

SAE S4

SITUATION

D'APPRENTISSAGE ET

D'ÉVALUATION

(Livrable de tests)

CAZOR--BONNET Adrian
CHESNIER Quentin
JUERY Clément
LADUREAU Baptiste

BUT Info2



Test sur DefaultComponentFactory	3
Tests sur les services	4
Résumé des tests pour Appel ApiService.....	4
Résumé des tests pour Connexion Service.....	4
Résumé des tests pour Facture Service Integration Test.....	5
Résumé des tests pour Facture Service Test.....	5
Résumé des tests pour Fournisseurs Service Test.....	6
Résumé des tests pour Notes De Frais Service Test.....	7
Tests sur les controllers	9
Résumé des tests pour HomeController.....	9
Résumé des tests pour Connexion Controller.....	9
Résumé des tests pour Affichage Palmare Fournisseurs Controller.....	10
Résumé des tests pour Affichage Liste Note Frais Controller Test.....	12
Résumé des tests pour Affichage Liste Fournisseurs Controller Test.....	14
Résumé des tests pour Affichage Facture Controller Test.....	16

Test sur DefaultComponentFactory

Tests de création de contrôleurs

Vérification de la création des contrôleurs suivants :

- HomeController
- ConnexionController
- AffichageListeNoteFraisController
- AffichageListeFournisseursController
- AffichagePalmairesFournisseursController
- Un test vérifie qu'une exception (NoControllerAvailableForNameException) est levée lorsqu'un contrôleur inexistant est demandé.

Tests de création de services :

Vérification de la création des services suivants :

- ConnexionService
- NotesDeFraisService
- FournisseursService
- FactureService
- Un test vérifie qu'une exception (NoServiceAvailableForNameException) est levée lorsqu'un service inexistant est demandé.

Objectif des tests :

S'assurer que la factory (DefaultComponentFactory) est capable de créer correctement les contrôleurs et services attendus.

Vérifier la levée d'exceptions en cas de demande d'un élément inexistant.

Tests sur les services

Résumé des tests pour Appel ApiService

- Tests d'appels API avec une clé valide :
- testValidApiKeyThirdparties : Vérifie que l'API retourne des données valides lorsqu'on interroge l'endpoint thirdparties avec une clé valide.
- testValidApiKeyExpenseReports : Vérifie que l'API retourne des données valides lorsqu'on interroge l'endpoint expensereports avec une clé valide.
- Tests d'échec des appels API :
- testInvalidApiKey : Vérifie que l'API ne retourne rien lorsqu'on utilise une clé invalide.
- testInvalidEndpoint : Vérifie que l'API ne retourne rien lorsqu'on interroge un endpoint inexistant.
- testInvalidUrl : Vérifie que l'appel à une URL incorrecte ne retourne aucune donnée.

Objectif des tests

- Vérifier que l'API répond correctement avec une clé valide.
- Vérifier que l'API rejette les requêtes incorrectes (clé invalide, mauvais endpoint, URL incorrecte).

Résumé des tests pour ConnexionService

Test de connexion avec des identifiants valides :

- testConnexionAPINominal : Vérifie que la connexion à l'API avec un login et un mot de passe valides retourne une réponse contenant la clé API correcte.
- Tests des cas limites (login/mot de passe vide)
- testConnexionAPILimiteLoginVide : Vérifie que l'API retourne une réponse vide si le login est vide.
- testConnexionAPILimitePasswordVide : Vérifie que l'API retourne une réponse vide si le mot de passe est vide.
- testConnexionAPILimiteLoginPasswordVide : Vérifie que l'API retourne une réponse vide si le login et le mot de passe sont vides.

Objectif des tests

- Vérifier que la connexion fonctionne correctement avec des identifiants valides.
- Vérifier que l'API rejette les tentatives de connexion incorrectes (login/mot de passe vide).

Résumé des tests pour FactureServiceIntegrationTest

Test de récupération des factures :

- testGetFacturesIntegration : Vérifie que l'appel à getFactures() retourne un tableau valide lorsqu'une URL et une clé API sont définies en session.

