

Riftbound

Riftbound

📅 Dernière mise à jour : 2026-05-10

TCG Riot Games (univers League of Legends).

Source

- **API Riot** : `https://riftbound.leagueoflegends.com/en-us/card-gallery` (JSON live, pas de bulk download)
- **Collection Qdrant** : `riftbound-cards`
- **Cache local** : `/app/data/riftbound-cards/cache.json` (post-fetch)

Code

Fichier	Rôle
<code>src/services/cards/sources/riftbound.ts</code>	CardSource (load depuis cache)
<code>scripts/riftbound-cards/fetch-cards.ts</code>	Appelle l'API Riot, écrit cache JSON
<code>scripts/riftbound-cards/normalize.ts</code>	Nettoyage HTML → domaines, énergie, types, tags
<code>scripts/riftbound-cards/ingest.ts</code>	Push Qdrant
<code>scripts/riftbound-cards/test-retrieve.ts</code>	Debug retrieval Qdrant
<code>scripts/riftbound-cards/link-game.ts</code>	Lie une ligne games
<code>services/meta/mobalytics-riftbound.ts</code>	Tier list Mobalytics (scrape)
<code>services/meta/riftboundstats.ts</code>	Tournois RiftboundStats (API)

Payload Qdrant riftbound-cards

```
{
  id: pointId,
  name: string,
  name_en: string,
  set_label: string,
  rarity: string,
  card_type: 'Unit' | 'Champion Unit' | 'Spell' | 'Gear' | 'Battlefield' | 'Legend' | 'Rune',
  card_domains: ('Fury' | 'Calm' | 'Mind' | 'Body' | 'Chaos' | 'Order')[],
  energy: number,
  might: number,           // valeur d'attaque
  text: string,            // abilities
  tags: string[],
  image_url: string,       // URL officielle Riot
}
```

Symboles UI

Token	Symbole	PNG
<code>:rb_might:</code>	Might (force)	<code>/riftbound-icons/might.png</code>
<code>[rune_fury]</code>	Domain Fury	<code>/riftbound-icons/fury.png</code>
<code>[rune_calm]</code>	Domain Calm	<code>/riftbound-icons/calm.png</code>
<code>[rune_mind]</code>	Domain Mind	<code>/riftbound-icons/mind.png</code>
<code>[rune_body]</code>	Domain Body	<code>/riftbound-icons/body.png</code>
<code>[rune_chaos]</code>	Domain Chaos	<code>/riftbound-icons/chaos.png</code>
<code>[rune_order]</code>	Domain Order	<code>/riftbound-icons/order.png</code>

PNG officiels Riot press-kit (cf. mémoire `project_riftbound_support.md`, commit `3de9cdd`).

Helper : `frontend/src/lib/riftbound-symbols.ts`. Activé si `game.hasCardDatabase === 'riftbound-cards'`.

Méta-game

Deux sources :

Mobalytics (tier list)

- `META_RIFTBOUND_ENABLED=true`
- Sync hebdo (fréquence configurable)
- Service `services/meta/mobalytics-riftbound.ts`

RiftboundStats (tournois)

- `META_RIFTBOUND_TOURNAMENT_ENABLED=true`
- Top placement max : `META_RIFTBOUND_TOURNAMENT_MAX_PLACEMENT=4`
- Format ID par défaut : `META_RIFTBOUND_TOURNAMENT_FORMAT_ID=2` (Spiritforged)
- Rate-limit : `META_RIFTBOUND_RATE_LIMIT_MS=300` (1 req/s safe)
- Service `services/meta/riftboundstats.ts`

Deckbuilding

- Spec par défaut : `40 mainboard + 0 sideboard + 12 runes + 3 battlefields + 3 gear, max 3 par carte`
- Pas de filtre Qdrant `legal_formats` (collection mono-format)
- Post-filtre TS : `card_domains ⊆ spec.domains` (multi-domain decks)
- 7 domaines à combiner : Fury / Calm / Mind / Body / Chaos / Order

Multi-query Riftbound

`retrieve/multi-query-riftbound.ts` : Haiku décompose la question en JSON `{domains, energyMax, types, themes, legend}` → filters Qdrant natifs.

Autocomplete @card

Pattern @ (mention-style) choisi 2026-04-13 (cf. mémoire `project_riftbound_cards_ux.md`). Première implémentation TCG du pattern, ensuite généralisée à MTG/FAB/Lorcana.

Workflow update

Source live API → workflow simple : `npm run cards:rftbound:fetch` puis `npm run cards:rftbound:ingest`. Pas de rebuild image Docker nécessaire (sauf si tu changes le code de normalisation).

Détails dans `mettre-a-jour-cartes.md`.

Revision #1

Created 2026-05-10 15:20:26 UTC by thymon

Updated 2026-05-10 15:20:26 UTC by thymon