SCANEE

LIVRE BLANC

Intégration API et webhooks Scanee

PUBLIÉ PAR

Scanee (scanee.ai)

© 2026 Scanee.ai — Tous droits réservés.

STRUCTURE DU DOCUMENT

Sommaire

	Périmètre	03
01	L'isolation stricte par projet	03
02	Le principe : trois temps	04
03	Étape 1 : récupérer votre URL et votre clé d'API	05
04	Étape 2 : envoyer un document	05
05	Étape 3 : configurer les champs à extraire	11
06	Étape 4 : recevoir les données extraites (webhooks)	12
07	Étape 5 : fiabilité, journal et erreurs	17
08	Le cycle de vie des fichiers	18
09	Aide-mémoire	19
	Conclusion : une boucle, trois appels, aucune saisie	20

© 2026 Scanee.ai — Tous droits réservés.

Périmètre

Ce guide couvre l'API d'ingestion et les webhooks sortants. Les exemples utilisent des clés de la forme `scanee_...` et les espaces réservés `<workspace>` et `<projet>` : remplacez-les par les valeurs de l'onglet « Réception » de votre projet.

En bref

- Chaque projet est un silo : sa clé d'API, ses champs, ses webhooks, son adresse de capture.
- Vous envoyez un document à l'adresse du projet. Scanee répond tout de suite par `202 Accepted`, avant l'extraction.
- Scanee extrait les champs définis dans le projet, puis pousse le résultat vers votre webhook au statut que vous choisissez.
- Six formats acceptés, 25 Mo par fichier, 25 documents par requête.

01 L'isolation stricte par projet

Dans Scanee, tout commence par la création d'un projet : « Factures fournisseurs », « Bons de commande », « Notes de frais », « Relevés bancaires ». Chaque projet est un silo autonome, avec sa propre configuration.

Ce qui est propre à chaque projet	Détail
Un point d'entrée HTTP	`https://<workspace>.docs.scanee.ai/<projet>/ingest`, construit à partir du sigle du projet
Ses clés d'API	Une clé peut être restreinte à un seul projet. Une clé du projet « Factures » ne peut pas déposer de document dans « Notes de frais » : la requête est rejetée en `403 KEY_PROJECT_MISMATCH`

© 2026 Scanee.ai — Tous droits réservés.

Ce qui est propre à chaque projet	Détail
Son schéma de données	Les champs à extraire : texte, montants, dates, listes, tableaux sur plusieurs pages
Sa consigne IA	Des directives métier en langage naturel, par exemple les règles fiscales québécoises pour un projet et ontariennes pour un autre
Ses webhooks et ses correspondances d'export	Où envoyer les données, avec quel schéma JSON et quelle authentification
Son adresse de capture par courriel	Sa propre adresse et sa liste d'expéditeurs approuvés

À retenir. Un projet, c'est une clé, un schéma, un webhook et une adresse de capture. Rien ne traverse la frontière du projet.

02 Le principe : trois temps

Scanee fonctionne en boucle asynchrone : votre système n'attend pas la fin de l'extraction, il est rappelé quand elle est faite.

Temps	Qui	Quoi
1	Votre système (ERP, script, application)	Envoie le fichier à l'adresse du projet : fichier, URL ou base64
2	Le projet Scanee	Extrait les données selon les champs configurés
3	Votre serveur (API, Zapier, Make, base de données)	Reçoit le JSON et les liens signés du fichier, au statut choisi

© 2026 Scanee.ai — Tous droits réservés.

La mise en œuvre tient en cinq étapes : récupérer l'URL et la clé (section 3), envoyer un document (section 4), configurer les champs (section 5), recevoir les données (section 6), gérer les erreurs (section 7).

03 Étape 1 : récupérer votre URL et votre clé d'API

1. Ouvrez votre projet dans Scanee, puis l'onglet **« Réception »**.
2. Vous y trouvez l'URL d'ingestion de votre projet :

```
https://<votre-workspace>.docs.scanee.ai/<votre-projet>/ingest
```

Exemple : `https://acme.docs.scanee.ai/factures/ingest`. L'alias `/documents` fonctionne aussi à la place de `/ingest`.

