Documentation Technique

API CollabDev - Plateforme Collaborative de Gestion de Projets

Version 1.0 | Spring Boot 3.x | API REST



Oue d'ensemble de l'application

API CollabDev est une plateforme collaborative permettant la gestion de projets participatifs avec un système de gamification basé sur des coins virtuels et des badges de récompense.

Acteurs principaux

- Contributeurs : Développeurs, designers qui participent aux projets
- Porteurs de projet : Initiateurs d'idées de projets
- Gestionnaires : Responsables de la gestion opérationnelle des projets
- Administrateurs : Gestionnaires de la plateforme

Concepts clés

- Système de coins : Monnaie virtuelle pour débloquer l'accès aux projets
- Badges : Récompenses basées sur les contributions validées
- Quiz d'évaluation : Tests de compétences pour la sélection des participants
- Templates de projets : Modèles prédéfinis pour faciliter la création

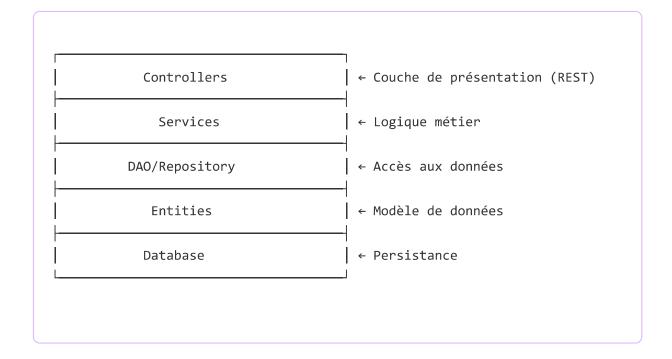
Architecture technique

Stack technologique

Framework : Spring Boot 3.xBase de données : JPA/Hibernate

Documentation: OpenAPI 3 (Swagger)
 Validation: Bean Validation (JSR-303)
 Architecture: REST API avec pattern MVC

Structure des couches



Packages principaux

Package	Description
controllers/	Endpoints REST
services/	Logique métier
dao/	Interfaces d'accès aux données
models/	Entités JPA et énumérations
dto/	Objets de transfert de données
config/	Configuration (OpenAPI, initialisation)
exception/	Gestion globale des erreurs

Workflow complet de l'application

1. Traitement d'une requête HTTP

```
Requête HTTP → Controller → Service → DAO → Database
  Response DTO ← Mapping ← Business Logic ← Entity
```

Séquence détaillée :

- 1. **Réception** : Le contrôleur reçoit la requête HTTP
- 2. Validation: Validation des données d'entrée (Bean Validation)
- 3. **Transformation** : Conversion DTO → Entité métier
- 4. **Traitement** : Exécution de la logique métier dans le service
- 5. **Persistance**: Interaction avec la base via les DAO
- 6. Notification : Envoi de notifications si nécessaire
- 7. **Réponse** : Transformation Entité → DTO de réponse
- 8. **Retour** : Envoi de la réponse HTTP avec le bon code de statut

2. Gestion des erreurs

```
Exception → GlobalExceptionHandler → ErrorResponse → HTTP Status
```

3. Système de notifications

Action métier → NotificationService → Création notification → Diffusion



Fonctionnalités principales



Authentification et gestion des utilisateurs

- Inscription avec attribution automatique de 100 coins et 10 points d'expérience
- Connexion avec validation des identifiants
- Gestion de profil et changement de mot de passe
- Types d'utilisateurs : Contributeur, Administrateur



Gestion des projets

- Proposition de projets par les contributeurs
- Validation administrative avec workflow d'approbation
- Filtrage par domaine (WEB, MOBILE, DESIGN) et secteur (SANTE, EDUCATION, etc.)
- **Gestion du cycle de vie** : EN ATTENTE → OUVERT → EN COURS → TERMINER
- Attribution de niveaux de complexité (DEBUTANT, INTERMEDIAIRE, AVANCE, EXPERT)

Système de participation

- Demandes de participation avec quiz d'évaluation
- Validation par gestionnaires avec scoring automatique
- Déverrouillage payant selon le niveau du projet
- Gestion des rôles : DEVELOPPEUR, DESIGNER, GESTIONNAIRE

Gestion des tâches et contributions

- **Définition de fonctionnalités** par les gestionnaires
- Réservation de tâches par les contributeurs
- Soumission de contributions (liens ou fichiers)
- Validation/rejet par les gestionnaires
- Suivi des statuts : A_FAIRE → EN_COURS → TERMINE



🙎 Système de gamification

- Coins virtuels pour débloquer l'accès aux projets
- Badges de récompense : DEBUTANT, BRONZE, ARGENT, OR, PLATINE

- Attribution automatique basée sur les contributions validées
- Paramétrage flexible des récompenses par les administrateurs



Système de quiz et évaluation

- Création de questionnaires pour projets ou templates
- Évaluation automatique avec scoring
- Détermination du niveau de compétence
- Commentaires personnalisés selon les performances



