

# Routes de l'API

## User Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (Obligatoires en gras)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/user/one		body	<b>userId:</b> <b>ObjectId</b>		Récupère les détails d'un utilisateur par ID	200 OK	Détails de l'utilisateur
GET	/user/all					Récupère tous les utilisateurs	200 OK	Liste des utilisateurs
POST	/user/login		body	<b>email:</b> string <b>password:</b> <b>string</b>		Connecte un utilisateur	200 OK	Token d'authentification
POST	/user/register		body	<b>username:</b> <b>string</b> <b>email:</b> string <b>password:</b> <b>string</b>	userVerif	Enregistre un nouvel utilisateur	201 Created	Détails de l'utilisateur nouvellement créé
PUT	/user/update	user	body	<b>userId:</b> <b>ObjectId</b> username: string email: string password: string	verifyToken	Met à jour les détails de l'utilisateur connecté	200 OK	Détails de l'utilisateur mis à jour
DELETE	/user/delete	user			verifyToken	Supprime l'utilisateur connecté	200 OK	Message de confirmation de suppression

Notes:

- Les champs en **gras** sont obligatoires.
- verifyToken est un middleware qui vérifie le token JWT pour authentifier l'utilisateur.
- userVerif est un middleware pour des vérifications supplémentaires lors de l'enregistrement.

## Movie Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (Obligatoires en gras)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/movie/one/:id		params	<b>id:</b> ObjectId		Récupère les détails d'un film par ID	200 OK	Film détaillé avec relations
GET	/movie/all					Récupère tous les films	200 OK	Liste des films avec relations
GET	/movie/search		query	<b>title:</b> string		Cherche des films par titre	200 OK	Liste des films correspondants

Méthode	Route	Header auth bearer token	Type de requête	Champs (Obligatoires en gras)	Middleware	Description	Statut de réponse	Type de données de réponse
POST	/movie/register	admin	body	<b>title: string</b> <b>synopsis: string</b> <b>image: string</b> director: ObjectId types: [ObjectId] adaptation: { idBook: ObjectId, name: string } actors: [ObjectId]	adminAuth	Enregistre un nouveau film	201 Created	Film détaillé avec relations
PUT	/movie/update	admin	body	<b>id: ObjectId</b> title: string synopsis: string image: string director: ObjectId types: [ObjectId] adaptation: { idBook: ObjectId, name: string } actors: [ObjectId]	adminAuth	Met à jour les détails d'un film	200 OK	Film mis à jour avec relations
DELETE	/movie/sup/:id	admin	params	<b>id: ObjectId</b>	adminAuth	Supprime un film	200 OK	Message de confirmation de suppression

Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.
- Les réponses incluent généralement des relations supplémentaires telles que le directeur, les types, les acteurs et, dans certains cas, les adaptations de livres.
- La colonne Header `auth` indique si la route nécessite une authentification pour accéder à la ressource.
- La colonne "Statut de réponse" précise le code HTTP attendu en réponse pour une demande réussie.

Admin Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (Obligatoires en gras)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/admin/one	admin	body	<b>id: ObjectId</b>	adminAuth	Récupère les détails d'un administrateur par ID	200 OK	Détails de l'administrateur
GET	/admin/all	admin			adminAuth	Récupère tous les administrateurs	200 OK	Liste des administrateurs

Méthode	Route	Header auth bearer token	Type de requête	Champs (Obligatoires en gras)	Middleware	Description	Statut de réponse	Type de données de réponse
POST	/admin/login		body	<b>email: string</b> <b>password: string</b>		Connecte un administrateur	200 OK	Token d'accès et détails de l'administrateur
POST	/admin/register	admin	body	<b>username: string</b> <b>email: string</b> <b>password: string</b>	adminAuth	Enregistre un nouvel administrateur	201 Created	Détails du nouvel administrateur
PUT	/admin/update	admin	body	<b>id: ObjectId</b> username?: string email?: string password?: string	adminAuth	Met à jour les détails de l'administrateur	200 OK	Détails mis à jour de l'administrateur
DELETE	/admin/delete	admin	body	<b>id: ObjectId</b>	adminAuth	Supprime l'administrateur	200 OK	Message de confirmation de suppression
GET	/admin/search	admin	body	username?: string email?: string	adminAuth	Cherche des administrateurs par nom ou email	200 OK	Liste des administrateurs trouvés

Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.