3. Cliquez sur **« Créer une clé API »**, donnez-lui un libellé (par exemple `ERP-Prod`) et copiez le jeton généré (format `scanee_...`).
4. Fournissez cette clé dans chaque requête, avec l'un de ces deux en-têtes :

```
X-API-Key: scanee_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
Authorization: Bearer scanee_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Traitez la clé comme un mot de passe : gardez-la côté serveur, jamais dans une application mobile ou une page web publique.

04 Étape 2 : envoyer un document

Toutes les méthodes visent le même point d'entrée, celui du projet.

© 2026 Scanee.ai — Tous droits réservés.

Méthode	Format	Quand l'utiliser
A — Envoi direct de fichier	`multipart/form-data`	Fichiers générés localement ou stockés sur votre serveur
B — URL distante ou présignée	JSON	Fichiers déjà sur AWS S3, Google Cloud Storage, Azure Blob ou générés par votre ERP
C — Base64	JSON	Applications mobiles, numériseurs connectés, fonctions serverless sans système de fichiers
D — Courriel entrant (sans code)	Courriel	Chaque projet a son adresse de capture et sa liste d'expéditeurs approuvés (onglet « Réception »)

L'API accepte aussi un fichier unique envoyé en flux binaire brut dans le corps de la requête.

Quotas et règles

Règle	Valeur
Formats acceptés	PDF, PNG, JPEG, WEBP, TIFF, HEIC (HEIF)
Taille maximale par fichier	25 Mo (après décodage pour le base64)
Documents par requête	25 au maximum

Pour les gros volumes, envoyez plusieurs requêtes, ou passez par des URL présignées (méthode B) plutôt que par le base64.

Paramètres communs. `autoProcess` lance l'extraction dès la réception ; il est activé par défaut, `false` le désactive. `metadata.externalId` conserve votre référence métier : elle vous est renvoyée telle quelle dans la réponse et à l'export.

© 2026 Scanee.ai — Tous droits réservés.

Méthode A — Envoi direct de fichier (multipart/form-data)

```bash
curl -X POST "https://<workspace>.docs.scanee.ai/<projet>/ingest" \
  -H "X-API-Key: scanee_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" \
  -F "files=@./facture_01.pdf" \
  -F 'metadata={"externalId": "ERP-INV-1092", "departement": "Comptabilite"}' \
  -F "autoProcess=true"
```

Champ	Rôle
`files`	Le ou les fichiers du lot. Répétez le champ pour envoyer plusieurs documents dans le même appel, jusqu'à 25
`metadata`	Objet JSON libre. `externalId` porte votre référence métier
`autoProcess`	Activé par défaut ; `false` dépose le document sans lancer l'extraction

Au-delà de 25 fichiers, la requête est rejetée en `400 BATCH_LIMIT_EXCEEDED` : scindez en lots de 25.

Node.js / TypeScript (Axios et form-data ; le fichier est lu en flux, sans être chargé entièrement en mémoire) :

```typescript
import axios from 'axios';
import FormData from 'form-data';
import fs from 'fs';

async function sendInvoice() {
  const form = new FormData();
  form.append('files', fs.createReadStream('./facture_01.pdf'));
  form.append('metadata', JSON.stringify({ externalId: 'INV-1092' }));
  form.append('autoProcess', 'true');

  const res = await axios.post(
    'https://<workspace>.docs.scanee.ai/<projet>/ingest',
    form,
    {
```

© 2026 Scanee.ai — Tous droits réservés.

```
      headers: {
        ...form.getHeaders(),
        'X-API-Key': process.env.SCANEE_API_KEY,
      },
    }
  );

  console.log('Document reçu par Scanee :', res.data);
}
```

Python (Requests) :

```python
import json
import os
import requests

url = "https://<workspace>.docs.scanee.ai/<projet>/ingest"
headers = {"X-API-Key": os.environ["SCANEE_API_KEY"]}

with open("facture_01.pdf", "rb") as f:
    files = [("files", ("facture_01.pdf", f, "application/pdf"))]
    data = {
        "metadata": json.dumps({"externalId": "INV-1092"}),
        "autoProcess": "true",
    }
    res = requests.post(url, headers=headers, files=files, data=data)

print(res.status_code, res.json())
```

Méthode B — URL distante ou présignée (JSON)

```bash
curl -X POST "https://<workspace>.docs.scanee.ai/<projet>/ingest" \
  -H "X-API-Key: scanee_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" \
  -H "Content-Type: application/json" \
  -d '{
    "autoProcess": true,
```

© 2026 Scanee.ai — Tous droits réservés.

