "credit": "0.00", "balance": "400.25" }
],
"totals": { "debit": "500.25", "credit": "300.25" },
"closingBalance": "200.00",
"outstanding": {
  "total": "200.00",
  "aging": { "current": "200.00", "d1_30": "0.00", "d31_60": "0.00", "d61_90": "0.00", "d90plus": "0.00" },
  "openItems": [ { "type": "INVOICE", "number": "INV-2026-000003", "dueDate": "2026-11-05T00:00:00.000Z",
    "daysOverdue": 0, "outstanding": "200.00" } ]
}
}

```



وصفات — المبيعات

يتطلب `POST /sales/invoices` الصلاحية `sales.invoice:create`. ويحدد `status` ما يحدث: `DRAFT` (الافتراضي) يحفظ فقط، و `ISSUED` يرقيم الفاتورة ويحرك المخزون ويحل القيد ويسجل `payments` ويضع الفاتورة الإلكترونية في الطابور — كل ذلك في معاملة واحدة. ويكون `type` افتراضياً `TAX` إذا كان للعميل رقم ضريبي و `SIMPLIFIED` في غير ذلك. وسعر السطر الافتراضي هو سعر بيع المنتج (أو الوحدة).

فاتورة مبسطة (للأفراد) شاملة الضريبة ومدفوعة بالبطاقة

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/sales/invoices" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{
  "type": "SIMPLIFIED",
  "status": "ISSUED",
  "pricesIncludeVat": true,
  "clientId": "09c3d295-3127-400b-ab9c-819ab51b011c",
  "notes": "Store order #10045",
  "lines": [
    { "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9", "quantity": "6" },
    { "productId": "6fe0f6fa-ed4d-4c4b-81ca-c8095504658f", "quantity": "1", "unitPrice": "40.00" }
  ],
  "payments": [ { "method": "CARD", "amount": "46.90", "reference": "PSP-7781" } ]
}'
```

JavaScript

```
import { randomUUID } from 'node:crypto';

const invoice = await sa4s('POST', '/sales/invoices', {
  type: 'SIMPLIFIED', // B2C — no customer needed
  status: 'ISSUED', // number, stock, ledger and the ZATCA queue happen now
  pricesIncludeVat: true, // unit prices below already contain 15 % VAT
  clientId: randomUUID(), // store it with your order BEFORE sending, reuse it on every retry
  notes: 'Store order #10045',
  lines: [
    { productId: WATER_ID, quantity: '6' }, // price taken from the product (1.15)
    { productId: DATES_ID, quantity: '1', unitPrice: '40.00' }, // or sent explicitly
  ],
  payments: [{ method: 'CARD', amount: '46.90', reference: 'PSP-7781' }],
});
console.log(invoice.number, invoice.total, invoice.paymentStatus); // INV-2026-000007 46.9 PAID
```

Python

```
import uuid

invoice = sa4s("POST", "/sales/invoices", {
  "type": "SIMPLIFIED",
  "status": "ISSUED",
  "pricesIncludeVat": True,
  "clientId": str(uuid.uuid4()),
  "notes": "Store order #10045",
  "lines": [
    {"productId": WATER_ID, "quantity": "6"},
    {"productId": DATES_ID, "quantity": "1", "unitPrice": "40.00"},
  ],
})
```



```

    ],
    "payments": [{ "method": "CARD", "amount": "46.90", "reference": "PSP-7781" }],
  })
  print(invoice["number"], invoice["total"], invoice["paymentStatus"])

```

PHP

```

<?php
$invoice = sa4s('POST', '/sales/invoices', [
    'type' => 'SIMPLIFIED',
    'status' => 'ISSUED',
    'pricesIncludeVat' => true,
    'clientId' => $order['sa4s_client_uuid'], // generated once and saved with the order
    'notes' => 'Store order #10045',
    'lines' => [
        [ 'productId' => $waterId, 'quantity' => '6' ],
        [ 'productId' => $datesId, 'quantity' => '1', 'unitPrice' => '40.00' ],
    ],
    'payments' => [[ 'method' => 'CARD', 'amount' => '46.90', 'reference' => 'PSP-7781' ]],
]);
echo $invoice['number'], ' ', $invoice['total'], "\n";

```

C#

```

var invoice = await Sa4s.Send(HttpMethod.Post, "sales/invoices", new
{
    type = "SIMPLIFIED",
    status = "ISSUED",
    pricesIncludeVat = true,
    clientId = order.Sa4sClientId, // Guid saved with the order before the first attempt
    notes = "Store order #10045",
    lines = new object[]
    {
        new { productId = waterId, quantity = "6" },
        new { productId = datesId, quantity = "1", unitPrice = "40.00" },
    },
    payments = new[] { new { method = "CARD", amount = "46.90", reference = "PSP-7781" } },
});
Console.WriteLine($"{invoice.GetProperty("number")} {invoice.GetProperty("total")}");

```

201 Created (abridged)

```

{
  "id": "442ea37d-b9ac-4535-b726-b3ad69832afe",
  "number": "INV-2026-000007",
  "type": "SIMPLIFIED",
  "status": "ISSUED",
  "paymentStatus": "PAID",
  "customerId": null,
  "branchId": "73afa37e-46d6-4c51-9a13-549d92cad970",
  "warehouseId": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0",
  "clientId": "09c3d295-3127-400b-ab9c-819ab51b011c",
  "uuid": "a14f90a9-de63-4d97-a093-6834846ba8d5",
  "issueDate": "2026-10-06T17:17:09.161Z",
  "pricesIncludeVat": true,
  "currency": "SAR",
  "subtotal": "40.78",
  "discountTotal": "0",
  "taxableTotal": "40.78",
  "vatTotal": "6.12",
  "total": "46.9",
  "paidAmount": "46.9",
  "zatcaStatus": "NOT_REQUIRED",
  "items": [

```



```
{
  "lineNo": 1, "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9", "sku": "SKU-2026-000001", "unitName": "قطعة",
  "quantity": "6", "baseQuantity": "6", "unitPrice": "1.15", "vatCategory": "S", "vatRate": "15", "taxableAmount": "6",
  "vatAmount": "0.9", "lineTotal": "6.9" },
  { "lineNo": 2, "productId": "6fe0f6fa-ed4d-4c4b-81ca-c8095504658f", "sku": "SKU-2026-000003", "unitName": "قطعة",
  "quantity": "1", "baseQuantity": "1", "unitPrice": "40", "vatCategory": "S", "vatRate": "15", "taxableAmount": "34.78",
  "vatAmount": "5.22", "lineTotal": "40" }
],
"payments": [
  { "id": "36f6d10d-95f4-40b7-8fd5-28cae04ccaa6", "number": "RCV-2026-000007", "direction": "IN", "method": "CARD",
  "amount": "46.9", "reference": "PSP-7781", "accountId": "0835961d-0dca-4896-99b4-11fa96c82eda" }
],
"zatca": null,
"amountDue": "0"
}
```

فاتورة ضريبية (بين المنشآت) غير شاملة الضريبة ومدفوعة جزئياً

يمكن تقسيم الدفع على عدة طرق (CASH و CARD و BANK_TRANSFER و CHEQUE و WALLET)، وما لم يُدفع يبقى على حساب العميل (paymentStatus بقيمة PARTIALLY_PAID أو UNPAID و amountDue). وتشير CREDIT صراحةً إلى بيع آجل.

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/sales/invoices" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{
  "type": "TAX",
  "status": "ISSUED",
  "customerId": "fab57752-483f-4b1a-8d80-8c0401809ffe",
  "pricesIncludeVat": false,
  "dueDate": "2026-11-05",
  "clientUuid": "fe66bdfc-781f-442d-96d0-3fa50ca52a0a",
  "lines": [
    { "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9", "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4",
    "quantity": "10", "unitPrice": "22.00" },
    { "productId": "7c6db833-d278-4537-87ed-86ef13f3801f", "quantity": "1" }
  ],
  "payments": [ { "method": "BANK_TRANSFER", "amount": "100.00", "reference": "TRX-55120" } ]
}'
```

JavaScript

```
const invoice = await sa4s('POST', '/sales/invoices', {
  type: 'TAX', // B2B: needs a customer (with a VAT number for ZATCA clearance)
  status: 'ISSUED',
  customerId: HORIZON_ID,
  pricesIncludeVat: false, // prices are net; the server adds 15 % VAT
  dueDate: '2026-11-05',
  clientUuid: order.clientUuid,
  lines: [
    { productId: WATER_ID, unitId: BOX_UNIT_ID, quantity: '10', unitPrice: '22.00' }, // 10 boxes = 240 pieces out of stock
    { productId: DELIVERY_SERVICE_ID, quantity: '1' }, // a SERVICE product (no stock)
  ],
  payments: [{ method: 'BANK_TRANSFER', amount: '100.00', reference: 'TRX-55120' }], // the rest stays on the customer account
});
console.log(invoice.total, invoice.paidAmount, invoice.amountDue); // 270.25 100 170.25
```

**Python**

```
invoice = sa4s("POST", "/sales/invoices", {
    "type": "TAX",
    "status": "ISSUED",
    "customerId": HORIZON_ID,
    "pricesIncludeVat": False,
    "dueDate": "2026-11-05",
    "clientUuid": order["client_uuid"],
    "lines": [
        {"productId": WATER_ID, "unitId": BOX_UNIT_ID, "quantity": "10", "unitPrice": "22.00"},
        {"productId": DELIVERY_SERVICE_ID, "quantity": "1"},
    ],
    "payments": [{"method": "BANK_TRANSFER", "amount": "100.00", "reference": "TRX-55120"}],
})
print(invoice["total"], invoice["amountDue"])
```

PHP

```
<?php
$invoice = sa4s('POST', '/sales/invoices', [
    'type' => 'TAX',
    'status' => 'ISSUED',
    'customerId' => $horizonId,
    'pricesIncludeVat' => false,
    'dueDate' => '2026-11-05',
    'clientUuid' => $order['sa4s_client_uuid'],
    'lines' => [
        ['productId' => $waterId, 'unitId' => $boxUnitId, 'quantity' => '10', 'unitPrice' => '22.00'],
        ['productId' => $deliveryServiceId, 'quantity' => '1'],
    ],
    'payments' => [['method' => 'BANK_TRANSFER', 'amount' => '100.00', 'reference' => 'TRX-55120']],
]);
echo $invoice['total'], ' / due ', $invoice['amountDue'], "\n";
```

C#

```
var invoice = await Sa4s.Send(HttpMethod.Post, "sales/invoices", new
{
    type = "TAX",
    status = "ISSUED",
    customerId = horizonId,
    pricesIncludeVat = false,
    dueDate = "2026-11-05",
    clientUuid = order.Sa4sClientUuid,
    lines = new object[]
    {
        new { productId = waterId, unitId = boxUnitId, quantity = "10", unitPrice = "22.00" },
        new { productId = deliveryServiceId, quantity = "1" },
    },
    payments = new[] { new { method = "BANK_TRANSFER", amount = "100.00", reference = "TRX-55120" } },
});
Console.WriteLine($"{invoice.GetProperty("total")} due {invoice.GetProperty("amountDue")}");
```

201 Created (abridged)

```
{
  "id": "1086f27d-620a-4e60-9dfe-0720fee48398",
  "number": "INV-2026-000002",
  "type": "TAX",
  "status": "ISSUED",
  "paymentStatus": "PARTIALLY_PAID",
  "customerId": "fab57752-483f-4b1a-8d80-8c0401809ffe",
```