Test de récupération des factures avec critères :

- testGetFacturesAvecCriteresIntegration : Vérifie que l'appel à getFacturesAvecCriteres() avec un critère (Test) retourne un tableau non vide.

Objectif des tests

- Tester l'intégration entre FactureService et Appel ApiService.
- Vérifier que la récupération des factures fonctionne avec et sans critères.

Résumé des tests pour FactureServiceTest

Tests sur getFactures :

- Cas nominal (testGetFacturesNominal)
 - Vérifie que la récupération des factures retourne un tableau contenant 8 éléments avec des valeurs spécifiques.
- Clé API invalide (testGetFacturesCleApiInvalide)
 - Vérifie que l'API retourne une réponse vide avec une clé invalide.
- Clé API non autorisée (testGetFacturesCleApiNonAutorisee)
 - Vérifie que l'API ne retourne rien si la clé API n'a pas les permissions nécessaires.

Tests sur getFacturesAvecCriteres :

- Cas nominal avec API réelle (testGetFacturesAvecCriteresWithRealApi)
 - Vérifie que l'API retourne bien un tableau en fournissant des critères précis.
- Aucun résultat trouvé (testGetFacturesAvecCriteresWithRealApiNoMatch)
 - Vérifie qu'une requête avec un critère inexistant retourne une réponse vide.
- Aucune clé API fournie (testGetFacturesAvecCriteresSansCleApi)
 - Vérifie que l'API retourne une réponse vide si aucune clé API n'est fournie.
- Tous les champs vides (testGetFacturesAvecCriteresWithAllFieldEmpty)
 - Vérifie que la récupération des factures retourne un résultat même si aucun critère n'est renseigné.

Tests sur getDocumentsLies :

- Cas nominal (testGetDocumentsLies)
 - Vérifie que l'API retourne bien un tableau avec 2 documents liés pour une facture donnée.

Aucune clé API fournie (testGetDocumentsLiesSansApiKey)

- Vérifie que l'API retourne une réponse vide si aucune clé API n'est fournie.

L'API ne retourne aucun document (testGetDocumentsLiesApiReturnsEmpty)

- Vérifie qu'une référence valide mais sans documents liés retourne une réponse vide.

Référence invalide (testGetDocumentsLiesWithInvalidReference)

- Vérifie que l'API retourne une réponse vide pour une référence de facture inexistante.

Objectif des tests :

Vérifier la récupération des factures avec ou sans critères.

- Tester les cas limites (clé API invalide, données absentes, référence erronée).
- Vérifier la récupération des documents liés à une facture.

Résumé des tests pour FournisseursServiceTest

Tests sur getFournisseurs :

- Cas nominal (testGetFournisseursNominal)
 - Vérifie que la récupération des fournisseurs retourne un tableau contenant 2 fournisseurs avec des chiffres d'affaires spécifiques.
- Aucun fournisseur trouvé (testGetFournisseursAucunFournisseur)
 - Vérifie que l'API retourne une réponse vide lorsqu'aucun fournisseur n'est trouvé.
- Chiffre d'affaires non défini (testGetFournisseursSansChiffreAffaire)
 - Vérifie que le chiffre d'affaires est considéré comme 0 si le champ est absent dans la réponse.
- Réponse invalide (testGetFournisseursReponseInvalide)
 - Vérifie que l'API retourne une réponse vide si la réponse reçue est invalide (null).
- API réelle avec clé valide (testGetFournisseursAvecApiReelle)
 - Vérifie que la récupération des fournisseurs fonctionne avec l'API réelle et que les résultats sont triés par chiffre d'affaires décroissant.
- API réelle sans clé API (testGetFournisseursAvecApiReelleSansCleApi)
 - Vérifie que l'API retourne une réponse vide si aucune clé API n'est fournie.

Tests sur getPalmaresFournisseurs :

- Cas nominal avec API réelle (testGetPalmaresFournisseursAvecApiReelle)
 - Vérifie que la récupération du palmarès des fournisseurs fonctionne correctement avec des dates de début et de fin définies.

