

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) ونقط البيع (POS) التابعة للممولين مع حلول الفاتورة الإلكترونية (eInvoicing) والإيصال الإلكتروني (eReceipt). تم تصميم هذا التوثيق لتمكين المطورين من أتمتة معالجة المستندات الضريبية بشكل كامل.¹

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

- مسار التكامل المباشر (**Direct API Integration**): هذا المسار، المؤتمن بشكل أساسي تحت قسم API، يمنح المطورين تحكمًا كاملاً ووصولاً مباشراً إلى جميع نقاط النهاية (Endpoints). يتطلب هذا المسار معالجة يدوية دقيقة لعمليات المصادقة، والتوفيق الرقمي للمستندات، والتحقق من صحة البيانات، وإدارة تدفقات العمليات غير المتزامنة.
- مسار حزمة أدوات التكامل (**Integration Toolkit**): هذا المسار، المؤتمن تحت قسم Integration Toolkit، يقدم مجموعة من الأدوات المساعدة التي تعمل كطبقة تجريد (Abstraction Layer) أو وكيل (Proxy) بين نظام الممول وواجهات برمجة التطبيقات المباشرة.¹ 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) ¹locally).

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

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

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 المصرية.¹

١.١ المبادئ التقنية الأساسية

يُستند 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 المترادفة (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 وبقليل 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 يضيقها 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 بحثة، وتتضمن تسجيل ملف رقمي لدى هيئة الضرائب (ETA) لإدارة الممثلين والأنظمة.
الحصول على بيانات اعتماد الد - **(API Client ID / Client Secret)**: بعد تسجيل نظام الد - ERP أو POS الخاص بالممول، يتم تزويده ببيانات اعتماد العميل (Client ID و Client Secret). تُستخدم هذه البيانات لمصادقة النظام نفسه عند طلب رموز الوصول (Access Tokens).
الحصول على شهادة الختم الإلكتروني (**eSeal X.509 Certificate**): هذا متطلب 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
 - الحصول على Access Token (كما هو موضح في الجزء الثاني). هذا الد - Token يثبت لمنصة الد - 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 به (Non-repudiation and Integrity).
الخطوة الأولى إدارية بحثة، وتتضمن تسجيل ملف رقمي لدى هيئة الضرائب (ETA) لإدارة الممثلين والأنظمة.
الحصول على بيانات اعتماد الد - **(API Client ID / Client Secret)**: بعد تسجيل نظام الد - ERP أو POS الخاص بالممول، يتم تزويده ببيانات اعتماد العميل (Client ID و Client Secret). تُستخدم هذه البيانات لمصادقة النظام نفسه عند طلب رموز الوصول (Access Tokens).
الحصول على شهادة الختم الإلكتروني (**eSeal X.509 Certificate**): هذا متطلب 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
 - الحصول على Access Token (كما هو موضح في الجزء الثاني). هذا الد - Token يثبت لمنصة الد - 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 المول المذكور ولم يتم التلاعب به (Non-repudiation and Integrity).

1.3 بيئات التشغيل (Environments)

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

الخاصة ببيانات 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 داخلياً (Internally Issued Certificates)¹. نتيجة لذلك، ستفشل جميع محاولات الاتصال بالـ API من بيئات التطوير القياسية بسبب عدم الثقة في مصدر الشهادة.

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

الخطوات (٤ - Windows 10):¹

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

تعد 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 لفواتير (eReceipt) والأخرى للإيصالات (eInvoicing).

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

هذا المسار مخصص لأنظمة الخافية (ERPs) التي تتعامل مع الفواتير الإلكترونية (eInvoicing).²

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

(Taxpayer System) جدول 2: معلومات طلب المصادقة

(مُستخرج من 2)

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

للوصول.

- ## **الاستجابة الناجحة:**

:الرمز المميز (JWT) لاستخدامه في استدعاءات الـ API القادمة. access_token

.token_type: Bearer

(أي ساعة واحدة). expires_in: 3600

استراتيجية إدارة الـToken

يشير التوثيق صراحة إلى أن صلاحية الـ Token هي ساعة واحدة (3600 ثانية)، وأن النظام يجب أن يكون "مسؤولًا" عن الحصول على رمز وصول جديد... قبل انتهاء صلاحية الـ Token الحالي". 2. هذا يستلزم بناء آلية تخزين مؤقت (Caching) للـ 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 ك وسيط (Intermediary)، يتطلب الطلب إضافة Header `onbehalfof` إضافي، 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 (Endpoint): POST <https://id.preprod.eta.gov.eg/connect/token> (نفس المسار، ولكن بأالية

٣. مختلفة تماماً

الآلية: تدفق client credentials مُعدّل ومتخصص.

يكشف هذا المسار عن اختلاف جوهرى عن التدفق القياسي. بدلاً من استخدام Authorization Header، يتم تمرير Client ID و Client Secret في Body، ويطلب أربعة Headers مخصصة إضافية خاصة بالـ POS³. هذا يعني أن المنطق البرمجي المكتوب لمصادقة eReceipt لن يعمل لمصادقة elnvoicing، ويجب بناء خدمتي مصادقة منفصلتين.

