



**AKADEMIA**  
FORMATION

PROGRAMME DE FORMATION

# Automatisation et Workflows IA avec Claude

---

DURÉE

**2 jours**

14 heures

FORMAT

**Inter · Intra**

Présentiel ou distanciel

PUBLIC

**Tout collaborateur**

CERTIFICATION

**Incluse**

À PROPOS DE LA FORMATION

# Automatisation et Workflows IA avec Claude.

DURÉE

**2 jours**

14 heures

FORMAT

**Inter · Intra**

Présentiel ou distanciel

PÉDAGOGIE

**Active**

Petits groupes

CERTIFICATION

**Incluse**

Attestation délivrée

## Objectifs pédagogiques

- Concevoir des architectures d'automatisation adaptées aux contraintes métier (volume, latence, coût)
- Implémenter des pipelines de traitement par lot avec le Batch API (100 000 requêtes, -50% coût)
- Orchestrer des workflows multi-étapes avec tool use, boucles agentiques et patterns de routing
- Intégrer Claude dans des plateformes no-code/low-code (n8n, Zapier, Make) via webhooks et API
- Automatiser le traitement de documents (emails, PDF, données) avec la Files API et le code execution
- Optimiser les coûts avec le prompt caching (jusqu'à -90% sur les lectures) et le choix de modèle adapté
- Mettre en place un monitoring robuste et une gestion d'erreurs pour des workflows de production

PROGRAMME

# Huit modules progressifs pour monter en compétences.

MODULE

## 01.

1H45H

### Patterns d'automatisation avec Claude

#### CONTENU PÉDAGOGIQUE

- Vue d'ensemble des patterns architecturaux : parallelization, chaining, routing, agent loops
- Choix du modèle selon le cas d'usage : Haiku 4.5 (volume/vitesse), Sonnet 4.6 (équilibre), Opus 4.6 (raisonnement)
- Architecture d'un workflow type : déclencheur → pré-traitement → appel Claude → post-traitement → stockage
- Gestion des rate limits, retries exponentiels et files d'attente
- Comparaison workflow déterministe vs agent autonome : arbitrages performance/flexibilité
- Exercice : Concevoir l'architecture d'un pipeline de traitement d'emails entrants avec Claude

## Batch API : traitement de masse à coût réduit

### CONTENU PÉDAGOGIQUE

- Principe du Message Batches API : jusqu'à 100 000 requêtes par batch, réduction de 50% sur tous les prix
- Création d'un batch : MessageCreateParamsNonStreaming, custom\_id, limites (256 MB)
- Tarification batch : Opus 4.6 à \$2.50/\$12.50 MTok, Sonnet 4.6 à \$1.50/\$7.50 MTok, Haiku 4.5 à \$0.50/\$2.50 MTok
- Cycle de vie : soumission → processing → ended (< 1h typiquement), expiration à 24h, résultats 29 jours
- Contenu batchable : vision, tool use, system messages, conversations multi-turn, fonctionnalités beta
- Atelier pratique : Traiter un lot de 50 documents PDF par batch et consolider les résultats en CSV

## Files API et Code Execution pour l'automatisation

### CONTENU PÉDAGOGIQUE

- Files API : upload unique, réutilisation via file\_id dans de multiples requêtes Messages
- Types supportés : PDF, texte, images (JPEG/PNG/GIF/WebP), datasets — 500 MB/fichier, 500 GB/org
- Code Execution tool : exécution Python/Bash en sandbox sécurisée côté Anthropic (v1 et v2)
- Combinaison Files API + Code Execution pour l'extraction de données et la génération de visualisations
- Server Tools Anthropic : web\_search, code\_execution, web\_fetch, tool\_search — exécutés côté serveur
- Exercice : Construire un pipeline d'extraction de données depuis des factures PDF → tableur structuré

