Admin

- Consulter les feedbacks
- Resync les cartes d'un TCG
- Valider un utilisateur

Consulter les feedbacks

Consulter les feedbacks

> *Dernière mise à jour : 2026-05-10*

Pourquoi

Voir où l'oracle se plante (et où il assure) pour orienter les améliorations RAG.

`/admin/feedback`

- **Tableau paginé** (20 par page)
- **Filtres** : vote (up/down), gameId
- **Stats agrégées** : total questions, count up/down, avg bestScore

Détail d'un feedback

Clic sur une ligne → panneau `AdminFeedbackDetail` se déploie en dessous :

- **Badges** : langue, intent (rules / synergy / meta / deckbuilding), ID court, date
- **Question + réponse rendue** (markdown)
- **Diagnostics** :
  - Top chunks retrouvés (snippet + page range + sectionTitle + isExtension + score reranker)
  - HyDE généré
  - Timings barre par étape (HyDE / embedding / search / reranker / LLM)

Bouton "Copier (md)"

Génère un dump markdown complet (`buildAnalysisMarkdown`) avec :

- Métadonnées
- Question
- Réponse
- Top chunks détaillés
- HyDE
- Timings

Tu peux le coller dans un éditeur ou dans une session Claude pour analyser un cas spécifique.

Export CSV

Bouton "Export CSV" en haut → POST `/api/admin/feedback/export` → fichier CSV avec toutes les questions filtrées (questionId, vote, gameId, question, answer, bestScore, intent, timestamps).

Utile pour bench RAG, dashboards externes, ou simplement archive.

Alimenter le banc d'éval

Les feedbacks votés down avec commentaire deviennent des cas-test naturels du banc `tests/rag/`. Le workflow propre serait :

1. Sélectionner les feedbacks down avec commentaire détaillé
2. Reformuler en cas-test (`question`, `expected behavior`)
3. Les ajouter dans `tests/rag/dataset.json` (à voir si le format évolue)
4. Lancer `npm run test:rag` pour mesurer la régression

(Pas encore automatisé — pour l'instant c'est manuel.)

Resync les cartes d'un TCG

Resync les cartes d'un TCG

> *Dernière mise à jour : 2026-05-11*

Pourquoi

Quand un TCG sort un nouveau set (ou que tu veux refresh la base après un fix de bug source ou une amélioration de traduction), il faut **re-synchroniser la collection Qdrant** avec la nouvelle source.

Depuis le 2026-05-11, **tout se fait depuis** `/admin` — plus besoin de SSH dans le container pour la majorité des TCG.

Où

`/admin` → section **Bases de cartes**.

Tu y trouves une ligne par TCG, avec :

- Le **nom du jeu** et le nombre de cartes actuellement indexées
- Un **badge de fraîcheur** coloré :
  - **Vert** — synchronisé il y a moins de 7 jours
  - **Jaune** — il y a moins de 30 jours
  - **Rouge** — il y a plus de 30 jours, ou jamais synchronisé
- La **durée de la dernière exécution** (ex : "dernière exécution : 24 min")
- Un **bouton d'action** (voir ci-dessous, deux variantes)

Les deux modes de resync

⚡ Mode "Live" — Riftbound uniquement

Bouton "Resync. (live)" : la source Riftbound est une API Riot interrogeable à chaud. Un clic suffit pour comparer la collection Qdrant avec ce que l'API renvoie en direct, et appliquer le diff (ajouts / mises à jour / suppressions).

La progression s'affiche **inline** sous le nom du deck : "Connexion à la source… → Diff calculé → Embedding 32/45 → Upsert Qdrant → Terminé : +3 ajouts, 12 modifs, 0 supprimé / 1 280 total".

C'est rapide (< 2 minutes) car aucune donnée n'est téléchargée localement.

⚙ Mode "Pipeline" — MTG, Lorcana, FAB, Terraforming Mars, Ark Nova

Bouton "Lancer le pipeline" : ces TCG nécessitent un téléchargement préalable (bulk Scryfall pour MTG, LorcanaJSON pour Lorcana, package npm pour FAB, données locales pour TM/Ark Nova). Un clic enchaîne **automatiquement** toutes les étapes :

1. **Téléchargement** de la source (sauf FAB qui est bundlé)
2. **Extraction / parsing** (filtrage FR, dédoublonnage, etc.)
3. **Embedding + ingestion Qdrant**

Le détail des étapes est affiché dans le **modal de progression** qui s'ouvre au clic. Tu y vois :

- La liste des étapes avec leur statut (en attente / en cours / terminé / erreur) et leur durée
- Une **console scrollable** qui streame en temps réel la sortie des scripts npm (logs `[ingest:mtg] 18 234 cartes chargées…`)
- Le **chrono total** dans le footer
- Un bouton **Fermer** en haut à droite

Tu peux fermer la fenêtre

Le pipeline continue côté serveur même si tu fermes le modal ou navigues ailleurs. Si tu reviens sur `/admin` plus tard (avant la fin du job ou dans les 10 minutes qui suivent), le modal se rouvre automatiquement avec l'historique complet des logs et l'étape en cours.

