

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 la méthode Abonnement() de la classe Abonnement pour voir si elle retourne l'abonnement.	Test unitaire lancé après avoir créé un objet de type Abonnement avec les propriétés « id, dateCommande, montant, dateFinAbonnement, idRevue, titre » contenant : « 00050 », «2023, 3,22 », « 50 », « 2023, 3,30 », « 10003 », « Challenges »	« 00050 », «2023, 3,22 », « 50 », « 2023, 3,30 », « 10003 », « Challenges »	OK
Contrôler la méthode Commande() de la classe Commande pour voir si elle retourne la commande.	Test unitaire lancé après avoir créé un objet de type Commande avec les propriétés « id, dateCommande, montant » contenant : « 00060 », «2023, 3, 22 », « 50 »	« 00060 », «2023, 3, 22 », « 50 »	OK
Contrôler la méthode CommandeDocument() de la classe CommandeDocument pour voir si elle retourne la commande concernant un document spécifique.	Test unitaire lancé après avoir créé un objet de type CommandeDocument avec les propriétés « id, dateCommande, montant, nbExemplaire, idLivreDvd, idSuivi, libelle » contenant : « 00070 », «2023, 3,22 », « 50 », « 5 », « 00016 », « 00001 » « en cours »	« 00070 », «2023, 3,22 », « 50 », « 5 », « 00016 », « 00001 », « en cours »	OK
Contrôler la méthode Document() de la classe Document pour voir si elle retourne un document.	Test unitaire lancé après avoir créé un objet de type Document avec les propriétés « id, titre, image, idGenre, genre, idPublic, lePublic, idRayon, rayon » contenant : « 20007 », «Le seigneur des anneaux : le retour du roi », « », « 10019 », « 00016 », « Fantasy », « 00003 », « Tous publics », « DF001 », « DVD films »	« 20007 », «Le seigneur des anneaux : le retour du roi », « », « 10019 », « 00016 », « Fantasy », « 00003 », « Tous publics », « DF001 », « DVD films »	OK
Contrôler la méthode Dvd() de la classe Dvd pour voir si elle retourne un dvd.	Test unitaire lancé après avoir créé un objet de type Dvd avec les propriétés « id, titre, image, duree, realisateur, synopsis, idGenre, genre, idPublic, lePublic, idRayon, rayon » contenant : « 20008 », «Police Academy», « », « 96 », « Hugh Wilson », «Académie de police», « 10013 », « Comédie », « 00003 », « Tous publics », « DF001 », « DVD films »	« 20008 », «Police Academy», « », « 96 », « Hugh Wilson », «Académie de police», « 10013 », « Comédie », « 00003 », « Tous publics », « DF001 », « DVD films »	OK

