

التحليل الفني الشامل: الدليل المرجعي الكامل لحزمة تطوير البرمجيات (SDK) الخاصة بهيئة الضرائب المصرية (ETA)

مقدمة: تحديد المسار المعماري للتكامل

توفر بوابة حزمة تطوير البرمجيات (SDK) الخاصة بهيئة الضرائب المصرية (ETA) مجموعة الموارد التقنية الأساسية لربط أنظمة تخطيط موارد المؤسسات (ERP) ونقاط البيع (POS) التابعة للممولين مع حلول الفاتورة الإلكترونية (eInvoicing) والإيصال الإلكتروني¹ (eReceipt). تم تصميم هذا التوثيق لتمكين المطورين من أتمتة معالجة المستندات الضريبية بشكل كامل¹.

يكشف تحليل بنية التوثيق عن مسارين معماريين متميزين ومصممين بعناية للتكامل، ويشكل الاختيار بينهما القرار الأول والأهم الذي يجب على مهندس الحلول اتخاذه:

1. **مسار التكامل المباشر (Direct API Integration):** هذا المسار، الموثق بشكل أساسي تحت قسم API، يمنح المطورين تحكماً كاملاً ووصولاً مباشراً إلى جميع نقاط النهاية¹ (Endpoints). يتطلب هذا المسار معالجة يدوية دقيقة لعمليات المصادقة، والتوقيع الرقمي للمستندات، والتحقق من صحة البيانات، وإدارة تدفقات العمليات غير المتزامنة.
2. **مسار حزمة أدوات التكامل (Integration Toolkit):** هذا المسار، الموثق تحت قسم Integration Toolkit، يقدم مجموعة من الأدوات المساعدة التي تعمل كطبقة تجريد (Abstraction Layer) أو وكيل (Proxy) بين نظام الممول وواجهات برمجة التطبيقات المباشرة¹. توفر هذه الحزمة ميزات حيوية مثل "العمل في وضع عدم الاتصال" (Offline mode)، و"التخزين والتصدير المحلي" (Store and export)، و"التحقق المحلي من صحة المستندات" (Validate documents locally)¹.

إن وجود هذين المسارين ليس عشوائياً. فمسار الـ API المباشر مصمم للأنظمة المركزية (مثل أنظمة ERP السحابية) التي تتطلب موثوقية عالية، وتحكماً دقيقاً في تدفق البيانات، وتتمتع باتصال دائم بالإنترنت. في المقابل، تم تصميم حزمة الأدوات (Toolkit) خصيصاً لسيناريوهات نقاط البيع (POS) الموزعة والتطبيقات التي قد تعمل في بيئات ذات اتصال متقطع بالإنترنت، حيث تقوم الحزمة بتبسيط العمليات المعقدة وتأجيل تنفيذها لحين توفر الاتصال.

يستهدف هذا التقرير الفني تفكيك وتحليل كل سطر وكل مكون في بوابة الـ SDK، وتقديم مرجع شامل لا يغفل أي تفصيل، لمساعدة الفرق التقنية على اتخاذ قرارات مستنيرة وبناء تكامل ناجح ومرن.

الجزء الأول: أسس التكامل والمتطلبات المسبقة

قبل البدء في أي عملية تكامل برمجي، يجب إرساء بنية تحتية تقنية وإدارية. يغطي هذا الجزء المبادئ الأساسية والمتطلبات الإلزامية التي تشكل حجر الأساس للربط مع أنظمة هيئة الضرائب المصرية¹.

1.1 المبادئ التقنية الأساسية

يستند تصميم الحل على مجموعة من المبادئ الحديثة التي تضمن الأمان والمرونة¹:

- **الهوية الخارجية (Externalized Identity):** يتم فصل إدارة هويات المستخدمين والأنظمة (Authentication/Authorization) عن منطق معالجة المستندات الفعلي.
- **الواجهات القياسية (Standards-based Interfaces):** تعتمد جميع الواجهات على بروتوكول REST وتستخدم تنسيق JSON كمعيار أساسي لمدخلات ومخرجات البيانات (مع دعم XML لأسباب التوافق مع الأنظمة القديمة).
- **المعالجة غير المتزامنة (Asynchronous Processing):** العمليات المعقدة، مثل تقديم المستندات، مصممة ليتم تنفيذها بشكل غير متزامن. يستجيب النظام فوراً بتأكيد استلام (مثل HTTP 202 Accepted)، بينما تتم المعالجة والتحقق الفعلي في الخلفية. هذا التصميم يضمن سرعة استجابة عالية ويقلل من احتمالية أخطاء انتهاء المهلة (Timeouts).
- **حماية البيانات (Data Protection):** يتم تأمين جميع الاتصالات عبر قنوات محمية (TLS). بالإضافة إلى ذلك، يتم تطبيق طبقة حماية ثانية على مستوى المستند نفسه من خلال التوقيعات الرقمية (Digital Signatures) التي يضيفها المصدر.