Notifications

- Notifications push pour les événements importants
- Système d'observateur pour la diffusion
- Historique des notifications par utilisateur



Guide d'utilisation des API REST

Scénario 1 : Inscription et première connexion

Étape 1 : Inscription d'un nouveau contributeur

```
POST /api/v1/auth/register Content-Type: application/json { "nom": "Dupont",
    "prenom": "Jean", "telephone": "0123456789", "email":
    "jean.dupont@email.com", "password": "motdepasse123" }
```

Étape 2 : Connexion

```
POST /api/v1/auth/login Content-Type: application/json { "email":
   "jean.dupont@email.com", "password": "motdepasse123" }
```

Scénario 2 : Création et validation d'un projet

Étape 1 : Proposer un nouveau projet

```
POST /api/v1/projets/contributeur/1 Content-Type: application/json { "titre":
"Application de gestion de tâches", "description": "Une application web pour
gérer les tâches quotidiennes", "domaine": "WEB", "secteur": "PRODUCTIVITE",
"urlCahierDeCharge": "https://example.com/cahier-charges.pdf" }
```

Étape 2 : Validation par un administrateur

PATCH /api/v1/projets/1/validate/admin/1

Étape 3 : Attribution d'un niveau de complexité

PATCH /api/v1/projets/1/niveau/admin/1?niveau=INTERMEDIAIRE

Scénario 3 : Participation à un projet

Étape 1 : Consulter les projets ouverts

GET /api/v1/projets/ouverts?domaine=WEBSeur=PRODUCTIVITE

Étape 2 : Envoyer une demande de participation

```
POST /api/v1/participants/projet/1/contributeur/2 Content-Type:
application/json { "profil": "DEVELOPPEUR", "scoreQuiz": 85 }
```

Étape 3 : Acceptation par le gestionnaire

PATCH /api/v1/participants/1/accept

Étape 4 : Déverrouillage de l'accès

PATCH /api/v1/participants/1/unlock

Scénario 4 : Contribution et validation

Étape 1 : Réserver une fonctionnalité

PATCH /api/v1/participants/1/fonctionnalite/1/reserve

Étape 2 : Soumettre une contribution

```
POST /api/v1/contributions/fonctionnalite/1/participant/1 Content-Type: application/json { "lienUrl": "https://github.com/user/repo", "fileUrl": "https://example.com/contribution.zip" }
```

Étape 3 : Validation par le gestionnaire

PUT /api/v1/contributions/1/status?status=VALIDE&gestionnaireId=1



Scénarios d'usage typiques

Scénario A : Porteur de projet (Idéateur uniquement)

- 1. **Inscription** et connexion
- 2. Proposition d'un projet avec cahier des charges
- 3. Attente de validation administrative (48h max)
- 4. Création d'un quiz pour sélectionner un gestionnaire
- 5. Sélection d'un gestionnaire après évaluation
- 6. Suivi de l'avancement du projet

Scénario B : Porteur de projet et Gestionnaire

- 1. **Inscription** et connexion
- 2. Proposition d'un projet en tant que gestionnaire
- 3. Attente de validation administrative
- 4. **Définition** des fonctionnalités du projet
- 5. Création de quiz pour les contributeurs
- 6. **Gestion** des demandes de participation
- 7. Attribution ou réservation de tâches
- 8. Validation des contributions

Scénario C : Contributeur participant

- 1. **Inscription** et connexion (100 coins offerts)
- 2. Consultation des projets ouverts
- 3. Passage du quiz d'évaluation
- 4. Attente de validation (48h max)
- 5. **Déverrouillage** de l'accès (coût en coins)
- 6. Réservation d'une tâche
- 7. **Développement** et soumission
- 8. Réception de récompenses (coins + badges)

Scénario D : Administration de la plateforme

- 1. Connexion administrateur
- 2. Validation des projets proposés
- 3. Attribution des niveaux de complexité
- 4. **Gestion** des paramètres de coins

- 5. **Création** de badges personnalisés
- 6. **Monitoring** de l'activité



Référence complète des endpoints



Authentification (/api/v1/auth)

Méthode	Endpoint	Description
POST	/register	Inscription d'un nouveau contributeur
POST	/login	Connexion utilisateur
POST	/logout	Déconnexion
GET	/profile/{id}	Récupération du profil
PATCH	/{id}/change-password	Changement de mot de passe

Projets (/api/v1/projets)

