

Contexte : MediaTek86

Situation professionnelle : Symfony

Application : mediatekformation (site de mise à disposition des auto-formations).

Plan de tests

Tests unitaires

But du test	Action de contrôle	Résultat attendu	Bilan
Contrôler la méthode getPublishedAtString() de la classe Formation pour voir si elle retourne la bonne date au bon format.	Test unitaire lancé avec la date : 20XX-01-04 17:00:12	04/01/20XX	OK

Tests d'intégration

But du test	Action de contrôle	Résultat attendu	Bilan
Contrôler avec testValidDateFormation que lors de l'ajout ou modification d'une date de formation valide la modification soit prise en compte.	Test lancé avec diverses dates valide, contrôlé par assertErrors.	Test valide, 0 erreurs	OK
Contrôler avec testNonValidDateFormation que lors de l'ajout ou modification d'une date de formation non valide la modification ne soit pas prise en compte.	Test lancé avec différentes dates non valide, contrôlé par assertErrors.	4 erreurs, ce qui est le résultat attendu	OK
Contrôler le nombre de formations avec testNbFormation.	Test lancé avec un count([]) sur le formationRepository , contrôlé par assertEquals.	Test valide, NbFormation doit être égale a 237	OK
Contrôler l'ajout d'une formation avec testAddFormation.	Test lancé avec une nouvelle formation pour contrôler avec assertEquals que la quantité de formations correspond.	Test valide, NbFormation + 1 doit être égale à la quantité d'éléments dans le repository	OK
Contrôler la suppression d'une formation avec testSupprFormation.	Test lancé avec une suppression de formation (remove) avec assertEquals pour contrôler que la suppression a été effectué.	Test valide, NbFormation - 1 doit être égale à la quantité d'éléments dans le repository	OK
Contrôler le nombre de playlists avec testNbPlaylist.	Test lancé avec un count([]) sur playlistRepository, contrôlé par assertEquals.	Test valide, NbPlaylist doit être égale à 27	OK
Contrôler l'ajout d'une playlist avec testAddPlaylist.	Test lancé avec une nouvelle playlist pour contrôler avec assertEquals que la quantité de playlists correspond.	Test valide, NbPlaylist + 1 doit être égale à la quantité d'éléments dans le repository	OK
Contrôler la suppression d'une playlist avec testSupprPlaylist.	Test lancé avec une suppression de playlist (remove) avec assertEquals pour contrôler que la suppression a été effectué.	Test valide, NbPlaylist - 1 doit être égale à la quantité d'éléments dans le repository	OK
Contrôler le nombre de catégories avec testNbCategorie.	Test lancé avec un count([]) sur CategorieRepository,	Test valide, NbCategorie doit être égale à 9	OK

	contrôlé par assertEquals		
Contrôler l'ajout d'une catégorie avec testAddCategorie.	Test lancé avec une nouvelle catégorie pour contrôler avec assertEquals que la quantité de playlists correspond.	Test valide, NbCategorie + 1 doit être égale à la quantité d'éléments dans le repository	OK
Contrôler la suppression d'une playlist avec testSupprCategorie.	Test lancé avec une suppression de playlist (remove) et assertEquals pour contrôler que la suppression a été effectué.	Test valide, NbCategorie - 1 doit être égale à la quantité d'éléments dans le repository	OK

Tests fonctionnels

But du test	Action de contrôle	Résultat attendu	Bilan
Contrôler l'accessibilité de la page d'accueil avec testAccesPage (AccueilControllerTest).	Contrôle que la page ne renvoie pas d'erreur avec assertEquals.	Test valide, la requête renvoie code 200 comme attendu	OK
Contrôler le contenu de la page d'accueil avec testContenuPage (AccueilControllerTest).	Contrôle que le premier « h3 » corresponde a l'affichage recherché, idem pour le premier « h5 » avec assertSelectorTextContains. Contrôle la quantité de « h5 » affiché par la page avec assertCount.	Test valide, les éléments correspondent	OK
Contrôler les liens de la page d'accueil avec testLinkFormation (AccueilControllerTest).	Contrôle que le 5ém lien « <a.. » de la page d'accueil renvoie bien sur /formations/formation/1 avec assertEquals.	Test valide, le lien renvoie bien sur la bonne page	OK
Contrôler l'accessibilité de la page /formations avec testAccesPage (FormationsControllerTest).	Contrôle que la page ne renvoie pas d'erreur avec assertEquals.	Test valide, la requête renvoie code 200	OK
Contrôler le contenu de la page /formations avec testContenuPage (FormationsControllerTest).	Contrôle que le premier « th » corresponde a l'affichage recherché avec assertSelectorTextContains. Contrôle la quantité de « th » affiché par la page avec assertCount	Test valide, les éléments correspondent	OK
Contrôler les liens de la page /formations avec testLinkFormation (FormationsControllerTest).	Contrôle que le 9ém lien « <a.. » de la page d'accueil renvoie bien sur /formations/formation/1 avec assertEquals	Test valide, le lien renvoie bien sur la bonne page	OK
Contrôle les filtres de la pages /formations avec testeFiltreFormation (FormationsControllerTest).	Après filtre recherché « Eclipse n°3... » : - contrôle la quantité de « h5 » avec assertCount - contrôle le texte du premier « h5 » avec assertSelectorTextContains.	Test valide, les éléments correspondent	OK
Contrôle les liens de trie de la page /formations avec testeSortFormation (FormationsControllerTest).	Contrôle que les que les textes des premiers « h5 » correspondent a celui attendu a chaque test avec assertSelectorTextContains.	Test valide, les éléments correspondent	OK
Contrôler l'accessibilité de la page /playlists avec testAccesPage (PlaylistsControllerTest).	Contrôle que la page ne renvoie pas d'erreur avec assertEquals	Test valide, la requête renvoie code 200	OK
Contrôler le contenu de la page /playlists avec testContenuPage (PlaylistsControllerTest).	Contrôle que le premier « th » corresponde a l'affichage recherché avec assertSelectorTextContains. Contrôle la quantité de « th » affiché par la page avec assertCount	Test valide, les éléments correspondent	OK
Contrôler les liens de la page /playlists avec testLinkFormation (PlaylistsControllerTest).	Contrôle que le 7ém lien « <a.. » de la page d'accueil renvoie bien sur /playlists/playlist/13 avec assertEquals	Test valide, le lien renvoie bien sur la bonne page	OK

Contrôle les filtres de la pages /playlists avec testeFiltreFormation (PlaylistsControllerTest).	Après filtre recherché « Cours » : - contrôle la quantité de « h5 » avec assertCount - contrôle le texte du premier « h5 » avec assertSelectorTextContains.	Test valide, les éléments correspondent	OK
Contrôle les liens de trie de la page /playlists avec testeSortFormation (PlaylistsControllerTest).	Contrôle que les que les textes des premiers « h5 » correspondent a celui attendu a chaque test avec assertSelectorTextContains.	Test valide, les éléments correspondent	OK

Tests de compatibilité

But du test	Action de contrôle	Résultat attendu	Bilan
Contrôle que l'application fonctionne sur Firefox, test l'affichage après tries, utilisations de liens etc	Contrôles de différentes pages et différentes fonction, le but a été de contrôler les liens et des filtres .	Test valide	OK
Contrôle que l'application fonctionne sur Chrome, test l'affichage après tries, utilisations de liens etc	Contrôle du bon déroulement du script précédent et validé sur Firefox, sur un autre navigateur.	Test valide	OK