

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 <code>getPublishedAtString()</code> de la classe <code>Formation</code> pour voir si elle retourne la bonne date au bon format. 20XX-01-04 17:00:12	Test unitaire lancé avec la date : 04/01/20XX	OK	
Contrôler la méthode <code>getPublishedAtString()</code> de la classe <code>Formation</code> lorsque la date est null.	Test unitaire appelé sans <code>setPublishedAt()</code> (<code>publishedAt = null</code>).	"" (chaîne vide)	OK
Vérifier que <code>getPublishedAtString()</code> retourne une chaîne au format <code>dd/mm/YYYY</code> . 2020-01-05	Test unitaire lancé avec la date : 05/01/2020	OK	

Tests d'intégration

But du test	Action de contrôle	Résultat attendu	Bilan
Vérifier que la contrainte <code>LessThanOrEqual('today')</code> rejette une date future dans le formulaire de <code>Formation</code> .	Validation Symfony appliquée à une date = date d'aujourd'hui + 1 jour.	1 violation retournée avec le message 'La date doit être antérieure ou égale à aujourd'hui.'	OK
Vérifier que la contrainte <code>LessThanOrEqual('today')</code> accepte la date du jour.	Validation Symfony appliquée à une date = date d'aujourd'hui.	0 violation (validation réussie).	OK
Vérifier que la contrainte <code>LessThanOrEqual('today')</code> accepte une date passée.	Validation Symfony appliquée à une date = date d'aujourd'hui - 1 an.	0 violation (validation réussie).	OK

But du test	Action de contrôle	Résultat attendu	Bilan
Vérifier que <code>findAllOrderBy('publishedAt', 'ASC')</code> retourne les formations triées du plus ancien au plus récent.	Insertion de 3 formations avec des dates différentes (2022, 2023, 2024), appel de <code>findAllOrderBy('publishedAt', 'ASC')</code> .	Le tableau retourné commence par la formation de 2022 et se termine par celle de 2024.	OK
Vérifier que <code>findAllOrderBy('publishedAt', 'DESC')</code> retourne les formations de la plus récente à la plus ancienne.	Même jeu de données, appel de <code>findAllOrderBy('publishedAt', 'DESC')</code> .	Le tableau retourné commence par la formation de 2024.	OK
Vérifier que <code>findByContainValue('title', '')</code> retourne toutes les formations.	Insertion de 2 formations, appel de <code>findByContainValue('title', '')</code> .	Le tableau retourné contient 2 formations.	OK
Vérifier que <code>findByContainValue('title', 'Symfony')</code> filtre correctement par titre.	Insertion de 3 formations dont 2 avec 'Symfony' dans le titre, appel de <code>findByContainValue('title', 'Symfony')</code> .	Le tableau retourné contient 2 formations. Les titres contiennent tous 'Symfony'.	OK
Vérifier que <code>findAllLasted(n)</code> retourne les n formations les plus récentes.	Insertion de 3 formations, appel de <code>findAllLasted(2)</code> .	Le tableau retourné contient 2 formations. La première est la plus récente.	OK
Vérifier que <code>findAllForOnePlaylist</code> retourne uniquement les formations liées à la playlist demandée.	Création d'une playlist avec 1 formation et d'une formation sans playlist. Appel de <code>findAllForOnePlaylist(id)</code> .	Le tableau retourné contient 1 formation correspondant à la playlist.	OK
Vérifier que <code>findAllOrderByName('ASC')</code> retourne les playlists triées par nom croissant.	Insertion de 3 playlists ('Zend Framework', 'Angular', 'MySQL'), appel de <code>findAllOrderByName('ASC')</code> .	La première playlist retournée est 'Angular', la dernière 'Zend Framework'.	OK
Vérifier que <code>findAllOrderByName('DESC')</code> retourne les playlists triées par nom décroissant.	Même jeu de données, appel de <code>findAllOrderByName('DESC')</code> .	La première playlist retournée est 'Zend Framework'.	OK
Vérifier que <code>findByContainValue('name', '')</code> retourne toutes les playlists.	Insertion de 2 playlists, appel de <code>findByContainValue('name', '')</code> .	Le tableau retourné contient 2 playlists.	OK

