

E-Receipts and E-Invoicing

Technical Guide

EGKits SaaS V10

Egyptian Tax Authority

Technical reference for software architects, developers, implementation engineers, IT administrators, and business stakeholders. This edition explains the e-Receipt and e-Invoicing integrations used by EGKits, including bulk submission, automation, digital signing, API constraints, and how to estimate throughput without treating an estimate as an official quota.

Field	Value
Document version	1.0
Publication date	26 October 2026
SDK pages retrieved	26 October 2026
Applies to	EGKits SaaS V10
Audience	Architecture, engineering, operations, and business owners
Distribution	Public. No taxpayer data, credentials, or production logs.
English file	EGKits_ETA_EReceipts_EInvoicing_Technical_Guide_EN.pdf
Arabic file	EGKits_ETA_EReceipts_EInvoicing_Technical_Guide_AR.pdf

Not confirmed

The retrieved SDK pages do not print a single package version number. Limits below are tied to the page URL and the retrieval date. A later SDK edit can change a rule. Re-read the cited page before changing production behavior.

500 receipts and 1.5 MB

Official batch ceiling, printed in the receipt FAQ.

No hourly submit quota

The governance table does not list Submit Documents or Submit Receipts.

A setting is not a quota

50 invoices per submission and 4 parallel receipt requests are EGKits controls.

Contents

1. How to read this guide	2	11. Bulk e-Invoice submission	10
2. Executive summary	2	12. Automatic signing and submission	11
3. Sources and environments	3	13. E-Invoice throughput	12
3.1 Environments	3	14. E-Receipts compared with e-invoices	12
3.2 Access tokens	3	15. Capacity planning	13
3.3 Published throttles	4	15.1 Preproduction load tests	13
4. E-Receipt architecture	4	16. EGKits V10 recommendations	14
4.1 What EGKits already does	5	17. Troubleshooting	14
4.2 Errors and monitoring	5	18. Implementation checklist	15
5. E-Receipt lifecycle	5	19. Glossary	15
5.1 Rules that change acceptance	5	20. Official references	15
5.2 What to persist	6		
6. Bulk e-Receipt submission	6		
6.1 How EGKits packs a batch	7		
7. Automatic e-Receipt submission	7		
7.1 Eligibility	7		
7.2 Queue, token, retry, and duplicates	7		
8. E-Receipt throughput	8		
9. E-Invoice lifecycle	8		
9.1 Cancellation and correction	9		
10. Authentication and digital signing	9		
10.1 EGKits.EtaSigningAgent	10		

1. How to read this guide

Three kinds of statement appear throughout. They are not interchangeable.

Label	Meaning
Official	The sentence is taken from an ETA SDK page retrieved on 26 October 2026. The source URL is in the references.
EGKits setting	A value configured or coded in EGKits V10. It is an implementation choice. It is not an ETA quota.
Estimate	A calculation from stated assumptions. It is not a measured production result and not an ETA guarantee.
Not confirmed	The available official pages retrieved for this edition do not state the number.

Bulk import into EGKits, bulk signing, bulk API submission, and bulk status retrieval are different operations. A spreadsheet that loads a thousand receipts does not mean the tax authority accepts those thousand receipts in one HTTP call, and a signing rate is not a submission rate.

2. Executive summary

EGKits submits e-receipts and e-invoices to the Egyptian Tax Authority over authenticated REST APIs. Both families accept more than one document in a single submission, return HTTP 202 while validation continues, and can accept some documents in that submission while rejecting others. The two families do not share a signing model, a payload ceiling, or a published request rate for the submit operation.

- E-receipts. POST /api/v1/receiptsubmissions. JSON only. The receipt issuance FAQ states a maximum submission of 500 receipts and 1.5 MB, a minimum of 1 receipt, and a maximum of 300 line items on a receipt. The same pages describe a batch CAdES-BES signature and also state that signature validation is not deployed until ETA decides to test and deploy it.
- E-invoices. POST /api/v1.0/documentsubmissions/. JSON or XML. One or more signed documents. Each document carries its own CAdES-BES signature. The submit page does not publish a fixed document count. A body that is too large returns HTTP 400 MaximumSizeExceeded, and the supported size is returned in the error rather than as a fixed number on that page.

- **Authentication.** OAuth 2.0 client credentials at POST /connect/token on the identity service. The sample token lifetime is 3,600 seconds. Integration practices say tokens are valid for 60 minutes by default and must be cached. Receipt authentication also sends POS serial, OS version, model framework, and pre-shared key headers.
- **Published rates.** The governance page lists per-ERP or per-POS limits for search, get, and recent-document operations. It does not list a per-hour limit for Submit Documents or Submit Receipts. Both submit pages say administrators configure throttling and that callers must not call the API too frequently.
- **Throughput.** No official receipts-per-hour or invoices-per-hour figure could be verified. Hourly capacity is estimated from latency, concurrency, and documents per request, and it must stay inside the published batch and query limits. EGKits has no production measurement in this edition.
- **Signing.** Invoice signing is per document, on an approved Egyptian eSeal or natural-person certificate, and in EGKits it is performed by EGKits.EtaSigningAgent when the key is on a local USB token. Receipt UUID generation is a SHA-256 of the canonical receipt. Receipt batch signing is specified separately, and the agent is not the receipt pipeline.

