

CUBE

Plan de test API

Juin 2022

1 - Tests unitaires

1.1 - ApplicationModel

Rappel de concept : `ApplicationModel` est la classe parente de tous les modèles existants dans le schéma de base de données (`Resource`, `Comment`, etc). Chaque `ApplicationModel` est le reflet de son `record` (modèle Prisma) sous-jacent et expose des méthodes supplémentaires dédiées à en faciliter les opérations dans le domaine métier.

Message envoyé	Valeur de retour attendue		Effet(s) de bord attendu(s)
id	Même valeur que pour record		Aucun
createdAt	Même valeur que pour record		Aucun
updatedAt	Même valeur que pour record		Aucun
isValid	Cas : l'objet testé est valide	true	La propriété errors est un tableau vide.
	Cas : l'objet testé est invalide	false	La propriété errors contient des erreurs de validation.
setAttributes	L'objet testé		Les propriétés de record correspondent à celles données en paramètres.
save	Cas : l'objet testé est invalide	N/A	Une exception InvalidRecordError a été levée et record n'est pas persisté.
	Cas : l'objet testé ne possède pas d'id	L'objet testé	record est persisté et les propriétés id, createdAt et updatedAt sont présentes et cohérentes.
	Cas : l'objet testé possède déjà un id	L'objet testé	record n'est pas persisté
	Cas : une erreur est survenue en base de données	N/A	
update	setAttributes suivi de save		
destroy	L'objet testé		record n'existe plus dans la base de donnée

1.2 - Resource

Message envoyé	Valeur de retour attendue		Effet(s) de bord attendu(s)
save	Cas : categoryId est présent	L'objet testé	categoryId n'as pas changé
	Cas : categoryId n'est pas présent		categoryId correspond à l'id de la catégorie par défaut

1.3 - Comment

Message envoyé	Valeur de retour attendue	Effet(s) de bord attendu(s)
userId	Même valeur que pour record	Aucun
resourceId	Même valeur que pour record	Aucun
parentUserId	Même valeur que pour record	Aucun
text	Même valeur que pour record	Aucun

1.4 - User

Message envoyé	Valeur de retour attendue	Effet(s) de bord attendu(s)
name	Même valeur que pour record	Aucun
firstName	Même valeur que pour record	Aucun
email	Même valeur que pour record	Aucun
profilePicture	Même valeur que pour record	Aucun
backgroundImage	Même valeur que pour record	Aucun
role	Même valeur que pour record	Aucun
bornedAt	Même valeur que pour record	Aucun
confirmedAt	Même valeur que pour record	Aucun

1.5 - AttachedFile

Message envoyé	Valeur de retour attendue	Effet(s) de bord attendu(s)
getPresignedUrl	Une promesse de chaîne de caractère : l'URL temporaire du fichier sur S3	Aucun
uploadAs	L'objet testé	L'objet est présent sur S3

2 - Tests d'intégration des contrôleurs

Endpoint	Code HTTP attendu		Contenu attendu
POST /users	Cas : l'utilisateur est correctement créé	201	result contenant les propriétés de l'utilisateur créé
	Cas : un ou plusieurs champs sont manquants	422	Un tableau errors avec les erreurs survenues
	Cas : une erreur est survenue	500	
POST /sessions	Cas : l'utilisateur est identifiable	200	Token JWT valide dans token
	Cas : l'utilisateur n'est pas identifiable	401	Un tableau errors avec les erreurs survenues
POST /resources	Cas : la ressource est correctement créée	201	result contenant les propriétés de la ressource créée
	Cas : un ou plusieurs champs sont manquants ou invalides	422	Un tableau errors avec les erreurs survenues
	Cas : une erreur est survenue	500	
	Cas : JWT manquant ou invalide	401	N/A
	Cas : action interdite pour l'utilisateur	403	
GET /resources	Les ressources sont listées	200	Un tableau result avec les ressources auxquelles a accès l'utilisateur
PATCH /resources/{id}	Cas : un ou plusieurs champs sont invalides	422	Un tableau errors avec les erreurs survenues
	Cas : une erreur est survenue	500	
	Cas : JWT manquant ou invalide	401	N/A
	Cas : action interdite pour l'utilisateur	403	
GET /resources/{id}	Cas : la ressource existe et est accessible pour l'utilisateur	200	La ressource dans la propriété result
	Cas : la ressource n'existe pas	404	N/A
	Cas : la ressource existe mais n'est pas accessible pour l'utilisateur	403	
	Cas : la ressource est privée et l'utilisateur n'est pas le créateur de la ressource	404	

CUBE – Plan de tests API

DELETE /resources/{id}	Cas : la ressource a été supprimée	204	N/A
	Cas : la ressource existe mais n'est pas accessible pour l'utilisateur	404	N/A
	Cas : l'utilisateur n'est pas le créateur de la ressource	403	N/A
Autres endpoints similaire pour les autres ressources...			