But du test	Action de contrôle	Résultat attendu	Bilan
Vérifier que findByContainValue filtre correctement les playlists par nom.	Insertion de 3 playlists dont 2 avec 'Symfony' dans le nom, appel de findByContainValue('name', 'Symfony').	Le tableau contient 2 playlists dont les noms contiennent tous 'Symfony'.	OK
Vérifier que findAllOrderByNombreFormations('ASC') trie par nombre de formations croissant.	Création d'une playlist avec 0 formation et une avec 3 formations, appel de findAllOrderByNombreFormations('ASC').	La première playlist retournée est celle sans formation.	OK
Vérifier que findAllOrderByNombreFormations('DESC') trie par nombre de formations décroissant.	Même jeu de données, appel de findAllOrderByNombreFormations('DESC').	La première playlist retournée est celle avec 3 formations.	OK
Vérifier que findByContainValueOrderByNombreFormations filtre et trie par nombre de formations.	Création de 2 playlists contenant 'Symfony' dans le nom, l'une avec 2 formations et l'autre sans. Appel en DESC.	Le tableau contient 2 playlists. La première a le plus grand nombre de formations.	OK
Vérifier que findAllForOnePlaylist(idPlaylist) retourne les catégories des formations de la playlist.	Création d'une playlist avec une formation liée aux catégories 'PHP' et 'Symfony', et d'une catégorie 'Laravel' non liée. Appel de findAllForOnePlaylist(id).	Le tableau retourné contient 2 catégories (PHP, Symfony). 'Laravel' est absent.	OK
Vérifier que findAllForOnePlaylist retourne les catégories triées par nom ASC.	Création d'une formation liée aux catégories 'Zend' et 'Angular', appel de findAllForOnePlaylist(id).	La première catégorie retournée est 'Angular'.	OK
Vérifier que upgradePassword met à jour le hash du mot de passe en base.	Création d'un utilisateur avec 'ancien_hash'. Appel de upgradePassword(\$user, 'nouveau_hash'). Rechargement depuis la base.	Le mot de passe stocké en base est 'nouveau_hash'.	OK
Vérifier que upgradePassword lève une UnsupportedUserException si l'entité n'est pas un User.	Appel de upgradePassword avec un objet implémentant PasswordAuthenticatedUserInterface mais pas User.	Une exception UnsupportedUserException est levée.	OK

But du test	Action de contrôle	Résultat attendu	Bilan
Vérifier que findOneBy(['email']) retourne le bon utilisateur.	Création de 2 utilisateurs avec des emails différents. Recherche par email du premier.	L'utilisateur retourné a l'email recherché et le rôle ROLE_ADMIN.	OK
Vérifier que findOneBy(['email']) retourne null si l'email est inexistant.	Appel de findOneBy(['email' => 'inconnu@test.com']) sur une base sans cet email.	null est retourné.	OK

Tests fonctionnels

But du test	Action de contrôle	Résultat attendu	Bilan
Vérifier que la page d'accueil (/) se charge correctement.	Requête GET sur /. Aucune authentification requise.	Réponse HTTP 200.	OK
Vérifier que la page d'accueil affiche le titre de bienvenue.	Requête GET sur /. Inspection du contenu HTML.	La balise <h3> contient le texte 'Bienvenue'.	OK
Vérifier que la page d'accueil propose des liens vers les formations et les playlists.	Requête GET sur /. Vérification des liens dans le HTML.	Les liens href='/formations' et href='/playlists' sont présents.	OK
Vérifier que le tri par titre ASC place la formation 'Bases de Python' en première ligne.	Base de test : 3 formations ('Eclipse PHP' 2024, 'Bases de Python' 2023, 'Symfony avancé' 2022). Clic sur le bouton '<' de la colonne formation.	La première ligne du tableau affiche 'Bases de Python'.	OK
Vérifier que le tri par titre DESC place la formation 'Symfony avancé' en première ligne.	Même base de test. Clic sur le bouton '>' de la colonne formation.	La première ligne du tableau affiche 'Symfony avancé'.	OK
Vérifier que le tri par date DESC place la formation la plus récente en première ligne.	Même base de test. Clic sur le bouton '>' de la colonne date.	La première ligne du tableau affiche 'Eclipse PHP' (publiée en 2024).	OK
Vérifier que le tri par date ASC place la formation la plus ancienne en première ligne.	Même base de test. Clic sur le bouton '<' de la colonne date.	La première ligne du tableau affiche 'Symfony avancé' (publiée en 2022).	OK