Estimate

Illustration only. If one receipt serializes to 8,000 bytes, the 1.5 MB ceiling fits 187 receipts, which is below the count ceiling of 500. At a hypothetical 1.0 second per request and one request at a time, that is 673,200 receipts per hour. That figure assumes every request succeeds, latency does not grow with payload, and no administrator throttle intervenes. It is not an ETA limit and not an EGKits production result.

3. Sources and environments

3.1 Environments

EGKits stores the API base URL and the identity base URL on the ETA configuration. The values below are the endpoints that configuration screen treats as the official preproduction and production pairs. Use preproduction for integration and load tests. Do not point a capacity test at production.

Role	Preproduction	Production
Invoicing API	https://api.preprod.invoicing.eta.gov.eg/	https://api.invoicing.eta.gov.eg/
Identity	https://id.preprod.eta.gov.eg/	https://id.eta.gov.eg/
SDK	https://sdk.invoicing.eta.gov.eg/	https://sdk.invoicing.eta.gov.eg/
Portal example used by the receipt QR	https://invoicing.eta.gov.eg/	https://invoicing.eta.gov.eg/

EGKits setting

EGKits calls the submit paths under api/v1 on the configured API host: api/v1/receiptsubmissions and api/v1/documentsubmissions. The SDK prints api/v1.0 for document submission and document cancellation. Keep the host on the official environment URL and treat a version-segment difference as something to confirm in preproduction, not as a second product API.

3.2 Access tokens

The identity service, not the invoicing host, issues the access token. The receipt page Authenticate POS documents POST /connect/token with grant_type client_credentials, client_id, and client_secret. A successful call returns HTTP 200, token_type Bearer, and expires_in. The published example of expires_in is 3,600, described as one hour. Integration practices say the default validity is 60 minutes and name a fresh token on every API call as an anti-pattern.

```
POST /connect/token
Content-Type: application/x-www-form-urlencoded
posserial: <POS serial>
pososversion: <POS OS version>
posmodelframework: <framework>
pressharedkey: <pre-shared key>
```

```
grant_type=client_credentials&client_id=<id>&client_secret=<secret>
```

The four POS headers are documented on Authenticate POS. Invoice system login uses the same token endpoint and client-credentials body. EGKits refreshes a token when a call returns HTTP 401, and it should keep using the cached token until that lifetime ends. Client secrets and pre-shared keys stay in the tenant ETA configuration. They are not written into this guide, into logs, or into the public website.

3.3 Published throttles

The governance page says the selective limiter applies per ERP or POS system. Exceeding a specific rate returns HTTP 429 with a Retry-After header. A platform-wide overload returns HTTP 503. The advised wait for HTTP 503 is 1 to 3 seconds. For HTTP 429 the wait must be longer than Retry-After. The published per-API table retrieved on 26 October 2026 is below. Submit Documents and Submit Receipts are absent from that table.

Area	API	Published limit
Common	Search my EGS code usage	1 request every 5 seconds
Common	Search Published Codes	1 request every 10 seconds
E-Invoicing	Get Recent Documents	3 requests every 1 second
E-Invoicing	Get Document	2 requests every 1 second
E-Invoicing	Get Document Printout	1 request every 5 seconds
E-Invoicing	Get Document Details	2 requests every 1 second
E-Invoicing	Search Documents	2 requests every 1 second
E-Receipts	Get Receipt Details	2 requests every 1 second
E-Receipts	Get Receipt	2 requests every 1 second
E-Receipts	Search Receipts	2 requests every 1 second
E-Receipts	Get Recent Receipts	3 requests every 1 second

Not confirmed

A per-hour cap on receipt submission or invoice submission is not confirmed in the available official documentation retrieved for this edition. The submit pages say throttling is configured by administrators. Handle HTTP 429 and HTTP 503. Do not invent the missing number.

4. E-Receipt architecture

An e-receipt is a B2C document issued by a taxpayer who is allowed to issue receipts, from a POS device that is registered and linked to the issuer registration number before the receipt is issued. The FAQ limits the POS serial to 100 characters and requires the taxpayer not to be blocked. The submit page says only taxpayers with the B2C tag may submit receipts, and that the POS should be activated before submission on any other channel.

EGKits receipt path

- 1 POS or sales posting creates a receipt for a tenant, company, branch, and point of sale.
- 2 EGKits validates core fields, codes, totals, and the previous-UUID chain for that device.
- 3 The canonical receipt is hashed with SHA-256. The 64-character hex digest is the receipt UUID.
- 4 Eligible receipts are packed into a JSON body of up to 500 receipts and 1.5 MB.
- 5 The receipt API client posts the body with the cached bearer token and POS headers.

6 The HTTP 202 body is stored: submission UUID, accepted receipts, and rejected receipts.

7 Later status reads use the submission or the receipt details API, inside the published query rates.

4.1 What EGKits already does

The e-receipt module keeps the receipt, its lines and taxes, validation errors, and an ETA submission log. Submission is grouped by ETA configuration, POS serial, and client id, because a POS submission must contain receipts from that POS. The automatic scanner reads receipts flagged for submission, skips a receipt that is already in flight or waiting for its next retry, and hands the rest to the same submission service used by an explicit submit. There is no second submission stack to add for this guide.

Company, branch, and POS configuration supply the registration number, branch code, device serial, client credentials, and environment URL. A receipt is not eligible merely because it exists in the database. The issuer on the receipt must match the token, the branch must be allowed, and the document type and version must be active. Those checks are in the official core-field rules and in the EGKits validator that runs before the HTTP call.