Objectif des tests :

- Tester les cas limites (réponse invalide, absence de clé API, aucun fournisseur trouvé).
- Assurer que les fournisseurs sont triés correctement.

Résumé des tests pour NotesDeFraisServiceTest

Tests sur getNotesDeFrais :

- Cas nominal (testGetNotesDeFraisIntegration)
 - Vérifie que la récupération des notes de frais retourne un tableau non vide avec une clé API valide.
- Clé API invalide (testGetNotesDeFraisWithInvalidApiKey)
 - Vérifie que la méthode retourne un tableau vide lorsqu'une clé API invalide est fournie.
- Clé API sans droits (testGetNotesDeFraisWithNoRights)
 - Vérifie que la récupération des notes de frais échoue (tableau vide) si la clé API n'a pas les droits nécessaires.
- Clé API non définie (testGetNotesDeFraisWithNoApiKey)
 - Vérifie que la méthode retourne un tableau vide lorsque la clé API est absente.

Tests sur getNotesDeFraisAvecCriteres :

- Cas nominal (testGetNotesDeFraisAvecAucunCriteresIntegration)
 - Vérifie que la récupération des notes de frais avec critères retourne un tableau non vide.
- Critères invalides (testGetNotesDeFraisAvecCriteresInvalideIntegration)
 - Vérifie que la récupération des notes de frais avec des critères invalides retourne un tableau vide.
- API sans clé (testGetNotesDeFraisAvecAucunCriteresIntegrationWithoutApiKey)
 - Vérifie que l'API retourne un tableau vide si aucune clé API n'est fournie.
- API avec clé sans droits (testGetNotesDeFraisAvecAucunCriteresIntegrationWithoutRights)
 - Vérifie que la récupération des notes de frais échoue (tableau vide) si la clé API ne possède pas les droits requis.
- API avec clé invalide (testGetNotesDeFraisAvecAucunCriteresIntegrationWithInvalidApiKey)
 - Vérifie que la récupération des notes de frais échoue (tableau vide) si la clé API est invalide.

Tests sur getNotesFraisByType :

- Cas nominal (testGetNotesFraisByTypeWithValidDates)
 - Vérifie que la récupération des statistiques par type retourne un tableau trié par montant décroissant.
- Comparaison des montants (testCompareNotesFraisTypesMontants)
 - Vérifie que les montants sont bien triés en ordre décroissant par type de note de frais.
- Autre méthode de récupération par type (testGetNotesFraisByType2WithValidDates)
 - Vérifie que la méthode alternative retourne un tableau valide de statistiques.

Tests sur getNotesFraisByMonth :

- Cas nominal (testGetNotesFraisByMonth)
 - Vérifie que la récupération des montants et du nombre de notes de frais par mois fonctionne correctement.

Tests sur getNotesFraisByDay :

- Cas nominal (testGetNotesFraisByDay)
 - Vérifie que la récupération des montants et du nombre de notes de frais par jour fonctionne correctement.

Objectifs des tests

- Vérifier la récupération des notes de frais avec ou sans critères.
- Tester les cas limites (clé API invalide, non définie, sans droits, critères invalides).
- Assurer que les statistiques sont correctement triées (par type, mois, jour).
- Vérifier le fonctionnement avec l'API réelle.

Tests sur les controllers

Résumé des tests pour HomeController

Tests de la méthode index :

- testIndex : Vérifie que lorsque la méthode index est appelée, une vue de type View est retournée.

Tests de la méthode redirectionAcceuil :

- testRedirectionAcceuil : Vérifie que si une clé API valide est présente dans la session, la méthode redirectionAcceuil retourne une vue de type View.
- testRedirectionAcceuilWithoutApiKey : Vérifie que si la clé API est absente ou invalide dans la session, une exception est levée avec le message "API key is not set in session."

Objectif des tests :

Vérifier la méthode index :

- Assurer que la méthode retourne correctement une vue de connexion.

Vérifier la méthode redirectionAcceuil avec une clé API valide :

- Assurer que lorsque la clé API est présente dans la session, la redirection vers l'accueil fonctionne correctement et retourne une vue.