```

"dueDate": "2026-11-05T00:00:00.000Z",
"pricesIncludeVat": false,
"subtotal": "235",
"taxableTotal": "235",
"vatTotal": "35.25",
"total": "270.25",
"paidAmount": "100",
"paymentMethod": "BANK_TRANSFER",
"zatcaStatus": "NOT_REQUIRED",
"items": [
  { "lineNo": 1, "sku": "SKU-2026-000001", "unitName": "علبة", "quantity": "10", "baseQuantity": "240", "unitPrice":
"22", "vatRate": "15", "taxableAmount": "220", "vatAmount": "33", "lineTotal": "253" },
  { "lineNo": 2, "sku": "SKU-2026-000002", "unitName": "قطعة", "quantity": "1", "baseQuantity": "1", "unitPrice":
"15", "vatRate": "15", "taxableAmount": "15", "vatAmount": "2.25", "lineTotal": "17.25" }
],
"customer": { "id": "fab57752-483f-4b1a-8d80-8c0401809ffe", "name": "شركة الأفق للتجارة", "vatNumber": "310123456700003",
"balance": "170.25", "creditLimit": "50000" },
"payments": [ { "number": "RCV-2026-000002", "method": "BANK_TRANSFER", "amount": "100", "reference": "TRX-55120" }
],
"amountDue": "170.25"
}

```

مسودة أولاً، ثم الإصدار عند تطابق الإجمالي

إذا كان لنظامك إجمالي خاص (مثل طلب المتجر)، فأنشئ مسودة، وقارن `total` الذي حسبه الخادم بإجماليك، ثم أصدرها مع الدفعة. الإصدار مرتين لا يضر: فالطلب الثاني يعود بـ `409 INVOICE_NOT_ISSUABLE`.

curl

```

# 1) draft – nothing is numbered, no stock moves, nothing reaches ZATCA
curl -X POST "https://erp.sa4s.com/api/v1/sales/invoices" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{ "pricesIncludeVat": true, "clientUuid": "6f0c9a3e-2d61-4f7e-9a0b-5b1c2d3e4f50",
      "lines": [ { "productId": "6fe0f6fa-ed4d-4c4b-81ca-c8095504658f", "quantity": "2", "unitPrice": "40.00" } ] }'

# 2) issue it with the payment = the total the server computed
curl -X POST "https://erp.sa4s.com/api/v1/sales/invoices/44d7970d-e564-42fc-a122-f70657e7dc17/issue" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{ "payments": [ { "method": "CASH", "amount": "80.00" } ] }'

```

JavaScript

```

const draft = await sa4s('POST', '/sales/invoices', { pricesIncludeVat: true, clientUuid, lines });
if (Math.round(Number(draft.total) * 100) !== Math.round(Number(order.total) * 100)) throw new Error(`total mismatch
${draft.total} ≠ ${order.total}`);
try {
  await sa4s('POST', `/sales/invoices/${draft.id}/issue`, { payments: [{ method: 'CARD', amount: draft.total }] });
} catch (e) {
  if (e.code !== 'INVOICE_NOT_ISSUABLE') throw e; // already issued by an earlier attempt → fine
}

```

Python

```

draft = sa4s("POST", "/sales/invoices", {"pricesIncludeVat": True, "clientUuid": client_uuid, "lines": lines})
try:
    sa4s("POST", f"/sales/invoices/{draft['id']}/issue", {"payments": [{"method": "CARD", "amount": draft["total"]}]}))
except Sa4sError as e:
    if e.code != "INVOICE_NOT_ISSUABLE":
        raise

```



PHP

```
<?php
$draft = sa4s('POST', '/sales/invoices', ['pricesIncludeVat' => true, 'clientId' => $clientId, 'lines' => $lines]);
try {
    sa4s('POST', "/sales/invoices/{$draft['id']}/issue", ['payments' => [['method' => 'CARD', 'amount' =>
    $draft['total']]]]);
} catch (Sa4sError $e) {
    if (($e->body['code'] ?? '') !== 'INVOICE_NOT_ISSUABLE') throw $e;
}
```

إعادة محاولة الفاتورة بأمان

أرسل الجسم نفسه مع `clientId` نفسه كلما احتجت — المثال الكامل في منع التكرار. تعيد الإعادة 201 بالفاتورة الأصلية مع `"idempotentReplay": true`.

تنزيل الفاتورة بصيغة PDF

— الصلاحية `sales.invoice:print` . `format=a4` (الافتراضي) أو `receipt` (إيصال حراري 80 مم)، و `lang=ar` أو `en` أو `both`، و `download=true` للتنزيل كمرفق بدل العرض. ويحمل الملف رمز الاستجابة السريعة (QR) الخاص بالهيئة متي توفّر للفاتورة.

curl

```
curl "https://erp.sa4s.com/api/v1/sales/invoices/442ea37d-b9ac-4535-b726-b3ad69832afe/pdf?
format=a4&lang=ar&download=true" \
-H "Authorization: Bearer $SA4S_API_KEY" \
-o INV-2026-000007.pdf
```

JavaScript

```
import { writeFile } from 'node:fs/promises';

const pdf = await sa4s('GET', `/sales/invoices/${invoiceId}/pdf?format=a4&lang=both`); // Buffer (application/pdf)
await writeFile(`${invoiceNumber}.pdf`, pdf);
```

Python

```
pdf = sa4s("GET", f"/sales/invoices/{invoice_id}/pdf", params={"format": "receipt", "lang": "ar"}) # bytes
with open(f"{invoice_number}.pdf", "wb") as f:
    f.write(pdf)
```

PHP

```
<?php
$pdf = sa4s('GET', "/sales/invoices/$invoiceId/pdf?format=a4&lang=en");
file_put_contents("$invoiceNumber.pdf", $pdf);
```

C#

```
var bytes = await Sa4s.Http.GetByteArrayAsync($"sales/invoices/{invoiceId}/pdf?format=a4&lang=ar");
await File.WriteAllBytesAsync($"{invoiceNumber}.pdf", bytes);
```



HTTP

```
HTTP/1.1 200 OK
Content-Type: application/pdf
Content-Length: 17138
Content-Disposition: attachment; filename="INV-2026-000007.pdf"
Cache-Control: private, no-store
```

حالة الفوترة الإلكترونية ورمز QR

يرسل SA4S الفواتير المُصدّرة إلى هيئة الزكاة والضريبة والجمارك تلقائياً (الإبلاغ للفواتير المبسّطة والاعتماد للفواتير الضريبية) بعد أن تكمل الشركة ربط الفوترة الإلكترونية من النظام. تستطيع المفاتيح قراءة النتيجة لا إدارة الربط:

zatcaStatus	المعنى
NOT_REQUIRED	مسودة، أو لم تُفَعَّل الفوترة الإلكترونية للشركة بعد
SUBMITTED / PENDING	في الطابور / أرسلت وتنتظر رد الهيئة
CLEARED / REPORTED	مقبولة (فاتورة مبسّطة / ضريبية)
REJECTED	مرفوضة — يشرح <code>zatca.errors</code> السبب، ويعرضه النظام للمحاسب

يعيد `GET /sales/invoices/{id}` الحقل `zatcaStatus` وكتلة `zatca` (`status` و `flow` و `icv` و `qrCode`) بترميز base64 TLV و `warnings` و `errors` و `submittedAt`) — وتكون `null` حتى تُعالج الفاتورة. صَفِّ القوائم بـ `zatcaStatus=...` أو اشترك في حدث `zatca.status`.

curl

```
# one invoice: zatcaStatus + the zatca block (QR, flow, warnings)
curl "https://erp.sa4s.com/api/v1/sales/invoices/442ea37d-b9ac-4535-b726-b3ad69832afe" -H "Authorization: Bearer $SA4S_API_KEY"

# everything ZATCA rejected this month
curl "https://erp.sa4s.com/api/v1/sales/invoices?zatcaStatus=REJECTED&from=2026-10-01&to=2026-10-31" -H "Authorization: Bearer $SA4S_API_KEY"
```

JavaScript

```
const inv = await sa4s('GET', `/sales/invoices/${invoiceId}`);
console.log(inv.zatcaStatus); // NOT_REQUIRED | PENDING | SUBMITTED | CLEARED | REPORTED | REJECTED
if (inv.zatca?.qrCode) renderQr(inv.zatca.qrCode); // base64 TLV – print it on your own receipt
if (inv.zatcaStatus === 'REJECTED') console.warn(inv.zatca.errors);
```

Python

```
inv = sa4s("GET", f"/sales/invoices/{invoice_id}")
zatca = inv.get("zatca") or {}
print(inv["zatcaStatus"], zatca.get("flow"), zatca.get("qrCode"))
```

PHP

```
<?php
$inv = sa4s('GET', "/sales/invoices/$invoiceId");
echo $inv['zatcaStatus'], ' ', $inv['zatca']['qrCode'] ?? '(no QR yet)', "\n";
```

**200 OK — shape of the zatca block after processing (illustrative values)**

```
{
  "id": "442ea37d-b9ac-4535-b726-b3ad69832afe",
  "number": "INV-2026-000007",
  "type": "SIMPLIFIED",
  "uuid": "a14f90a9-de63-4d97-a093-6834846ba8d5",
  "zatcaStatus": "REPORTED",
  "zatca": {
    "status": "REPORTED",
    "flow": "reporting",
    "icv": 7,
    "qrCode": "<base64 TLV of the ZATCA QR>",
    "warnings": [],
    "errors": [],
    "submittedAt": "2026-10-06T17:17:12.402Z"
  }
}
```

الاسترجاع أو الإرجاع: الإشعار الدائن

الصلاحية `sales.invoice:refund`. اختر أسطراً بعينها (`originalItemId`) و (`quantity`) أو كل الفاتورة (`full: true`)، ويمكنك إعادة البضاعة إلى المخزون (`restock`) وردّ المبلغ (`refund`). يأخذ الإشعار رقمه الخاص (`CN-...`) ويُرسَل إلى الهيئة مثل أي فاتورة.

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/sales/invoices/1086f27d-620a-4e60-9dfe-0720fee48398/credit-note" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{
  "reason": "Damaged boxes returned",
  "lines": [ { "originalItemId": "2cfc59f1-ae9f-4868-8920-0b264ac071b2", "quantity": "2" } ],
  "restock": true,
  "refund": { "method": "BANK_TRANSFER", "reference": "RF-1001" },
  "clientId": "d99f7cd1-d612-40bf-8a74-aeabad8c9e9c"
}'
```

JavaScript

```
const original = await sa4s('GET', `/sales/invoices/${invoiceId}`);
const line = original.items.find((i) => i.productId === WATER_ID);
const note = await sa4s('POST', `/sales/invoices/${invoiceId}/credit-note`, {
  reason: 'Damaged boxes returned',
  lines: [{ originalItemId: line.id, quantity: '2' }], // in the unit of the original line (boxes)
  restock: true, // goods go back to stock at the original cost
  refund: { method: 'BANK_TRANSFER', reference: 'RF-1001' }, // amount defaults to the note total (capped by what was paid)
  clientId: refund.clientId,
});
console.log(note.number, note.total); // CN-2026-000001 50.6
```

Python

```
original = sa4s("GET", f"/sales/invoices/{invoice_id}")
line = next(i for i in original["items"] if i["productId"] == WATER_ID)
note = sa4s("POST", f"/sales/invoices/{invoice_id}/credit-note", {
    "reason": "Damaged boxes returned",
    "lines": [{"originalItemId": line["id"], "quantity": "2"}],
})
```



```
"restock": true,
"refund": {"method": "BANK_TRANSFER", "reference": "RF-1001"},
"clientId": refund_uuid,
})
```

PHP

```
<?php
// full refund of everything still creditable on the invoice
$note = sa4s('POST', "/sales/invoices/$invoiceId/credit-note", [
    'reason' => 'Order returned',
    'full' => true,
    'restock' => true,
    'refund' => ['method' => 'CARD', 'reference' => $pspRefundId],
    'clientId' => $refundUuid,
]);
```

201 Created (abridged)

```
{
  "id": "d6111f48-8d38-4375-8786-805627995cb5",
  "number": "CN-2026-000001",
  "type": "CREDIT_NOTE",
  "status": "ISSUED",
  "paymentStatus": "PAID",
  "originalInvoiceId": "1086f27d-620a-4e60-9dfe-0720fee48398",
  "noteReason": "Damaged boxes returned",
  "taxableTotal": "44",
  "vatTotal": "6.6",
  "total": "50.6",
  "paidAmount": "50.6",
  "items": [ { "originalItemId": "2cfc59f1-ae9f-4868-8920-0b264ac071b2", "unitName": "علبة", "quantity": "2",
  "baseQuantity": "48", "unitPrice": "22", "lineTotal": "50.6" } ]
}
```

إلغاء فاتورة

مع سبب `reason` — الصلاحية `sales.invoice:cancel`. يقتصر على الفواتير التي ليس لها دفعات أو إشعارات ولم تصل إلى الهيئة؛ وفي غير ذلك أصدر إشعاراً دائماً (`INVOICE_HAS_PAYMENTS` و `ZATCA_REPORTED_USE_CREDIT_NOTE`).