## Prompt caching et optimisation des coûts

### CONTENU PÉDAGOGIQUE

- Principe du prompt caching : `cache_control ephemeral`, préfixes cachés pour tâches répétitives
- Deux TTL : 5 min (écriture 1.25x base) et 1h (écriture 2x base), lecture à 0.1x base = -90%
- Tokens minimum cachables : 4096 pour Opus 4.6/Sonnet 4.6/Haiku 4.5, 2048 pour Sonnet 4.5
- Caching automatique (top-level) vs breakpoints explicites (block-level) — maximum 4 breakpoints
- Cache sur les définitions d'outils : optimiser les workflows multi-tools récurrents
- Fenêtre de lookback 20 blocs, règles d'invalidation et monitoring (`cache_read/cache_creation` tokens)
- Atelier pratique : Optimiser un workflow existant avec le caching et mesurer les économies réalisées

## Intégration no-code/low-code : n8n, Zapier, Make

### CONTENU PÉDAGOGIQUE

- Architecture d'intégration : webhooks entrants, appels API sortants, transformation de données
- n8n : nœud HTTP Request vers l'API Claude, workflows event-driven, gestion des credentials et secrets
- Zapier et Make : configuration des Zaps/Scenarios avec Claude comme étape de traitement IA
- Patterns courants : email reçu → analyse Claude → action CRM, formulaire → génération document → envoi
- Webhooks : recevoir des événements externes et déclencher des workflows Claude automatiquement
- Exercice : Créer un workflow n8n complet — réception email → extraction par Claude → classification → action CRM

## Tool Use et boucles agentiques pour l'orchestration

### CONTENU PÉDAGOGIQUE

- Tool Use avancé : définition de schémas JSON (name, description, input\_schema, input\_examples)
- Contrôle avec tool\_choice (auto/any/tool/none), strict mode pour garantir la conformité des inputs
- Boucle agentique : message → tool\_use → tool\_result → analyse → nouvelle action ou réponse finale
- Multi-tools : orchestrer plusieurs outils dans une même conversation (CRM + email + calendrier + base de données)
- Bonnes pratiques : descriptions détaillées (3-4 phrases), namespacing (github\_list\_prs), consolidation d'outils
- Atelier pratique : Développer un agent avec 4+ outils qui collecte des données, les analyse et produit un rapport

## Cas d'usage métier : emails, documents, données

### CONTENU PÉDAGOGIQUE

- Automatisation email : tri intelligent, extraction d'informations clés, génération de réponses contextuelles
- Traitement documentaire : analyse de contrats (PDF + citations), extraction de clauses, comparaison de versions
- Pipeline de données : nettoyage, enrichissement, classification et structuration avec Claude
- Génération de documents par lot : rapports personnalisés, lettres types, synthèses via Batch API
- RAG automatisé : search\_result\_block pour des réponses avec citations depuis une base documentaire interne
- Démonstration : Pipeline complet de traitement de dossiers clients (réception → analyse → synthèse → action)

## Monitoring, erreurs et mise en production

### CONTENU PÉDAGOGIQUE

- Gestion d'erreurs robuste : retries exponentiels, fallbacks entre modèles, circuit breaker pattern
- Logging et traçabilité : journaliser chaque appel API (tokens, latence, coût, statut, cache hit ratio)
- Monitoring des coûts : tableaux de bord, alertes de dépassement, optimisation continue
- Sécurité en production : gestion des clés API, rotation, permissions par workspace
- Scalabilité : dimensionner les workflows pour la montée en charge (batch + async + queue)
- Atelier final : Mettre en production un workflow complet avec monitoring, alertes et documentation opérationnelle

PASSONS À L'ACTION

## Construisons ensemble votre session sur-mesure.

Dites-nous vos contraintes (format, lieu, dates, nombre de participants) et recevez une proposition personnalisée sous 24 heures ouvrées.

### Akademia Formation

SERVICE ADMINISTRATION DES  
VENTES

[adv@akademiaformation.com](mailto:adv@akademiaformation.com)

[www.akademiaformation.com](http://www.akademiaformation.com)

### Devis personnalisé

RÉPONSE SOUS 24 H OUVRÉES

Format inter · intra

Présentiel ou distanciel

— FIN DU PROGRAMME —