4.2 Errors and monitoring

Persist the HTTP status, the raw response, the submission UUID, and the per-receipt acceptance or rejection. A transport failure before any response is not the same as HTTP 202, and it is not the same as HTTP 422 DuplicateSubmission. The duplicate rule hashes the payload and compares it with submissions from the same taxpayer during the past 10 minutes. A retry of an identical body inside that window is refused. The caller waits for the Retry-After value and then decides from the earlier response, instead of assuming the receipt was never received.

Operational monitoring should show queued, in-flight, accepted, rejected, and exhausted-retry counts per tenant and per POS. Alerts belong on authentication failures, repeated HTTP 429, payload-too-large receipts, and a device whose previous-UUID chain is broken. Do not alert on a single rejected line inside an otherwise accepted submission without showing the property path returned by ETA.

5. E-Receipt lifecycle

From issue to a final status

1 Create the receipt with issuer, receiver, POS, lines, taxes, and internal receipt number.

2 Validate offline: structure, codes, core fields, and the national-ID threshold when it applies.

3 Set previous UUID to the prior receipt of the same POS, or empty for the first receipt.

4 Serialize with an empty UUID, normalize, SHA-256, and write the 64-character hex UUID.

5 Submit inside 24 hours unless the receipt is a resubmission that carries the old UUID.

6 Read HTTP 202. Store submission UUID, long id, and receipt number mapping.

7 Poll the submission, or accept a callback, after asynchronous validation finishes.

8 On a content correction, issue a new UUID, keep the same previous UUID, and set reference old UUID.

5.1 Rules that change acceptance

- Issuance time cannot be in the future, and submission time cannot be in the future.

- The receipt UUID is a hash of the serialized receipt excluding the UUID itself. Previous UUID is also 64 hexadecimal characters.
- All receipts in a POS submission are issued by the same POS. A retired or permanently retired device rejects the document and, in the FAQ wording, the submission is eventually rejected.
- National ID is mandatory for a person buyer when the total reaches 50,000 EGP on receipt versions 1.0 and 1.1, and 150,000 EGP on version 1.2 and newer.
- A return receipt may be issued up to 540 days after the sales receipt. Its reference UUID must point at the original receipt type.
- The FAQ states amendment with reference as 96 hours and amendment without reference as 72 hours. Late submission uses a yearly quota, a window in days, and a period in hours, and the token must be refreshed after a late-submission request is added to the profile.
- Content type is application/json. The branch must be allowed. The document type and version must be valid and active. A late submission requires a reason.

5.2 What to persist

Keep the immutable canonical JSON used for the hash, the UUID, the previous UUID, the reference old UUID when present, the internal receipt number, the POS serial, the submission UUID, the long id, the HTTP status, the raw ETA body, the attempt number, the time the claim was taken, and the final status. EGKits already stores the receipt aggregate and an ETA receipt submission log. Correlation uses the receipt number returned on accepted and rejected items. Do not replace a stored UUID after acceptance. A corrected receipt is a new hash and must point at the invalid UUID through the reference-old-UUID field, as the FAQ example shows.

The printed QR code is not the submission API. The FAQ template is the portal URL, then #Total:, the total, then ,IssuerRIN: and the seller registration number. The URL shape is the portal host, /receipts/search/, the UUID, /share/, and the issuance time in UTC. The example retrieved from the FAQ used an http scheme; use the portal host configured for the environment.

```
https://invoicing.eta.gov.eg/receipts/search/{UUID}/share/{ReceiptDateAndTimeUtc}#Total:{Total},IssuerRIN:
{RegistrationNumber}
```

6. Bulk e-Receipt submission

ETA supports multiple receipts in one API request. The submit signature is POST /api/v1/receiptsubmissions. The body has a receipts array of at least one document. Receipt types in that array are the receipt, the return receipt, and the return receipt without reference. The FAQ, which says the limits are managed by the IT operator, states the maximum submission size as 500 receipts (1.5 MB) and the minimum as 1 receipt.

Official

Maximum submission size is 500 receipts (1.5 MB). Minimum submission size is 1 receipt. Maximum line items per receipt is 300. Average number of lines per receipt is stated as 10. Source: Receipt Issuance FAQ, retrieved 26 October 2026.

Sequential submission of one receipt is valid because the minimum is one. Parallel submission of independent requests is not given a numeric concurrency allowance on the submit page. The page only says callers should submit in batches and should not call too frequently, and that administrators configure throttling. Background queues are an EGKits mechanism. The ETA integration toolkit also documents a local batch submitter with maximumSubmissionDocumentCount defaulting from the initialize API notes to 500. That toolkit is a separate channel. EGKits submits through the receipt submission API, not through the toolkit batch file.

Question	Answer from the retrieved pages
Several receipts in one request?	Yes. receipts array, at least one.
Official count ceiling	500 receipts, FAQ.
Official payload ceiling	1.5 MB, FAQ. HTTP 400 MaximumSizeExceeded also exists on the submit page, with the supported size in the error body.

Question	Answer from the retrieved pages
Partial success	HTTP 202 returns acceptedDocuments and rejectedDocuments.
Whole-request failure	BadStructure, IncorrectSubmitter, MaximumSizeExceeded, DuplicateSubmission.
Identical retry	Duplicate submission for 10 minutes, based on a hash of the payload. Honor Retry-After.
Requests per hour	Not confirmed for this API.