Contrôler la méthode Etat() de la classe Etat pour voir si elle retourne l'état.	Test unitaire lancé après avoir créé un objet de type Etat, avec le libelle contenant: « neuf »	« neuf »	OK
Contrôler la méthode Exemplaire() de la classe Exemplaire pour voir si elle retourne un exemplaire.	Test unitaire lancé après avoir créé un objet de type Exemplaire avec les propriétés « numero, dateAchat, photo, idEtat, id, libelle, » contenant : « 23 », «2023, 3, 22», « », « 00001», « 00003 », «neuf»	« 23 », «2023, 3, 22», « », « 00001», « 00003 », «neuf»	OK
Contrôler la méthode Genre() de la classe Genre pour voir si elle retourne le genre.	Test unitaire lancé après avoir créé un objet de type Genre avec l'id et le libelle contenant: «10000 » , «Humour »	«10000 » , «Humour »	OK
Contrôler la méthode Livre() de la classe Livre pour voir si elle retourne le livre.	Test unitaire lancé après avoir créé un objet de type Livre avec les propriétés « id, titre, image, isbn, auteur, collection, idGenre, genre, idPublic, lePublic, idRayon, rayon » contenant : « 00023 », «Les 100», « », « 1111111111 », « Kass Morgan », «Laffont», « 10002 », « Science Fiction », « 00003 », « Tous publics », « JN002 », « Jeunesse romans »	« 00023 », «Les 100», « », « 1111111111 », « Kass Morgan », «Laffont», « 10002 », « Science Fiction », « 00003 », « Tous publics », « JN002 », « Jeunesse romans »	OK
Contrôler la méthode Public() de la classe Public pour voir si elle retourne le public.	Test unitaire lancé après avoir créé un objet de type Public avec l'id et le libelle contenant: «10001 » , «Jeunesse »	«10001 » , «Jeunesse »	OK
Contrôler la méthode Rayon() de la classe Rayon pour voir si elle retourne le rayon.	Test unitaire lancé après avoir créé un objet de type Rayon avec l'id et le libelle contenant: «BD001 » , «BD Adultes »	«BD001 » , «BD Adultes »	OK
Contrôler la méthode Revue() de la classe Revue pour voir si elle retourne la revue.	Test unitaire lancé après avoir créé un objet de type Revue avec les propriétés « id, titre, image, idGenre, genre, idPublic, lePublic, idRayon, rayon, periodicite, delaiMiseADispo » contenant : « 10001 », «Arts Magazine», « », « 10016 », « Presse Culturelle », «00002», « Adultes», « PR002 », « 00003 », «Magazines », « MS », « 52 »	« 10001 », «Arts Magazine», « », « 10016 », « Presse Culturelle », «00002», « Adultes», « PR002 », « 00003 », «Magazines », « MS », « 52 »	OK
Contrôler la méthode Service() de la classe Service pour voir si elle retourne le service.	Test unitaire lancé après avoir créé un objet de type Service avec l'id et le libelle contenant: «00001 » , «Administratif »	«00001 » , «Administratif »	OK
Contrôler la méthode Suivi() de la classe Suivi pour voir si elle retourne le suivi.	Test unitaire lancé après avoir créé un objet de type Suivi avec l'id et le libelle contenant: «00001 » , «en cours »	«00001 » , «en cours »	OK
Contrôler la méthode Utilisateur() de la classe Utilisateur pour voir si elle retourne l'utilisateur.	Test unitaire lancé après avoir créé un objet de type Utilisateur avec les propriétés « login, password, idService, libelle » contenant: «ZakariZotto », «ZakariZotto430», « 2 », « administratif »	«ZakariZotto », «ZakariZotto430», « 2 », « administratif »	OK

Tests fonctionnels avec Specflow sur les recherches (onglet livre)

But du test	Action de contrôle	Résultat attendu	Bilan
<p>Scénarios 1 et 5 :</p> <p>Contrôler la recherche d'un livre sur le clic du bouton « Rechercher », de la vue « FrmMediatek », pour voir si la méthode retourne le livre et les informations détaillées du livre telles que le titre.</p>	<p>Test fonctionnel lancé après avoir créé une simulation :</p> <p>De saisie d'une valeur dans le champ « txbLivresNumRecherche » du formulaire « frmMediatek » avec la propriété « valeur » contenant: «00003»</p> <p>De saisie d'un titre dans le champ «txbLivresTitreRecherche » du formulaire « frmMediatek » avec la propriété « titreAttendu » contenant: «Et je danse»</p> <p>De clic sur le bouton « Rechercher »</p>	<p>« 00003 »</p> <p>« Et je danse »</p>	<p>Erreur d'exécution du test</p>
<p>Scénario 2 :</p> <p>Contrôler la recherche de livres sur la sélection d'un genre de la comboBox «cbxLivresGenres » de la vue « FrmMediatek », pour voir si elle retourne la liste de livres associés au genre.</p>	<p>Test fonctionnel lancé après avoir créé une simulation :</p> <p>De sélection d'une valeur « Policier » dans le champ «cbxLivresGenres » du formulaire « frmMediatek » avec la propriété « listeAttendue » contenant: [8] (nombre de livres possédant le genre)</p>	<p>[8]</p>	<p>Erreur d'exécution du test</p>
<p>Scénario 3 :</p> <p>Contrôler la recherche de livres sur la sélection d'un public de la comboBox «cbxLivresPublics » de la vue « FrmMediatek », pour voir si elle retourne la liste de livres associés au public.</p>	<p>Test fonctionnel lancé après avoir créé une simulation :</p> <p>De sélection d'une valeur « Adultes » dans le champ «cbxLivresPublics » du formulaire « frmMediatek » avec la propriété « listeAttendue » contenant: [12] (nombre de livres possédant le public)</p>	<p>[12]</p>	<p>Erreur d'exécution du test</p>
<p>Scénario 4 :</p> <p>Contrôler la recherche de livres sur la sélection d'un rayon de la comboBox «cbxLivresRayons » de la vue « FrmMediatek », pour voir si elle retourne la liste de livres associés au rayon.</p>	<p>Test fonctionnel lancé après avoir créé une simulation :</p> <p>De sélection d'une valeur « Sciences » dans le champ «cbxLivresSciences » du formulaire « frmMediatek » avec la propriété « listeAttendue » contenant: [0] (nombre de livres possédant le rayon)</p>	<p>[0]</p>	<p>Erreur d'exécution du test</p>