جدول 3: معلمات طلب المصادقة (POS)

(مستخرج من 3)

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

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

رموز الخطأ (Error Responses) :

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

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) وإصداراتها.¹
- 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 لتكوين سلعه وخدماته).¹ وتشمل:
 - Create EGS Code Usage
 - Search my EGS code usage requests
 - Request Code Reuse
 - Search Published Codes (البحث في أكواد EGS و GS1).
 - Get Code Details by Item Code
 - Update EGS Code Usage / Update Code
- Notification APIs: لاستعلام النظام عن الإشعارات التي تم استلامها سابقاً.

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

3.2 واجهات الفاتورة الإلكترونية (eInvoicing API)

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

3.3 واجهات الإيصال الإلكتروني (eReceipt API)

على عكس eInvoicing، يوفّر التوثيق المستخرج تفاصيل شاملة لنقطة النهاية³ eReceipt، والتي يمكن الوصول إليها عبر: https://sdk.invoicing.eta.gov.eg/ereceiptapi/.²

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

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

	{d}	
إرجاع معلومات عن عملية إرسال دفعه كاملة (Batch)	GET /api/v1/receiptsubmissions/{ submissionUUID}	جلب حالة الإرسال
البحث في الإيصالات المرسلة أو المستلمة (يدعم معلمات 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 (المعرفة إذا كانت جاهزة للتنزيل).	GET /api/v1/receipts/package	جلب حالة طلبات الحزم
تنزيل الحزمة الجاهزة (تنسيق 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. يتجلّى هذا النمط بوضوح في تدفقيْن رئيسين:

1. تدفق إرسال الإيصال (Receipt Submission Flow):³

- عند استدعاء POST /api/v1/receiptsubmissions، لا يقوم النظام بالتحقق الكامل من الإيصال فوراً.
- الاستجابة الفورية هي HTTP 202 Accepted³ وليس 200 OK. هذا الرمز يعني "لقد استلمنا طلبك وسنقوم بمعالجته لاحقاً".

التحقق الفوري يقتصر على البنية الأساسية (مثل خطأ³ BadStructure).

يجب على المطور تخزين submissionUUID المُرجع 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):
 - هذا التدفق مثالي لتوضيح النمط غير المترافق 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 التأكيد من جاهزيتها.
- يجب تصميم أنظمة التكامل كـ "آلات حالة" (State Machines) قادرة على تتبع حالة الطلبات غير المترافق، بدلاً من توقيع 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 تصميماً مرناً وقابلً 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`.
 - هذه الحقول تحتوي على مخطط (Schema) التحقق الكامل كـ "Base64 encoded string".⁵
 - يوضح التوثيق أن الإصدارات تُستخدم لتطوير الحقول بمرور الوقت، مثل "تحويل حقل اختياري إلى إلزامي".
- 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 ببرمجياً باستدعاء Get Document Type Version، وفك ترميز الدالة `jsonSchema`، واستخدام مكتبة JSON Schema Validator لتحقق من صحة المستندات قبل إرسالها. هذا يضمن أن التكامل لن ينقطع إذا قامت هيئة الضوابط بتحديث المخطط (Schema) في المستقبل.

4.2 تحليل البنية (Schema Analysis)

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

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

العنصر	النوع	الوصف	ملاحظة هامة
issuer	Issuer	هيكل يمثل معلومات المصدر.	(هيكل فرعي معقد)
receiver	Receiver	هيكل يمثل معلومات المستلم.	(هيكل فرعي معقد)
documentType	String	اسم نوع المستند. a (الفاتورة)، c (دائن)، d (دين).	الحقل المميز لنوع المستند.
documentTypeVersion	String	اسم إصدار نوع المستند. يجب أن يكون 1.0.	
dateTimeIssued	Date	تاريخ ووقت 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 UUIDs بـ للفواتير السابقة التي يتم الإشارة إليها.	مثال: S98L2CP1SMVBIU
invoiceLines	Invoice Line	بنود الفاتورة. يجب أن تحتوي على بند واحد على الأقل.	(هيكل فرعي معقد)
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)

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

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

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

كما ذكرنا في المقدمة، تقدم هيئة الضرائب مساراً بديلاً للتكامل المباشر، وهو "حزمة أدوات التكامل" (Integration Toolkit).¹ 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 بيانات التطوير¹ :
 - .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 الجوهيرية. 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.
 - التتحقق من صحة المستندات محلياً (Validate documents locally).
 - تخزين وتتصدير الإيصالات المصدرة¹ (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.

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

(مستخرج من 1)

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

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

5.2 الخيار 1: حاوية Docker (Docker Container)

هذا الخيار يحولـ Toolkit إلى خدمة مصغرة (Microservice) تعمل محلياً بدلاً من أن يقوم تطبيقـ (بأي لغة برمجة) بالاتصال بـ [http://localhost:8020/toolkit/api/....](http://localhost:8020/toolkit/api/)⁸, <https://api.eta.gov.eg>

- التثبيت والتهيئة:
 - يتطلب بيئة تشغيل Docker Desktop أو⁸ Rancher Desktop.
 - يتم التهيئة عبر ملف⁸ docker-compose.yml.
 - يجب على المطور تعديل متغيرات البيئة (Environment Variables) في ملف YAML، وأهمها IdentityServiceUrl ToolkitConfig_EtaConfig_PosSerial IdentityServiceUrl للـ ToolkitConfig_EtaConfig_PosSerial وباقي عناوين URL الخاصة بالبيئة (الإنتاج أو الاختبار).

