

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 l'instanciation de la classe Abonnement.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe CommandeDocument.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe Commande.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe Document.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe DVD.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler la méthode ToString() de la classe Etat pour voir si elle retourne le libelle	Test unitaire lancé après avoir créé un objet de type Etat avec libelle contenant : "Court écrit satirique, diffamatoire."	"Court écrit satirique, diffamatoire."	OK
Contrôler l'instanciation de la classe Exemple.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe Genre.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe Livre.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe Public.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe Rayon.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe Revue.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK
Contrôler l'instanciation de la classe Service, voir si elle retourne le libelle.	Test unitaire lancé après avoir créé un objet.	« Fenouil »	OK
Contrôler l'instanciation de la classe Suivi, voir si elle retourne le libelle.	Test unitaire lancé après avoir créé un objet.	« en cours »	OK
Contrôler l'instanciation de la classe Utilisateur.	Test unitaire lancé après avoir créé un objet.	Assert.AreEqual	OK

Tests fonctionnels avec Specflow sur les recherches (onglet livre)

But du test	Action de contrôle	Résultat attendu	Bilan
Test recherche id	<p>Given je saisie la valeur dans le champs de recherche de l'id : "00024"</p> <p>When je clique sur le bouton de recherche</p> <p>Then la vue affiche le livre possédant l'id : "00024"</p>	result.Should().BeTrue();	OK
Test recherche titre	<p>Given je saisie la valeur dans le champs de recherche des titres: "La planète des singes"</p> <p>Then la vue affiche le livre possédant le titre : "La planète des singes"</p>	result.Should().BeTrue();	OK
Test rayon	<p>Given je selectionne le rayon : "BD Adultes"</p> <p>Then la vue affiche les livres du rayon : "BD Adultes"</p>	result.Should().BeTrue();	OK

Tests fonctionnels dans Postman (fonctionnalités de l'API)

But du test	Action de contrôle	Résultat attendu	Bilan
Tester la connexion avec identifiant valide	<p>GET</p> <p>http://localhost/rest_mediatekdocuments/utilisateur/</p> <p>{"mail":"Paul","password":"0558687dff97785f3391d655a2b9a38ec99a8e0dc20d9ef192cc3861bd2347f8"}</p>	<p>L'API doit renvoyer</p> <pre>"code": 200, "message": "OK", "result": [{ "id": "0001", "nom": "Paul", "prenom": "Paul", "mail": "paul@mail.com", "idservice": "0001", "service": "compta" }]</pre>	OK
Tester la connexion avec non identifiant valide	<p>GET</p> <p>http://localhost/rest_mediatekdocuments/utilisateur/</p> <p>{"mail":"Paul","password":"0558687dff97785f3391d644a2b9a38ec99a8e0dc20d9ef192cc3861"}</p>	<p>L'API doit renvoyer un tableau vide</p> <pre>{ "code": 200, "message": "OK", }</pre>	OK

	bd2347f8"} }	"result": [] }	
Tester les livres	GET http://localhost/rest_mediatekdocuments/livre	L'API doit renvoyer code : 200 , message : OK , et un tableau contenant tous les livres	OK
Tester les exemplaire d'un livre	http://localhost/rest_mediatekdocuments/exemplaire/{id:"00007"}	L'API doit renvoyer 200, OK et un tableau contenant 5 exemplaire	OK
Tester la suppression d'un exemplaire	DELETE et exemplaire/{ "Numero":8, "Photo": "", "DateAchat": "2024-01-24", "IdEtat": "00004", "Id": "00007" }	L'API doit renvoyer 200, OK	OK
Tester la modification d'un exemplaire	PUT et exemplaire/00007/{ "Numero":1, "Photo": "", "DateAchat": "2023-10-10", "IdEtat": "00002", "Id": "00007" }	L'API doit renvoyer 200, OK	OK
Tester la modification d'un livre	PUT et livre/00017/{ "Isbn": "", "Auteur": "Philippe Maçon", "Collection": "", "Id": "00017", "Titre": "Catastrophes au Brésil", "Image": "", "IdGenre": "10018", "Genre": "Actualités", "IdPublic": "00004", "Public": "Ados", "IdRayon": "BL001", "Rayon": "Beaux Livres" }	L'API doit renvoyer 200, OK	OK
Tester les Dvd	GET et dvd	L'API doit renvoyer code : 200 , message : OK , et un tableau contenant tous les Dvd	OK
Tester la modification d'un DVD	PUT et dvd/20003/{ "Duree":128, "Realisateur": "Steven Spielberg", "Synopsis": "Un milliardaire et des généticiens créent des dinosaures à partir de clonage.", "Id": "20003", "Titre": "JurassiQue ParQue", "Image": "", "IdGenre": "10018", "Genre": "Actualités", "IdPublic": "00003", "Public": "Tous publics", "IdRayon": "DF001", "Rayon": "DVD films" }	L'API doit renvoyer code : 200 , message : OK	OK
Tester la suppression d'un DVD	Delete et dvd/{ "Duree":128, "Realisateur": "Steven Spielberg", "Synopsis": "Un milliardaire et des généticiens créent des dinosaures à partir de clonage.", "Id": "20003", "Titre": "JurassiQue ParQue", "Image": "", "IdGenre": "10018", "Genre": "Actualités", "IdPublic": "00003", "Public": "Tous publics", "IdRayon": "DF001", "Rayon": "DVD	L'API doit renvoyer code : 200 , message : OK	OK