Book Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (Obligatoires en gras)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/book/one/:id		params	<b>id: ObjectId</b>		Récupère les détails d'un livre par ID	200 OK / 404 Livre non trouvé	Détails du livre avec relations
GET	/book/all					Récupère tous les livres	200 OK	Liste des livres avec relations
GET	/book/search		query	<b>title: string</b>		Cherche des livres par titre	200 OK	Liste des livres correspondants avec relations
POST	/book/register	admin	body	<b>title: string</b> <b>summary: string</b> image: string author: [ObjectId] types: [ObjectId] adaptation: {	adminAuth	Enregistre un nouveau livre	201 Created / 400 Mauvaise requête	Livre détaillé avec relations / Message d'erreur

Méthode	Route	Header auth bearer token	Type de requête	Champs (Obligatoires en gras)	Middleware	Description	Statut de réponse	Type de données de réponse
				adapt: boolean, id_movie: ObjectId }				
PUT	/book/update	admin	body	<b>id: ObjectId</b> title: string summary: string image: string author: [ObjectId] types: [ObjectId] adaptation: { adapt: boolean, id_movie: ObjectId }	adminAuth	Met à jour les détails d'un livre	200 OK / 404 Livre non trouvé	Livre mis à jour avec relations / Message d'erreur
DELETE	/book/sup/:id	admin	params	<b>id: ObjectId</b>	adminAuth	Supprime un livre	200 OK / 404 Livre non trouvé	Message de confirmation de suppression / Message d'erreur

Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.

Actor Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/actor/one/:id		params	<b>id: ObjectId</b>		Récupère les détails d'un acteur par ID	200 OK / 404 Acteur non trouvé	Détails de l'acteur avec films associés
GET	/actor/all					Récupère tous les acteurs	200 OK	Liste des acteurs avec films associés
GET	/actor/search		query	<b>name: string</b>		Cherche des acteurs par nom	200 OK	Liste des acteurs correspondants
POST	/actor/register	admin	body	<b>name: string</b> <b>surname: string</b> image: string movies: [ObjectId]	adminAuth	Enregistre un nouvel acteur	201 Created / 400 Mauvaise requête	Acteur détaillé avec films associés / Message d'erreur
PUT	/actor/update	admin	body	<b>id: ObjectId</b> name: string surname: string image: string	adminAuth	Met à jour les détails d'un acteur	200 OK / 404 Acteur non trouvé	Acteur mis à jour avec films associés / Message d'erreur

Méthode	Route	Header auth bearer token	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
				movies: [ObjectId]				
DELETE	/actor/sup/:id	admin	params	<b>id: ObjectId</b>	adminAuth	Supprime un acteur	200 OK / 404 Acteur non trouvé	Message de confirmation de suppression / Message d'erreur

Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.

Director Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/director/one/:id		params	<b>id: ObjectId</b>		Récupère les détails d'un directeur par ID	200 OK / 404 Directeur non trouvé	Détails du directeur avec films associés
GET	/director/all					Récupère tous les directeurs	200 OK	Liste des directeurs avec films associés
GET	/director/search		query	<b>director: string</b>		Cherche des directeurs par nom	200 OK	Liste des directeurs correspondants
POST	/director/register	admin	body	<b>director: string</b> image: string movies: [ObjectId]	adminAuth	Enregistre un nouveau directeur	201 Created / 400 Mauvaise requête	Directeur enregistré / Message d'erreur
PUT	/director/update	admin	body	<b>id: ObjectId</b> director: string image: string movies: [ObjectId]	adminAuth	Met à jour les détails d'un directeur	200 OK / 404 Directeur non trouvé	Directeur mis à jour avec films associés / Message d'erreur
DELETE	/director/sup/:id	admin	params	<b>id: ObjectId</b>	adminAuth	Supprime un directeur	200 OK / 404 Directeur non trouvé	Message de confirmation de suppression / Message d'erreur

Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.

## Type Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/type/one/:id		params	<b>id: ObjectId</b>		Récupère les détails d'un type par ID	200 OK / 404 Type non trouvé	Détails du type
GET	/type/all					Récupère tous les types	200 OK	Liste des types
GET	/type/search		query	<b>title: string</b>		Cherche des types par titre	200 OK	Liste des types correspondants
POST	/type/register	admin	body	<b>type: string</b>	adminAuth	Enregistre un nouveau type	201 Created / 500 Erreur interne	Type enregistré / Message d'erreur
PUT	/type/update	admin	body	<b>id: ObjectId</b> type: string	adminAuth	Met à jour les détails d'un type	200 OK / 404 Type non trouvé / 500 Erreur interne	Type mis à jour / Message d'erreur
DELETE	/type/sup/:id	admin	params	<b>id: ObjectId</b>	adminAuth	Supprime un type	200 OK / 404 Type non trouvé / 500 Erreur interne	Message de confirmation de suppression / Message d'erreur

Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.