3 - Tests end-to-end avec Insomnia

Il est possible d'intégrer des tests E2E avec l'outil de développement d'API Insomnia. Les tests doivent s'effectuer sur un serveur isolé et une base de donnée vierge. Avec Insomnia CLI, cette procédure peut s'intégrer facilement dans une CI. Cette fonctionnalité est disponible uniquement avec les « design document » répondant au standard Open API (ce qui permet d'avoir une documentation auto générée pour l'API de manière automatisé).

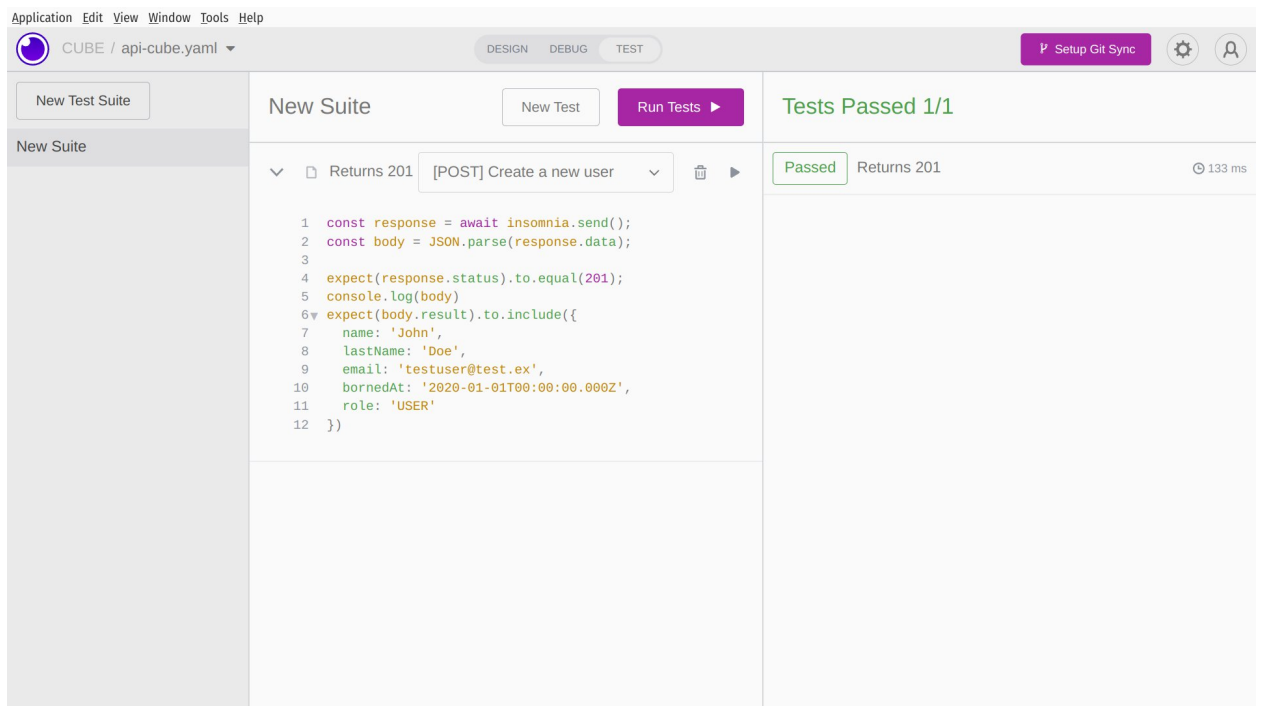


Figure 1 Capture d'écran d'un test Insomnia

La suite de tests Insomnia reproduit un scénario d'utilisation classique avec les étapes suivantes dans l'ordre :

Action	Endpoint	Notes
Créer un utilisateur	POST /users	
Créer une session pour l'utilisateur	POST /sessions	
Lister les ressources existantes	GET /resources	La liste est vide
Créer une ressource	POST /sessions	
Créer un commentaire sur la ressource créée	POST /comments	
Visualiser la ressource créée	GET /resources/{id}	Le commentaire doit être présent
Supprimer la ressource créée	DELETE /resources/{id}	

Il pourra être imaginé des scénarios plus complexes sur une base de données avec un jeu de test initial conséquent.