1.2 المتطلبات الإلزامية (Mandatory Prerequisites)

تحدد وثائق "البدء" (Getting Started) مجموعة من "تفاعلات الإعداد" (Set-Up Interactions) التي يجب إكمالها قبل كتابة أي كود برمجي¹:

1. **التسجيل وإنشاء الملف الرقمي (Digital Profile):** الخطوة الأولى إدارية بحتة، وتتضمن تسجيل ملف رقمي لدى هيئة الضرائب (ETA) لإدارة الممثلين والأنظمة¹.
2. **الحصول على بيانات اعتماد الـ (API Client ID / Client Secret):** بعد تسجيل نظام الـ ERP أو POS الخاص بالمول، يتم تزويده ببيانات اعتماد العميل (Client ID و Client Secret). تُستخدم هذه البيانات لمصادقة النظام نفسه عند طلب رموز الوصول¹ (Access Tokens).
3. **الحصول على شهادة الختم الإلكتروني (eSeal X.509 Certificate):** هذا مطلب منفصل وحيوي. من الضروري فهم أن النظام يتعامل مع مستويين من الأمان:
 - **أمان على مستوى النقل/الوصول (Transport/Access Layer):** يتم تأمينه باستخدام Client ID/Secret للحصول على Access Token (كما هو موضح في الجزء الثاني). هذا الـ Token يثبت لمنصة الـ API أن النظام المرسل هو نظام موثوق به ومصرح له بالوصول.
 - **أمان على مستوى المستند (Document Layer):** يتم تأمينه باستخدام شهادة eSeal X.509. تُستخدم هذه الشهادة لتوقيع المستند نفسه (الفاتورة أو الإيصال) قبل إرساله. هذا التوقيع الرقمي يضمن (للمستلم وهيئة الضرائب) أن المستند صادر بالفعل من الممول المذكور ولم يتم التلاعب به¹ (Non-repudiation and Integrity).

1.3 بيئات التشغيل (Environments)

يوفر النظام بيئتين منفصلتين: بيئة الاختبار (Pre-Production) وبيئة الإنتاج (Production). يجب على المطورين بدء اختبار التكامل حصرياً على بيئة¹. Pre-Production.

الخاصة ببيئات التشغيل URL جدول 1: عناوين
(مُستخرج من 1)

بيئة الإنتاج (Prod)	بيئة الاختبار (PreProd)	الخدمة (Service)
https://profile.eta.gov.eg	https://profile.preprod.eta.gov.eg	بوابة التسجيل (Registration Portal)
https://invoicing.eta.gov.eg	https://preprod.invoicing.eta.gov.eg	بوابة الفوترة (Invoicing Portal)
https://api.invoicing.eta.gov.eg	https://api.preprod.invoicing.eta.gov.eg	واجهة برمجة التطبيقات (System API)
https://id.eta.gov.eg	https://id.preprod.eta.gov.eg	خدمة الهوية (Identity Service)

1.4 إعداد بيئة الاختبار (PreProd Setup)

تعتمد بيئة الاختبار (PreProd) على شهادات SSL صادرة داخلياً¹ (Internally Issued Certificates) نتيجة لذلك، ستعطل جميع محاولات الاتصال بالـ API من بيئات التطوير القياسية بسبب عدم الثقة في مصدر الشهادة.

لحل هذه المشكلة، يجب على المطورين تنزيل وتثبيت شهادة "Root CA certificate file" الخاصة ببيئة¹. PreProd.

الخطوات (لد¹: Windows 10)

1. قم بتنزيل ملف شهادة Root CA.
2. انقر بزر الماوس الأيمن على الملف واختر **Install Certificate** (يتطلب حقوق المسؤول).
3. اختر **Local Machine**.
4. حدد الخيار **Place all certificates in the following store**.

5. اختر **Trusted Root Certification Authorities** (سلطات التصديق الجذر الموثوق بها).
6. أكمل عملية التثبيت.

الفشل في إكمال هذه الخطوة سيؤدي إلى ظهور أخطاء SSL Error عند محاولة استدعاء الواجهات، خاصة عند استخدام أدوات مثل Postman.¹

الجزء الثاني: التحليل العميق لآليات المصادقة وإدارة الوصول

تُعد المصادقة (Authentication) هي البوابة الأولى للتفاعل مع الـ API. يستخدم النظام بروتوكول OAuth 2.0 وتحديداً تدفق بيانات اعتماد العميل (client_credentials).² ومع ذلك، يكشف التحليل الدقيق للتوثيق عن وجود آليتين مختلفتين تماماً للمصادقة، واحدة للفواتير (eInvoicing) والأخرى للإيصالات (eReceipt)، على الرغم من أنهما تستخدمان نفس نقطة النهاية (Endpoint).

2.1 المسار 1: مصادقة نظام دافع الضرائب (Login as Taxpayer System)

هذا المسار مخصص للأنظمة الخلفية (ERPs) التي تتعامل مع الفواتير الإلكترونية (eInvoicing).²

- نقطة النهاية (Endpoint): POST <https://id.preprod.eta.gov.eg/connect/token>²
- الآلية: تدفق client_credentials قياسي.