6.1 How EGKits packs a batch

The packer closes a request when the next receipt would exceed 500 documents or 1,500,000 bytes. A single receipt that cannot fit alone is not posted. With an 8,000-byte receipt the byte ceiling binds first: 187 receipts. With a 2,500-byte receipt the count ceiling binds first: 500 receipts. The wrapper used for that arithmetic is the 15 bytes of `{"receipts":[]}` plus one comma between receipts. Real documents vary, so the packer measures UTF-8 length rather than assuming the example.

EGKits setting

EReceiptAutoSubmit:EtaBatchSize defaults to 500 and cannot be set above 500. MaxParallelism defaults to 4. MaxReceiptsPerTenant defaults to 2,000 per scanner tick. The cron default is every minute on the integrations queue. These are EGKits controls.

Importing a file into EGKits can create many receipt rows. That import is not an ETA submission. Signing a receipt batch, if a channel requires the CADES-BES signatures element, is not the HTTP post. Asking for status with Search Receipts or Get Recent Receipts is not the submit, and those two reads are rate limited. Use the submission id returned by HTTP 202 for reconciliation. Integration practices say Get Recent Receipts should usually be replaced by Search Receipts, and that checking every document with the retrieval APIs during submission is an anti-pattern.

7. Automatic e-Receipt submission

Automatic submission means EGKits posts an eligible receipt without a person pressing Submit on that receipt. It does not mean a USB token signed the receipt. The invoice signing agent is a different product path. The receipt SDK describes a batch signature over the canonical receipts array, and the same pages say signature validation will not be deployed until ETA decides to test and deploy the component. EGKits V10 builds the receipt UUID locally and posts the receipts array through the receipt API client. It does not lease those receipts to EGKits.EtaSigningAgent.

7.1 Eligibility

A new receipt inherits automatic submission from the point-of-sale AutoSendToETA flag and the ETA configuration AutoSubmit flag. An explicit submit sets the same ingestion flag. The scanner then requires that flag, a submittable status (draft, validated, queued, or invalid when submission is still allowed), a retry time that is due, a retry count inside the automatic maximum, and no active in-flight claim. Statuses that represent a receipt already submitted, valid, or returned are not sent again.

7.2 Queue, token, retry, and duplicates

- The recurring job is Hangfire on the integrations queue. The default cron is every minute. Hangfire does not schedule a smaller interval. One tick processes a bounded number of tenants and a bounded number of receipts per tenant.
- The worker reuses the cached access token and refreshes it on HTTP 401. It does not request a token per receipt.
- A claim is held for 5 minutes. The HTTP client budget is shorter than that hold, so a slow call is not overtaken by a second submission. A claim older than the wire budget can be recovered after a process restart.
- Automatic retries default to 8. Backoff leaves the receipt queued until nextRetryAt. After the maximum, the receipt stays queued and needs a person.
- HTTP 202 is success for the transport step even though validation is still running. Do not post that payload again.
- HTTP 422 DuplicateSubmission means the authority already saw that payload inside 10 minutes. Reconcile. Do not

treat it as a validation failure of the receipt content.

- HTTP 429 and HTTP 503 use Retry-After when present, otherwise the short wait documented for overload. They are not reasons to split a valid batch into smaller identical copies.
- An uncertain outcome is a timeout or a dropped connection after the request was written. The receipt stays claimed until the hold expires, then a lookup by submission or receipt UUID runs before another post.

Audit history is the submission log plus the receipt status. Alerts should fire when a POS exhausts retries, when authentication fails for a device, and when a tick leaves a growing queued backlog. They should not include receipt totals that identify a shopper, and they should not include the client secret or the pre-shared key.

8. E-Receipt throughput

The question is how many e-receipts EGKits can submit per hour. The official pages do not answer it with a single number. They answer the size of one request, and they answer the rate of the read APIs. The rest is an estimate.

Figure	Value	Class
Submit requests per hour	Not confirmed	Not in the governance table
Receipts per request	1 to 500, and 1.5 MB	Official, FAQ
One receipt per request at 1.0 s	3,600 / hour	Estimate, sequential
187 receipts of 8 KB at 1.0 s, concurrency 1	673,200 / hour	Estimate
Same batch at 2.0 s	336,600 / hour	Estimate
Same batch, concurrency 2, 1.0 s	1,346,400 / hour	Estimate, not an allowance
500 receipts of 2.5 KB at 0.5 s	3,600,000 / hour	Estimate, optimistic
Production measurement	Not measured for this edition	Not a result
Get Receipt or Get Receipt Details	2 requests every 1 second, 7,200 requests per hour if sustained	Official query limit, not a submit quota

One receipt per request does not establish a maximum receipts-per-hour figure. At a hypothetical average of 1.0 second it produces 3,600 requests an hour, and at 2.0 seconds it produces 1,800. The hour changes as soon as the latency changes. Batching multiplies documents per successful request. It also makes each request slower and more likely to hit the 1.5 MB ceiling, so the single-receipt rate and the full-batch rate are not both achievable at the latency assumed for the other.

Controlled concurrency multiplies the sequential rate. Nothing retrieved for this edition authorizes a particular concurrency for Submit Receipts. The EGKits default of 4 in-flight requests per tenant is a product setting. Raise it only after a preproduction run shows that HTTP 429 is absent, and stop at the first sustained throttle. A practical production number is not stated here because this edition did not measure one.

Estimate

A safe planning illustration, not a target to run unmeasured: if preproduction shows a stable 1.0 second for the 187-receipt batch and concurrency 1, take about 70 percent of 673,200, which is 471,239. Leave the rest for retries, status reads, and other tenants on the same ERP credential. Replace the inputs with measured latency before using the number operationally.