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/sales/invoices/015727de-9146-41eb-839b-fd477dc9e9a8/cancel" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{ "reason": "Customer cancelled the order" }'
```

JavaScript

```
await sa4s('POST', `/sales/invoices/${invoiceId}/cancel`, { reason: 'Customer cancelled the order' });
```

Python

```
sa4s("POST", f"/sales/invoices/{invoice_id}/cancel", {"reason": "Customer cancelled the order"})
```

PHP

```
<?php
sa4s('POST', "/sales/invoices/$invoiceId/cancel", ['reason' => 'Customer cancelled the order']);
```



وصفات — المدفوعات

تسجيل سند قبض من عميل

POST /payments/receipts — الصلاحية payments:create. وُزّع المبلغ على فاتورة مفتوحة أو أكثر (allocations)، أو دُع SA4S يوزعه على أقدم الفواتير المفتوحة (autoAllocate: true)، أو أتركه دفعة مقدمة غير موزعة. ويحدد accountId الصندوق أو الحساب البنكي (الافتراضي: أول صندوق نشط لـ CASH، وأول حساب بنكي لغيره)؛ واعرض الحسابات بـ GET /financial-accounts (الصلاحية cash:view).

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/payments/receipts" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{
  "customerId": "fab57752-483f-4b1a-8d80-8c0401809ffe",
  "method": "BANK_TRANSFER",
  "amount": "170.25",
  "reference": "TRX-55190",
  "clientId": "c8b5212a-1b50-4864-9d45-7b562575f3fa",
  "allocations": [ { "invoiceId": "1086f27d-620a-4e60-9dfe-0720fee48398", "amount": "170.25" } ]
}'
```

JavaScript

```
const receipt = await sa4s('POST', '/payments/receipts', {
  customerId: HORIZON_ID,
  method: 'BANK_TRANSFER', // CASH | CARD | BANK_TRANSFER | CHEQUE | WALLET
  amount: '170.25',
  reference: 'TRX-55190', // your bank / PSP reference
  clientId: transfer.clientId, // idempotency key: one receipt per bank transfer
  allocations: [{ invoiceId: INVOICE_ID, amount: '170.25' }], // or autoAllocate: true (oldest open invoices first)
});
console.log(receipt.number, receipt.allocations.map((a) => `${a.invoiceId}=${a.amount}`));
```

Python

```
receipt = sa4s("POST", "/payments/receipts", {
    "customerId": HORIZON_ID,
    "method": "BANK_TRANSFER",
    "amount": "170.25",
    "reference": "TRX-55190",
    "clientId": transfer["client_uuid"],
    "allocations": [{"invoiceId": INVOICE_ID, "amount": "170.25"}],
})
print(receipt["number"])
```

PHP

```
<?php
$receipt = sa4s('POST', '/payments/receipts', [
    'customerId' => $horizonId,
    'method' => 'BANK_TRANSFER',
    'amount' => '170.25',
    'reference' => 'TRX-55190',
```



```
'clientId' => $transfer['client_uuid'],
'allocations' => [['invoiceId' => $invoiceId, 'amount' => '170.25']],
]);
echo $receipt['number'], "\n";
```

C#

```
var receipt = await Sa4s.Send(HttpMethod.Post, "payments/receipts", new
{
    customerId = horizonId,
    method = "BANK_TRANSFER",
    amount = "170.25",
    reference = "TRX-55190",
    clientId = transfer.ClientUuid,
    allocations = new[] { new { invoiceId, amount = "170.25" } },
});
Console.WriteLine(receipt.GetProperty("number"));
```

201 Created

```
{
  "id": "f7e14210-3cbc-43ba-8b39-ae5c2b207898",
  "number": "RCV-2026-000004",
  "direction": "IN",
  "partyType": "CUSTOMER",
  "partyId": "fab57752-483f-4b1a-8d80-8c0401809ffe",
  "method": "BANK_TRANSFER",
  "amount": "170.25",
  "date": "2026-10-06T17:01:02.845Z",
  "accountId": "0835961d-0dca-4896-99b4-11fa96c82eda",
  "reference": "TRX-55190",
  "notes": null,
  "branchId": null,
  "clientId": "c8b5212a-1b50-4864-9d45-7b562575f3fa",
  "journalEntryId": "b42f4c6e-154d-4a8a-b04e-62dca3465d57",
  "cancelledAt": null,
  "createdAt": "2026-10-06T17:01:02.846Z",
  "allocations": [
    { "id": "9152b27c-05d7-4c9a-9989-1e0fe31b6e3a", "invoiceType": "SALES_INVOICE", "invoiceId": "1086f27d-620a-4e60-9dfe-0720fee48398", "amount": "170.25" }
  ]
}
```

400 Bad Request

```
{
  "statusCode": 400,
  "code": "ALLOCATION_EXCEEDS_PAYMENT",
  "message": "Allocations exceed the payment amount",
  "requestId": "54c3a873-10f7-4dc1-97e4-52de241723c0"
}
```

عرض الفواتير غير المسددة

تحميل `sales.invoice:view` - (أو `PARTIALLY_PAID`) - `GET /sales/invoices?status=ISSUED&paymentStatus=UNPAID`
صفوف القائمة `balanceDue` ، ويسميه مسار الفاتورة الواحدة `amountDue` .

**curl**

```
curl "https://erp.sa4s.com/api/v1/sales/invoices?customerId=fab57752-483f-4b1a-8d80-8c0401809ffe&status=ISSUED&paymentStatus=UNPAID&sort=issueDate&order=asc" \
-H "Authorization: Bearer $SA4S_API_KEY"
# partially paid ones: paymentStatus=PARTIALLY_PAID (run both to get every open invoice)
```

JavaScript

```
const open = [];
for (const paymentStatus of ['UNPAID', 'PARTIALLY_PAID']) {
  const q = new URLSearchParams({ customerId, status: 'ISSUED', paymentStatus, pageSize: '200', sort: 'issueDate',
    order: 'asc' });
  open.push(...(await sa4s('GET', `/${sales}/invoices?${q}`).items));
}
const due = open.reduce((sum, inv) => sum + Math.round(Number(inv.balanceDue) * 100), 0) / 100; // compare in halalas
```

Python

```
from decimal import Decimal

open_invoices = []
for status in ("UNPAID", "PARTIALLY_PAID"):
    open_invoices += sa4s("GET", "/sales/invoices", params={
        "customerId": customer_id, "status": "ISSUED", "paymentStatus": status, "pageSize": 200, "sort": "issueDate",
        "order": "asc",
    })["items"]
total_due = sum(Decimal(i["balanceDue"]) for i in open_invoices)
```

PHP

```
<?php
$open = [];
foreach (['UNPAID', 'PARTIALLY_PAID'] as $ps) {
    $q = http_build_query(['customerId' => $customerId, 'status' => 'ISSUED', 'paymentStatus' => $ps, 'pageSize' =>
    200, 'sort' => 'issueDate', 'order' => 'asc']);
    $open = array_merge($open, sa4s('GET', "/sales/invoices?$q")['items']);
}
$due = array_reduce($open, fn ($s, $i) => bcadd($s, $i['balanceDue'], 2), '0');
```

C#

```
var open = new List<JsonElement>();
foreach (var ps in new[] { "UNPAID", "PARTIALLY_PAID" })
{
    var page = await Sa4s.Send(HttpMethod.Get, $"sales/invoices?customerId={customerId}&status=ISSUED&paymentStatus={ps}&pageSize=200&sort=issueDate&order=asc");
    open.AddRange(page.GetProperty("items").EnumerateArray());
}
decimal due = open.Sum(i => decimal.Parse(i.GetProperty("balanceDue").GetString()!,
System.Globalization.CultureInfo.InvariantCulture));
```

200 OK (abridged)

```
{
  "items": [
    {
      "id": "46f95e34-0748-4b23-a08b-8765c6ad5d58",
      "number": "INV-2026-000003",
      "type": "TAX",
      "status": "ISSUED",
      "paymentStatus": "UNPAID",
```



```

    "issueDate": "2026-10-06T17:01:02.393Z",
    "dueDate": "2026-11-05T00:00:00.000Z",
    "total": "230",
    "paidAmount": "0",
    "paymentMethod": "CREDIT",
    "customer": { "id": "fab57752-483f-4b1a-8d80-8c0401809ffe", "name": "شركة الأفق للتجارة", "nameEn": "Horizon Trading Co", "vatNumber": "310123456700003" },
    "balanceDue": "230"
  }
],
"total": 1,
"page": 1,
"pageSize": 25
}

```

الدفع لمورد

الحساب الدافع ما لم تسمح الشركة بالرصيد النقدي السالب. `payments:create` — الصلاحية `POST /payments/disbursements`، والتوزيعات تشير إلى فواتير مشتريات مرحلة. يجب أن يكفي رصيد

curl

```

curl -X POST "https://erp.sa4s.com/api/v1/payments/disbursements" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{
  "supplierId": "3788b574-54f8-4ed9-9a30-f7d38199ff05",
  "method": "BANK_TRANSFER",
  "amount": "100.00",
  "clientId": "ed9df166-3fee-44b6-8a0d-d006e00e427e",
  "allocations": [ { "invoiceId": "274f3053-5728-4d84-931f-a42ef0ebdc55", "amount": "100.00" } ]
}'

```

JavaScript

```

const payment = await sa4s('POST', '/payments/disbursements', {
  supplierId: SUPPLIER_ID, method: 'BANK_TRANSFER', amount: '100.00', clientId: payout.clientId,
  allocations: [{ invoiceId: PURCHASE_INVOICE_ID, amount: '100.00' }],
});
console.log(payment.number); // PAY-2026-000001

```

Python

```

payment = sa4s("POST", "/payments/disbursements", {
    "supplierId": SUPPLIER_ID, "method": "BANK_TRANSFER", "amount": "100.00", "clientId": payout["client_uuid"],
    "allocations": [{"invoiceId": PURCHASE_INVOICE_ID, "amount": "100.00"}],
})

```

PHP

```

<?php
$payment = sa4s('POST', '/payments/disbursements', [
    'supplierId' => $supplierId, 'method' => 'BANK_TRANSFER', 'amount' => '100.00', 'clientId' =>
    $payout['client_uuid'],
    'allocations' => [['invoiceId' => $purchaseInvoiceId, 'amount' => '100.00']],
]);