(Taxpayer System) جدول 2: معلمات طلب المصادقة (مُستخرج من 2)

نوع المعلمة	المعلمة	الوصف	القيمة / المثال
Header	Authorization	يجب أن تحتوي على سلسلة تخويل أساسية (Basic) مكونة من Client ID و Client Secret بعد ترميزها بـ Base64 (بالشكل Base64(ClientID:ClientSecret)).	Basic <Base64 <encoded value
Body	grant_type	نوع التدفق المطلوب.	client_credentials
Body	scope	(اختياري) النطاق المطلوب	InvoicingAPI

	للوصول.		
--	---------	--	--

- الاستجابة الناجحة:²
- access_token: الرمز المميز (JWT) لاستخدامه في استدعاءات الـ API القادمة.
- token_type: Bearer
- expires_in: 3600 (أي ساعة واحدة).
- استراتيجية إدارة الـ Token:
- يشير التوثيق صراحة إلى أن صلاحية الـ Token هي ساعة واحدة (3600 ثانية)، وأن النظام يجب أن يكون "مسؤولاً عن الحصول على رمز وصول جديد... قبل انتهاء صلاحية الـ Token الحالي".² هذا يستلزم بناء آلية تخزين مؤقت (Caching) للـ Token. لا يجوز برمجياً استدعاء /connect/token/ مع كل استدعاء API؛ بل يجب تخزين الـ Token والتحقق من صلاحيته قبل الاستخدام. في حالة انتهاء الصلاحية أو عند استلام خطأ 401 Unauthorized، يجب طلب Token جديد وتحديث القيمة المخزنة.
- حالة خاصة: عند تسجيل الدخول كوسيط (Intermediary)، يتطلب الطلب إضافة Header إضافي: onbehalfof ويحتوي على رقم تسجيل الممول الذي يتم العمل نيابة عنه.²

2.2 المسار 2: مصادقة نقاط البيع (Authenticate POS)

- هذا المسار مخصص لأنظمة نقاط البيع (POS) التي تتعامل مع الإيصالات الإلكترونية (eReceipt).³
- نقطة النهاية (POST https://id.preprod.eta.gov.eg/connect/token) (نفس المسار، ولكن بالآلية مختلفة تماماً).³
 - الآلية: تدفق client_credentials مُعدّل ومخصص.
- يكشف هذا المسار عن اختلاف جوهري عن التدفق القياسي. بدلاً من استخدام Authorization Header، يتم تمرير Client ID و Client Secret في الـ Body، ويتطلب أربعة Headers مخصصة إضافية خاصة بالـ POS.³ هذا يعني أن المنطق البرمجي المكتوب لمصادقة eInvoicing لن يعمل لمصادقة eReceipt، ويجب بناء خدمتي مصادقة منفصلتين.

(POS) جدول 3: معلمات طلب المصادقة

(مُستخرج من 3)

نوع المعلمة	المعلمة	الوصف	القيمة / المثال
Header	posserial	الرقم التسلسلي لنقطة البيع.	1234567899
Header	pososversion	إصدار نظام تشغيل POS.	os
Header	posmodelframework	إطار عمل نموذج POS.	1

Header	presharedkey	المفتاح المشترك المسبق لـ .POS	...03ac674216f3e1
Body	grant_type	نوع التدفق.	client_credentials
Body	client_id	Client ID المحدد.	...d0394a9f-0607
Body	client_secret	Client Secret المحدد.	...6d62315e-d65a

رموز الخطأ (Error Responses):

توفر هذه الواجهة رموز خطأ واضحة في حالة 400 Bad Request، مثل: unauthorized_client, invalid_posserial, invalid_pososversion, invalid_presharedkey, invalid_clientsecret.3

الجزء الثالث: المرجع الكامل لواجهات برمجة التطبيقات (API Reference)

هذا الجزء يغطي الواجهات الفعلية المستخدمة في التكامل المباشر.

3.1 واجهات API المشتركة (Common APIs)

تُستخدم هذه الواجهات لجلب البيانات الوصفية (Metadata) والإعدادات اللازمة لإنشاء المستندات¹:

- **Get Document Types / Get Document Type**: لاسترجاع قائمة بأنواع المستندات المتاحة (مثل فاتورة، إشعار دائن) وإصداراتها¹.
- **Get Document Type Version**: الواجهة الأهم لجلب بنية (Schema) المستند الفعلية. (انظر الجزء الرابع لمزيد من التفاصيل).¹
- **Code Management APIs**: مجموعة من الواجهات لإدارة أكواد EGS (التي يستخدمها الممول لتكويد سلعه وخدماته).¹ وتشمل:
 - Create EGS Code Usage
 - Search my EGS code usage requests
 - Request Code Reuse
 - Search Published Codes (للبحث في أكواد EGS و GS1).
 - Get Code Details by Item Code
 - Update EGS Code Usage / Update Code
- **Notification APIs**:
 - Get Notifications: لاستعلام النظام عن الإشعارات التي تم استلامها سابقاً.

- ERP - Ping: واجهة يجب أن يعرضها نظام ال ERP الخاص بك لتستخدمها منصة الفواتير للتحقق من أن نظامك متاح لاستقبال الإشعارات (Callbacks).