9. E-Invoice lifecycle

An e-invoice submission carries one or more signed documents. The SDK lists invoice, credit note, debit note, export invoice, export credit note, and export debit note. Each document must match the document-type version that is active when it is issued. JSON and XML are both accepted, and every document in one submission must use the same format. The content type header says which one. EGKits submits JSON.

EGKits invoice path

- 1 The user or a posting flow finalizes the invoice for a tenant, company, and branch.

- 2 Schema and business rules run before any signature is requested.
- 3 EGKits stores the canonical text that will be signed and does not change it afterwards.
- 4 A signing session is leased to the device that holds the issuer certificate.
- 5 The agent signs that text. The server verifies the signature before ETA sees it.
- 6 A separate submission job posts one or more signed documents.
- 7 HTTP 202 stores the submission UUID, the ETA document UUID, the long id, and any rejection.
- 8 Status is read from the submission. Cancellation or a credit note follows the rules for that document.

Synchronous validation checks structure, that the taxpayer who submits is the issuer, and that the document matches its type version. HTTP 202 means the submission was accepted for further processing, not that every document is valid. `acceptedDocuments` maps the internal id to the ETA uuid and `longId`. The ETA uuid on this API is a 26-character identifier assigned by the authority. It is not the 64-character SHA-256 the receipt uses as its own UUID. `rejectedDocuments` returns the internal id and the error, including nested details so the caller can fix all of them before trying again.

9.1 Cancellation and correction

Cancel Document is PUT `/api/v1.0/documents/state/{UUID}/state`. The body status must be cancelled, and a reason is required. The SDK examples of reasons are wrong buyer details and wrong invoice details. The cancellation window depends on the document type and is returned by Get Document Type in the workflow parameters. The page says the period is configured by ETA according to the law. After the window, the documented correction is a credit note that references the invoice, not a second cancellation.

- HTTP 400 `OperationPeriodOver`: the cancellation window has ended.
- HTTP 400 `IncorrectState`: the document is already rejected or invalid.
- HTTP 400 `ActiveReferencingDocuments`: cancel the credit note or other referencing document first.
- HTTP 400 `TryingToCancelFreezedDocument`: the document is marked frozen.
- HTTP 403 when the caller does not own the document, or when the profile has the B2B Deny ERP Cancellation/Rejection tag.

Store the internal id, the canonical payload, the signature value, the certificate identity used to produce it, the submission UUID, the ETA uuid, the long id, the HTTP status, and the raw response. Signing metadata is not a substitute for the ETA response. A valid signature can still be rejected by a business rule, and an accepted submission can still fail later validation.

10. Authentication and digital signing

Invoice signing and invoice submission are separate steps, and neither step is acceptance. The Document Signature Creation page specifies the algorithm for one document.

Signature of one invoice

- 1 Build the document JSON or XML with no signature element yet.
- 2 Canonicalize it so insignificant whitespace cannot change the hash.

- 3 SHA-256 the UTF-8 bytes of that canonical form.
- 4 Sign the hash as CAdES-BES with an approved Egyptian eSeal or natural-person certificate.
- 5 Place the Base64 signature on the original document. Issuer type is I. Service-provider type S is optional and signs the document including the issuer signature.
- 6 Add the signed documents to the documents array and submit.

The invoice v1.0 signature rules require a Base64 value that is a valid CAdES-BES structure, a signing certificate that is valid at the submission date, and a certificate issued to the taxpayer registration number on the document or to an active representative at the issuance time. The signature must be a valid RSA signature from an approved Egyptian eSeal or natural-person certificate. EGKits checks the signature over the stored canonical text, the certificate hash, validity, key usage, the issuer tax number, and that the certificate is one the device reported, before the document is eligible to send.

10.1 EGKits.EtaSigningAgent

The SaaS server cannot see a USB token plugged into a customer machine. The agent runs on that machine. It leases documents with a device credential. A lease lasts 5 minutes so two computers do not sign the same document, and a document is retried at most 5 times. The agent produces a detached CMS signature, SHA-256, with the ESS signing-certificate-v2 attribute, content type 1.2.840.113549.1.7.5. It uploads the signature. Failures are reported on the session and appear on the invoice audit trail.

- Register the computer from ETA signing devices. The registration key starts with EGK1., works once, and expires after 24 hours.
- The PIN is entered on the signing computer and stored with Windows DPAPI. The server does not return PINs. The PIN is not written to appsettings, logs, or this guide.
- One computer can register for several tenants. Each tenant uses its own token and its own credentials. Tokens are not shared.
- Several companies on one computer are matched by tax registration number. The PIN is tried only on the issuer token. If no attached token holds that certificate, signing stops.
- Not every middleware accepts a silent PIN. EnterSafe ePass2003 can show a PIN dialog unless the agent logs in through PKCS#11. Unattended signing is therefore a per-token capability, not a universal property of USB tokens.
- Certificate expiry warnings are sent to signing-device administrators at 30, 7, and 1 days, and on the day after expiry. An expired certificate fails signature validation.
- Revocation and renewal are operational duties. EGKits refuses a revoked device on its next request. Replace the certificate before the old one lapses, and re-enroll a lost computer so the old secret dies.

EGKits setting

The agent design notes describe USB signing as sequential, roughly 0.5 to 2 seconds per signature, and about 1,500 to 3,000 signatures per hour per token when the queue is drained. That range is an engineering note in the agent documentation. It is not an ETA limit and it is not a measured production result published in this guide.