Vérifier la méthode redirectionAcceuil sans clé API valide :

- Assurer que si la clé API est absente ou invalide, l'application lève une exception avec le bon message d'erreur.

Résumé des tests pour ConnexionController

Tests des méthodes de ConnexionController :

Tests de la méthode getUser :

- test GetUser : Vérifie que lorsque la méthode getUser est appelée avec des informations de connexion valides, elle retourne une vue de type View.
- test GetUserWithoutApiKey : Vérifie que lorsque la méthode getUser est appelée sans clé API valide, elle retourne une vue de type View (connexions).

Tests de la méthode deconnexion :

- testDeconnexion : Vérifie que lorsque la méthode deconnexion est appelée, elle retourne une vue de type View.

Objectif des tests :

Vérifier la méthode getUser avec des informations valides :

- Assurer que la méthode retourne correctement une vue de type View lorsque les informations de connexion sont valides.

Vérifier la méthode getUser sans clé API :

- Assurer que même sans une clé API valide, la méthode retourne correctement une vue de type View pour la connexion.

Vérifier la méthode deconnexion :

- Assurer que la méthode deconnexion retourne une vue de type View lorsqu'elle est appelée.

Résumé des tests pour AffichagePalmaresFournisseurs Controller

Tests de la méthode checkSession :

- testCheckSession : Vérifie que la méthode checkSession retourne une vue de type View lorsque la session n'est pas authentifiée (clé API absente).

Tests de la méthode afficherVueDebut :

- testAfficherVueDebut : Vérifie que la méthode afficherVueDebut retourne une vue de type View lorsque la session est authentifiée.

Tests de la méthode recherche :

- testRecherche : Vérifie que la méthode recherche retourne une vue de type View avec les données du palmarès des fournisseurs basées sur les paramètres GET spécifiés (date, topSelector, etc.).
- testRechercheSansCleApi : Vérifie que la méthode recherche retourne une vue de type View même sans clé API valide dans la session.

Tests de la méthode remiseAZero :

- testRemiseAZero : Vérifie que la méthode remiseAZero réinitialise les paramètres de la recherche (date, topSelector) à leurs valeurs par défaut et retourne une vue de type View.

Tests des méthodes de redirection :

- testRedirectionAcceuil : Vérifie que la méthode redirectionAcceuil retourne une vue de type View avec le template approprié.
- testRedirectionNoteFrais : Vérifie que la méthode redirectionNoteFrais retourne une vue de type View.
- testRedirectionFournisseurs : Vérifie que la méthode redirectionFournisseurs retourne une vue de type View.

Objectif des tests :

Vérifier la méthode checkSession :

- Assure que la méthode retourne une vue de type View lorsqu'il n'y a pas de clé API dans la session.

Vérifier la méthode afficherVueDebut :

- Assure que la méthode retourne correctement une vue de type View lorsque la session est authentifiée.

Vérifier la méthode recherche :

- Assure que la méthode retourne une vue de type View avec les paramètres GET utilisés (dates, topSelector).

Vérifier le comportement lorsque la clé API est absente (retourne une vue sans données spécifiques).

Vérifier la méthode remiseAZero :

- Assure que la méthode réinitialise correctement les paramètres de recherche à leurs valeurs par défaut et retourne une vue de type View.

Vérifier les méthodes de redirection :

- Assure que les méthodes de redirection (redirectionAcceuil, redirectionNoteFrais, redirectionFournisseurs) retournent correctement une vue de type View avec le bon template.

Résumé des tests pour AffichageListeNoteFraisControllerTest

Tests de la méthode recherche :

- testRecherche : Vérifie que la méthode recherche retourne une instance de View et que les critères de recherche sont bien passés à la vue (référence de la note, dates, montant, type de frais, etc.).

Tests de la méthode remiseAZero :

- testRemiseAZero : Vérifie que la méthode remiseAZero réinitialise les critères de recherche et retourne une instance de View avec les critères réinitialisés.