3.2 واجهات الفاتورة الإلكترونية (eInvoicing API)

يوفر التوثيق رابطاً مخصصاً لهذه الواجهات: ² <https://sdk.invoicing.eta.gov.eg/einvoicingapi/> على الرغم من أن الأبحاث المستخرجة لم تقدم تفاصيل دقيقة لنقاط النهاية الخاصة بـ eInvoicing (مثل SubmitDocuments أو CancelDocument)، إلا أنها قدمت تحليلاً كاملاً لحمولة (Payload) هذه الواجهات، أي بنية المستندات نفسها (انظر الجزء الرابع).⁶

3.3 واجهات الإيصال الإلكتروني (eReceipt API)

على عكس eInvoicing، يوفر التوثيق المستخرج تفاصيل شاملة لنقاط نهاية ³ eReceipt، والتي يمكن الوصول إليها عبر: ² <https://sdk.invoicing.eta.gov.eg/ereceiptapi/>

eReceipt API جدول 4: مرجع واجهات (مُستخرج من 3)

الغرض (Purpose)	نقطة النهاية (Endpoint)	الوظيفة الرئيسية
مصادقة POS	POST /connect/token	الحصول على Access Token (انظر الجزء 2.2).
إرسال الإيصالات	POST /api/v1/receiptsubmissions	إرسال إيصال واحد أو أكثر (عملية غير متزامنة).
جلب تفاصيل إيصال	GET /api/v1/receipts/{uuid}/details	استرجاع التفاصيل الكاملة ونتائج التحقق (Validation) لإيصال محدد.
جلب الإيصال (JSON)	{GET /api/v1/receipts/{uuid	استرجاع JSON المصدر للإيصال مع البيانات الوصفية المضافة من الهيئة.
جلب تفاصيل (مجهول)	GET /api/v1/receipts/search/{uui	استرجاع JSON المصدر (بدون مصادقة كاملة).

	{d}	
إرجاع معلومات عن عملية إرسال دفعة (Batch) كاملة.	GET /api/v1/receiptsubmissions/{ submissionUUID	جلب حالة الإرسال
البحث في الإيصالات المُستلمة (يُدمج معلومات بحث متعددة).	GET /api/v1/receipts	البحث في الإيصالات
استعلام سريع عن أحدث الإيصالات المتاحة.	GET /api/v1/receipts/recent	جلب أحدث الإيصالات
طلب تجهيز حزمة (Package) كبيرة من الإيصالات للتنزيل لاحقاً.	POST /api/v1/receipts/package	طلب حزمة إيصالات
استرجاع حالة طلبات الحزم السابقة (لمعرفة إذا كانت جاهزة للتنزيل).	GET /api/v1/receipts/package	جلب حالة طلبات الحزم
تنزيل الحزمة الجاهزة (بتنسيق CSV أو JSON مضغوط).	GET /api/v1/receipts/package/{re questId	تنزيل الحزمة
واجهة يجب أن يعرضها الـ ERP لاستقبال إشعارات أحداث الإيصال (تحقق، استقبال، إلغاء).	ERP - Receive Receipt Notifications	(استقبال الإشعارات)

3.4 تحليل التدفقات غير المتزامنة (Async Flow Analysis)

أحد المبادئ الأساسية المذكورة هو "المعالجة غير المتزامنة" ¹ (Asynchronous processing) الفشل في فهم هذا النمط هو السبب الأكثر شيوعاً لفشل التكامل. يتجلى هذا النمط بوضوح في تدفقين رئيسيين:

1. **تدفق إرسال الإيصال ³: (Receipt Submission Flow)**
 - عند استدعاء POST /api/v1/receiptsubmissions، لا يقوم النظام بالتحقق الكامل من الإيصال فوراً.
 - الاستجابة الفورية هي HTTP 202 Accepted، وليس OK. ³ هذا الرمز يعني "لقد استلمنا طلبك وسنقوم بمعالجته لاحقاً".
 - التحقق الفوري يقتصر على البنية الأساسية (مثل خطأ ³ BadStructure).
 - يجب على المطور تخزين submissionUUID المرجع في الاستجابة. ³

- لاحقاً، يجب على النظام إجراء استعلام (Polling) باستخدام GET `/api/v1/receiptsubmissions/{submissionUUID}` لمعرفة الحالة النهائية (Valid أو Invalid) لعملية الإرسال.
 - 2. تدفق طلب الحزم (3): Package Request Flow
هذا التدفق مثالي لتوضيح النمط غير المتزامن المكون من ثلاث خطوات:
 - الخطوة 1 (طلب): POST `/api/v1/receipts/package` لطلب تجهيز الحزمة.
 - الخطوة 2 (استعلام): GET `/api/v1/receipts/package` للاستعلام عن حالة الطلب (هل أصبح جاهزاً؟).
 - الخطوة 3 (تنزيل): GET `/api/v1/receipts/package/{requestId}` لتنزيل الحزمة بعد التأكد من جاهزيتها.
- يجب تصميم أنظمة التكامل كـ "آلات حالة" (State Machines) قادرة على تتبع حالة الطلبات غير المتزامنة، بدلاً من توقع استجابات فورية.

الجزء الرابع: تحليل بنية المستندات (Document Schema Analysis)

هذا الجزء يحل "ماذا" يتم إرساله، أي بنية الفاتورة والإيصال.