11. Bulk e-Invoice submission

The supported model is one HTTP request that contains a documents array of one or more already signed documents. The official signature is POST /api/v1.0/documentsubmissions/. There is no separate asynchronous batch-file API on the invoicing submit page. The receipt toolkit batch API is not the invoice API.

Question	Retrieved documentation
Multiple invoices in one request	Yes. The documents array must contain at least one document.

Question	Retrieved documentation
Official maximum documents	Not confirmed. The page says one or more. It does not print a count.
Official payload size	Not printed as a fixed byte count. HTTP 400 MaximumSizeExceeded returns the supported size in the error. The caller must split and resubmit.
Signature	Each document is signed before it enters the array.
Partial failure	HTTP 202 can list accepted and rejected documents together.
Duplicate	HTTP 422 DuplicateSubmission for an identical payload from the same taxpayer within 10 minutes.
Order	Not specified as a sequencing requirement across requests. Inside one submission, map by internalId, not by array position alone.
Submit rate	Not confirmed. Administrators configure throttling. The page recommends batches and warns against calling too frequently.

EGKits setting

EtaSubmissionBatch:MaxDocumentsPerSubmission defaults to 50. MaxPayloadBytes defaults to 4,194,304. DelayBetweenSubmissionsMilliseconds defaults to 500. Throttle retries default to 3, with an initial backoff of 5 seconds and a maximum of 60 seconds. Submission status page size defaults to 100. None of these numbers is an official ETA quota. A batch that still exceeds the size ETA returns is split and retried.

Importing invoices, validating them, signing them, submitting them, and polling their status are five operations. Import can be large. Validation is local. Signing is limited by the token. Submission is limited by payload size, by the unpublished administrator throttle, and by the 10-minute duplicate hash. Status polling of each document through Get Document is officially limited to 2 requests per second per ERP or POS. Get Document Printout is 1 request every 5 seconds and is not a mass-PDF API. Prefer the submission UUID.

12. Automatic signing and submission

Automatic signing and automatic submission are independent switches. AutoSignInvoices allows a registered device to sign sessions for that configuration without a person at the browser. AutoSubmitSignedInvoices sends a document after the signature has been verified. With submission turned off, the signed invoice waits for Submit signed. Turning signing on does not turn submission on, and a configuration that does not use client-side signing does not gain a token.

Production sequence

- 1 Finalize and validate the invoice.
- 2 Persist the immutable canonical payload.
- 3 Create a durable signing job and lease it to one device.
- 4 Resolve tenant, company, branch, certificate, and token.
- 5 Sign on the local provider. Do not bypass a middleware that requires a person.
- 6 Verify the signature against the stored payload.
- 7 Persist the signature and the certificate metadata.
- 8 Queue a submission job that is distinct from the signing job.
- 9 Submit through the existing ETA client.

10 Record the actual response. Reconcile timeouts before a second post.

11 Update the invoice status and the audit history.

- Explicit approval remains available. A user can sign from the browser through the localhost bridge, which accepts a session id only. The PIN and the document text do not pass through the browser.
- Multiple tax ids on one computer are selected by the issuer on the document. A missing token is an error for that taxpayer, not a fallback to another company's key.
- Device availability and certificate expiry are checked before the job is treated as retryable forever. An expired certificate is not retried until the certificate is replaced.
- Restart recovery uses the lease. A lease that expires returns the document to the queue. A signature already verified is not signed again. A submission that received HTTP 202 is not posted again.
- Idempotency is the internal id plus the hash of the payload. The authority also rejects an identical body for 10 minutes. Both checks stay in place.

13. E-Invoice throughput

How many e-invoices EGKits can submit per hour depends on which stage is slowest. Signing capacity and submission capacity are different quantities. A token that signs 3,600 invoices an hour does not submit 3,600 HTTP calls if those invoices travel in batches, and a submission API that could carry 50 documents does not move 50 documents until each one has a verified signature.

Figure	Value	Class
Submit requests per hour	Not confirmed	Absent from the governance table
Invoices per request	One or more. No fixed official maximum.	Official
Payload bytes	Returned inside MaximumSizeExceeded. No fixed number on the submit page.	Official behavior, unpublished constant
Signing at 2.0 s, one token	1,800 signatures / hour	Estimate
Signing at 1.0 s, one token	3,600 signatures / hour	Estimate
Signing at 0.5 s, one token	7,200 signatures / hour	Estimate
Agent design note	About 1,500 to 3,000 signatures / hour / token	Engineering note, not a measurement
Submission of 50 already signed documents at 1.5 s, concurrency 1	120,000 documents / hour	Estimate, and 50 is an EGKits default
Get Document if misused for status	2 per second, 7,200 requests / hour	Official query limit
Production measurement	Not measured for this edition	Not a result

The pipeline rate is the minimum of validation, signing, submission, and the status method you actually call. In the illustration above, one token at 1.0 second signs 3,600 invoices an hour, while a 50-document submission at 1.5 seconds could carry 120,000 already signed documents an hour. Signing is the bottleneck. Adding submission concurrency does not remove it. A second token on a second agent is what adds signing capacity, because the token itself is sequential.

Estimate

Recommended safe throughput for a single unattended token: plan inside the agent note of 1,500 to 3,000 signatures per hour, then keep headroom of about 30 percent for PIN failures, certificate prompts, and retries. That is a planning band of roughly 1,050 to 2,100 invoices per hour per token until a preproduction measurement replaces it. Do not publish it as an ETA guarantee.