Tests d'affichage de vues :

- testAffichageListeNoteFraisVide : Vérifie que la méthode affichageListeNoteFraisVide retourne une instance de View.
- testAffichageSectorielNotesFrais : Vérifie que la méthode affichageSectorielNotesFrais retourne une instance de View.
- testAffichageEvolutionNotesFrais : Vérifie que la méthode affichageEvolutionNotesFrais retourne une instance de View.

Tests des méthodes de redirection :

- testRedirectionMethods : Vérifie que les méthodes de redirection (redirectionAccueil, redirectionNoteFrais, redirectionFournisseurs) retournent toutes une instance de View.

Tests de la méthode affichageListeNoteFrais :

- testAffichageListeNoteFrais : Vérifie que la méthode affichageListeNoteFrais retourne une instance de View et que la vue contient une liste de notes de frais sous forme de tableau.

Tests de la méthode checkSession :

- testCheckSession : Vérifie que la méthode checkSession retourne la vue de connexion si la session n'est pas authentifiée ou si la clé API est vide, et null si l'utilisateur est authentifié.

Objectif des tests :

Vérifier la méthode recherche :

- Assure que la méthode retourne une vue de type View et que les paramètres de recherche sont bien passés à la vue.

Vérifier la méthode remiseAZero :

- Assure que la méthode réinitialise correctement les critères de recherche et retourne une vue de type View avec les valeurs réinitialisées.

Vérifier les méthodes d'affichage de vues :

- Assure que les méthodes affichageListeNoteFraisVide, affichageSectorielNotesFrais, et affichageEvolutionNotesFrais retournent correctement une vue de type View.

Vérifier les méthodes de redirection :

- Assure que les méthodes de redirection retournent correctement une vue de type View.

Vérifier la méthode affichageListeNoteFrais :

- Assure que la méthode retourne une vue contenant la liste des notes de frais sous forme de tableau.

Vérifier la méthode checkSession :

- Assure que la méthode retourne la vue de connexion lorsque la session est non authentifiée, et null lorsque l'utilisateur est authentifié.

Résumé des tests pour AffichageListeFournisseursControllerTest

Test de la méthode checkSession :

- testCheckSession : Vérifie que la méthode checkSession retourne une instance de View lorsque la session est vide. Cela signifie que l'utilisateur n'est pas authentifié.

Test de la méthode afficherFournisseurs sans clé API :

- testAfficherFournisseursSansCle : Vérifie que la méthode afficherFournisseurs retourne une vue de type View lorsque la session est vide, ce qui simule une session non authentifiée.

Test de la méthode afficherFournisseurs avec clé API :

- testAfficherFournisseursAvecCle : Vérifie que la méthode afficherFournisseurs retourne une vue de type View lorsque la session est authentifiée (clé API présente).

Test de la méthode recherche sans clé API :

- testRecherche : Vérifie que la méthode recherche retourne une vue de type View lorsque la session est vide, ce qui simule une session non authentifiée.

Test de la méthode recherche avec paramètres de recherche sans clé API :

- testRechercheWithParamsSansCle : Vérifie que la méthode recherche retourne une vue de type View même si des paramètres de recherche sont définis, mais que la session est vide (utilisateur non authentifié).

Test de la méthode recherche avec paramètres de recherche avec clé API :

- testRechercheWithParamsAvecCle : Vérifie que la méthode recherche retourne une vue de type View lorsque la session est authentifiée (clé API présente) et que des paramètres de recherche sont définis.

Test de la méthode remiseAZero sans clé API :

- testRemiseAZeroSansCle : Vérifie que la méthode remiseAZero retourne une vue de type View lorsque la session est vide, simulant ainsi une session non authentifiée.

Test de la méthode remiseAZero avec clé API :

- testRemiseAZeroAvecCle : Vérifie que la méthode remiseAZero retourne une vue de type View et réinitialise correctement les paramètres de recherche lorsque la session est authentifiée (clé API présente). Les paramètres de recherche doivent être vides.

Test de la méthode afficherVueDebut sans clé API :

- testAfficherVueDebutSansCle : Vérifie que la méthode afficherVueDebut retourne une vue de type View lorsque la session est vide (utilisateur non authentifié).