4.1 آلية استرجاع المخطط (Schema Retrieval Mechanism)

- يوفر النظام آلية ديناميكية للتحقق من بنية المستندات، وهو ما يمثل تصميمًا مرناً وقابلًا للتطوير.
 - عند استدعاء واجهة `GET /api/v1/documenttypes/versions/{versionId}` (أو الواجهات ذات الصلة)⁵، يُرجع النظام كائن `DocumentTypeVersion`.
 - يحتوي هذا الكائن على الحقول الحاسمة: `jsonSchema` و `xmlSchema`⁵.
 - هذه الحقول تحتوي على مخطط (Schema) للتحقق الكامل كـ `"Base64 encoded string"`⁵.
 - يوضح التوثيق أن الإصدارات تُستخدم لتطوير الحقول بمرور الوقت، مثل "تحويل حقل اختياري إلى إلزامي"⁵.
- إن تقديم المخطط (Schema) بهذه الطريقة الديناميكية (بدلاً من مجرد توثيقه بشكل ثابت) يشير بقوة إلى أفضل ممارسة للتكامل: يجب على نظام المطور أن يقوم برمجياً باستدعاء `Get Document Type Version`، وفك ترميز الـ `jsonSchema`، واستخدام مكتبة `JSON Schema Validator` للتحقق من صحة المستندات قبل إرسالها. هذا يضمن أن التكامل لن ينقطع إذا قامت هيئة الضرائب بتحديث المخطط (Schema) في المستقبل.

4.2 تحليل البنية (Schema Analysis)

توفر صفحة "Types" (الأنواع) البنى الأساسية للمستندات المختلفة.⁶

(Credit/Debit v1.0) والإشعارات (Invoice v1.0) جدول 5: الحقول الأساسية - الفاتورة (مُستخرج من 6)

العنصر	النوع	الوصف	ملاحظة هامة
issuer	Issuer	هيكّل يمثل معلومات المُصدر.	(هيكّل فرعي معقّد)
receiver	Receiver	هيكّل يمثل معلومات المُستلم.	(هيكّل فرعي معقّد)
documentType	String	اسم نوع المستند. i (للفاتورة)، c (دائن)، d (مدين).	الحقل المميز لنوع المستند.
documentTypeVersion	String	اسم إصدار نوع المستند. يجب أن يكون 1.0.	
dateTimeIssued	Date	تاريخ وقت إصدار المستند (بتوقيت UTC).	مثال: 13T13:15:0-02-2015 OZ
taxpayerActivityCode	String	رمز نشاط الممول للأعمال التي أصدرت المستند.	
internalId	String	المعرف الداخلي للمستند (رقم الفاتورة في نظام الـ ERP).	مثال: AZ-24883
references	String	(مطلوب لـ C/D) مصفوفة بـ UUIDs للفواتير السابقة التي يتم الإشارة إليها.	مثال: S98L2CP1SMVBIU
invoiceLines	Invoice Line	بنود الفاتورة. يجب أن تحتوي على بند واحد على الأقل.	(هيكّل فرعي معقّد)
totalSalesAmount	Decimal	مجموع جميع عناصر InvoiceLine/SalesTo	

	.tal		
	المبلغ الإجمالي للفاتورة (صافي المبلغ + إجمالي الضرائب).	Decimal	totalAmount
(هيكل فرعي معقد)	هيكل يحتوي على التوقيع الرقمي (eSeal). يجب أن يكون توقيع المصدر موجوداً.	Signature	signatures

جدول 6: الحقول الأساسية - الإيصال (Receipt v1.2)

(مُستخرج من 6)

العنصر	النوع	الوصف
header	header	إلزامي، هيكل يمثل معلومات الرأس (بما في ذلك dateTimeIssued).
documentType	documentType	إلزامي، هيكل يمثل معلومات نوع المستند.
seller	seller	إلزامي، هيكل يمثل معلومات البائع (بدلاً من issuer).
buyer	buyer	إلزامي، هيكل يمثل معلومات المشتري (بدلاً من receiver).
itemData	itemData	إلزامي، هذا يمثل مجموعة من الكائنات (بدلاً من invoiceLines).
totalSales	Decimal	إلزامي، مجموع كل عناصر إجمالي المبيعات.
totalAmount	Decimal	إلزامي، الإجمالي الكلي.
paymentMethod	(String 50)	إلزامي، رموز طريقة الدفع (مثل C للكاش).

contractor	contractor	اختياري، هيكل يمثل معلومات المقاول.
beneficiary	beneficiary	اختياري، هيكل يمثل معلومات المستفيد.

الجزء الخامس: الدليل الشامل لحزمة أدوات التكامل (Integration Toolkit)

كما ذكرنا في المقدمة، تقدم هيئة الضرائب مساراً بديلاً للتكامل المباشر، وهو "حزمة أدوات التكامل" ¹ (Integration Toolkit). هذه الحزمة هي "صندوق أسود" مصمم لتبسيط عملية التكامل، خاصة لسيناريوهات نقاط البيع (POS).