14. E-Receipts compared with e-invoices

Topic	E-Receipt	E-Invoice
Format	JSON only	JSON or XML, one format per submission
Document id	Taxpayer SHA-256, 64 hex characters	ETA assigns a 26-character uuid at accept time
Authentication	Client credentials plus POS headers and pre-shared key	Client credentials on the identity service
Who may submit	B2C tag, registered POS, allowed branch	Issuer taxpayer. B2B deny tags can forbid ERP submission or cancellation.
Signing	Batch CAdES-BES is specified. Validation is documented as not deployed yet.	Per-document CAdES-BES is required by the invoice v1.0 rules.
EGKits signer	Not the USB signing agent	EGKits.EtaSigningAgent when the key is on a token
Submit API	POST /api/v1/receiptsubmissions	POST /api/v1.0/documentsubmissions/
Bulk in one request	Yes	Yes
Documented batch size	500 receipts and 1.5 MB	Not confirmed as a count. Size comes back in the error.
Published submit rate	Not confirmed	Not confirmed
Published read rate	2 per second for get and search; 3 per second for recent	2 per second for get, details, and search; 3 per second for recent; printout 1 per 5 seconds
Partial success	HTTP 202 accepted and rejected lists	HTTP 202 accepted and rejected lists
Duplicate prevention	10-minute payload hash, plus the receipt UUID chain	10-minute payload hash, plus EGKits signature and submission state
Automation	POS AutoSend and configuration AutoSubmit	Auto sign and auto submit are separate
Main bottleneck	Payload size, unpublished submit throttle, and query limits if status is polled per receipt	USB token signing, then payload size and the unpublished submit throttle
EGKits approach	Reuse the receipt submission service, packer, and Hangfire scanner	Reuse signing sessions, the agent, and the invoice batcher

Not confirmed

Do not copy an invoice limit onto receipts or a receipt limit onto invoices. The 500-receipt and 1.5 MB figures are receipt FAQ figures. The invoice submit page does not repeat them.

15. Capacity planning

```

Requests per hour = 3600 / Average request duration in seconds
Estimated throughput = Concurrent requests x 3600 / Average request duration in seconds
Estimated documents per hour = Documents per request x Successful requests per hour
Signing operations per hour = 3600 / Average signing duration in seconds
Pipeline throughput = minimum(validation, signing, submission, required status processing)

```

These formulas are simplified estimates. Apply the documented batch size, the documented payload ceiling, the documented query rates, the failure rate, and the measured latency. A formula result that exceeds a documented limit is not usable. For receipts the documents-per-request input cannot exceed 500 and the body cannot exceed 1.5 MB. For invoices the documents-per-request input has no retrieved official count, so the EGKits default of 50 must stay labeled as a setting.

Scenario	Inputs	Result	Class
Receipt, sequential, one document	1.0 s, concurrency 1, 1 receipt	3,600 / hour	Hypothetical
Receipt, payload bound	1.0 s, concurrency 1, 187 receipts	673,200 / hour	Hypothetical
Receipt, slower	2.0 s, concurrency 1, 187 receipts	336,600 / hour	Hypothetical
Invoice signing	1.0 s, one token	3,600 / hour	Hypothetical
Invoice submission after signing	1.5 s, concurrency 1, 50 documents	120,000 / hour	Hypothetical, 50 is a setting

15.1 Preproduction load tests

- Use the preproduction API and identity hosts only.

- Use one test taxpayer, one client id, and synthetic documents that pass validation.
- Start at concurrency 1. Record latency, payload bytes, HTTP status, and Retry-After.
- Increase concurrency only while HTTP 429 and HTTP 503 stay absent.
- Stop at the first sustained throttle. That observation belongs to that client id and that environment. It is not a universal ETA limit.
- Do not poll Get Recent Documents or Get Document Printout to manufacture load.
- Do not retry an HTTP 202 body. Do not send an empty client id.

16. EGKits V10 recommendations

Keep the current split. EReceiptSubmissionService and EtaReceiptApiClient own receipts. EtaSubmissionService and EtaApiClient own invoices. EGKits.EtaSigningAgent owns invoice signatures that require a local token. The Hangfire integrations queue already runs the receipt scanner. Do not add a second job system for this guide, and do not change production signing or submission behavior in order to publish it.

- Map each company and branch to its own ETA configuration, client credentials, and environment URL. Map each POS to its serial, pre-shared key, and AutoSend flag.
- Map each invoice issuer tax id to the token that holds its certificate. Do not share a token across tenants.
- Keep signing state and submission state separate on the invoice. A signed document is not a submitted document, and a submitted document is not a valid document.
- Keep the receipt packer on the official 500 and 1.5 MB ceilings. Keep the invoice batcher configurable, and split when ETA returns MaximumSizeExceeded.
- Honor Retry-After on HTTP 429 and HTTP 422. Use 1 to 3 seconds as the documented starting wait for HTTP 503, then back off.
- Bound concurrency per tenant and per client id. The receipt default of 4 is a ceiling to test under, not a quota to fill.
- Store secrets in the tenant configuration and the agent DPAPI store. Do not log tokens, PINs, client secrets, or pre-shared keys.
- Separate Development, Testing, and Production by environment URL. A preproduction success is evidence for that environment only.
- Measure before raising batch size or parallelism. Record the measurement next to the formula inputs so the next estimate can be reproduced.

