NextGes

Document technique du service - Version 1

DELOEIL Tristan

ESGI - École Supérieure de Génie Informatique
Département Informatique

Reims, 25 avril 2025

Table des matières

Présentation

1.1 Introduction

Ce document présente la conception, l'architecture et les détails techniques de l'**API NextGes**, un back-end servant d'interface à l'intranet d'une organisation scolaire. L'objectif est de fournir un outil centralisé pour l'accès aux informations essentielles (étudiants, cours, classes, logs, etc.), tout en garantissant sécurité, scalabilité et maintenabilité.

Cette API est développée dans le cadre d'un projet pour les matières **MongoDB** et **Express** pour l'établissement **ESGI**.

Le code de l'application est disponible sur GitHub au lien suivant `https://github.com/FlenderGit/esgi-nextges-api`

1.2 Contexte et objectifs

- **Contexte** : Intranet universitaire existant nécessitant modernisation des interfaces et centralisation des données.
- **Objectifs** :
  - Simplifier l'accès aux données via une API RESTful.
  - Implémenter un système de permissions granulaire (scopes JWT).
  - Assurer la robustesse, la sécurité et la conformité aux meilleures pratiques.

1.3 Architecture générale

1.3.1 Diagramme d'architecture

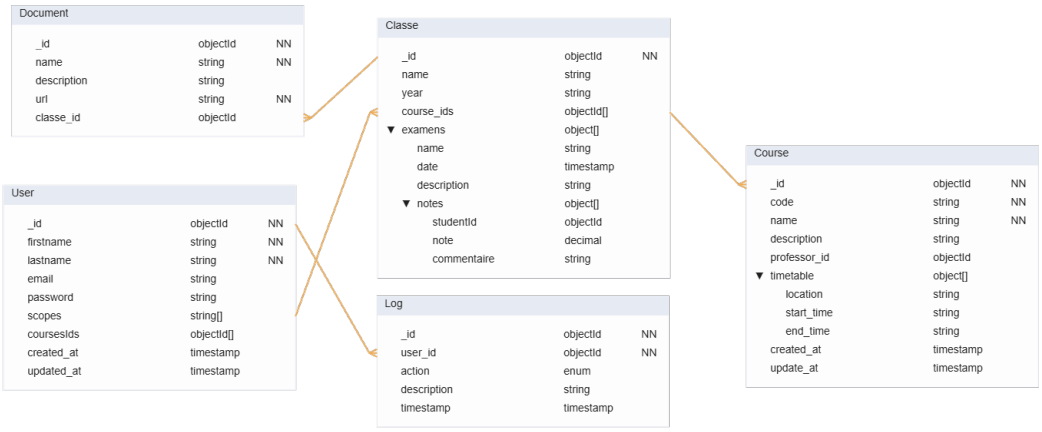

FIGURE 1.1 – Architecture générale de l'API

Cette architecture de base de données illustre la structure de l'API NextGes. Elle est conçue pour être évolutive et flexible, permettant d'ajouter facilement de nouvelles fonctionnalités et de nouveaux types de données à l'avenir. Elle utilise des relations entre les tables ainsi que les arrays imbriqués pour les données indépendantes. Cela permet de réduire le nombre de requêtes nécessaires pour récupérer les données et d'améliorer les performances globales de l'application.

1.3.2 Composants principaux

- **API RESTful** (Express 5.1.0)
- **Serveur** (Node.js 20.4.0)
- **Base de données** (MongoDB 6.0.5)

1.4 Stack Tecknologique

1.5 Système d'Authentification et d'Autorisation

1.5.1 Authentification JWT

Le système d'authentification repose sur le protocole JWT (JSON Web Token). Lorsqu'un utilisateur se connecte, un token est généré et envoyé au client. Ce token contient des informations sur l'utilisateur et ses permissions (scopes). Le client doit inclure ce token dans chaque requête pour accéder aux ressources protégées de l'API.