But du test	Action de contrôle	Résultat attendu	Bilan
Vérifier que le tri par nom ASC place la playlist 'Algorithmes' en première ligne.	Base de test : 2 playlists ('Algorithmes' avec 2 formations, 'Débutants Python' avec 1). Clic sur '<' de la colonne playlist.	La première ligne du tableau affiche 'Algorithmes'.	OK
Vérifier que le tri par nom DESC place 'Débutants Python' en première ligne.	Même base de test. Clic sur '>' de la colonne playlist.	La première ligne du tableau affiche 'Débutants Python'.	OK
Vérifier que le tri par nombre de formations ASC place la playlist la moins fournie en premier.	Même base de test. Clic sur '<' de la colonne nombre de formations.	La première ligne affiche 'Débutants Python' (1 formation).	OK
Vérifier que le tri par nombre de formations DESC place la playlist la plus fournie en premier.	Même base de test. Clic sur '>' de la colonne nombre de formations.	La première ligne affiche 'Algorithmes' (2 formations).	OK
Vérifier que le filtre par titre retourne uniquement les formations correspondantes.	Même base de test (3 formations). Saisie de 'Eclipse' dans le champ de filtre de la colonne formation, clic sur 'filtrer'.	Le tableau affiche 1 ligne. La première ligne contient 'Eclipse PHP'.	OK
Vérifier que le filtre par titre sans résultat affiche un tableau vide.	Saisie de 'inexistant_xyz' dans le filtre titre.	Le tableau ne contient aucune ligne de données.	OK
Vérifier que le filtre par playlist retourne toutes les formations de cette playlist.	Saisie de 'Algorithmes' dans le filtre playlist.	Le tableau affiche 2 lignes (Eclipse PHP et Symfony avancé).	OK
Vérifier que le filtre par nom retourne uniquement la playlist correspondante.	Saisie de 'Algo' dans le champ de filtre de la colonne playlist.	Le tableau affiche 1 ligne contenant 'Algorithmes'.	OK
Vérifier que le filtre playlists sans résultat affiche un tableau vide.	Saisie de 'inexistant_xyz' dans le filtre nom.	Le tableau ne contient aucune ligne de données.	OK
Vérifier que le filtre playlists avec une valeur vide retourne toutes les playlists.	Saisie d'une chaîne vide dans le filtre nom, clic sur 'filtrer'.	Le tableau affiche les 2 playlists.	OK

But du test	Action de contrôle	Résultat attendu	Bilan
Vérifier que le clic sur la miniature d'une formation ouvre la page de détail correcte.	Tri par titre ASC. Clic sur le lien miniature de la première ligne ('Bases de Python').	La page de détail s'affiche (HTTP 200). Le titre 'Bases de Python' est présent dans la balise <h4>.	OK
Vérifier que la page de détail d'une formation affiche la playlist associée.	Navigation vers la page de détail d'une formation liée à 'Débutants Python'.	Le nom 'Débutants Python' est visible dans le contenu de la page.	OK
Vérifier que le bouton 'Voir détail' d'une playlist ouvre la page de détail correcte.	Tri par nom ASC. Clic sur 'Voir détail' de la première ligne ('Algorithmes').	La page de détail de la playlist s'affiche (HTTP 200). Le nom 'Algorithmes' est présent.	OK
Vérifier que la page de détail d'une playlist affiche ses formations associées.	Navigation vers la page de détail de la playlist 'Algorithmes' (2 formations).	Les titres 'Eclipse PHP' et 'Symfony avancé' sont présents dans le contenu de la page.	OK

Tests de compatibilité

But du test	Action de contrôle	Résultat attendu	Bilan
Vérifier que l'application s'affiche et fonctionne correctement sur Google Chrome.	Ouverture de l'application dans Google Chrome (dernière version stable). Navigation sur les pages formations, playlists, détail. Utilisation des tris et filtres.	Toutes les pages s'affichent correctement. Les tris, filtres et liens fonctionnent sans erreur. La mise en page Bootstrap est respectée.	OK
Vérifier que l'application s'affiche et fonctionne correctement sur Mozilla Firefox.	Ouverture de l'application dans Mozilla Firefox (dernière version stable). Navigation identique à celle effectuée sur Chrome.	Toutes les pages s'affichent correctement. Les tris, filtres et liens fonctionnent sans erreur. La mise en page Bootstrap est respectée.	OK
Vérifier que l'application s'affiche et fonctionne correctement sur Apple Safari.	Ouverture de l'application dans Safari (dernière version stable sous macOS). Navigation identique aux tests précédents.	Toutes les pages s'affichent correctement. Les tris, filtres et liens fonctionnent sans erreur. La mise en page Bootstrap est respectée.	OK