## Author Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/author/one/:id		params	<b>id: ObjectId</b>		Récupère les détails d'un auteur par ID	200 OK / 404 Auteur non trouvé	Détails de l'auteur
GET	/author/all					Récupère tous les auteurs	200 OK	Liste des auteurs
GET	/author/search		query	<b>author: string</b>		Cherche des auteurs par nom	200 OK	Liste des auteurs correspondants
GET	/author/:id/all/books		params	<b>id: ObjectId</b>		Récupère tous les livres d'un auteur spécifique	200 OK / 404 Auteur non trouvé	Liste des livres de l'auteur

Méthode	Route	Header auth bearer token	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
POST	/author/register	admin	body	<b>author:</b> <b>string</b> image: string books: [ObjectId] (lié aux livres écrits par l'auteur)	adminAuth	Enregistre un nouvel auteur	201 Created / 400 Informations manquantes / 500 Erreur interne	Auteur enregistré / Message d'erreur
PUT	/author/update	admin	body	<b>id: ObjectId</b> author: string image: string books: [ObjectId]	adminAuth	Met à jour les détails d'un auteur	200 OK / 404 Auteur non trouvé / 500 Erreur interne	Auteur mis à jour / Message d'erreur
DELETE	/author/sup/:id	admin	params	<b>id: ObjectId</b>	adminAuth	Supprime un auteur	200 OK / 404 Auteur non trouvé / 500 Erreur interne	Message de confirmation de suppression / Message d'erreur

Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.

## Reviews Routes

Méthode	Route	Header auth bearer token	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/reviews/one/:id		params	<b>id: ObjectId</b>		Récupère les détails d'une critique par ID	200 OK / 404 Critique non trouvée	Détails de la critique
GET	/reviews/all					Récupère toutes les critiques	200 OK	Liste des critiques
GET	/reviews/all/user/:userId		params	<b>userId:</b> <b>ObjectId</b>		Récupère toutes les critiques d'un utilisateur spécifique	200 OK / 404 Utilisateur non trouvé / 404 e critique trouvée	Liste des critiques de l'utilisateur
POST	/reviews/register	Oui	body	<b>message:</b> <b>string</b> <b>user:</b> <b>ObjectId</b> movie: ObjectId	verifyToken	Enregistre une nouvelle critique	201 Created / 400 Informations manquantes / 500 Erreur interne	Critique enregistrée / Message d'erreur

Méthode	Route	Header <small>auth bearer token</small>	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
				book: ObjectId				
PUT	/reviews/update	Oui	body	<b>id: ObjectId</b> message: string user: ObjectId movie: ObjectId book: ObjectId	verifyToken	Met à jour les détails d'une critique	200 OK / 404 Critique non trouvée / 500 Erreur interne	Critique mise à jour / Message d'erreur
DELETE	/reviews/sup/:id	Oui	params	<b>id: ObjectId</b>	verifyToken	Supprime une critique	200 OK / 404 Critique non trouvée / 500 Erreur interne	Message de confirmation de suppression / Message d'erreur

Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.
- `verifyToken` est un middleware qui vérifie le token JWT pour authentifier l'utilisateur.

Rating Routes

Méthode	Route	Header <small>auth bearer token</small>	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
GET	/rating/one/:id	Oui	params	<b>id: ObjectId</b>	verifyToken	Récupère les détails d'une évaluation par ID	200 OK / 404 Évaluation non trouvée	Détails de l'évaluation
GET	/rating/all/user/:userId	Oui	params	<b>userId: ObjectId</b>	verifyToken	Récupère toutes les évaluations d'un utilisateur spécifique	200 OK	Liste des évaluations de l'utilisateur
POST	/rating/register	Oui	body	<b>user: ObjectId</b> <b>value: number (1-5)</b> <b>movie: ObjectId (optionnel)</b> <b>book: ObjectId (optionnel)</b>	verifyToken	Enregistre une nouvelle évaluation	201 Created / 400 Données manquantes ou invalides / 400 Note invalide	Nouvelle évaluation créée / Message d'erreur



Méthode	Route	Header auth bearer token	Type de requête	Champs (obligatoire)	Middleware	Description	Statut de réponse	Type de données de réponse
				<i>Note: soit un film soit un livre</i>				
PUT	/rating/update	Oui	body	<b>ratingId:</b> <b>ObjectId</b> <b>value:</b> <b>number (1-5)</b> <b>movie:</b> <b>ObjectId</b> <b>(optionnel)</b> <b>book:</b> <b>ObjectId</b> <b>(optionnel)</b> <i>Note: soit un film soit un livre</i>	verifyToken	Met à jour une évaluation	200 OK / 400 Données manquantes / 400 Note invalide / 404 Évaluation non trouvée	Évaluation mise à jour / Message d'erreur
DELETE	/rating/sup/:id	Oui (admin)	params	<b>id: ObjectId</b>	adminAuth	Supprime une évaluation	200 OK / 404 Évaluation non trouvée	Message de suppression réussie / Message d'erreur
GET	/rating/admin/user/:userId	Oui (admin)	params	<b>userId:</b> <b>ObjectId</b>	adminAuth	Récupère toutes les évaluations d'un utilisateur spécifique (version admin)	200 OK	Liste des évaluations de l'utilisateur
GET	/rating/all	Oui (admin)			adminAuth	Récupère toutes les évaluations (version admin)	200 OK	Liste de toutes les évaluations

#### Notes:

- Les champs en **gras** sont obligatoires.
- `adminAuth` est un middleware qui vérifie les droits d'administration avant d'exécuter certaines actions.
- `verifyToken` est un middleware qui vérifie le token JWT pour authentifier l'utilisateur.