

# Documentation API – Fiche de Présence

## 1. Présentation du projet

Cette application permet de gérer la présence des élèves ("ninjas") dans des sessions CoderDojo, via une API REST sécurisée avec JWT.

### *Fonctionnalités principales :*

- Gestion des lycées
- Gestion des équipes CoderDojo par lycée
- Gestion des mentors (profils)
- Gestion des ninjas (élèves)
- Gestion des sessions de formation
- Gestion des présences avec contrôle de doublons

## 2. Modèles principaux

Modèle	Champs	Type	Description
Lycée	nom	CharField	Nom du lycée
	adresse	CharField	Adresse du lycée (optionnel)
Equipe	nom	CharField	Nom de l'équipe CoderDojo
	lycee	OneToOneField	Lycée associé à l'équipe
Profil	user	OneToOneField	Utilisateur Django associé
	equipe	ForeignKey	Équipe du mentor
Ninja	nom	CharField	Nom de l'élève
	prenom	CharField	Prénom de l'élève
	lycee	ForeignKey	Lycée de l'élève
	classe	CharField	Classe de l'élève
Session	theme	CharField	Thème de la session
Presence	ninja	ForeignKey	Ninja concerné
	session	ForeignKey	Session concernée
	lycee	ForeignKey	Lycée associé
	present	BooleanField	Présent ou absent
	fait_par	ForeignKey	Mentor ayant enregistré la présence
	date	DateField	Date de la présence
	timestamp	DateTimeField	Date et heure d'enregistrement automatique

### *Contraintes importantes :*

- Un ninja ne peut être enregistré plusieurs fois pour la même session, même à des dates différentes.

- Le ninja doit appartenir au même lycée que le mentor pour enregistrer sa présence.

### 3. Endpoints API

Endpoint	Méthodes	Description
/api/lycees/	GET, POST	Liste et création des lycées
/api/ninjas/	GET, POST	Liste et création des ninjas filtrés par le lycée du mentor connecté
/api/sessions/	GET, POST	Liste et création des sessions
/api/presences/	GET, POST	Liste et création des présences avec contrôle de doublon et cohérence d
/api/profils/	GET, POST	Liste et création des profils de mentors
/api/token/	POST	Obtenir un token JWT pour l'authentification
/api/token/refresh/	POST	Rafraîchir un token JWT

### 4. Détails des endpoints

#### 4.1 Lycees

GET /api/lycees/ : Liste tous les lycées. POST /api/lycees/ : Créer un nouveau lycée.

##### Exemple JSON (POST)

```
{
  "nom": "Lycée Saint Joseph",
  "adresse": "Antananarivo"
}
```

#### 4.2 Ninjas

GET /api/ninjas/ : Liste des ninjas du lycée du mentor connecté. POST /api/ninjas/ : Créer un nouveau ninja.

##### Exemple JSON (POST)

```
{
  "nom": "Rakoto",
  "prenom": "Jean",
  "lycee": 1,
  "classe": "CM2"
}
```

#### 4.3 Sessions

GET /api/sessions/ : Liste toutes les sessions. POST /api/sessions/ : Créer une nouvelle session.

##### Exemple JSON (POST)

```
{
  "theme": "Initiation à Python"
}
```

### 4.4 Presences

GET /api/presences/ : Liste des présences filtrées par le lycée du mentor connecté. POST /api/presences/ : Créer une nouvelle présence.

#### Exemple JSON (POST)

```
{
  "ninja": 1,
  "session": 2,
  "lycee": 1,
  "present": true
}
```

#### Réponse en cas d'erreur (doublon)

```
{
  "detail": "■■■ Ce ninja est déjà enregistré pour cette session."
}
```

#### Réponse en cas d'erreur (cohérence lycée)

```
{
  "detail": "Le ninja n'appartient pas à ce lycée."
}
```

### 4.5 Profils

GET /api/profils/ : Liste tous les profils de mentors. POST /api/profils/ : Créer un profil pour un mentor.

#### Exemple JSON (POST)

```
{
  "user": 3,
  "equipe": 1
}
```

### 4.6 Authentification JWT

POST /api/token/ : Obtenir un token d'accès et un token de rafraîchissement. POST /api/token/refresh/ : Rafraîchir un token d'accès.

#### Exemple JSON (POST)

```
{
  "username": "mentor1",
  "password": "motdepasse"
}
```

## 5. Notes supplémentaires

- Tous les endpoints sont RESTful et sécurisés avec JWT.
- Les champs timestamp et fait\_par sont automatiques et en lecture seule.

- Les erreurs sont retournées au format JSON avec un message clair.
- Les données sont filtrées pour chaque mentor en fonction de son lycée.