

| | | |
|--|---|----------------------|
|  ÉCOLE MAROCAINE DES SCIENCES DE L'INGÉNIEUR <small>Membre de</small> HONORIS UNITED UNIVERSITIES | Contrôle Architecture JEE - TP | Durée : 1h30' |
| | Filière : 4 Année IIR / Classe : | 18/04/2025 |

Mini service de gestion de campagnes de dons et transaction :

1. Quelques consignes :

- **Environnement** : IDE IntelliJ, Java 17+, Spring Boot, Spring Data JPA et base H2 en mémoire
- **Livrables** : code source Git avec le fichier README indiquant – pour chaque point – ce qui a été fait.

2. Liste des tâches :

- Modélisation : Entités JPA
 - **Campagne**{ id, nom, objectifMontant, dateDebut, dateFin }
 - **Donation**{ id, campagne (ManyToOne), nomDonateur, montant, date }
- Répertoires :
 - **CampagneJPA** (hérite de JpaRepository)
 - **DonationJPA** (hérite de JpaRepository)
- DTOs :
 - **DonDTO** { id, nomCampagne, nomDonateur, montant, date }
- Projection :
 - **CampagneResume**{ Long getId(); String getNom(); BigDecimal getObjectifMontant(); }
- Services :
 - **ServiceCampagne** : récupérer les campagnes actives
 - **ServiceDon** : enregistrer un don, transformer une entité en DonDTO
- API REST :
 - GET */api/campagnes/actives* → Liste des campagnes actives (projection)

- POST */api/campagnes/{id}/dons* → Enregistrer un don (reçoit un DonDTO)
- Validation & gestion des erreurs :
 - Validation avec @Valid (ex : montant > 0, nom non vide)
- Tests attendus :
 - Test unitaire du ServiceCampagne
 - Test d'intégration de l'API
- Documentation avec Swagger (bonus !)

Détails des tâches à réaliser :

Entités JPA :

- Campagne { id, nom, objectifMontant, dateDebut, dateFin }
- Don { id, campagne (ManyToOne), nomDonateur, montant, date }

Répertoires :

- CampagneRepository (hérite de JpaRepository)
- DonRepository (hérite de JpaRepository)

Services :

- ServiceCampagne : récupérer les campagnes actives
- ServiceDon : enregistrer un don, transformer une entité en DonDTO

DTOs :

- DonDTO { id, nomCampagne, nomDonateur, montant, date }

Projection :

- CampagneRésumé { Long getId(); String getNom(); BigDecimal getObjectifMontant(); }

API REST :

- GET /api/campagnes/actives → Liste des campagnes actives (projection)
- POST /api/campagnes/{id}/dons → Enregistrer un don (reçoit un DonDTO)

Validation & gestion des erreurs :

- Validation avec @Valid (ex : montant > 0, nom non vide)
- Gestion des erreurs avec @ControllerAdvice

Tests attendus :

- Test unitaire du ServiceCampagne
- Test d'intégration de l'API DonController