```



409 Conflict

```
{
  "statusCode": 409,
  "code": "INSUFFICIENT_FUNDS",
  "message": "The balance of \"Main Bank Account\" is lower than the amount",
  "details": { "accountId": "f8aa56c8-9796-455f-b072-967a40ef7954", "accountName": "Main Bank Account", "balance":
"0.00", "amount": "100.00" },
  "requestId": "f40c8732-1f1a-4be3-9c16-7714b34c865e"
}
```

curl

```
# receipts (IN) of October, newest first
curl "https://erp.sa4s.com/api/v1/payments?direction=IN&from=2026-10-01&to=2026-10-31&sort=date&order=desc" \
-H "Authorization: Bearer $SA4S_API_KEY"
```

JavaScript

```
const { items } = await sa4s('GET', '/payments?direction=IN&from=2026-10-01&to=2026-10-31&sort=date&order=desc');
```

Python

```
items = sa4s("GET", "/payments", params={"direction": "IN", "from": "2026-10-01", "to": "2026-10-31", "sort": "date",
"order": "desc"})["items"]
```

PHP

```
<?php
$items = sa4s('GET', '/payments?direction=IN&from=2026-10-01&to=2026-10-31&sort=date&order=desc')['items'];
```



وصفات — المشتريات

إنشاء فاتورة مشتريات

عند `POST /purchases/invoices` — الصلاحية `purchases.invoice:create`. فاتورة المورد المستقلة تستلم البضاعة في `warehouseId` عند ترحيلها، ويمكن بناؤها أيضاً من استلام بضاعة (`receiptId`) أو أمر شراء (`orderId`). والحقل `unitPrice` إلزامي في أسطر المشتريات. ويتطلب الترحيل (`POST /purchases/invoices/{id}/post` أو `post: true` عند الإنشاء) الصلاحية `purchases.invoice:approve`: فهو يُدخل المخزون بالتكلفة، ويقيّد ضريبة المدخلات والمستحق للمورد. ويُرفض `supplierInvoiceNo` المستخدم مسبقاً للمورد نفسه (`DUPLICATE_SUPPLIER_INVOICE`)، ما يجعل إعادة المحاولة آمنة.

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/purchases/invoices" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{
  "supplierId": "3788b574-54f8-4ed9-9a30-f7d38199ff05",
  "warehouseId": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0",
  "supplierInvoiceNo": "NWF-88231",
  "pricesIncludeVat": false,
  "items": [
    { "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9", "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4",
      "quantity": "20", "unitPrice": "14.40" },
    { "productId": "6fe0f6fa-ed4d-4c4b-81ca-c8095504658f", "quantity": "50", "unitPrice": "28.00" }
  ]
}'

# post it (stock in at cost, input VAT, supplier payable) – needs purchases.invoice:approve
curl -X POST "https://erp.sa4s.com/api/v1/purchases/invoices/274f3053-5728-4d84-931f-a42ef0ebdc55/post" \
-H "Authorization: Bearer $SA4S_API_KEY"
```

JavaScript

```
const bill = await sa4s('POST', '/purchases/invoices', {
  supplierId: SUPPLIER_ID,
  warehouseId: WAREHOUSE_ID,           // where the goods are received
  supplierInvoiceNo: 'NWF-88231',      // the supplier's own number (keep it unique on your side)
  pricesIncludeVat: false,
  items: [
    { productId: WATER_ID, unitId: BOX_UNIT_ID, quantity: '20', unitPrice: '14.40' }, // unitPrice is required on
    purchases
    { productId: DATES_ID, quantity: '50', unitPrice: '28.00' },
  ],
  // post: true → create and post in one call
});
const posted = await sa4s('POST', `/purchases/invoices/${bill.id}/post`);
console.log(posted.number, posted.status, posted.total, posted.balanceDue); // PI-2026-000001 POSTED 1941.2 1941.2
```

Python

```
bill = sa4s("POST", "/purchases/invoices", {
    "supplierId": SUPPLIER_ID,
    "warehouseId": WAREHOUSE_ID,
    "supplierInvoiceNo": "NWF-88231",
    "pricesIncludeVat": False,
    "items": [
        {"productId": WATER_ID, "unitId": BOX_UNIT_ID, "quantity": "20", "unitPrice": "14.40"},
        {"productId": DATES_ID, "quantity": "50", "unitPrice": "28.00"},
    ]
})
```



```
],
"post": True,
})
print(bill["number"], bill["status"])
```

PHP

```
<?php
$bill = sa4s('POST', '/purchases/invoices', [
    'supplierId' => $supplierId,
    'warehouseId' => $warehouseId,
    'supplierInvoiceNo' => 'NWF-88231',
    'pricesIncludeVat' => false,
    'items' => [
        ['productId' => $waterId, 'unitId' => $boxUnitId, 'quantity' => '20', 'unitPrice' => '14.40'],
        ['productId' => $datesId, 'quantity' => '50', 'unitPrice' => '28.00'],
    ],
    'post' => true,
]);
echo $bill['number'], ' ', $bill['status'], "\n";
```

201 Created (abridged)

```
{
  "id": "274f3053-5728-4d84-931f-a42ef0ebdc55",
  "number": "PI-2026-000001",
  "supplierInvoiceNo": "NWF-88231",
  "supplierId": "3788b574-54f8-4ed9-9a30-f7d38199ff05",
  "warehouseId": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0",
  "status": "DRAFT",
  "date": "2026-10-06T00:00:00.000Z",
  "dueDate": "2026-11-05T00:00:00.000Z",
  "pricesIncludeVat": false,
  "subtotal": "1688",
  "taxableTotal": "1688",
  "vatTotal": "253.2",
  "total": "1941.2",
  "paidAmount": "0",
  "receivesStock": true,
  "items": [
    { "productId": "6fe0f6fa-ed4d-4c4b-81ca-c8095504658f", "unitId": "0757ef16-15fa-4b58-996d-ebb3d8bb5fcb",
      "quantity": "50", "baseQuantity": "50", "unitPrice": "28", "vatCategory": "S", "vatRate": "15", "taxableAmount":
      "1400", "vatAmount": "210", "lineTotal": "1610" },
    { "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9", "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4",
      "quantity": "20", "baseQuantity": "480", "unitPrice": "14.4", "vatCategory": "S", "vatRate": "15", "taxableAmount":
      "288", "vatAmount": "43.2", "lineTotal": "331.2" }
  ],
  "balanceDue": "1941.2"
}
```

إنشاء أمر شراء

POST /purchases/orders — الصلاحية purchases.order:create ، ويبدأ الأمر بحالة DRAFT .

curl

```
curl -X POST "https://erp.sa4s.com/api/v1/purchases/orders" \
-H "Authorization: Bearer $SA4S_API_KEY" -H "Content-Type: application/json" \
-d '{
  "supplierId": "3788b574-54f8-4ed9-9a30-f7d38199ff05",
  "warehouseId": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0",
  "expectedDate": "2026-10-20",
```



```
"items": [ { "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9", "unitId": "b82dbc97-8f7b-43c0-8b73-8b4aed1ebdb4",
"quantity": "30", "unitPrice": "14.40" } ]
}'
# → 201 { "number": "PO-2026-000001", "status": "DRAFT", "total": "496.8", ... }
# then POST /purchases/orders/{id}/submit (DRAFT → PENDING) and POST /purchases/orders/{id}/approve (→ APPROVED)
```

JavaScript

```
const po = await sa4s('POST', '/purchases/orders', {
  supplierId: SUPPLIER_ID, warehouseId: WAREHOUSE_ID, expectedDate: '2026-10-20',
  items: [{ productId: WATER_ID, unitId: BOX_UNIT_ID, quantity: '30', unitPrice: '14.40' }],
});
await sa4s('POST', `/purchases/orders/${po.id}/submit`); // needs purchases.order:edit
await sa4s('POST', `/purchases/orders/${po.id}/approve`); // needs purchases.order:approve
```

Python

```
po = sa4s("POST", "/purchases/orders", {
  "supplierId": SUPPLIER_ID, "warehouseId": WAREHOUSE_ID, "expectedDate": "2026-10-20",
  "items": [{"productId": WATER_ID, "unitId": BOX_UNIT_ID, "quantity": "30", "unitPrice": "14.40"}],
})
```

PHP

```
<?php
$po = sa4s('POST', '/purchases/orders', [
  'supplierId' => $supplierId, 'warehouseId' => $warehouseId, 'expectedDate' => '2026-10-20',
  'items' => [['productId' => $waterId, 'unitId' => $boxUnitId, 'quantity' => '30', 'unitPrice' => '14.40']],
]);
```



وصفات — المخزون

أرصدة المخزون حسب المستودع

الصلاحية `inventory:view` . صف لكل منتج × مستودع فيه `onHand` و `reserved` و `available` `onHand - reserved`، وكلها بالوحدة الأساسية؛ ولا يظهر `avgCost` و `value` إلا مع `inventory.cost:view`.

curl

```
curl "https://erp.sa4s.com/api/v1/inventory/balances?warehouseId=1ae3935b-6fae-4c7f-803a-bf405dbc0ea0&page=1&pageSize=200&sort=sku&order=asc" \
-H "Authorization: Bearer $SA4S_API_KEY"
# other filters: productId, categoryId, search (SKU / name), status=all|low|over|out|dead|in_stock
```

JavaScript

```
// full snapshot of one warehouse: { sku → available }
async function stockSnapshot(warehouseId) {
  const out = new Map();
  for (let page = 1; ; page++) {
    const res = await sa4s('GET', `/inventory/balances?warehouseId=${warehouseId}&page=${page}&pageSize=200&sort=sku&order=asc`);
    for (const row of res.items) out.set(row.product.sku, row.available); // decimal strings, base units
    if (page * res.pageSize >= res.total) return out;
  }
}
```

Python

```
def stock_snapshot(warehouse_id):
    out, page = {}, 1
    while True:
        res = sa4s("GET", "/inventory/balances", params={"warehouseId": warehouse_id, "page": page, "pageSize": 200, "sort": "sku", "order": "asc"})
        out.update({row["product"]["sku"]: row["available"] for row in res["items"]})
        if page * res["pageSize"] >= res["total"]:
            return out
        page += 1
```

PHP

```
<?php
function stock_snapshot(string $warehouseId): array
{
    $out = [];
    for ($page = 1; ; $page++) {
        $res = sa4s('GET', '/inventory/balances?' . http_build_query(['warehouseId' => $warehouseId, 'page' => $page, 'pageSize' => 200, 'sort' => 'sku', 'order' => 'asc']));
        foreach ($res['items'] as $row) $out[$row['product']['sku']] = $row['available'];
        if ($page * $res['pageSize'] >= $res['total']) return $out;
    }
}
```

**C#**

```
var stock = new Dictionary<string, string>();
for (int page = 1; ; page++)
{
    var res = await Sa4s.Send(HttpMethod.Get, $"inventory/balances?warehouseId={warehouseId}&page={page}&pageSize=200&sort=sku&order=asc");
    foreach (var row in res.GetProperty("items").EnumerateArray())
        stock[row.GetProperty("product").GetProperty("sku").GetString()!] = row.GetProperty("available").GetString();
    if (page * res.GetProperty("pageSize").GetInt32() >= res.GetProperty("total").GetInt32()) break;
}
```

200 OK (abridged)

```
{
  "items": [
    {
      "warehouseId": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0",
      "warehouse": { "id": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0", "code": "WH-MAIN", "nameAr": "المستودع الرئيسي",
"nameEn": "Main Warehouse" },
      "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9",
      "product": { "id": "51f771bf-cfb5-4814-8e79-55b990be69c9", "sku": "SKU-2026-000001", "nameAr": "مياه معدنية 330 مل",
"nameEn": "Mineral water 330 ml", "type": "PHYSICAL", "trackBatch": false, "baseUnit": "PCS" },
      "onHand": "282",
      "reserved": "0",
      "available": "282",
      "minStock": "48",
      "maxStock": null,
      "reorderLevel": null,
      "flags": { "low": false, "over": false, "out": false },
      "lastMovementAt": "2026-10-06T17:01:02.314Z",
      "updatedAt": "2026-10-06T17:01:02.284Z",
      "avgCost": "0.6",
      "value": "169.20"
    }
  ],
  "total": 2,
  "page": 1,
  "pageSize": 200,
  "summary": { "rows": 2, "totalOnHand": "325", "totalValue": "1373.20" }
}
```

curl