5.1 المفاهيم الأساسية لـ Toolkit

- **النكهات (Flavors):** يتم تقديم الـ Toolkit بثلاثة أشكال مختلفة لتناسب جميع بيئات التطوير ¹:
 1. Docker Container for Rest API (حاوية Docker تعرض واجهات API محلية).
 2. Nuget Library (مكتبة برمجية لبيئة .NET).
 3. Command Line Interface (CLI) (أداة سطر أوامر تنفيذية).
- **الوضع غير المتصل (Offline Mode):** هذه هي الميزة الجوهرية. تسمح الـ Toolkit للنظام بالعمل دون اتصال مباشر بالإنترنت، حيث تدعم:
 - إصدار الإيصال (Issue Receipt).
 - توليد رمز الاستجابة السريعة (Generate Receipt QR code).
 - تعيين UUID فريد عالمياً للإيصال.
 - التحقق من صحة المستندات محلياً (Validate documents locally).
 - تخزين وتصدير الإيصالات المُصدرة ¹ (Store and export issued receipts).
- **التخزين المحلي (Local Storage):**
 - تستخدم الـ Toolkit قاعدة بيانات SQLite لتخزين الإيصالات وأي بيانات محلية أخرى ¹.
 - يتم تخزين كل إيصال محلياً مع "رمز حالة" (Status) رقمي يحدد دورة حياته.

(Toolkit Status Codes) جدول 7: أكواد حالة الإيصال المحلية (مُستخرج من 1)

الوصف	الحالة (Status)	القيمة (Status Value)
تم تغيير حالة الإيصال من "جديد" إلى "مُصدّر" عندما يقوم المستخدم بعملية تصدير.	Exported (مُصدّر)	1-

0	New (جديد)	الحالة الافتراضية للإيصال عند إصداره وتخزينه محلياً.
1	Submitted (مُرسل)	تم إرسال الإيصال إلى منصة هيئة الضرائب (ينتظر التحقق).
2	Valid (صالح)	تم استلام تأكيد بنجاح الإرسال وصحة الإيصال.
3	Invalid (غير صالح)	تم استلام رد بأن الإرسال به مشكلات في التحقق (أخطاء).
4	Cancelled (مُلغى)	تم إلغاء الإيصال بواسطة المراجع (Reviewer).

5.2 الخيار 1: حاوية Docker (Docker Container)

هذا الخيار يحول الـ Toolkit إلى خدمة مصغرة (Microservice) تعمل محلياً. بدلاً من أن يقوم تطبيقك (بأي لغة برمجة) بالاتصال بـ <https://api.eta.gov.eg>، فإنه يتصل بـ <http://localhost:8020/toolkit/api/....>⁸

● التثبيت والتهيئة:

- يتطلب بيئة تشغيل Docker (مثل Docker Desktop أو Rancher Desktop).
- يتم التهيئة عبر ملف `docker-compose.yml`⁸.
- يجب على المطور تعديل متغيرات البيئة (Environment Variables) في ملف `YAML`، وأهمها `ToolkitConfig__EtaConfig__PosSerial` (الرقم التسلسلي للـ POS)، بالإضافة إلى `IdentityServiceUrl` وباقي عناوين URL الخاصة بالبيئة (الإنتاج أو الاختبار).⁸

● الاستخدام:

- بمجرد تشغيل الحاوية، فإنها تعرض واجهة Swagger UI لتوثيق واختبار جميع واجهاتها المحلية، ويمكن الوصول إليها عبر: <http://localhost:8020/toolkit/index.html>.⁸

(المحلية) Toolkit Docker API جدول 8: مرجع واجهات

(مُستخرج من 8)

الواجهة (API)	الغرض
Toolkit Initialize API	تهيئة الـ Toolkit (يجب أن يكون أول استدعاء).

مصادقة الـ POS (تجريد لعملية المصادقة المباشرة).	Toolkit token API
تحديث بيانات الكاش المحلية (مثل أكواد الضرائب والأنشطة).	Toolkit refresh cache API
توليد UUID فريد للإيصال.	Toolkit UUID API
توليد QR code للإيصال.	Toolkit QR code API
التحقق من صحة الإيصال وتخزينه في قاعدة SQLite المحلية (يصبح بحالة 0 (New)).	Issue receipt API
تصدير الإيصالات المخزنة (للحالة 1- (Exported)).	Get export receipts API
إرسال الإيصالات المخزنة محلياً (التي بحالة New) إلى هيئة الضرائب (تتحول إلى 1 Submitted)).	Batch submission API
مزامنة حالة الإيصالات المرسله (جلب الحالة النهائية Valid 2 أو 3 Invalid)).	Sync submission API
البحث في قاعدة بيانات SQLite المحلية.	Search Receipts API

5.3 الخيار 2: مكتبة (.NET) (Nuget)

هذا الخيار يدمج منطق الـ Toolkit مباشرة داخل تطبيق الـ NET. (سواء كان NET Core أو Framework)، مما يوفر أداءً عالياً (In-Process).⁹⁾

● التثبيت والتهيئة:¹⁰⁾

1. تثبيت الحزمة من nuget.org باسم: ETA.eReceipt.IntegrationToolkit.
2. إضافة قسم ToolkitConfig إلى ملف appsettings.json (يحتوي نفس إعدادات Docker، مثل PosSerial و IdentityServiceUrl).¹⁰⁾
3. تسجيل الخدمة في ملف program.cs (أو services.AddSingleton<IToolkitHandler, Startup.cs>: services.AddSingleton<IToolkitHandler, ToolkitHandler>());⁹⁾

● الاستخدام:⁹⁾

- يتم حقن (Inject) واجهة IToolkitHandler في الكلاس الخاص بالمطور.
- مثال (#C) لدورة حياة إيصال كاملة:

#C

// (يتم حقن toolkitHandler_ في المُنشئ)