Méthode	Endpoint	Description
GET	/	Liste tous les projets (filtrage par statut)
GET	/ouverts	Projets ouverts (filtrage domaine/secteur)
GET	/domaine/{domaine}	Projets par domaine
GET	/secteur/{secteur}	Projets par secteur
GET	/{id}	Détails d'un projet
POST	/contributeur/{idPorteur}	Proposer un nouveau projet
PATCH	<pre>/{id}/validate/admin/{idAdmin}</pre>	Valider un projet
DELETE	<pre>/{id}/reject/admin/{idAdmin}</pre>	Rejeter un projet
PATCH	/{id}/cahier-charges	Modifier le cahier des charges
PATCH	/{id}/niveau/admin/{idAdmin}	Attribuer un niveau
PATCH	/{id}/start	Démarrer un projet
PATCH	/{id}/complete	Terminer un projet

Participants (/api/v1/participants)

Méthode	Endpoint	Description
POST	<pre>/projet/{idProjet}/contributeur/{idContributeur}</pre>	Demande de participation
PATCH	/{id}/accept	Accepter une demande
PATCH	/{id}/reject	Refuser une demande
PATCH	/{id}/unlock	Débloquer l'accès
PATCH	<pre>/{idParticipant}/fonctionnalite/{idFonctionnalite}/reserve</pre>	Réserver une tâche
PATCH	<pre>/{idParticipant}/fonctionnalite/{idFonctionnalite}/assign</pre>	Attribuer une tâche
GET	/{id}/historique	Historique d'acquisition
GET	/{id}/contributions	Contributions du participant
GET	/projet/{idProjet}	Participants d'un projet
GET	/{id}/badges/progression	Progression des badges

© Contributions (/api/v1/contributions)

Méthode	Endpoint	Description
GET	/	Liste des contributions (filtrage par statut)
GET	/{id}	Détails d'une contribution
POST	<pre>/fonctionnalite/{idFonctionnalite}/participant/{idParticipant}</pre>	Soumettre une contribution
PUT	/{id}/status	Valider/rejeter une contribution
GET	/participant/{participantId}	Contributions d'un participant
GET	/fonctionnalite/{fonctionnaliteId}	Contributions d'une fonctionnalité



Badges (/api/v1/badges)

Méthode	Endpoint	Description
GET	/	Liste tous les badges
GET	/{id}	Détails d'un badge
POST	/admin/{idAdmin}	Créer un badge
PUT	/{id}	Modifier un badge
PATCH	/{id}	Modification partielle
DELETE	/{id}	Supprimer un badge

Paramètres Coins (/api/v1/parametres-coins)

Méthode	Endpoint	Description
GET	/	Liste des paramètres
GET	/{id}	Détails d'un paramètre
POST	/admin/{idAdmin}	Créer un paramètre
PUT	/{id}	Modifier un paramètre
DELETE	/{id}	Supprimer un paramètre



Questionnaires (/api/v1/questionnaires)

Méthode	Endpoint	Description
POST	/projet/{idProjet}/createur/{idCreateur}	Créer pour un projet
POST	<pre>/template/{idTemplate}/admin/{idAdmin}</pre>	Créer pour un template
POST	/{idQuestionnaire}/questions	Ajouter une question
POST	/{idQuestionnaire}/evaluer	Évaluer un quiz
GET	/{id}	Détails d'un questionnaire
GET	/projet/{idProjet}	Questionnaires d'un projet
GET	<pre>/template/{idTemplate}</pre>	Questionnaires d'un template

Templates (/api/v1/templates)

Méthode	Endpoint	Description
GET	/	Liste des templates
GET	/{id}	Détails d'un template
POST	/admin/{idAdmin}	Créer un template
PUT	/{id}	Modifier un template
DELETE	/{id}	Supprimer un template



Notifications (/api/v1/notifications)

Méthode	Endpoint	Description
GET	/utilisateur/{idUtilisateur}	Notifications d'un utilisateur
PATCH	/{id}/read	Marquer comme lue
DELETE	/{id}	Supprimer une notification



Configuration et initialisation

Initialisation automatique

L'application initialise automatiquement :

- Paramètres de coins par défaut (inscription, contributions, déverrouillages)
- Badges système (DEBUTANT à PLATINE)
- Administrateur système si nécessaire

Variables d'environnement

Les configurations peuvent être personnalisées via les propriétés Spring Boot dans application.properties.

Codes de statut HTTP

Code	Signification	Usage
200	ОК	Opération réussie
201	Created	Ressource créée
204	No Content	Suppression réussie
400	Bad Request	Données invalides
401	Unauthorized	Authentification requise
404	Not Found	Ressource introuvable
409	Conflict	Conflit (ex: email déjà utilisé)
500	Internal Error	Erreur serveur

▲ Points d'attention

- Vérifier la configuration de la base de données avant le premier démarrage
- S'assurer que les ports nécessaires sont disponibles
- Configurer les paramètres SMTP pour l'envoi d'emails
- Définir les variables d'environnement de production

Cette documentation fournit une vue complète du fonctionnement de l'API CollabDev, permettant aux développeurs de comprendre l'architecture, d'utiliser les endpoints correctement et d'implémenter les scénarios d'usage typiques de la plateforme collaborative.

API CollabDev - Documentation Technique v1.0 | Spring Boot 3.x