```
# one product across every warehouse (+ batches when the product tracks them)
curl "https://erp.sa4s.com/api/v1/inventory/products/51f771bf-cfb5-4814-8e79-55b990be69c9/stock" -H "Authorization: Bearer $SA4S_API_KEY"
# → { "product": { "sku": "SKU-2026-000001", ... }, "warehouses": [ { "warehouseId": "...", "onHand": "282", "reserved": "0", "available": "282" } ],
#    "totals": { "onHand": "282", "reserved": "0", "available": "282" } }
```

JavaScript

```
const s = await sa4s('GET', `/inventory/products/${productId}/stock`);
console.log(s.product.sku, s.totals.available, s.warehouses.map((w) => `${w.warehouse.code}=${w.available}`));
```

Python

```
s = sa4s("GET", f"/inventory/products/{product_id}/stock")
```



PHP

```
<?php
$s = sa4s('GET', "/inventory/products/$productId/stock");
```

حركات المخزون منذ تاريخ

GET /inventory/movements — الصلاحية inventory:view. دفتر المخزون غير القابل للتعديل: كل بيع وشراء وإرجاع وتسوية وتحويل وجرد ككمية موقّعة qtyChange بالوحدة الأساسية، مع الرصيد الناتج balanceAfter والمستند المصدر (refType و refId و note). الأنواع: OPENING و PURCHASE_RECEIPT و PURCHASE_RETURN و SALE و SALE_RETURN و ADJUSTMENT_IN و ADJUSTMENT_OUT و TRANSFER_IN و TRANSFER_OUT و DAMAGE و EXPIRED و COUNT_ADJUSTMENT و COMPOSITE_CONSUME.

curl

```
curl "https://erp.sa4s.com/api/v1/inventory/movements?from=2026-10-06&warehouseId=1ae3935b-6fae-4c7f-803a-bf405dbc0ea0&order=asc&page=1&pageSize=200" \
-H "Authorization: Bearer $SA4S_API_KEY"
# other filters: to, productId, type (SALE, PURCHASE_RECEIPT, ...), refType, refId, batchId
```

JavaScript

```
// every movement since a date, oldest first; skip the ones you already processed
async function movementsSince(day, seen) {
  const changed = new Set();
  for (let page = 1; ; page++) {
    const res = await sa4s('GET', `/inventory/movements?from=${day}&order=asc&page=${page}&pageSize=200`);
    for (const m of res.items) if (!seen.has(m.id)) { seen.add(m.id); changed.add(`${m.productId}:${m.warehouseId}`); }
    if (page * res.pageSize >= res.total) return changed;
  }
}
```

Python

```
def movements_since(day, seen):
    changed, page = set(), 1
    while True:
        res = sa4s("GET", "/inventory/movements", params={"from": day, "order": "asc", "page": page, "pageSize": 200})
        for m in res["items"]:
            if m["id"] not in seen:
                seen.add(m["id"])
                changed.add((m["productId"], m["warehouseId"]))
        if page * res["pageSize"] >= res["total"]:
            return changed
        page += 1
```

PHP

```
<?php
function movements_since(string $day, array &$amp;seen): array
{
    $changed = [];
    for ($page = 1; ; $page++) {
        $res = sa4s('GET', '/inventory/movements?' . http_build_query(['from' => $day, 'order' => 'asc', 'page' => $page, 'pageSize' => 200]));
        foreach ($res['items'] as $m) {
            if (isset($seen[$m['id']])) continue;
            $seen[$m['id']] = true;
            $changed["{$m['productId']}:{m['warehouseId']}"] = true;
        }
    }
}
```



```

    }
    if ($page * $res['pageSize'] >= $res['total']) return array_keys($changed);
  }
}

```

200 OK (abridged)

```

{
  "items": [
    {
      "id": "3002b923-48af-4d7a-b5b3-325a4fbaec75",
      "createdAt": "2026-10-06T17:01:02.058Z",
      "type": "PURCHASE_RECEIPT",
      "warehouseId": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0",
      "productId": "51f771bf-cfb5-4814-8e79-55b990be69c9",
      "product": { "sku": "SKU-2026-000001", "nameAr": "مياه معدنية 330 مل", "nameEn": "Mineral water 330 ml" },
      "batchId": null,
      "qtyChange": "480",
      "balanceAfter": "480",
      "refType": "PURCHASE_INVOICE",
      "refId": "274f3053-5728-4d84-931f-a42ef0ebdc55",
      "note": "PI-2026-000001",
      "unitCost": "0.6",
      "value": "288.00"
    },
    {
      "id": "16d83e5d-d936-4dff-a48a-b2e059637c92",
      "createdAt": "2026-10-06T17:01:03.681Z",
      "type": "SALE",
      "warehouseId": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0",
      "productId": "6fe0f6fa-ed4d-4c4b-81ca-c8095504658f",
      "qtyChange": "-1",
      "balanceAfter": "41",
      "refType": "SALES_INVOICE",
      "refId": "015727de-9146-41eb-839b-fd477dc9e9a8",
      "note": "INV-2026-000006",
      "unitCost": "28",
      "value": "-28.00"
    }
  ],
  "total": 8,
  "page": 1,
  "pageSize": 200
}

```



وصفات — التقارير

قائمة التقارير

GET /reports — الصلاحية reports:view. تسرد التقارير المتاحة للمفتاح مع مرشحاتها وأعمدتها وصيغها. ويحتاج كل تقرير أيضاً صلاحية بياناته (مثل sales.invoice:view لتقارير المبيعات و accounting:view للقوائم المالية).

curl

```
# only the reports this key may run, with their filters and columns
curl "https://erp.sa4s.com/api/v1/reports" -H "Authorization: Bearer $SA4S_API_KEY"
```

JavaScript

```
const { items } = await sa4s('GET', '/reports');
console.log(items.map((r) => `${r.name}: ${r.filters.map((f) => f.name).join(', ')}`));
```

Python

```
for r in sa4s("GET", "/reports")["items"]:
    print(r["name"], [f["name"] for f in r["filters"]])
```

PHP

```
<?php
foreach (sa4s('GET', '/reports')['items'] as $r) echo $r['name'], PHP_EOL;
```

200 OK (abridged)

```
{
  "groups": [
    { "key": "sales", "labelAr": "المبيعات", "labelEn": "Sales", "reports": ["sales-daily", "sales-monthly", "sales-by-product"] }
  ],
  "items": [
    {
      "name": "sales-daily",
      "group": "sales",
      "titleAr": "المبيعات اليومية",
      "titleEn": "Daily sales",
      "permissions": ["reports:view", "sales.invoice:view"],
      "paginated": true,
      "dateMode": "range",
      "defaultSort": { "key": "date", "order": "asc" },
      "filters": [ { "name": "dateFrom", "type": "date", "labelAr": "من تاريخ", "labelEn": "From date" }, { "name": "dateTo", "type": "date" }, { "name": "branchId", "type": "uuid" } ],
      "columns": [ { "key": "date", "labelAr": "التاريخ", "labelEn": "Date", "type": "date" } ],
      "formats": ["json", "csv", "xlsx", "pdf"]
    }
  ],
  "canExport": true,
  "canSeeCost": true
}
```



تشغيل تقرير بصيغة JSON

GET /reports/{name} مع مرشحات التقرير (warehouseId و branchId و period و asOf و dateTo و dateFrom) و customerId و ... و page و pageSize (حتى 500). ويعيد format=csv أو xlsx أو pdf ملفاً ويتطلب reports:export. ومن الأسماء المتاحة: sales-daily و sales-monthly و sales-by-product و sales-by-customer و sales-by-payment-method و product-profitability و stock-balance و stock-movement و low-stock و expiring-batches و purchases-by- و supplier و customer-balances و receivables-aging و profit-loss و balance-sheet و trial-balance و cash-flow.

curl

```
curl "https://erp.sa4s.com/api/v1/reports/sales-daily?dateFrom=2026-10-01&dateTo=2026-10-31" \
-H "Authorization: Bearer $SA4S_API_KEY"

# the same report as a file (needs reports:export): format=csv | xlsx | pdf, lang=ar | en | both
curl "https://erp.sa4s.com/api/v1/reports/sales-daily?dateFrom=2026-10-01&dateTo=2026-10-31&format=xlsx&lang=en" \
-H "Authorization: Bearer $SA4S_API_KEY" -o sales-october.xlsx
```

JavaScript

```
const r = await sa4s('GET', '/reports/sales-daily?dateFrom=2026-10-01&dateTo=2026-10-31');
for (const row of r.rows) console.log(row.date, row.invoices, row.netSales, row.vat, row.total);
console.log('month total', r.totals.total);
```

Python

```
r = sa4s("GET", "/reports/stock-balance", params={"warehouseId": WAREHOUSE_ID, "status": "low", "pageSize": 500})
for row in r["rows"]:
    print(row["sku"], row["onHand"], row["minStock"])
```

PHP

```
<?php
$r = sa4s('GET', '/reports/sales-by-product?' . http_build_query(['dateFrom' => '2026-10-01', 'dateTo' => '2026-10-31']));
foreach ($r['rows'] as $row) echo json_encode($row, JSON_UNESCAPED_UNICODE), "\n";
```

200 OK (abridged)

```
{
  "report": { "name": "sales-daily", "group": "sales", "titleAr": "المبيعات اليومية", "titleEn": "Daily sales" },
  "filters": { "dateFrom": "2026-10-01", "dateTo": "2026-10-31", "branchId": null, "warehouseId": null },
  "currency": "SAR",
  "generatedAt": "2026-10-06T17:01:03.112Z",
  "columns": [ { "key": "date", "labelAr": "التاريخ", "labelEn": "Date", "type": "date" } ],
  "rows": [
    { "date": "2026-10-06", "invoices": 4, "creditNotes": 0, "quantity": "254", "grossSales": "511.08", "returns": "0.00", "netSales": "511.08", "vat": "76.67", "total": "587.75", "cost": "343.60", "profit": "167.48", "marginPercent": "32.77" }
  ],
  "totals": { "invoices": 4, "netSales": "511.08", "vat": "76.67", "total": "587.75", "profit": "167.48" },
  "summary": { "netSales": "511.08", "total": "587.75", "invoices": 4, "creditNotes": 0 },
  "page": 1,
  "pageSize": 50,
  "total": 1,
  "pages": 1
}
```



Webhooks

ترسل ال Webhooks الأحداث إلى عنوان HTTPS لديك لحظة وقوعها، فلا تحتاج إلى الاستطلاع المتكرر.

الاشتراك

1. في النظام افتح الإعدادات ← التكامل البرمجي (API) ← الويب هوك ← نقطة استقبال جديدة (تُدار ال Webhooks من النظام لا عبر المفاتيح).
2. أدخل عنوان HTTPS واختر الأحداث (أو * لكلها) ثم احفظ. حتى 20 عنواناً لكل شركة.
3. انسخ سر التوقيع (whsec_...) — يُعرض مرة واحدة — واحفظه بجوار المستقبل لديك.
4. اضغط إرسال تجريبي: يملك فوراً حدث webhook.test موقع، ويعرض النظام ردك.

صيغة الإرسال

كل إرسال طلب POST جسمه JSON بالشكل { id, event, createdAt, data } مع هذه الترويسات:

الترويسة	المحتوى
X-SA4S-Event	اسم الحدث، مثل invoice.issued
X-SA4S-Delivery	معرف الإرسال — يساوي id في الجسم ويبقى ثابتاً في كل إعادة: استخدمه لمنع المعالجة المكررة
X-SA4S-Timestamp	وقت هذه المحاولة بثواني Unix
X-SA4S-Signature	sha256= متبوعاً بتجزئة HMAC-SHA256 الست عشرية ل الجسم الخام + "." + timestamp بمفتاح سر العنوان

يحمل data المعرفات مع لقطة صغيرة من السجل، فلا تحتاج أغلب الأنظمة إلى طلب إضافي؛ واطلب السجل الكامل عبر الواجهة عند الحاجة. تُسلّم الأحداث مرة واحدة على الأقل وقد تصل بغير ترتيبها — اجعل المعالجة غير متأثرة بالتكرار، وإن كان الترتيب مهماً فأعد قراءة الحالة الحالية.

HTTP