C'est utile car certains pipelines sont longs :

TCG	Durée typique
FAB	~5-10 min

TCG	Durée typique
Lorcana	~5-10 min
Terraforming Mars	~5 min
Ark Nova	~5 min
MTG	**~25-40 min** (download bulk Scryfall + 30k embeddings)
Riftbound (mode live)	~1-2 min

Une seule resync à la fois

Pour ne pas saturer le GPU d'embedding, **un seul job tourne à la fois** (toutes collections confondues). Si MTG est en cours et que tu cliques sur FAB, le bouton FAB est désactivé et un message t'indique qu'un autre pipeline tourne déjà. Attends la fin du premier (ou ferme le modal et fais autre chose pendant que ça mouline).

Vérifier que ça a marché

À la fin du pipeline :

1. Le badge de fraîcheur passe au **vert** ("À l'instant" / "Il y a 1 min")
2. Le **compteur de cartes** est mis à jour (ex : 1 280 → 1 312)
3. La **durée** s'affiche ("dernière exécution : 24 min")

Pour valider concrètement :

- Va sur `/play` du jeu en question, ouvre une question et tape `@<nom d'une carte du nouveau set>` → la carte doit apparaître dans l'autocomplete
- Pose une question synergy : "*Quelles sont les nouvelles cartes du set X ?*" → l'oracle doit pouvoir les citer

En cas d'erreur

Si une étape échoue, le modal affiche :

- L'**étape en erreur** en rouge dans la liste
- Le **message d'erreur** dans le footer (ex : "Erreur après 3 min — `cards:mtg:download` terminé avec code 1")
- Les **logs complets** dans la console (scroll vers le bas pour voir le traceback)

Le badge de fraîcheur passe au rouge avec un tooltip qui montre les 60 premiers caractères de l'erreur.

Que faire ?

1. Lis l'erreur dans la console : la plupart du temps c'est un service externe en panne (Scryfall down, LorcanaJSON inaccessible) ou un quota dépassé (Haiku rate-limit pour MTG).
2. Attends quelques minutes, relance le pipeline.
3. Si ça persiste, ouvre les logs container côté Unraid pour voir si une dépendance backend est cassée (Qdrant, TEI embeddings).

Cas particulier FAB (toujours utile à savoir)

Le package `@flesh-and-blood/cards` est **embarqué dans l'image Docker**. Le bouton "Lancer le pipeline" ré-ingère ce que ce package contient, mais **pas plus** : si un nouveau set est sorti et que le package n'a pas été bumpé côté repo, le resync ne ramènera rien de neuf.

Workflow pour vraiment ajouter de nouvelles cartes FAB :

1. Update le package localement : `npm update @flesh-and-blood/cards @flesh-and-blood/types`
2. Commit + push → CI Gitea build l'image
3. Pull + restart le container sur Unraid
4. **Ensuite** clic sur "Lancer le pipeline" depuis `/admin`

Valider un utilisateur

Valider un utilisateur

> *Dernière mise à jour : 2026-05-10*

Pourquoi

Tous les nouveaux comptes (self-register ou créés par admin) sont en statut `pending` par défaut. Tant qu'ils sont pending, ils peuvent se connecter mais sont redirigés vers `/pending` sans accès aux features.

Comment

1. Va sur `/admin` (compte admin requis)
2. Section **Utilisateurs**
3. Filtre éventuellement par username
4. Clique **Confirmer** sur le compte concerné
5. La ligne `users.role` passe de `pending` à `user`

À partir de là, l'utilisateur peut accéder à `/`, `/play`, `/history`, etc.

Permissions complémentaires

- **`canAddGames`** : par défaut `false` pour les nouveaux users. Toggle disponible sur le compte (`/admin` → Users → bouton "Autoriser ajout de jeux"). Quand `true`, l'utilisateur a accès à `/add-game`. Les admins ont toujours `canAddGames` à `true` implicitement (bypass).

Notification mail

Au moment du register, l'app envoie automatiquement un mail de notification à toi (l'admin) si SMTP configuré (`SMTP_HOST` non vide). Le mail contient le username + lien direct vers `/admin`.

Pour rappel, la conf SMTP est optionnelle — si tu n'as pas configuré `SMTP_HOST` dans `.env`, aucun mail n'est envoyé, tu dois checker `/admin` manuellement.

Supprimer un utilisateur

`/admin` → Users → bouton Supprimer (avec ConfirmDialog).

Effets :

- Suppression de la ligne `users`
- Suppression de toutes ses lignes `questions` (cascade)
- Réassignation des jeux qu'il avait ajouté (`addedBy`) à toi (l'admin courant)