```
    "documents": [
      {
        "url": "https://mon-bucket.s3.ca-central-1.amazonaws.com/factures/inv_1092.pdf?X-Amz-
Signature=...",
        "filename": "inv_1092.pdf",
        "externalId": "INV-1092"
      }
    ]
  }'
```

Ce que fait Scanee à la réception :

1. **Téléchargement serveur à serveur.** Scanee fait un `GET` sur l'URL fournie, avec un délai maximal de 30 secondes et un plafond de 25 Mo.
2. **Contrôle du type réel.** Scanee ne se fie pas seulement à l'extension ou au type annoncé : il lit les premiers octets du fichier (signature `%PDF` pour un PDF, en-têtes d'image) pour confirmer le format.
3. **Copie dans l'espace du projet.** Le fichier est enregistré, chiffré, dans l'espace isolé du projet.
4. **Réponse** `202 Accepted` avec le lot et les tâches créées.

Si l'URL ne répond pas, ou renvoie une erreur, la requête est rejetée en `422`.

> **Avantage sécurité.** Vous n'ouvrez ni votre réseau ni un accès SFTP. Une URL présignée de 5 ou 10 minutes suffit : une fois le `202 Accepted` reçu, Scanee a sa copie de travail et l'URL peut expirer.

Méthode C — Base64 (JSON)

```bash
curl -X POST "https://<workspace>.docs.scanee.ai/<projet>/ingest" \
  -H "X-API-Key: scanee_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" \
  -H "Content-Type: application/json" \
  -d '{
```

© 2026 Scanee.ai — Tous droits réservés.

```
    "autoProcess": true,
    "documents": [
      {
        "filename": "recu_caisse.jpg",
        "data": "...",
        "externalId": "REC-8801"
      }
    ]
  }'
```

Le champ `data` est une URL de données complète, préfixe MIME inclus (``), suivie de la charge encodée. Le champ `base64` accepte aussi la charge seule. Mêmes limites : 25 Mo par fichier après décodage, 6 formats, 25 documents par requête.

La réponse immédiate : 202 Accepted

Quelle que soit la méthode, la réponse a la même forme :