Test de la méthode afficherVueDebut avec clé API :

- testAfficherVueDebutAvecCle : Vérifie que la méthode afficherVueDebut retourne une vue de type View lorsque la session est authentifiée (clé API présente).

Test des méthodes de redirection :

- testRedirectionMethods : Vérifie que les méthodes de redirection (redirectionAccueil, redirectionNoteFrais, redirectionFournisseurs) retournent toutes une instance de View.

Objectif des tests :

Vérifier la méthode checkSession :

- Assure que la méthode retourne une vue de type View lorsque la session est vide, indiquant que l'utilisateur n'est pas authentifié.

Vérifier la méthode afficherFournisseurs sans clé API :

- Assure que la méthode retourne une vue lorsque la session est vide (utilisateur non authentifié).

Vérifier la méthode afficherFournisseurs avec clé API :

- Assure que la méthode retourne une vue lorsque la session est authentifiée (clé API présente).

Vérifier la méthode recherche sans clé API :

- Assure que la méthode retourne une vue même si des paramètres sont définis, mais que la session est vide.

Vérifier la méthode recherche avec paramètres de recherche avec clé API :

- Assure que la méthode retourne une vue et que les paramètres de recherche sont correctement pris en compte lorsque la session est authentifiée.

Vérifier la méthode remiseAZero sans clé API :

- Assure que la méthode retourne une vue lorsque la session est vide (utilisateur non authentifié).

Vérifier la méthode remiseAZero avec clé API :

- Assure que la méthode réinitialise les paramètres de recherche correctement et retourne une vue lorsque la session est authentifiée.

Vérifier la méthode afficherVueDebut sans clé API :

- Assure que la méthode retourne une vue lorsque la session est vide (utilisateur non authentifié).

Vérifier la méthode afficherVueDebut avec clé API :

- Assure que la méthode retourne une vue lorsque la session est authentifiée.

Vérifier les méthodes de redirection :

- Assure que les méthodes de redirection retournent correctement une vue de type View.

Résumé des tests pour AffichageFactureControllerTest

Test de la méthode affichageFacture :

- testAfficherFactures : Vérifie que la méthode affichageFacture retourne une instance de View lorsqu'elle est appelée.

Test de la méthode redirectionAcceuil :

- testRedirectionAcceuil : Vérifie que la méthode redirectionAcceuil retourne une instance de View lorsqu'elle est appelée.

Test de la méthode redirectionNoteFrais :

- testRedirectionNoteFrais : Vérifie que la méthode redirectionNoteFrais retourne une instance de View lorsqu'elle est appelée.

Test de la méthode redirectionFournisseurs :

- testRedirectionFournisseurs : Vérifie que la méthode redirectionFournisseurs retourne une instance de View lorsqu'elle est appelée. Cela garantit que la redirection vers la liste des fournisseurs fonctionne correctement.

Objectif des tests :

Vérifier la méthode affichageFacture :

- Assure que la méthode retourne une vue de type View lors de son appel, garantissant ainsi que l'affichage des factures fonctionne correctement.

Vérifier la méthode redirectionAcceuil :

- Assure que la méthode retourne une vue de type View lors de la redirection vers la page d'accueil, indiquant que la redirection fonctionne comme prévu.

Vérifier la méthode redirectionNoteFrais :

- Assure que la méthode retourne une vue de type View lors de la redirection vers la page des notes de frais, garantissant ainsi la bonne redirection.

Vérifier la méthode redirectionFournisseurs :

- Assure que la méthode retourne une vue de type View lors de la redirection vers la page des fournisseurs, garantissant le bon fonctionnement de cette redirection.

Conclusion

Les tests existants couvrent bien les fonctionnalités de base, mais pour assurer une couverture complète et robuste, il est essentiel d'ajouter des tests qui prennent en compte les cas d'erreur, les limites, ainsi que des scénarios de sécurité et de performance. Par exemple, tester l'application avec des volumes de données élevés permettra de s'assurer que les algorithmes réagissent correctement en cas de surcharge. Ces ajouts contribueront à rendre les tests plus fiables, garantissant ainsi que l'application se comporte correctement même dans des situations imprévues.