// 1. التهيئة (يُستدعى مرة واحدة عند بدء تشغيل التطبيق)

```
var initializeRequestDto = new InitializeRequestDto { ... };  
await _toolkitHandler.Initialize(initializeRequestDto);
```

// 2. المصادقة (إذا لم يتم حفظ بيانات الاعتماد)

```
var authRequestDto = new AuthenticateRequestDto { ClientId = "...", ClientSecret = "..."  
};  
await _toolkitHandler.Authenticate(authRequestDto);
```

// 3. توليد UUID

```
var responseUuid = await _toolkitHandler.GenerateUuid(receipt.JsonWithoutUuid);  
var updatedReceiptJson = responseUuid.UpdatedReceiptJson?.ToString();
```

// 4. إصدار الإيصال (التحقق والتخزين المحلي)

// (يُرجع أخطاء التحقق أو رمز QR)

```
var responseIssue = await _toolkitHandler.IssueReceipt(updatedReceiptJson);
```

// 5. إرسال الإيصالات (إرسال الإيصالات الجديدة إلى هيئة الضرائب)

```
var submitRequestDto = new SubmitReceiptsRequestDto { ReceiptCount = 5 };  
await _toolkitHandler.SubmitReceipts(submitRequestDto);
```

// 6. المزامنة (التحقق من حالة الإيصالات المُرسلة)

```
var syncRequestDto = new SyncSubmissionRequestDto { All = true };  
;(await _toolkitHandler.SyncSubmission(syncRequestDto
```

5.4 الخيار 3: واجهة الأوامر السطرية (CLI)

هذا الخيار هو الأنسب للاختبارات السريعة، أو لعمليات الـ Scripting، أو للتكامل مع لغات برمجة غير NET (مثل Python أو

PHP) عبر استدعاءات `shell_exec`.⁷

• **التثبيت:**¹¹

1. تنزيل الملف المضغوط IntegrationToolKit.Cli.zip.

2. استخراج الملف التنفيذي toolkit.exe.

• **التهيئة:** يتطلب وجود ملف appsettings.json (بنفس بنية Nuget و Docker) في نفس مجلد الملف التنفيذي.⁷

• **الاستخدام:**

CLI الـ (Commands) جدول 9: مرجع أوامر

(مُستخرج من 7)

الأمر (Command)	الوصف	مثال الاستخدام
-----------------	-------	----------------

help--	عرض المساعدة لجميع الأوامر أو لأمر محدد.	toolkit --help (أو toolkit --init --help)
init--	تهيئة الـ Toolkit (تحديد مدة الكاش، إلخ).	toolkit --init --save-credential true
authenticate--	مصادقة الـ POS.	toolkit --authenticate --client-id "..." "..." --client-secret
refresh-cache--	تحديث الكاش المحلي من خوادم هيئة الضرائب.	toolkit --refresh-cache
generate-uuid--	توليد UUID لإيصال JSON.	toolkit --generate-uuid --input-json-path "in.json" --output-json-path "out.json"
generate-qr-code--	توليد QR code لإيصال (يتطلب وجود UUID).	toolkit --generate-qr-code --input-json-path "in_with_uuid.json"
issue-receipt--	التحقق من الإيصال وحفظه في SQLite (الحالة 0).	toolkit --issue-receipt --input-json-path "in_with_uuid.json"
submit--	إرسال الإيصالات (الحالة 0) إلى هيئة الضرائب (الحالة 1).	toolkit --submit --receipts-count 5
sync--	مزامنة حالة الإيصالات (الحالة 1) لجلب الحالة (2 أو 3).	toolkit --sync --all true
export--	تصدير الإيصالات (الحالة 1).	toolkit --export --receipts-count 10
search--	البحث في قاعدة بيانات SQLite المحلية.	toolkit --search --uuid "..."

● الاختيار بين نكهات الـ Toolkit:

يعتمد الاختيار بين نكهات الـ Toolkit الثلاث على البنية المعمارية للنظام القائم:

- **Docker**: هو "خدمة" (Service). مستقل عن لغة البرمجة (Language-agnostic) ومثالي لمعمارية الخدمات المصغرة (Microservices).
- **Nuget**: هي "مكتبة" (Library). يتم دمجها داخل التطبيق. هي الأسرع (لا يوجد حمل زائد لـ HTTP) ولكنها تقتصر على بيئة .NET.
- **CLI**: هو "برنامج تنفيذي" (Executable). هو الأبطأ (يتم تشغيل عملية جديدة مع كل أمر) ولكنه الأكثر مرونة للتكامل مع أنظمة قديمة أو scripts.

الجزء السادس: موارد المطورين والاختبار

لتسريع عملية التطوير والاختبار، توفر البوابة ملفات Postman جاهزة.¹

6.1 استخدام Postman للاختبار السريع

1. **المتطلب الأول**: تأكد من إكمال الخطوة 1.4 (تنصيب شهادة PreProd Root CA).
2. **تنزيل الملفات**: قم بتنزيل ملفات الـ Collection والـ Environment.