- الاستخدام:
 - بمجرد تشغيلـ الحاوية، فإنـها تعرض واجهة UI Swagger لتوثيقـ واختبارـ جميعـ واجهـاتها المحليةـ، ويمكنـ الوصولـ إليها عبرـ <http://localhost:8020/toolkit/index.html>.⁸

(المحليةـ) Toolkit Docker API جدول 8: مرجعـ واجـهـاتـ (مستخرجـ منـ 8)

الغرض	الواجهةـ (API)
تهيئةـ الـ Toolkit (يـجبـ أنـ يكونـ أولـ استـدعاءـ).	Toolkit Initialize API

مصادقة الـ POS (تجري لعملية المصادقة المباشرة).	Toolkit token API
تحديث بيانات الكاش المحلية (مثل أ��واض 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 حالة 0 (New)).	Issue receipt API
تصدير الإيصالات المخزنة (الحالة -1 (Exported)).	Get export receipts API
إرسال الإيصالات المخزنة محلياً (التي بحالة New) إلى هيئة الضرائب (تنحول إلى 1 (Submitted)).	Batch submission API
مزامنة حالة الإيصالات المرسلة (جلب الحالة النهائية Valid (2) أو Invalid (3)).	Sync submission API
البحث في قاعدة بيانات SQLite المحلية.	Search Receipts API

5.3 الخيار 2: مكتبة Nuget (.NET)

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

• التثبيت والتهدئة:

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 Startup.cs): services.AddSingleton<IToolkitHandler>();⁹

• الاستخدام:

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

```
#C
// يتم حقن toolkitHandler في المنشئ
```

```

1 // التهيئة (يُستدعي 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(receiptJsonWithoutUuid);
var updatedReceiptJson = responseUuid.UpdatedReceiptJson?.ToString();

4 // إصدار الإيصال (التحقق والتخزين المحلي)
// (يرجع 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 المرسلة)
var syncRequestDto = new SyncSubmissionRequestDto { All = true };
;(await _toolkitHandler.SyncSubmission(syncRequestDto)

```

5.4 الخيار 3: واجهة الأوامر السطحية (CLI)

هذا الخيار هو الأنسب لاختبارات السريعة، أو لعمليات الـ Scripting، أو للتكامل مع لغات برمجة غير .NET (مثل Python أو PHP) عبر استدعاءات⁷ shell_exec.

- التثبيت:¹¹
- تنزيل الملف المضغوط IntegrationToolKit.Cli.zip.
- استخراج الملف التنفيذي toolkit.exe.
- التهيئة: يتطلب وجود ملف appsettings.json (بنفس بنية Docker و Nuget) في نفس مجلد الملف التنفيذي.⁷
- الاستخدام:

الـ CLI اوامر Commands جدول 9: مرجع (7) مستخرج من

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

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

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

- Docker: هو "خدمة" (Service). مستقل عن لغة البرمجة (Language-agnostic) ومثالي لعمارية الخدمات المصغرة (Microservices).
- Nuget: هي "مكتبة" (Library). يتم دمجها داخل التطبيق. هي الأسرع (لا يوجد حمل زائد لـ HTTP) ولكنها تقصر 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: تأكيد من إكمال الخطوة 1.4 (تثبيت شهادة CA 1.4). (PreProd Root CA)
- .2 تنزيل الملفات: قم بتنزيل ملفات Collection و Environment Collection.

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

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

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

الجزء السابع: الملاحق وجدائل 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 ها عند 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 وتحصيات معمارية

يقوم SDK هيئة الضرائب المصرية نظاماً شاملأً 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

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

1. إدارة مستويين من الأمان: رموز الوصول (Access Tokens) لـ OAuth 2.0، وشهادات الختم الإلكتروني (eSeal X.509) لتوفيق 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 динамики من المخطط (Dynamic Schema Validation) عن طريق استرجاع وتخزين jsonSchema API. هذا المسار يوفر أقصى درجات التحكم والمرؤنة.

(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 (abstracting) أجزاء التكامل (المعالجة غير المتزامنة، التخزين المحلي، التحقق من الصحة) خلف واجهة بسيطة، سواء كانت (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/erceiptapi/>
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 eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/toolkit/nuget/usage/>
 - 10. Toolkit Nuget Installation - Egyptian eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/toolkit/nuget/installation/>
 - 11. Toolkit CLI Installation - Egyptian eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/toolkit/cli/installation/>
 - 12. Frequently asked questions - Egyptian eInvoicing & eReceipt SDK, accessed November 10, 2025, <https://sdk.invoicing.eta.gov.eg/faq/>