Tests fonctionnels dans Postman (fonctionnalités de l'API)

But du test	Action de contrôle	Résultat attendu	Bilan
Contrôler la méthode « selectOne(\$table, \$id) » de la classe « AccessBDD » pour voir si elle retourne l'élément d'une table.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/document/20001 »	{ "id": "20001", "titre": "Star Wars 5 L'empire contre attaque", "image": "", "idRayon": "DF001", "idPublic": "00003", "idGenre": "10002" }	OK
Contrôler la méthode « selectAllLivres() » de la classe « AccessBDD » pour voir si elle retourne la liste de livres.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/livre »	Liste de tout les livres	OK
Contrôler la méthode « selectAllDvd() » de la classe « AccessBDD » pour voir si elle retourne la liste de dvd.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/dvd »	Liste de tout les dvd	OK
Contrôler la méthode « selectAllRevue » de la classe « AccessBDD » pour voir si elle retourne la liste des revues.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/revue »	Liste de toutes les revues	OK
Contrôler la méthode « selectAllExemplairesRevue(\$id) » de la classe « AccessBDD » pour voir si elle retourne la liste des exemplaires d'une revue.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/exemplaire/id » Avec l'id contenant : « 10001 »	{ "id": "10001", "numero": "1", "dateAchat": "2023-03-07", "photo": "", "idEtat": "00001" }, { "id": "10001", "numero": "2", "dateAchat": "2023-03-03", "photo": "", "idEtat": "00001" }	OK
Contrôler la méthode « selectAllCommandesDocument(\$id) » de la classe « AccessBDD » pour voir si elle retourne la liste des commandes d'un document.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/commandedocument/id » Avec l'id contenant : « 00002 »	{ "nbExemplaire": "2", "idLivreDvd": "00002", "idSuivi": "00002", "libelle": "livrée", "id": "00002", "dateCommande": "2023-03-07", "montant": "30" }	OK

		}	
Contrôler la méthode « selectAllAbonnementsRevue(\$id) » de la classe « AccessBDD » pour voir si elle retourne la liste des abonnements d'une revue.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/abonnement/id Avec l'id contenant : « 10001 »	{ "id": "00021", "dateCommande": "2022-12-27", "montant": "23", "dateFinAbonnement": "2023-05-19", "idRevue": "10001" }	OK
Contrôler la méthode « selectAllAbonnementsEcheance » de la classe « AccessBDD » pour voir si elle retourne la liste des abonnements arrivants à échéance dans moins de 30 jours.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/abonnementsecheance »	{ "dateFinAbonnement": "2023-03-21", "idRevue": "10001", "titre": "Arts Magazine" }, { "dateFinAbonnement": "2023-03-31", "idRevue": "10001", "titre": "Arts Magazine" }	OK
Contrôler la méthode « selectAllExemplairesDocument(\$id) » de la classe « AccessBDD » pour voir si elle retourne la liste des exemplaires d'un document.	Test fonctionnel lancé après avoir créé la requête http GET « http://localhost/rest_mediatekdocuments/exemplairesdocument/id Avec l'id contenant : « 00002 »	{ "id": "00002", "numero": "1", "dateAchat": "2023-03-07", "photo": "", "idEtat": "00001", "libelle": "neuf" }, { "id": "00002", "numero": "2", "dateAchat": "2023-03-07", "photo": "", "idEtat": "00001", "libelle": "neuf" }	OK