```

POST /sa4s/webhooks HTTP/1.1
Host: shop.example.com
Content-Type: application/json
User-Agent: SA4S-Webhooks/1.0
X-SA4S-Event: invoice.issued
X-SA4S-Delivery: 2a695809-856b-43c4-ae7e-ba876cb05daf
X-SA4S-Timestamp: 1791305956
X-SA4S-Signature: sha256=bf28b879951e61d1a313f917e5f07f7bdd884e7e98c46620c724a639165f3cd5

{"id":"2a695809-856b-43c4-ae7e-ba876cb05daf","event":"invoice.issued","createdAt":"2026-10-06T17:01:03.551Z","data":{...}}
```



قائمة الأحداث

الحدث	مقي	data
invoice.issued	صدرت فاتورة مبيعات أو إشعار دائن أو مدين	invoiceId و لقطه invoice
invoice.cancelled	ألغيت فاتورة مبيعات مُصدّرة	invoiceId و لقطه invoice
payment.recorded	سُجِّل قبض من عميل أو دفع لمورد — أو أُلغِي (cancelled: true)	paymentId و direction و invoiceId / amount و الاختياريان و لقطه payment
stock.changed	تغيّرت الكمية المتوفرة لمنتج في مستودع	productId و warehouseId و balance (onHand و reserved)
zatca.status	تغيّرت حالة الفاتورة لدى الهيئة	invoiceId و status و لقطه invoice
webhook.test	ضغطت إرسال تجريبي في النظام	message و endpointId و triggeredBy

invoice.issued

```
{
  "id": "2a695809-856b-43c4-ae7e-ba876cb05daf",
  "event": "invoice.issued",
  "createdAt": "2026-10-06T17:01:03.551Z",
  "data": {
    "invoiceId": "aafe08be-d24b-47a4-9058-61fce56397b6",
    "invoice": {
      "id": "aafe08be-d24b-47a4-9058-61fce56397b6",
      "number": "INV-2026-000005",
      "type": "SIMPLIFIED",
      "status": "ISSUED",
      "paymentStatus": "PAID",
      "customerId": null,
      "branchId": "73afa37e-46d6-4c51-9a13-549d92cad970",
      "issueDate": "2026-10-06T17:01:03.475Z",
      "total": "40"
    }
  }
}
```

invoice.cancelled

```
{
  "id": "57109e0d-53a4-463e-816b-59fc33da1f36",
  "event": "invoice.cancelled",
  "createdAt": "2026-10-06T17:01:03.787Z",
  "data": {
    "invoiceId": "015727de-9146-41eb-839b-fd477dc9e9a8",
    "invoice": {
      "id": "015727de-9146-41eb-839b-fd477dc9e9a8",
      "number": "INV-2026-000006",
      "type": "SIMPLIFIED",
      "status": "CANCELLED",
      "paymentStatus": "UNPAID",
      "customerId": "b4382d00-46be-436d-af45-c3dcd8b0222c",
      "branchId": "73afa37e-46d6-4c51-9a13-549d92cad970",
      "issueDate": "2026-10-06T17:01:03.635Z",
      "total": "40"
    }
  }
}
```

**payment.recorded**

```
{
  "id": "c6ff3cf3-3f1b-4564-bcb7-91181e19eece",
  "event": "payment.recorded",
  "createdAt": "2026-10-06T17:01:03.551Z",
  "data": {
    "paymentId": "1662f93b-01fc-4abc-a45d-96aeda8d6f60",
    "direction": "IN",
    "invoiceId": "aafe08be-d24b-47a4-9058-61fce56397b6",
    "amount": "40",
    "payment": {
      "id": "1662f93b-01fc-4abc-a45d-96aeda8d6f60",
      "number": "RCV-2026-000006",
      "direction": "IN",
      "partyType": "CUSTOMER",
      "partyId": null,
      "method": "CASH",
      "amount": "40",
      "date": "2026-10-06T17:01:03.518Z",
      "reference": null
    }
  }
}
```

stock.changed

```
{
  "id": "48fb6802-8ac4-47f7-87de-b2483fd0e19b",
  "event": "stock.changed",
  "createdAt": "2026-10-06T17:01:03.552Z",
  "data": {
    "productId": "6fe0f6fa-ed4d-4c4b-81ca-c8095504658f",
    "warehouseId": "1ae3935b-6fae-4c7f-803a-bf405dbc0ea0",
    "balance": { "onHand": "42", "reserved": "0" }
  }
}
```

حمولة **zatca.status** بشكل **invoice.issued** نفسه مع الحالة الجديدة **status** (القيم أدناه توضيحية):

zatca.status

```
{
  "id": "0d3c1f7e-5b52-4c8a-9a51-3f8f1d3e2b10",
  "event": "zatca.status",
  "createdAt": "2026-10-06T17:17:12.410Z",
  "data": {
    "invoiceId": "442ea37d-b9ac-4535-b726-b3ad69832afe",
    "status": "REPORTED",
    "invoice": {
      "id": "442ea37d-b9ac-4535-b726-b3ad69832afe",
      "number": "INV-2026-000007",
      "type": "SIMPLIFIED",
      "status": "ISSUED",
      "paymentStatus": "PAID",
      "customerId": null,
      "branchId": "73afa37e-46d6-4c51-9a13-549d92cad970",
      "issueDate": "2026-10-06T17:17:09.161Z",
      "total": "46.9"
    }
  }
}
```

**webhook.test**

```
{
  "id": "7a8dad24-9ac7-48d7-82d3-62b396f7bdfb",
  "event": "webhook.test",
  "createdAt": "2026-10-06T16:59:16.516Z",
  "data": { "message": "This is a test delivery from SA4S", "endpointId": "b07d0348-c275-41e7-81ae-831247444a00",
  "triggeredBy": "Owner" }
}
```

التحقق من التوقيع

احسب HMAC على البايتات الخام التي استلمتها — فتحليل JSON ثم إعادة تسلسله يغيّر البايتات ويفسد التوقيع. وقارن بزمن ثابت.

Node.js

```
import { createHmac, timingSafeEqual } from 'node:crypto';

/** rawBody: the exact bytes received (Buffer) – never a re-serialised JSON object */
export function verifySa4s(headers, rawBody, secret, toleranceSec = 300) {
  const ts = headers['x-sa4s-timestamp'];
  const given = String(headers['x-sa4s-signature'] ?? '');
  if (!ts || Math.abs(Date.now() / 1000 - Number(ts)) > toleranceSec) return false; // replay window
  const expected = 'sha256=' + createHmac('sha256', secret).update(`${ts}.`).update(rawBody).digest('hex');
  return given.length === expected.length && timingSafeEqual(Buffer.from(given), Buffer.from(expected));
}
```

Python

```
import hashlib, hmac, time

def verify_sa4s(headers, raw_body: bytes, secret: str, tolerance=300) -> bool:
    ts = headers.get("X-SA4S-Timestamp", "")
    if not ts.isdigit() or abs(time.time() - int(ts)) > tolerance:
        return False
    expected = "sha256=" + hmac.new(secret.encode(), ts.encode() + b"." + raw_body, hashlib.sha256).hexdigest()
    return hmac.compare_digest(expected, headers.get("X-SA4S-Signature", ""))
```

PHP

```
<?php
function verify_sa4s(string $rawBody, string $secret, int $tolerance = 300): bool
{
    $ts = $_SERVER['HTTP_X_SA4S_TIMESTAMP'] ?? '';
    if (!ctype_digit($ts) || abs(time() - (int) $ts) > $tolerance) return false;
    $expected = 'sha256=' . hash_hmac('sha256', $ts . '.' . $rawBody, $secret);
    return hash_equals($expected, $_SERVER['HTTP_X_SA4S_SIGNATURE'] ?? '');
}

$raw = file_get_contents('php://input');
if (!verify_sa4s($raw, getenv('SA4S_WEBHOOK_SECRET')) { http_response_code(401); exit; }
$event = json_decode($raw, true);
```

C#

```
using System.Security.Cryptography;
using System.Text;

static bool VerifySa4s(HttpRequest req, byte[] rawBody, string secret, int toleranceSec = 300)
{
    string ts = req.Headers["X-SA4S-Timestamp"].ToString();
    if (!long.TryParse(ts, out var unix) || Math.Abs(DateTimeOffset.UtcNow.ToUnixTimeSeconds() - unix) > toleranceSec)
```



```
return false;
using var hmac = new HMACSHA256(Encoding.UTF8.GetBytes(secret));
byte[] signed =[.. Encoding.UTF8.GetBytes(ts + "."), .. responseBody];
string expected = "sha256=" + Convert.ToHexString(hmac.ComputeHash(signed)).ToLowerInvariant();
return CryptographicOperations.FixedTimeEquals(Encoding.UTF8.GetBytes(expected),
Encoding.UTF8.GetBytes(req.Headers["X-SA4S-Signature"].ToString()));
}
```

Java

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.util.HexFormat;

static boolean verifySa4s(String timestamp, String signature, byte[] responseBody, String secret) throws Exception {
    if (timestamp == null || !timestamp.matches("\\d+")) return false;
    if (Math.abs(System.currentTimeMillis() / 1000 - Long.parseLong(timestamp)) > 300) return false;
    Mac mac = Mac.getInstance("HmacSHA256");
    mac.init(new SecretKeySpec(secret.getBytes(StandardCharsets.UTF_8), "HmacSHA256"));
    mac.update((timestamp + ".").getBytes(StandardCharsets.UTF_8));
    String expected = "sha256=" + HexFormat.of().formatHex(mac.doFinal(responseBody));
    return signature != null && MessageDigest.isEqual(expected.getBytes(StandardCharsets.UTF_8),
signature.getBytes(StandardCharsets.UTF_8));
}
```

Go

```
import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/hex"
    "math"
    "net/http"
    "strconv"
    "time"
)

func VerifySa4s(r *http.Request, responseBody []byte, secret string) bool {
    ts := r.Header.Get("X-SA4S-Timestamp")
    unix, err := strconv.ParseInt(ts, 10, 64)
    if err != nil || math.Abs(float64(time.Now().Unix()-unix)) > 300 {
        return false
    }
    mac := hmac.New(sha256.New, []byte(secret))
    mac.Write([]byte(ts + "."))
    mac.Write(responseBody)
    expected := "sha256=" + hex.EncodeToString(mac.Sum(nil))
    return hmac.Equal([]byte(expected), []byte(r.Header.Get("X-SA4S-Signature")))
}
```

الحماية من إعادة الإرسال

الطابع الزمني جزء من المحتوى الموقع فلا يمكن تغييره. ارفض أي إرسال يبعد `X-SA4S-Timestamp` فيه أكثر من 5 دقائق عن ساعتك (كما في الأمثلة أعلاه)، واحفظ معرّفات `X-SA4S-Delivery` المعالجة (يكفي فهرس فريد): فلا يستطيع من التخطي إرسالاً أن يعيده لاحقاً ولا أن يجعلك تعالجه مرتين. وأبقى ساعة خادمك متزامنة (NTP).