```JSON
{
  "status": "success",
  "batchId": "batch_9a4f21de88b0",
  "project": "factures",
  "tenantId": "acme",
  "totalFiles": 1,
  "tasks": [
    {
      "taskId": "1024",
      "filename": "facture_01.pdf",
      "status": "pending",
      "trackingUrl": "https://app.scanee.ai/projects/factures/tasks/67d8f1e2700bd45d79cc1918",
      "externalId": "INV-1092"
    }
  ]
}
```

- `taskId` : numéro séquentiel court attribué au document dans votre projet.
- `trackingUrl` : lien direct vers le document dans l'interface Scanee.

© 2026 Scanee.ai — Tous droits réservés.

- `externalId` : votre référence métier, renvoyée telle quelle.

05 Étape 3 : configurer les champs à extraire

Vous ne codez pas de règles d'OCR ni de modèles. Vous définissez les champs voulus dans l'onglet **« Champs »** du projet.

Type de champ	Usage	Exemple de valeur extraite
Input / TextArea	Texte simple ou multiligne	`"INV-2026-089"`, `"Fournitures ACME inc."`
Number	Montant ou quantité	`1437.50` (séparateurs normalisés)
DatePicker	Date	`2026-09-22T00:00:00.000Z` (ISO)
Select / Radio	Choix dans une liste	Devise : `CAD`, `USD`, `EUR`
List (référentiel)	L'IA rapproche le texte de votre liste officielle (fournisseurs, comptes)	`"Hydro-Qc"` → `"Hydro-Québec"`
Table (multipages)	Chaque ligne d'article : description, quantité, prix, total	`[{"desc": "Article A", "qty": 2, "price": 50}]`
Tags (zones 2D)	Coordonnées de la donnée sur le document	`[ymin, xmin, ymax, xmax]` normalisés

Consigne IA du projet. Dans l'onglet « Général » du projet, ajoutez une consigne métier en langage naturel. Par exemple : « Pour les factures avec taxes canadiennes, isoler la TPS (5 %) et la TVQ (9,975 %). Si le numéro de bon de commande figure dans les notes, l'extraire dans po_number. »

© 2026 Scanee.ai — Tous droits réservés.

06 Étape 4 : recevoir les données extraites (webhooks)

Dès qu'un document atteint le statut choisi, Scanee le pousse sur votre serveur par une requête HTTP `POST`.

Configurer le webhook

1. Ouvrez l'onglet **« Exports »** du projet.
2. Cliquez sur **« + Ajouter une configuration »** et choisissez le format **« API / Zapier / Make »** (webhook).
3. Renseignez le nom (par exemple `Webhook ERP Production`), l'URL (par exemple `https://api.votre-domaine.com/webhooks/scanee`) et l'authentification.

Authentification	Effet
Aucune	Pas d'en-tête d'authentification
Bearer	Ajoute `Authorization: Bearer <jeton>`
Clé d'API	Ajoute un en-tête personnalisé, par défaut `x-api-key`, ou le nom de votre choix (par exemple `X-Webhook-Secret`)

Le déclenchement automatique

Statut	Quand
Nouveau (`new`)	Dès que le document entre dans le projet, avant l'extraction
En révision (`reviewing`)	L'IA a extrait les données ; le document est prêt à valider. Pour une automatisation complète, sans intervention
Terminé (`completed`)	Une personne a vérifié le document et l'a passé à Terminé : c'est la validation

Vous pouvez aussi déclencher l'envoi à la main, document par document.

© 2026 Scanee.ai — Tous droits réservés.

La correspondance des champs

Dans l'éditeur d'export, vous associez chaque champ de Scanee à la clé JSON que votre API attend.

Champ source dans Scanee	Clé JSON cible (exemple)	Rôle
ID	`scanee_doc_id`	Numéro séquentiel court (par exemple 1024)
`metadata.externalId`	`external_id`	Votre référence d'origine
`details.<id_fournisseur>`	`vendor_name`	Nom du fournisseur extrait
`details.<id_montant_total>`	`total_amount`	Montant total
`details.<id_lignes>`	`items`	Lignes d'articles, en tableau d'objets ; `table.<id_lignes>.<colonne>` donne seulement les valeurs d'une colonne
`fileData.signedUrl`	`document_download_url`	Lien signé pour télécharger le PDF original
`fileData.annotatedSignedUrl`	`annotated_pdf_url`	Lien du PDF avec les zones de détection

Les clés de sortie sont libres : chaque projet définit son propre schéma. Le même document peut alimenter un ERP avec un schéma, et un logiciel de paie avec un autre, sans transformation de votre côté.

© 2026 Scanee.ai — Tous droits réservés.

Ce que votre webhook reçoit

Le corps est un tableau JSON, un objet par document :

```JSON
[
  {
    "scanee_doc_id": 1024,
    "external_id": "INV-1092",
    "invoice_number": "FA-2026-0042",
    "invoice_date": "2026-09-22",
    "vendor_name": "Fournitures ACME inc.",
    "subtotal": 1250.00,
    "gst": 62.50,
    "qst": 124.69,
    "total_amount": 1437.19,
    "currency": "CAD",
    "items": [
      {
        "description": "Licences logicielles annuelles",
        "quantity": 5,
        "unit_price": 200.00,
        "line_total": 1000.00
      },
      {
        "description": "Frais d'installation",
        "quantity": 1,
        "unit_price": 250.00,
        "line_total": 250.00
      }
    ],
    "document_download_url": "https://storage.scanee.ai/storage/export/...?sig=...",
    "annotated_pdf_url": "https://storage.scanee.ai/storage/export-annotated/665f1c...?
sig=..."
  }
]
```

© 2026 Scanee.ai — Tous droits réservés.

Les liens de téléchargement. `document_download_url` et `annotated_pdf_url` sont signés par Scanee (signature HMAC) : votre serveur ou votre ERP peut faire un `GET` dessus sans s'authentifier à nouveau. Ils ne sont pas devinables, mais ils n'expirent pas : traitez-les comme des données confidentielles et ne les exposez pas publiquement.

Récepteur Node.js / Express

JAVASCRIPT