17. Troubleshooting

Symptom	What it usually means	What to do
HTTP 401	The access token expired or the POS headers do not match the device.	Refresh the token once. Confirm serial, OS version, framework, and pre-shared key. Do not request a token per document.
HTTP 400 BadStructure	The body is not a single receipts or documents collection.	Send the documented JSON shape. Do not mix XML and JSON in one invoice submission.
HTTP 400 MaximumSizeExceeded	The body is larger than the size named in the error.	Split and submit the pieces. Read the supported size from the error.
HTTP 403 IncorrectSubmitter	The token taxpayer is not the issuer, or the intermediary lacks permission.	Submit with the issuer credentials. Check the B2C tag for receipts.
HTTP 422 DuplicateSubmission	The same payload was sent within 10 minutes.	Wait for Retry-After and reconcile the earlier submission UUID. Do not edit the payload only to defeat the hash.
HTTP 429 or 503	The ERP or the platform is over the configured rate.	Wait for Retry-After, or 1 to 3 seconds for 503, then retry the same unsent batch.
Receipt chain rejected	Previous UUID does not match the prior receipt of that POS.	Rebuild the chain for that device. A corrected receipt needs a new UUID and referenceOldUUID.

Symptom	What it usually means	What to do
Invoice stays unsigned	No device, no certificate for that tax id, or the PIN dialog is blocking.	Register the agent, set the PIN through PKCS#11 when the middleware requires it, and confirm the certificate dates.
Cancellation refused	The window ended, the state is wrong, or another document references this one.	Read <code>OperationPeriodOver</code> , <code>IncorrectState</code> , or <code>ActiveReferencingDocuments</code> . Use a credit note when the window has ended.

18. Implementation checklist

Check	Done when
Environment	Preproduction and production URLs are the official hosts, stored per configuration.
Credentials	Client id, secret, and POS pre-shared key are present and are not in source control or in this PDF.
POS	Serial is registered, at most 100 characters, linked to the issuer, and activated.
Receipt batch	Requests stay within 500 receipts and 1.5 MB, and a single oversized receipt is held back.
Receipt UUID	SHA-256 of the canonical receipt, previous UUID chained per device.
Invoice signature	CAdES-BES verified locally before submit, certificate matches the issuer tax id.
Automation switches	Receipt auto-send is per POS. Invoice auto-sign and auto-submit are independent.
Duplicates	HTTP 202 is not retried. HTTP 422 is reconciled. Claims expire and recover.
Status	Submission UUID is stored. Per-document get APIs are not used as the hot path.
Throttle	HTTP 429 and 503 back off. Concurrency is capped per tenant.
Capacity	Any hourly number in operations is labeled with its latency, concurrency, and batch size.

19. Glossary

Term	Meaning in this guide
ETA	Egyptian Tax Authority.
CAdES-BES	The signature format the SDK specifies for invoice documents and for a receipt batch.
eSeal	An approved Egyptian electronic seal certificate used to sign for a taxpayer.
internalId	The identifier from the taxpayer system, returned so the ERP can map the ETA uuid.
longId	The longer identifier used for anonymous document or receipt access.
submissionUUID	The identifier of one HTTP submission, which may contain many documents.
Previous UUID	The receipt UUID of the prior receipt from the same POS.
Reference old UUID	The UUID of an invalid receipt that a corrected receipt replaces.
RIN	The seller registration number printed in the receipt QR payload.
B2C	The taxpayer tag the submit-receipt page requires for receipt submission.

20. Official references

All pages below were retrieved on 26 October 2026 from the Egyptian Tax Authority eInvoicing SDK. The pages do not display one package version number.

Page	URL	Used for
API governance	https://sdk.invoicing.eta.gov.eg/apigovernance/	HTTP 429 and 503, and the per-API rate table
Integration practices	https://sdk.invoicing.eta.gov.eg/integrationpractices/	Token lifetime, callbacks, polling, and anti-patterns
Submit documents	https://sdk.invoicing.eta.gov.eg/einvoicingapi/01-submit-documents/	Invoice submission, HTTP 202, size and duplicate errors
Cancel document	https://sdk.invoicing.eta.gov.eg/einvoicingapi/02-cancel-document/	Cancellation window and error codes
Signature creation	https://sdk.invoicing.eta.gov.eg/signature-creation/	Per-document canonicalization and CAdES-BES

Page	URL	Used for
Invoice v1.0	https://sdk.invoicing.eta.gov.eg/documents/invoice-v1-0/	Signature validation rules
Submit receipt	https://sdk.invoicing.eta.gov.eg/ereceiptapi/02-submit-receipt/	Receipt submission, signatures element, and errors
Authenticate POS	https://sdk.invoicing.eta.gov.eg/ereceiptapi/01-authenticate-pos/	Token request and POS headers
Receipt issuance FAQ	https://sdk.invoicing.eta.gov.eg/receiptissuancefaq/	500 receipts, 1.5 MB, UUID, QR, late submission
Receipt batch signature	https://sdk.invoicing.eta.gov.eg/receipt-batch-signature-creation/	Batch canonicalization, and the note that validation is not deployed
Toolkit initialize	https://sdk.invoicing.eta.gov.eg/toolkitapi/01-initialize/	maximumSubmissionDocumentCount of 500 on the toolkit channel

EGKits behavior cited in this guide is taken from the V10 e-receipt submission service, the receipt packer, the receipt auto-submit options, the invoice batch options, the invoice API client, and the ETA signing agent. Those citations are implementation facts. They are marked as EGKits settings wherever a number is involved.

Not confirmed

This edition does not state a measured production throughput for any tenant. A successful test on one taxpayer, in one environment, is not a universal ETA limit.