

Documentation du Projet SCHOOLO

Table of Contents

Introduction	1
Guide d'Installation et Configuration	1
Structure des Applications	1
Application : quizz	1
Application : users	2
Application : lessons	2
Déploiement	3
Diagrammes	3

Introduction

Ceci est la documentation générée automatiquement pour le projet SCHOOLO.

Guide d'Installation et Configuration

Pour installer et configurer le projet, assurez-vous d'avoir Python et les dépendances listées dans requirements.txt. Voici les principales étapes d'installation :

1. Clonez le dépôt et installez les dépendances.
2. Configurez les variables d'environnement essentielles (comme `DEBUG`, `DATABASES`, etc.).

Structure des Applications

Application : quizz

Modèles

Aucun modèle trouvé.

Vues

- quizz_list(request):
 - Pas de description disponible.
- quizz_detail(request):
 - Pas de description disponible.
- add_quizz(request):

- Pas de description disponible.

Endpoints d'API

- `path('', views.quiz_list, name='quiz_list')`,
- `path('<int:quiz_id>/', views.quiz_detail, name='quiz_detail')`,
- `path('add/', views.add_quiz, name='add_quiz')`,

Application : users

Modèles

Aucun modèle trouvé.

Vues

- `list_teachers(request):`
 - Pas de description disponible.

Endpoints d'API

- `path('api/teachers/', views.list_teachers, name='list-teachers')`,

Application : lessons

Modèles

Aucun modèle trouvé.

Vues

- `subjects_list(request):`
 - Pas de description disponible.
- `lessons_list(request):`
 - Pas de description disponible.
- `lesson_detail(request):`
 - Pas de description disponible.
- `add_lesson(request):`
 - Pas de description disponible.
- `modify_lesson(request):`
 - Pas de description disponible.

Endpoints d'API

- `path('api/lessonslist/', views.LessonListView.as_view(), name='lesson-list')`,
- `path('api/lessonslist/subject/<str:subject>', views.LessonListView.as_view(), name='lessons-by-subject')`,
- `path('api/lessonslist/detail/<int:lesson_id>', views.LessonDetailView.as_view(), name='lesson-detail')`,
- `path('api/lessonslist/add', views.CreateLessonView.as_view(), name='create-lesson')`,
- `path('', views.subjects_list, name='subjects_list')`, # Liste des sujets sur l'entrée dans l'app
- `path('list', views.lessons_list, name='lessons_list')`, # Liste des leçons
- `path('<int:lesson_id>', views.lesson_detail, name='lesson_detail')`, # Détail d'une leçon
- `path('add/', views.add_lesson, name='add_lesson')`, # ajout d'une leçon
- `path('lessons/modify/<int:lesson_id>', views.modify_lesson, name='modify_lesson')`, # modifier une leçon

Déploiement

Pour déployer le projet en production :

1. Configurez `DEBUG=False` et `ALLOWED_HOSTS`.
2. Utilisez `python manage.py migrate` pour appliquer les migrations.
3. Utilisez un serveur compatible WSGI, tel que Gunicorn, et configurez un proxy inverse comme Nginx pour gérer les requêtes HTTP.

Diagrammes

Diagramme des relations de modèle :