```
const express = require('express');
const app = express();
app.use(express.json());

app.post('/webhooks/scanee', (req, res) => {
  // 1. Vérifier le secret (authentification « Clé d'API » avec l'en-tête X-Webhook-Secret)
  const webhookSecret = req.headers['x-webhook-secret'];
  if (webhookSecret !== process.env.SCANEE_WEBHOOK_SECRET) {
    return res.status(401).json({ error: 'Unauthorized' });
  }

  // 2. Traiter les documents reçus
  for (const doc of req.body) {
    console.log(`Facture reçue : ${doc.invoice_number}, total : ${doc.total_amount}`);
    // Insérer dans votre ERP ou votre base de données
  }

  // 3. Répondre 200 rapidement (moins de 15 secondes)
  res.status(200).json({ status: 'received' });
});

app.listen(3000, () => console.log('Récepteur de webhooks prêt sur le port 3000'));
```

© 2026 Scanee.ai — Tous droits réservés.

Récepteur Python / FastAPI

```python
import os
from typing import Any, Dict, List

from fastapi import FastAPI, Header, HTTPException, status

app = FastAPI()
SECRET_TOKEN = os.environ["SCANEE_WEBHOOK_TOKEN"]

@app.post("/webhooks/scanee")
async def receive_scanee_webhook(
    payload: List[Dict[str, Any]],
    authorization: str | None = Header(default=None),
):
    # 1. Vérifier le jeton Bearer
    if authorization != f"Bearer {SECRET_TOKEN}":
        raise HTTPException(status_code=status.HTTP_401_UNAUTHORIZED, detail="Invalid token")

    # 2. Traiter les données
    for doc in payload:
        print(f"Document {doc.get('scanee_doc_id')} - montant : {doc.get('total_amount')}")
        # Enregistrer dans votre système