Postman جدول 10: روابط التنزيل المباشر لملفات (مُستخرج من 12)

الملف (File)	الرابط المباشر (Direct Download Link)
Postman Collection	https://sdk.invoicing.eta.gov.eg/files/Egyptian%20eInvoicing%20SDK.postman_collection.json
PreProd Environment	https://sdk.invoicing.eta.gov.eg/files/EEI%20-%20PREPROD%20Env.postman_environment.json
Production Environment	https://sdk.invoicing.eta.gov.eg/files/EEI%20-%20PRD%20Env.postman_environment.json

3. **الاستيراد**: قم باستيراد ملف الـ Collection وملف بيئة PreProd إلى Postman.
4. **استكشاف الأخطاء**: إذا واجهت خطأ SSL Error: Self signed certificate ... على الرغم من تثبيت الشهادة، يمكنك كحل أخير تعطيل "SSL Certificate Verification" من إعدادات Postman.¹

الجزء السابع: الملاحق وجداول الأكواد المرجعية

لا يقبل النظام إدخال بيانات نصية حرة (free text) في العديد من الحقول الحيوية (مثل نوع الضريبة، العملة، البلد). بدلاً من ذلك، يجب استخدام أكواد محددة مسبقاً. يوفر التوثيق قسماً كاملاً لـ "Codes" (الأكواد).²

جدول 11: جداول الأكواد المرجعية المتاحة
(مُستخرج من 2)

الوصف	جدول الأكواد (Code Table)
قائمة أنواع الأنشطة المسموح بها عند إرسال المستند.	Activity Types
معلومات عن فروع الممول التنظيمية لتحديد إصدار المستندات.	Branches
قائمة أكواد الدول المسموح بها في معلومات العنوان.	Country Codes
قائمة أكواد العملات المسموح بها في المستند.	Currency Codes
قائمة أوضاع تسليم الطلبات.	Order Delivery Modes
قائمة طرق الدفع التي يمكن اختيارها لدفع الإيصال.	Payment Methods
قائمة الأسباب المستخدمة لإرجاع الإيصالات بدون مرجع.	Return with No Reference Reasons Types
قائمة أنواع الضرائب والأنواع الفرعية المسموح بها كجزء من المستند.	Tax Types
قائمة أنواع الوحدات (مثل "قطعة"، "كيلو") المسموح بها في بنود الفاتورة.	Unit Types
قائمة أنواع وحدات الوزن المسموح بها في بنود المستند.	Weight Unit Types

خاتمة وتوصيات معمارية

يقدم SDK هيئة الضرائب المصرية نظاماً شاملاً وناضجاً يتطلب مستوى عالٍ من الدقة التقنية للتكامل. التحليل المعماري لهذا التوثيق يقود إلى توصيتين أساسيتين بناءً على المسارين المتاحين:

(Direct API) التوصية الأولى: مسار التكامل المباشر

يُطلب هذا المسار فريق تطوير لديه. (Large-scale, centralized ERPs) يُنصح بهذا المسار للأنظمة المركزية واسعة النطاق:
القدرة على

1. إدارة مستويين من الأمان: رموز الوصول (Access Tokens) لـ OAuth 2.0، وشهادات الختم الإلكتروني (eSeal X.509) لتوقيع المستندات.
 2. بناء آلات حالة (State Machines) قادرة على التعامل مع التدفقات غير المتزامنة (Asynchronous Flows) بشكل فعال، خاصة لعمليات الإرسال وطلبات الحزم.
 3. تطبيق آلية للتحقق الديناميكي من المخطط (Dynamic Schema Validation) عن طريق استرجاع وتخزين jsonSchema من الـ API.
- هذا المسار يوفر أقصى درجات التحكم والمرونة.

Integration Toolkit (التوصية الثانية: مسار حزمة أدوات التكامل)

أو التطبيقات الموزعة، أو أي سيناريو يتطلب العمل في بيئة ذات اتصال متقطع، (POS) يُنصح بشدة بهذا المسار لأي نظام نقاط بيع بتجريد Toolkit للفرق التي ترغب في تسريع عملية التكامل. تقوم الـ (Fast-Track) "بالإنترنت. كما أنه يمثل "المسار السريع أعقد أجزاء التكامل (المعالجة غير المتزامنة، التخزين المحلي، التحقق من الصحة) خلف واجهة بسيطة، سواء كانت (abstracting) (Docker API, Nuget Library, or CLI).

إن الاختيار بين هذين المسارين هو قرار معماري أساسي سيحدد ملامح مشروع التكامل بأكمله.

Works cited

1. Egyptian eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/>
2. Application Programming Interface (API), accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/api/>
3. eReceipt API - Egyptian eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/ereceiptapi/>
4. Getting started... - Egyptian eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/start/>
5. Get Document Type Version - Egyptian eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/api/04-get-document-type-version/>
6. Types - Egyptian eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/types/>
7. Toolkit Command Line Interface(CLI) Usage, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/toolkit/cli/usage/>
8. Toolkit Docker container usage - Egyptian eInvoicing & eReceipt SDK, accessed

- November 10, 2025, <https://sdk.invoicing.eta.gov.eg/toolkit/docker/usage/>
9. Toolkit Nuget Usage - Egyptian elInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/toolkit/nuget/usage/>
 10. Toolkit Nuget Installation - Egyptian elInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/toolkit/nuget/installation/>
 11. Toolkit CLI Installation - Egyptian elInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/toolkit/cli/installation/>
 12. Frequently asked questions - Egyptian elInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/faq/>