إعادة المحاولة وإعادة الإرسال

أجب بأي حالة **2xx** خلال **10 ثوانٍ** — أكد الاستلام أولاً ثم نفذ العمل في الخلفية. ولا تتبّع إعادات التوجيه. وأي رد آخر (حالة خطأ، أو انتهاء مهلة، أو فشل اتصال) يُعاد بتراجع أُشّي — بعد نحو 30 ثانية، ثم 1 و 2 و 4 و 8 دقائق — بإجمالي **6 محاولات**، كل منها موقّعة من جديد بطابع زمني جديد وبمعرّف الإرسال نفسه. وبعد المحاولة الأخيرة يُعلّم الإرسال متوقّفاً. يعرض النظام كل إرسال بحالته ومحاولاته وآخر رد منك، ويتيح من **سجل الإرسال** زر **إعادة الإرسال** لأي منها. والعنوان الذي تتوقف إرسالته باستمرار (25 متتالية) يُعلّق تلقائياً؛ أصلحه ثم فُعله من النظام.

Node.js (Express)

```
import express from 'express';
import { verifySa4s } from './verify.mjs';

const app = express();
const seen = new Set(); // use your DB (unique index on the delivery id) in production

app.post('/sa4s/webhooks', express.raw({ type: 'application/json' })), (req, res) => {
  if (!verifySa4s(req.headers, req.body, process.env.SA4S_WEBHOOK_SECRET)) return res.sendStatus(401);
  const delivery = req.get('X-SA4S-Delivery');
  if (seen.has(delivery)) return res.sendStatus(200); // retries re-send the same delivery id
  seen.add(delivery);
  const event = JSON.parse(req.body.toString('utf8'));
  queue.push(event); // answer within 10 s - do the real work asynchronously
  res.sendStatus(204);
});

async function handle(event) {
  switch (event.event) {
    case 'invoice.issued': return onInvoice(event.data.invoiceId); // fetch GET /sales/invoices/{id} for
    full details
    case 'payment.recorded': return onPayment(event.data.paymentId, event.data.cancelled === true);
    case 'stock.changed': return onStock(event.data.productId, event.data.warehouseId, event.data.balance);
    case 'zatca.status': return onZatca(event.data.invoiceId, event.data.status);
    default: return; // ignore unknown events (new ones may be added)
  }
}

app.listen(3000);
```

Python (Flask)

```
import json, os
from flask import Flask, request

app = Flask(__name__)
SECRET = os.environ["SA4S_WEBHOOK_SECRET"]

@app.post("/sa4s/webhooks")
def sa4s_webhook():
    raw = request.get_data() # exact bytes
    if not verify_sa4s(request.headers, raw, SECRET):
        return "", 401
    delivery = request.headers["X-SA4S-Delivery"]
    if already_processed(delivery): # unique key in your DB
        return "", 200
    enqueue(json.loads(raw)) # process asynchronously, answer fast
    mark_processed(delivery)
    return "", 204
```

الاختبار محلياً

يجب أن يكون العنوان متاحاً عبر HTTPS عام. أثناء التطوير اكشف المستقبل المحلي عبر نفق HTTPS، واستخدم زر **إرسال تجريبي**، وأعد إرسال أجسام ملتقطة بتوقيع محسوب من جديد:



Shell

```
# 1) run your receiver locally
node receiver.mjs          # listens on http://localhost:3000/sa4s/webhooks

# 2) expose it over HTTPS with any tunnel you trust, e.g.
cloudflared tunnel --url http://localhost:3000
#   → https://<random>.trycloudflare.com (use https://<random>.trycloudflare.com/sa4s/webhooks as the endpoint URL)

# 3) ERP → Settings → API & Integrations → Webhooks → Send test: a signed "webhook.test" is delivered immediately

# 4) replay a captured body against your local receiver with a fresh signature (no SA4S needed)
TS=$(date +%s)
BODY='{ "id": "7a8dad24-9ac7-48d7-82d3-62b396f7bdf", "event": "webhook.test", "createdAt": "2026-10-06T16:59:16.516Z", "data": {} }'
SIG=$(printf '%s.%s' "$TS" "$BODY" | openssl dgst -sha256 -hmac "$SA4S_WEBHOOK_SECRET" -hex | sed 's/^.* //')
curl -X POST http://localhost:3000/sa4s/webhooks -H "Content-Type: application/json" \
  -H "X-SA4S-Event: webhook.test" -H "X-SA4S-Delivery: 7a8dad24-9ac7-48d7-82d3-62b396f7bdf" \
  -H "X-SA4S-Timestamp: $TS" -H "X-SA4S-Signature: sha256=$SIG" -d "$BODY"
```



مثال متكامل: طلبات المتجر الإلكتروني ← فواتير SA4S

مزامنة طلبات كاملة وغير متأثرة بالتكرار لمتجر من نوع WooCommerce أو سلة. شغلها من Webhook "تم الدفع" في المتجر أو من مهمة مجدولة على الطلبات المدفوعة الأخيرة — وتشغيلها مرتين للطلب نفسه لا ينشئ فاتورة ثانية أبداً.

Flow

```
store "order paid" event
  ↳ map every line SKU → SA4S productId      GET /products/lookup?code={sku}      (cached)
  ↳ B2B order? find / create the customer    GET /customers?search={vat} • POST /customers
  ↳ clientId = UUIDv5(store URL + order id) (same order → same key on every retry / duplicate event)
  ↳ draft invoice, server computes VAT       POST /sales/invoices      { pricesIncludeVat: true, clientId,
lines }
  ↳ compare draft.total with the store total (±0.05) — stop and flag the order when they differ
  ↳ issue + record the store payment         POST /sales/invoices/{id}/issue { payments: [{ method: "CARD", amount:
draft.total }] }
  ↳ save invoice number / PDF on the order   GET /sales/invoices/{id}/pdf
```

قرارات تصميم تجعلها متينة:

- **clientId** حتمي — UUID v5 من عنوان المتجر ورقم الطلب: إشعارات المتجر المكررة، والأعطال، والعمليات المتوازية كلها تصل إلى الفاتورة نفسها.
- **أسعار المتجر شاملة الضريبة** (pricesIncludeVat: true) وأسعار وحدات نهائية (بعد الخصم، فلا يحتاج المفتاح صلاحية sales.discount:manage)، والشحن سطر لمنتج من نوع SERVICE.
- **مسودة ← مقارنة ← إصدار** — اختلاف السعر أو الفئة الضريبية يوقف الطلب للمراجعة بدل إصدار فاتورة خاطئة.
- الطلبات بين المنشآت التي فيها رقم ضريبي تحصل على عميل وفاتورة ضريبية، وطلبات الأفراد على فاتورة مبسطة.

JavaScript

```
// order-sync.mjs — paid store orders → issued SA4S invoices. Idempotent: safe to re-run for the same order.
import { createHash } from 'node:crypto';
import { sa4s, Sa4SError } from './sa4s.mjs';

const SHOP = 'https://shop.example.com';
const SHIPPING_SKU = 'SRV-DELIVERY'; // a SERVICE product created once in SA4S for shipping fees
const URL_NAMESPACE = '6ba7b811-9dad-11d1-80b4-00c04fd430c8'; // RFC 4122

/** RFC 4122 version-5 UUID: deterministic, so the same order always maps to the same clientId */
export function uuidV5(name, namespace = URL_NAMESPACE) {
  const h = createHash('sha1').update(Buffer.from(namespace.replace(/-/g, ''), 'hex')).update(name, 'utf8').digest();
  h[6] = (h[6] & 0x0f) | 0x50;
  h[8] = (h[8] & 0x3f) | 0x80;
  const x = h.subarray(0, 16).toString('hex');
  return `${x.slice(0, 8)}-${x.slice(8, 12)}-${x.slice(12, 16)}-${x.slice(16, 20)}-${x.slice(20)}`;
}

const productIds = new Map();
async function productIdOf(sku) {
  if (!productIds.has(sku)) productIds.set(sku, (await sa4s('GET', `/products/lookup?code=${encodeURIComponent(sku)}`).product.id);
  return productIds.get(sku);
}

async function customerIdOf(billing) {
  if (!billing.vatNumber) return undefined; // B2C → simplified invoice without a customer
  const { items } = await sa4s('GET', `/customers?search=${billing.vatNumber}&pageSize=5`);
  const found = items.find((c) => c.vatNumber === billing.vatNumber);
  if (found) return found.id;
}
```



```

return (await sa4s('POST', '/customers', {
  type: 'COMPANY', name: billing.company, vatNumber: billing.vatNumber, phone: billing.phone, email: billing.email,
  city: billing.city,
})).id;
}

export async function syncOrder(order) {
  const clientId = uuidV5(`${SHOP}/orders/${order.id}`);
  const lines = [];
  for (const item of order.items) {
    // discounts need a permission API keys do not have: send the final (VAT-inclusive) unit price instead
    lines.push({ productId: await productIdOf(item.sku), quantity: String(item.quantity), unitPrice:
item.finalUnitPrice, description: item.name });
  }
  if (Number(order.shippingTotal) > 0) lines.push({ productId: await productIdOf(SHIPPING_SKU), quantity: '1',
unitPrice: order.shippingTotal });
  const customerId = await customerIdOf(order.billing);

  const draft = await sa4s('POST', '/sales/invoices', {
    type: customerId ? 'TAX' : 'SIMPLIFIED', customerId, pricesIncludeVat: true, clientId, notes: `Store order
#${order.number}`, lines,
  });
  if (draft.status !== 'DRAFT') return draft; // issued by an earlier run (idempotent replay)
  if (Math.abs(Number(draft.total) - Number(order.total)) > 0.05) {
    throw new Error(`order ${order.number}: SA4S total ${draft.total} ≠ store total ${order.total} – check prices / VAT
categories`);
  }
  try {
    return await sa4s('POST', `/sales/invoices/${draft.id}/issue`, { payments: [{ method: 'CARD', amount: draft.total,
reference: order.paymentId }] });
  } catch (e) {
    if (e instanceof Sa4sError && e.code === 'INVOICE_NOT_ISSUABLE') return sa4s('GET', `/sales/invoices/${draft.id}`);
  }
  // a parallel run won
  throw e;
}
}

```

Python

```

# order_sync.py – paid store orders → issued SA4S invoices (idempotent)
import uuid
from decimal import Decimal
from sa4s import sa4s, Sa4sError

SHOP = "https://shop.example.com"
SHIPPING_SKU = "SRV-DELIVERY"
_product_ids = {}

def product_id_of(sku):
    if sku not in _product_ids:
        _product_ids[sku] = sa4s("GET", "/products/lookup", params={"code": sku})["product"]["id"]
    return _product_ids[sku]

def customer_id_of(billing):
    vat = billing.get("vat_number")
    if not vat:
        return None
    found = [c for c in sa4s("GET", "/customers", params={"search": vat, "pageSize": 5})["items"] if c["vatNumber"] ==
vat]
    if found:
        return found[0]["id"]
    return sa4s("POST", "/customers", {"type": "COMPANY", "name": billing["company"], "vatNumber": vat, "phone":
billing["phone"]})["id"]

def sync_order(order):