    return {"status": "ok"}
```

Zapier, Make ou n8n, sans écrire de code

1. Créez un déclencheur **Webhooks by Zapier** (ou *Custom Webhook* dans Make ou n8n).
2. Copiez l'URL fournie.
3. Collez-la dans le champ **URL du webhook** de l'onglet **Exports** de Scanee.
4. Envoyez un document de test : les variables mappées apparaissent, prêtes à partir vers Google Sheets, Slack, Airtable, Notion ou votre ERP.

© 2026 Scanee.ai — Tous droits réservés.

07 Étape 5 : fiabilité, journal et erreurs

Codes HTTP de l'API d'ingestion

Code	Cause	Action
`202`	Documents acceptés et mis en file	Conserver le `taskId` ou le `batchId`
`400`	Requête vide, document sans `url` ni `data`, ou plus de 25 documents (`BATCH_LIMIT_EXCEEDED`)	Scinder en lots de 25, vérifier le corps
`401`	Clé absente (`MISSING_API_KEY`) ou invalide, révoquée (`INVALID_API_KEY`)	Vérifier l'en-tête `X-API-Key` ou `Authorization`
`403`	Clé d'un autre espace (`KEY_TENANT_MISMATCH`) ou non autorisée sur ce projet (`KEY_PROJECT_MISMATCH`)	Vérifier le workspace et le sigle du projet dans l'URL
`404`	Espace de travail ou projet introuvable	Vérifier l'URL du projet dans l'onglet « Réception »
`413`	Fichier de plus de 25 Mo	Réduire la résolution ou scinder le document
`415`	Format non accepté	Fournir un PDF, PNG, JPEG, WEBP, TIFF ou HEIC
`422`	URL distante inaccessible (méthode B)	Vérifier que l'URL est valide et pas encore expirée

Si votre serveur webhook ne répond pas

1. **Nouvelles tentatives automatiques.** Si votre serveur répond par une erreur `5xx`, ne répond pas dans les 15 secondes ou est injoignable, Scanee garde le message et réessaie automatiquement, jusqu'à 5 tentatives au total.
2. **Pas de nouvelle tentative sur une erreur 4xx.** Une réponse `4xx` (par exemple `401` ou `404`) est considérée comme définitive : corrigez la configuration, puis renvoyez à la main.

© 2026 Scanee.ai — Tous droits réservés.

3. **Journal complet.** Chaque envoi est enregistré dans l'historique du projet : date et heure, code HTTP de réponse, contenu envoyé et message d'erreur.
4. **Renvoi manuel.** Si votre serveur était arrêté pendant une maintenance, ouvrez l'historique des webhooks dans Scanee et renvoyez la charge utile.

Votre récepteur doit donc répondre `200` vite, idéalement avant de faire le traitement lourd, et savoir recevoir deux fois le même document : dédoublonnez sur `scanee_doc_id` ou `external_id`.

08 Le cycle de vie des fichiers

Quand Scanee reçoit votre document, il en garde une copie de travail chiffrée, isolée dans le projet, pour l'extraction. Vous décidez de ce qu'elle devient.

Suppression automatique

La suppression automatique est désactivée par défaut : les documents restent dans le projet jusqu'à ce que vous les supprimiez. Dans l'onglet **« Stockage »** du projet, section « Rétention et confidentialité des données », vous pouvez l'activer et fixer une période en jours (7 par défaut ; par exemple 30 ou 90), avec l'un de ces deux modes :

Mode	Effet
Supprimer uniquement les fichiers	Le PDF est détruit ; les métadonnées et les données extraites restent consultables
Tout purger	Le fichier, les données extraites et l'historique du document sont supprimés

Votre propre stockage (BYOS)

Vous pouvez brancher votre propre stockage : un dossier Google Drive, Microsoft OneDrive, SharePoint ou Dropbox, ou, au forfait Scale, AWS S3, Google Cloud Storage, Azure Blob Storage ou un SFTP d'entreprise.

© 2026 Scanee.ai — Tous droits réservés.

#	Étape	Détail
1	Extraction	Le traitement du document est terminé
2	Délai avant transfert	Exprimé en heures (24 par défaut), et peut être fixé à 0
3	Cycle de transfert	Les transferts partent par cycles horaires, ce qui laisse le temps de les annuler
4	Transfert	Le document part vers votre compartiment ou votre dossier, puis la copie de travail interne est supprimée

À retenir. Le document finit sa vie chez vous sans disparaître de Scanee : une fois transféré, il reste visible dans le projet, qui conserve le lien vers le fichier. Les connexions suivent le protocole du fournisseur : OAuth, compte de service ou clé privée.

09 Aide-mémoire

Envoyer un document :

```bash
curl -X POST "https://<workspace>.docs.scanee.ai/<projet>/ingest" \
  -H "X-API-Key: scanee_..." \
  -F "files=@mon_document.pdf" \
  -F 'metadata={"externalId": "REF-001"}'
```

En-têtes d'authentification acceptés : `X-API-Key: scanee_...` ou `Authorization: Bearer scanee_...`

Formats et limites : PDF, PNG, JPEG, WEBP, TIFF, HEIC. 25 Mo par fichier, 25 documents par requête.

© 2026 Scanee.ai — Tous droits réservés.

Statuts de déclenchement des webhooks :

Valeur	Statut	Quand
`new`	Nouveau	Dès la réception du document
`reviewing`	En révision	Prêt à valider, pour l'automatisation complète
`completed`	Terminé	Après la vérification par une personne

Conclusion : une boucle, trois appels, aucune saisie

Vous envoyez un document à l'adresse du projet. Scanee l'extrait selon les champs que vous avez définis. Votre serveur reçoit le JSON et les liens signés du fichier. Le reste (nouvelles tentatives, journal, isolation des projets) est pris en charge par la plateforme.

Prochaine étape. Ouvrez l'onglet « Réception » de votre projet, créez une clé `ERP-Prod` et envoyez un premier document de test avec la commande de l'aide-mémoire. Le webhook se branche ensuite dans l'onglet « Exports ».

© 2026 Scanee.ai — Tous droits réservés.

SCANEE

scanee.ai

© 2026 Scanee.ai — Tous droits réservés.