Contexte : MediaTek86

Application : MediatekDocuments (application de bureau C#) exploitant rest\_mediatekdocuments (API REST en PHP)

# Plan de tests

## Tests unitaires sur les classes du package model

|  |  |  |  |
| --- | --- | --- | --- |
| But du test | Action de contrôle | Résultat attendu | Bilan |
| Contrôler la méthode ToString() de la classe Categorie pour voir si elle retourne le libelle. | Test unitaire lancé après avoir créé un objet de type Genre avec libelle contenant :  "Horreur" | "Horreur" | OK |
| Contrôler le constructeur Categorie() de la classe Categorie pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Genre avec id contenant ‘00001’ et libelle contenant ‘Horreur’ | Toutes les propriétés valorisées | OK |
| Contrôler la méthode ToString() de la classe Suivi pour voir si elle retourne le libelle. | Test unitaire lancé après avoir créé un objet de type Suivi avec libelle contenant :  "en cours" | ‘en cours’ | OK |
| Contrôler le constructeur Suivi() de la classe Suivi pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Suivi avec id contenant ‘1’ et libelle contenant ‘en cours’ | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Rayon() de la classe Rayon pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Rayon avec id contenant ‘1234’ et libelle contenant ‘Voyages’ | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Public() de la classe Public pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Public avec id contenant ‘blabla’ et libelle contenant ‘Enfants’ | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Genre() de la classe Genre pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Genre avec id contenant ‘blabla’ et libelle contenant ‘Enfants’ | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Document() de la classe Document pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Document contenant toutes les propriétés : id, titre, image, idGene, genre, idPublic, public, idRayon, rayon | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Livre() de la classe Livre pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Livre contenant toutes les propriétés : id, titre, image, idGene, genre, idPublic, public, idRayon, rayon, isbn, auteur, collection | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Dvd() de la classe Dvdpour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Dvd contenant toutes les propriétés : id, titre, image, idGene, genre, idPublic, public, idRayon, rayon, réalisateur, synospsis, duree | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Commande() de la classe Commande pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Commande contenant toutes les propriétés : id 🡪 ‘00025’, dateCommande 🡪 ‘2025-04-26’, montant 🡪 12.5 | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur CommandeDocument() de la classe CommandeDocument pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Commande contenant toutes les propriétés : id, dateCommande , montant, idLivreDvd, nbExemplaire idSuivi, suivi | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Abonnement() de la classe Abonnement pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Abonnement contenant toutes les propriétés : id, dateCommande , montant, dateFinAbonnement, idRevue | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Service() de la classe Service pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Service avec id contenant ‘1’ et libelle contenant ‘admin’ | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Users() de la classe Users pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Users avec toutes les propirétés : id, login et idService | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Revue() de la classe Revue pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Revue avec toutes les propriétés. | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Etat() de la classe Etat pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Etat avec id contenant ‘00005’ et libelle contenant ‘poubelle’ | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Profil() de la classe Profil pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Profil avec login contenant ‘sorciere’ et pwd contenant ‘pwdsorciere’ | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur AbonnFinProche() de la classe AbonnFinProche pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type AbonnFinProche avec titreRevue contenant ‘Le canard enchaine’ et dateFinAbonn contenant ‘2025-07-15’ | Toutes les propriétés valorisées | OK |
| Contrôler le constructeur Exemplaire() de la classe Exemplaire pour voir s’il valorise les propriétés. | Test unitaire lancé après avoir créé un objet de type Exemplaire avec toutes les propriétés | Toutes les propriétés valorisées | OK |

## Tests fonctionnels dans Postman (fonctionnalités de l'API)

|  |  |  |  |
| --- | --- | --- | --- |
| But du test | Action de contrôle | Résultat attendu | Bilan |
| Contrôler la méthode selectAllLivres() | Test fonctionnel réalisé avec une requête GET sur la table livre | Liste des livres contenus dans la BDD | OK |
| Contrôler la méthode selectAllDvd() | Test fonctionnel réalisé avec une requête GET sur la table dvd | Liste des Dvd contenus dans la BDD | OK |
| Contrôler la méthode selectAllRevues() | Test fonctionnel réalisé avec une requête GET sur la table revue | Liste des revues contenues dans la BDD | OK |
| Contrôler la méthode selectExemplairesRevue($champs) | Test fonctionnel réalisé avec une requête GET sur la table exemplaire avec idRevue = ‘10002’ | Liste des exemplaires d’une revue | OK |
| Contrôler la méthode selectAllCommandes() | Test fonctionnel réalisé avec une requête GET sur la table commande | Liste des commande contenues dans la BDD | OK |
| Contrôler la méthode selectAllUsers() | Test fonctionnel réalisé avec une requête GET sur la table users | Liste des users contenus dans la BDD | OK |
| Contrôler la méthode selectAbonnementFinProche() | Test fonctionnel réalisé avec une requête GET sur la table abonnement | Liste des revues avec dateFinAbonnement, titre de la revue et id de la revue | OK |
| Contrôler la méthode selectCommandesDocument($champs) | Test fonctionnel réalisé avec une requête GET sur la table commandedocument avec idLivreDvd = ‘00003’ | Liste des commandes pour un document | OK |
| Contrôler la méthode selectAbonnementRevue($champs) | Test fonctionnel réalisé avec une requête GET sur la table abonnement avec idRevue = ‘10006’ | Liste des abonnements d’une revue | OK |
| Contrôler la méthode selectTableSimple($table) | Test fonctionnel réalisé avec une requête GET sur la table genre | Liste du contenu de la table genre | OK |
| Contrôler la méthode insertCommandeDocument($champs) | Test fonctionnel réalisé avec l’insertion d’une commande pour le document contenant l’id : ‘00018’ | 1 ligne insérée dans la table commande et 1 ligne insérée dans la table commandedocument | OK |
| Contrôler la méthode insertAbonnementRevue($champs) | Test fonctionnel réalisé avec l’insertion d’une commande pour la revue contenant l’id : ‘10003’ | 1 ligne insérée dans la table commande et 1 ligne insérée dans la table abonnement | OK |
| Contrôler la méthode updateCommandeDocument($id, $champs) | Test fonctionnel réalisé avec la mise à jour du suivi de la commande contenant l’id : ‘00012’, nouveau idSuivi = ‘2’ | 1 ligne insérée | OK |
| Contrôler la méthode deleteTuplesOneTable($table, $champs) | Test fonctionnel réalisé avec la suppression de la commande contenant l’id : ‘00011’ | 1 ligne supprimée | OK |