Voici les informations contenues dans le token :

- **iss** : Identifiant de l'émetteur (l'API).
- **aud** : Identifiant du destinataire (l'utilisateur).
- **exp** : Date d'expiration du token.
- **iat** : Date de création du token.
- **sub** : Identifiant de l'utilisateur.
- **scopes** : Liste des permissions accordées à l'utilisateur, sous forme de scope. Voir 1.5.2 pour plus de détails.
- **roles** : Rôle de l'utilisateur (étudiant, enseignant, administrateur).
- **email** : Adresse e-mail de l'utilisateur.

Le token est signé avec uen clef secrète et l'algorithme HS256.

1.5.2 Gestion des scopes

Le système de permissions repose sur des scopes, qui sont des chaînes de caractères sous le format `action:ressource`. Par exemple, `read:student` permet à un étudiant de lire ses propres informations. Les scopes sont définis dans le fichier de configuration de l'API et peuvent être modifiés selon les besoins. Le système inclue aussi les jokers `"*"` et `"all"` pour permettre de donner accès à toutes les actions d'un scope ou à tous les scopes d'un utilisateur. Par exemple `*:student` permet d'accéder à toutes les actions du scope `student`, et `all:all` permet d'accéder à toutes les actions de tous les scopes.

1.6 Définition des Endpoints API

Voir le document relatif aux endpoints pour plus de détails.

1.7 Schéma de la base de données

Toutes les models de données ont les attributs **id** (identifiant unique), **createdAt** (date de création) et **updatedAt** (date de mise à jour). Ces attributs sont gérés automatiquement.

1.7.1 Utilisateur

- **firstName** : Prénom de l'utilisateur.
- **lastName** : Nom de l'utilisateur.
- **email** : Adresse e-mail de l'utilisateur.

- **password** : Mot de passe de l'utilisateur (hashé par brypt avec un secret).
- **role** : Rôle de l'utilisateur (étudiant, enseignant, administrateur).
- **permissions** : Liste des permissions accordées à l'utilisateur, sous forme de scope.
- **classeId** : Identifiant de la classe à laquelle l'utilisateur appartient (pour les étudiants sinon null).

Sur cette table, il y a un index unique sur l'attribut **email** pour éviter les doublons. Et des indexs sur les attributs **role** et **classeId** pour améliorer les performances des requêtes.

1.7.2 Classe

Une classe représente un groupe d'étudiants. Elle est définie par les attributs suivants :

- **name** : Nom de la classe.
- **year** : Année scolaire de la classe.
- **studentIds** : Liste des identifiants des étudiants appartenant à la classe.
- **examens** : Liste imbriquée des examens de la classe. Les examens sont définis par les attributs suivants :
  - **date** : Date de l'examen.
  - **name** : Matière de l'examen.
  - **description** : Durée de l'examen.
  - **notes** : Liste imbriquée des notes des étudiants. Les notes sont définies par les attributs suivants :
    - **studentId** : Identifiant de l'étudiant.
    - **note** : Note de l'étudiant.
    - **commentaire** : Commentaire de l'enseignant sur la note.

1.7.3 Cours

Un cours représente une matière enseignée dans une classe. Il est défini par les attributs suivants :

- **name** : Nom du cours.
- **code** : Code du cours. Défini sous le format XXX-000. Par exemple, `INF-101`.
- **description** : Description du cours.
- **enseignantId** : Identifiant de l'enseignant responsable du cours.
- **classeId** : Identifiant de la classe à laquelle le cours est associé.
- **timetable** : Liste imbriquée des horaires du cours. Les horaires sont définis par les attributs suivants :
  - **date** : Date du cours.
  - **startTime** : Heure de début du cours.
  - **endTime** : Heure de fin du cours.
  - **location** : Salle de cours.

1.7.4 Document

Un document représente un fichier associé à un cours. Il peut être défini pour un classe spéfique ou pour tous les étudiants d'une classe. Il est défini par les attributs suivants :

- **title** : Titre du document.
- **description** : Description du document.
- **url** : Chemin d'accès au document.
- **uploadBy** : Identifiant de l'utilisateur ayant téléchargé le document.
- **classeId** : Identifiant de la classe à laquelle le document est associé. Si null, le document est associé à tous les étudiants de la classe.