```



```

client_uuid = str(uuid.uuid5(uuid.NAMESPACE_URL, f"{SHOP}/orders/{order['id']}"))
lines = [{"productId": product_id_of(i["sku"]), "quantity": str(i["quantity"]), "unitPrice": i["final_unit_price"]}
for i in order["items"]]
if Decimal(order["shipping_total"]) > 0:
    lines.append({"productId": product_id_of(SHIPPING_SKU), "quantity": "1", "unitPrice": order["shipping_total"]})
customer_id = customer_id_of(order["billing"])
body = {"type": "TAX" if customer_id else "SIMPLIFIED", "pricesIncludeVat": True, "clientId": client_uuid,
        "notes": f"Store order #{order['number']}", "lines": lines}
if customer_id:
    body["customerId"] = customer_id
draft = sa4s("POST", "/sales/invoices", body)
if draft["status"] != "DRAFT":
    return draft
if abs(Decimal(draft["total"]) - Decimal(order["total"])) > Decimal("0.05"):
    raise ValueError(f"order {order['number']}: SA4S {draft['total']} != store {order['total']}")
try:
    return sa4s("POST", f"/sales/invoices/{draft['id']}/issue",
               {"payments": [{"method": "CARD", "amount": draft["total"], "reference": order["payment_id"]}]}))
except Sa4sError as e:
    if e.code == "INVOICE_NOT_ISSUABLE":
        return sa4s("GET", f"/sales/invoices/{draft['id']}")
    raise

```

PHP

```

<?php
// order-sync.php — paid store orders → issued SA4S invoices (idempotent)
require 'sa4s.php';

function uuid_v5(string $name, string $ns = '6ba7b811-9dad-11d1-80b4-00c04fd430c8'): string
{
    $h = sha1(hex2bin(str_replace('-', '', $ns)) . $name);
    $h = substr($h, 0, 12) . '5' . substr($h, 13, 3) . dechex(hexdec($h[16]) & 0x3 | 0x8) . substr($h, 17, 15);
    return vsprintf('%s-%s-%s-%s-%s', [substr($h, 0, 8), substr($h, 8, 4), substr($h, 12, 4), substr($h, 16, 4),
    substr($h, 20, 12)]);
}

function sync_order(array $order): array
{
    static $productIds = [];
    $pid = function (string $sku) use (&$productIds) {
        return $productIds[$sku] ??= sa4s('GET', '/products/lookup?code=' . urlencode($sku))['product']['id'];
    };
    $lines = array_map(fn ($i) => ['productId' => $pid($i['sku']), 'quantity' => (string) $i['quantity'], 'unitPrice'
=> $i['final_unit_price']], $order['items']);
    if ((float) $order['shipping_total'] > 0) {
        $lines[] = ['productId' => $pid('SRV-DELIVERY'), 'quantity' => '1', 'unitPrice' => $order['shipping_total']];
    }
    $draft = sa4s('POST', '/sales/invoices', [
        'type' => 'SIMPLIFIED',
        'pricesIncludeVat' => true,
        'clientId' => uuid_v5("https://shop.example.com/orders/{$order['id']}"),
        'notes' => "Store order #{order['number']}",
        'lines' => $lines,
    ]);
    if ($draft['status'] !== 'DRAFT') return $draft;
    if (abs((float) $draft['total'] - (float) $order['total']) > 0.05) {
        throw new RuntimeException("order {$order['number']}: SA4S {$draft['total']} != store {$order['total']}");
    }
    try {
        return sa4s('POST', "/sales/invoices/{$draft['id']}/issue", ['payments' => [['method' => 'CARD', 'amount' =>
$draft['total'], 'reference' => $order['payment_id']]]]);
    } catch (Sa4sError $e) {

```



```
if (($e->body['code'] ?? '') === 'INVOICE_NOT_ISSUABLE') return sa4s('GET', "/sales/invoices/${draft['id']}");  
throw $e;  
}  
}
```

بعد ذلك احفظ رقم الفاتورة `number` في الطلب، وقدم ملف PDF (`GET /sales/invoices/{id}/pdf`) للعميل بوصفه فاتورته الضريبية. وللإسترجاع استخدم إشعاراً دائماً بـ `clientId` حتمي خاص به (مثل v5 UUID لرقم عملية الإسترجاع).



مثال متكامل: مزامنة المخزون

انشر مخزون SA4S في المتجر على ثلاث طبقات: لقطة كاملة، وWebhooks لحظية، ومطابقة دورية تصلح ما قد يتركه حدث فائت أو متأخر.

Flow

- A. first run / nightly GET /inventory/balances?warehouseId={wh}&pageSize=200 (all pages) → set every store quantity by SKU
- B. real time webhook stock.changed { productId, warehouseId, balance.onHand, balance.reserved }
→ available = onHand - reserved → update that one SKU
- C. safety net (15 min) GET /inventory/movements?from={yesterday}&order=asc → products you have not seen
→ GET /inventory/products/{productId}/stock → update those SKUs

JavaScript

```
// stock-sync.mjs — keep store quantities equal to the SA4S "available" stock of one warehouse
import { sa4s } from './sa4s.mjs';

const WAREHOUSE_ID = process.env.SA4S_WAREHOUSE_ID;
const skuCache = new Map();
const skuOf = async (productId) => {
  if (!skuCache.has(productId)) skuCache.set(productId, (await sa4s('GET', `/products/${productId}`).sku);
  return skuCache.get(productId);
};
const toStoreQty = (decimal) => Math.max(0, Math.floor(Number(decimal))); // base units, never negative

// A) full sync
export async function fullSync(store) {
  for (let page = 1; ; page++) {
    const res = await sa4s('GET', `/inventory/balances?warehouseId=${WAREHOUSE_ID}&page=${page}&pageSize=200&sort=sku&order=asc`);
    for (const row of res.items) await store.setQuantity(row.product.sku, toStoreQty(row.available));
    if (page * res.pageSize >= res.total) break;
  }
}

// B) webhook handler for stock.changed (see "Webhooks")
export async function onStockChanged(store, { productId, warehouseId, balance }) {
  if (warehouseId !== WAREHOUSE_ID || !balance) return;
  await store.setQuantity(await skuOf(productId), toStoreQty(Number(balance.onHand) - Number(balance.reserved)));
}

// C) reconciliation — catches anything a missed / delayed webhook did not deliver
export async function reconcile(store, seen = new Set()) {
  const yesterday = new Intl.DateTimeFormat('en-CA', { timeZone: 'Asia/Riyadh' }).format(new Date(Date.now() - 86_400_000));
  const touched = new Set();
  for (let page = 1; ; page++) {
    const res = await sa4s('GET', `/inventory/movements?from=${yesterday}&warehouseId=${WAREHOUSE_ID}&order=asc&page=${page}&pageSize=200`);
    for (const m of res.items) if (!seen.has(m.id)) { seen.add(m.id); touched.add(m.productId); }
    if (page * res.pageSize >= res.total) break;
  }
  for (const productId of touched) {
    const s = await sa4s('GET', `/inventory/products/${productId}/stock`);
    const row = s.warehouses.find((w) => w.warehouseId === WAREHOUSE_ID);
    await store.setQuantity(s.product.sku, toStoreQty(row?.available ?? '0'));
  }
}
```



Python

```
# stock_sync.py
import math, os
from datetime import datetime, timedelta
from zoneinfo import ZoneInfo
from sa4s import sa4s

WAREHOUSE_ID = os.environ["SA4S_WAREHOUSE_ID"]
to_store_qty = lambda d: max(0, math.floor(float(d)))

def full_sync(store):
    page = 1
    while True:
        res = sa4s("GET", "/inventory/balances", params={"warehouseId": WAREHOUSE_ID, "page": page, "pageSize": 200,
"sort": "sku", "order": "asc"})
        for row in res["items"]:
            store.set_quantity(row["product"]["sku"], to_store_qty(row["available"]))
        if page * res["pageSize"] >= res["total"]:
            return
        page += 1

def on_stock_changed(store, data):
    # webhook "stock.changed"
    if data["warehouseId"] != WAREHOUSE_ID or "balance" not in data:
        return
    sku = sa4s("GET", f"/products/{data['productId']}")["sku"]
    store.set_quantity(sku, to_store_qty(float(data["balance"]["onHand"]) - float(data["balance"]["reserved"])))

def reconcile(store, seen):
    yesterday = (datetime.now(ZoneInfo("Asia/Riyadh")) - timedelta(days=1)).strftime("%Y-%m-%d")
    touched, page = set(), 1
    while True:
        res = sa4s("GET", "/inventory/movements", params={"from": yesterday, "warehouseId": WAREHOUSE_ID, "order":
"asc", "page": page, "pageSize": 200})
        for m in res["items"]:
            if m["id"] not in seen:
                seen.add(m["id"])
                touched.add(m["productId"])
        if page * res["pageSize"] >= res["total"]:
            break
        page += 1
    for product_id in touched:
        s = sa4s("GET", f"/inventory/products/{product_id}/stock")
        row = next((w for w in s["warehouses"] if w["warehouseId"] == WAREHOUSE_ID), {"available": "0"})
        store.set_quantity(s["product"]["sku"], to_store_qty(row["available"]))
```

PHP

```
<?php
// stock-sync.php - webhook part (A and C work like the JavaScript tab)
require 'sa4s.php';

$raw = file_get_contents('php://input');
if (!verify_sa4s($raw, getenv('SA4S_WEBHOOK_SECRET'))) { http_response_code(401); exit; }
$event = json_decode($raw, true);
if ($event['event'] === 'stock.changed' && $event['data']['warehouseId'] === getenv('SA4S_WAREHOUSE_ID')) {
    $d = $event['data'];
    $sku = sa4s('GET', "/products/{${d['productId']}}")['sku'];
    $qty = max(0, (int) floor((float) $d['balance']['onHand'] - (float) $d['balance']['reserved']));
    store_set_quantity($sku, $qty); // your store's API
}
http_response_code(204);
```

**ملاحظة**

الكميات بالوحدة الأساسية. فإن كان المتجر يبيع بالكرتون فاقسم على معامل الوحدة `factor` من بيانات المنتج. واستخدم `available` (المتوفر ناقص المحجوز) لا `onHand` لمعرفة الكمية القابلة للبيع.



سجل التغييرات والإصدارات

الإصدار	التاريخ	التغييرات
1.1.0	2026-10-06	إعادة كتابة الدليل من مصدر واحد للنظام وملف PDF؛ وصفات لكل وحدة باستجابات حقيقية؛ حمولة كل حدث Webhook؛ التحقق من التوقيع بست لغات؛ أمثلة متكاملة للمتاجر؛ تصحيح المرشحات (تستخدم التقارير <code>dateTo</code> / <code>dateFrom</code> ، و PDF يقبل <code>format=a4 receipt</code>).
1.0.0	2026-10	إطلاق الواجهة البرمجية: مفاتيح API (فعلي/تجربة، صلاحيات، تدوير، قائمة عناوين IP، حدود الطلبات)، وال Webhooks، ومرجع OpenAPI.

يبقى إصدار الواجهة **v1** ما دامت التغييرات إضافية (انظر العنوان الأساسي والإصدارات). ووثيقة OpenAPI الحالية متاحة دائماً على `GET` <https://erp.sa4s.com/api/v1/developer/openapi.json>.



الدعم

- ابدأ برمز الخطأ `code` وجدول رموز الأخطاء؛ فأغلب المشكلات صلاحية ناقصة أو رسالة تحقق.
- عند التواصل مع الدعم أرفق `requestId` والمسار والوقت (مع المنطقة الزمنية) و**بإدئة** المفتاح - ولا ترسل المفتاح السري كاملاً أبداً.
- قناة الدعم: بيانات التواصل التي تنشرها SA4S لاشتراكك (اسأل مالك الحساب لديكم).

يُولد هذا الدليل من المصدر نفسه الذي يعرضه النظام. المرجع التفاعلي متاح في الإعدادات - التكامل البرمجي (API).

<https://erp.sa4s.com/api/v1/developer/openapi.json>