

Modelisation conceptuelle orientee NoSQL

1. Objectif du modele

Cette modelisation a pour objectif de structurer les donnees du projet GeoChallenge Tracker dans un environnement NoSQL, plus particulierement MongoDB. Elle permet de représenter les entites principales de l'application, leurs proprietes et leurs relations logiques, tout en respectant la souplesse propre aux bases orientees documents.

2. Approche retenue

Plutot qu'un MCD classique issu de la methode Merise, nous adoptons ici une approche orientee document, centree sur :

- Les collections (equivalents d'entites)
- Les champs (attributs)
- Les relations logiques via des references (_id)
- Les regles metier (obligations de structure, coherence, unicite)

Cette approche vise a rendre la modelisation plus proche des besoins applicatifs reels et des schemas de requetage.

3. Collections principales

- * users : comptes utilisateurs avec role
- * caches : geocaches connues (metadonnees)
- * found_caches : caches deja trouvees par utilisateur
- * challenges : defis geocaching crees par la communaute
- * tasks : conditions individuelles composant un challenge
- * progress : suivi d'un utilisateur sur une tache
- * targets : caches encore disponibles utiles a une tache

4. Relations logiques (vue simplifiee)

- * found_caches.user_id -> users._id (1 -> N)
- * found_caches.cache_id -> caches._id (1 -> N)

Modelisation conceptuelle orientee NoSQL

- * challenges.user_id -> users._id (1 -> N)
- * tasks.challenge_id -> challenges._id (1 -> N)
- * progress.user_id -> users._id (1 -> N)
- * progress.task_id -> tasks._id (1 -> 1)
- * targets.user_id -> users._id (1 -> N)
- * targets.task_id -> tasks._id (1 -> N)
- * targets.cache_id -> caches._id (1 -> 1)

5. Contraintes metier

- Un challenge doit comporter au moins une tache.
- Une progression est toujours liee a une tache.
- Une tache est toujours liee a un challenge.
- Les caches trouvees sont exclues des cibles (targets).
- Les series temporelles sont stockees dans progress.history.