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 |
| Contrôle la methode AbonnementTest() de classe Abonnement pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis. | Test unitaire lancé après avoir crée un objet de type Abonnement | Passé | OK |
| Contrôle la methode CommandeTest() de classe Commande pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis. | Test unitaire lancé après avoir crée un objet de type Commande | Passé | OK |
| Contrôle la methode CommandeDocumentTest() de classe CommandeDocument pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis. | Test unitaire lancé après avoir crée un objet de type CommandeDocument | Passé | OK |
| Contrôle la methode DocumentTest() de classe Document pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis. | Test unitaire lancé après avoir crée un objet de type Document | Passé | OK |
| Contrôle la methode LivreTest() de classe Livre pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis. | Test unitaire lancé après avoir crée un objet de type Livre | Passé | OK |
| Contrôle la methode DvdTest() de classe Dvd pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Dvd | Passé | OK |
| Contrôle la methode EtatTest() de classe Etat pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Etat | Passé | OK |
| Contrôle la methode CategorieTest() de classe Categorie pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Categorie | Passé | OK |
| Contrôle la methode GenreTest() de classe Genre pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Genre | Passé | OK |
| Contrôle la methode RayonTest() de classe Rayon pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Rayon | Passé | OK |
| Contrôle la methode RevueTest() de classe Revue pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Revue | Passé | OK |
| Contrôle la methode LivreDvdTest() de classe LivreDvd pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type LivreDvd | Passé | OK |
| Contrôle la methode SuiviTest() de classe Suivi pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Suivi | Passé | OK |
| Contrôle la methode ServiceTest() de classe Service pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Service | Passé | OK |
| Contrôle la methode UtilisateurTest() de classe Utilisateur pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Utilisateur | Passé | OK |
| Contrôle la methode PublicTest() de classe Public pour vérifier que le constructeur de cette classe initialise correctement tous les attributs à partir des paramètres fournis | Test unitaire lancé après avoir crée un objet de type Public | Passé | OK |

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

|  |  |  |  |
| --- | --- | --- | --- |
| But du test | Action de contrôle | Résultat attendu | Bilan |
| Contrôle la methode selectAllLivres() de l’Api pour vérifier que l’Api retourne tous les livres de la base de données | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/livre | Tout les livres de BDD, 26 en totale, sont retourner en liste JSON | OK |
| Contrôle la methode selectAllDvd() de l’Api pour vérifier que l’Api retourne tous les dvds de la base de données | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/dvd | Tous les dvd de BDD, 4 en totale, sont retourner en liste JSON | OK |
| Contrôle la methode selectAllLivreDvd() de l’Api pour vérifier que l’Api retourne tous les livredvds de la base de données | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/livredvd | Tous les livredvds de BDD, 4 en totale, sont retourner en liste JSON | OK |
| Contrôle la methode selectAllCommandes() de l’Api pour vérifier que l’Api retourne tous les commandes de la base de données | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/commande | Tous les commandes de BDD sont retournées en liste JSON | OK |
| Contrôle la methode selectAllRevues() de l’Api pour vérifier que l’Api retourne toutes les revues de la base de données | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/revue | Toutes les revues de BDD, 11 en totale, sont retournées en liste JSON | OK |
| Contrôle la methode selectAllServices() de l’Api pour vérifier que l’Api retourne tous les services de la base de données | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/service | Toutes les services de BDD, 4 en totale, sont retournées en liste JSON | OK |
| Contrôle la methode selectAllAbonnement() de l’Api pour vérifier que l’Api retourne tous les abonnement de la base de données que si la date finale de l’abonnement est proche a celle de sont achat (30 jours). | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/abonnementecheance | Toutes les abonnements qui ont moins de 30 jours entre l’achat et la fin de l’abonnement, 3 au temps du test, son retourner en liste JSON | OK |
| Contrôle la methode selectAllCommandeDocument() de l’Api pour vérifier que l’Api retourne toutes les informations concernant le document concerne | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/commandedocument  dans le body de la requête :  {      "idLivreDvd": "00003"  } | [          {              "id": "333",              "nbExemplaire": 3,              "idLivreDvd": "00003",              "idSuivi": "00002",              "dateCommande": "2025-03-03",              "montant": 333,              "Libelle": "livrée"          }      ] | OK |
| Contrôle la methode selectUtilisateur() de l’Api pour vérifier que l’Api retourne un seul utilisateur, en forme de liste d’attribtuts, si le login et password sont correct | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/utilisateur  dans le body de la requête :  {      "login": "admin",      "password": "P@$$word1"  } | [          {              "login": "admin",              "password": "P@$$word1",              "nom": "Mukuri Maka",              "prenom": "Jonas",              "idService": "00004"          }      ] | OK |
| Contrôle la methode SelectAllAbonnementsRevue() de l’Api pour vérifier que l’Api retourne tous les commandes d’abonnement fait pour une revue spécifique | Une requête GET est faite sur postman :  http://localhost/rest\_mediatekdocuments/abonnement  dans le body de la requête :  {      "id": "10003"  } | [          {              "id": "38421",              "dateCommande": "2025-04-01",              "montant": 34,              "dateFinAbonnement": "2025-04-04",              "idRevue": "10003"          }      ] | OK |
| Contrôle la methode insertCommandeDocument() de l’Api pour inserer une nouvelle commande(commande) et le document liee a cette commande(commandedocument) dans la base de données | Une requête POST est faite sur postman :  http://localhost/rest\_mediatekdocuments/commandedocument  dans le body de la requête :  {  "id": "98765",  "dateCommande": "2025-04-01",  "montant": 30,  "nbExemplaire": 3,  "idLivreDvd": "00003",  "idSuivi": "00001",  "Libelle": "en cours"  } | "code": 200,  "message": "OK"  La nouvelle commande et son document(commandedocument) sont insérer dans la BDD | OK |
| Contrôle la methode updateCommandeDocument() de l’Api pour faire une modification de l’etape de suivi d’une commande | Une requête PUT est faite sur postman :  http://localhost/rest\_mediatekdocuments/commandedocument  dans le body de la requête :  {      "id": "98765",      "idSuivi": "00002"  } | "code": 200,  "message": "OK"  La commandedocument en question a un nouveau idSuivi, "00002", et le libelle est change automatiquement | OK |
| Contrôle la methode deleteCommandeDocument() de l’Api pour supprimer une commande et son document dans la base de données | Une requête DELETE est faite sur postman :  http://localhost/rest\_mediatekdocuments/commandedocument  dans le body de la requête :  {      "id": "98765"  } | "code": 200,  "message": "OK"  La commande et son document(commandedocument) sont supprimer de la BDD | OK |
| Contrôle la methode insertAbonnement() de l’Api pour inserer une nouveau abonnement dans la base de données | Une requête POST est faite sur postman :  http://localhost/rest\_mediatekdocuments/abonnement  dans le body de la requête :  {      "id": "55555",      "dateCommande": "2025-02-01",      "montant": 30,      "dateFinAbonnement": "2025-04-01",      "idRevue": "10002"  } | "code": 200,  "message": "OK"  La nouvelle commande (id, dateCommande, montant) et son abonnement (id, dateFinAbonnement, idRevue) sont insérer dans la BDD | OK |
| Contrôle la methode insertAbonnement() de l’Api pour inserer une nouveau abonnement dans la base de données | Une requête POST est faite sur postman :  http://localhost/rest\_mediatekdocuments/abonnement  dans le body de la requête :  {      "id": "55555"  } | "code": 200,  "message": "OK"  La commande et son abonnement sont supprimer dans la BDD | OK |