Rapport de tests
—
Projet « Pur Beurre »

Table des matières

1. Plan de tests

Le plan de tests concerne le traitement **côté serveur**, c'est-à-dire le code Python qui s'appuie sur le framework Django.

Voici l'arborescence du répertoire regroupant les tests :

```
tests
|-------  functional_tests
|         |-------  tests_functional.py
|-------  integration_tests
|         |-------  tests_integration.py
|-------  unit_tests_auth
|         |-------  tests_my_user.py
|-------  unit_tests_off_sub
|         |-------  tests_context_processors.py
|         |-------  tests_db_init.py
|         |-------  tests_db_update.py
|         |-------  tests_models_category.py
|         |-------  tests_models_product.py
|         |-------  tests_models_store.py
|         |-------  tests_views.py
```

Ce répertoire se situe au même niveau que les applications « off_sub » et « auth » dans l'arborescence du projet :

```
pur_beurre
|-------  auth
|-------  off_sub
|-------  pur_beurre
|-------  tests
|-------  venv
|-------  …
```

1.1. Tests unitaires

1.1.1. Application « auth » — tests du module « models »

Classe de test : tests.units_tests_auth.tests_my_users.MyUserTestCase

Target class	Target method	Test method	Test purpose	Test id
MyUser	__str__	test_str_user	Test if user printing is correct.	1
	has_perm	test_user_has_perm	Test if user method 'has_perm' is correct.	2
	has_module_perms	test_user_has_module_perms	Test if user method 'has_module_perms' is correct.	3
	is_staff	test_user_is_not_staff	Test if user is not admin.	4
MyUserManager	create_super_user	test_create_superuser	Test if one record has been added to the 'MyUser' table (my_auth_myuser) with superuser permissions.	5
	create_user	test_create_user	Test if one record has been added to the 'MyUser' table (my_auth_myuser).	6
		test_create_user_without_email	Test if a ValueError is raised if no email is provided.	7

1.1.2. Application « off_sub » — tests du module « context_processors »

Classe de test : tests.units_tests_off_sub.tests_context_processors.ContextProcessorsTestCase

Target function	Test method	Test purpose	Test id
pur_beurre_all_products	test_off_sub_cp_all_products_results_type_is_dict	Test that the results is a dict.	8
	test_off_sub_cp_all_products_value_type_is_list_of_strings	Test that, in the returned dict, the value for key 'all_products', is a list of strings.	9
	test_off_sub_cp_all_products_really_all	Test that all products are in the returned dict, used as context in views.	10

1.1.3. Application « off_sub » — tests du module « management.commands.db_init »

Classe de test : tests.units_tests_off_sub.tests_db_init.DatabaseInitializationTestCase

Target object	Target method	Test method	Test purpose	Test id
Command	handle	test_get_all_products	Test if all products are returned.	11
		test_get_code_product_a_and_product_b	Test if product codes are correct.	12
		test_get_product_name_product_a	Test if product name is correct (with a given name).	13
		test_get_product_name_product_b	Test if product name is correct (without any given name).	14
		test_nutriscore_grade_product_a_and_product_b	Test if product nutriscore grades are correct.	15
		test_nutriscore_score_product_a_and_product_b	Test if product nutriscore scores are correct.	16
		test_get_url_product_a	Test if product url is correct (with a given url).	17
		test_get_url_product_b	Test if product url is correct (without any given url).	18
		test_get_image_url_product_a	Test if product image url is correct (with a given url).	19
		test_get_image_url_product_b	Test if product image url is correct (without any given url).	20
		test_get_fat_product_a	Test if fat data are correct (with given data).	21
		test_get_fat_product_b	Test if fat data are correct (without any given data).	22
		test_get_saturated_fat_product_a	Test if saturated fat data are correct (with given data).	23
		test_get_saturated_fat_product_b	Test if saturated fat data are correct (without any given data).	24
		test_get_sugars_product_a	Test if sugars data are correct (with given data).	25
		test_get_sugars_product_b	Test if sugars data are correct (without any given data).	26
		test_get_salt_product_a	Test if salt data are correct (with given data).	27
		test_get_salt_product_b	Test if salt data are correct (without any given data).	28
		test_get_all_stores	Test if all stores are returned.	29
		test_get_stores_names	Test if stores names are properly parsed.	30
		test_stores_linked_to_product_a	Test if all links between product and stores are correct (if applicable).	31
		test_no_store_linked_to_product_b	Test if there are no store link to a product (if applicable).	32

1.1.4. Application « off_sub » — tests du module « management.commands.db_update »

Classe de test : tests.units_tests_off_sub.tests_db_init.DatabaseInitializationTestCase

Target object	Target method	Test method	Test purpose	Test id
Command	handle	test_update_product_name	Test if product name is updated.	105
		test_update_nutriscore_grade	Test if product nutriscore grade is correct.	106
		test_update_nutriscore_score	Test if product nutriscore score is updated.	107
		test_update_url	Test if product url is updated.	108
		test_update_image_url	Test if product image url is updated.	109
		test_update_fat	Test if fat data are updated.	110
		test_update_saturated_fat	Test if saturated fat data are updated.	111
		test_update_sugars	Test if sugars data are updated.	112
		test_update_salt	Test if salt data are updated.	113

1.1.5. Application « off_sub » — tests du module « models »

Classe de test : tests.units_tests_off_sub.tests_models_category.CategoryTestCase

Target object	Target method	Test method	Test purpose	Test id
Category	get_url_250_products	test_get_url_250_products_right_category	Test if the returned URL is well as expected.	33
		test_get_url_250_products_wrong_category	Test if the returned URL is an empty string.	34
	add_category_to_db	test_add_category_to_db_one_new_record	Test if one record has been added to the 'Category' table.	35
		test_add_category_to_db_already_exists	Test if the 'Category' table stays unchanged if the category is already recorded.	36
	get_categories_list	test_get_categories_list	Test if the categories list is well as expected.	37

Classe de test : tests.units_tests_off_sub.tests_models_store.StoreTestCase

Target object	Target method	Test method	Test purpose	Test id
Store	add_store_to_db	test_add_store_to_db_one_new_record	Test if one record has been added to the 'Store' table.	38
		test_add_store_to_db_already_exists	Test if the 'Store' table stays unchanged if the store is already recorded.	39

Classe de test : tests.units_tests_off_sub.tests_models_product.ProductTestCase

Target object	Target method	Test method	Test purpose	Test id
Product	add_product_category_to_db	test_add_product_category_to_db_one_new_record	Test if one record has been added to the 'ProductCategory' table (off_sub_product_categories).	40
		test_add_product_category_to_db_already_exists	Test if the 'ProductCategory' table (off_sub_product_categories) stays unchanged if the link between product and category is already recorded.	41
	add_product_store_to_db	test_add_product_store_to_db_one_new_record	Test if one record has been added to the 'ProductStore' table (off_sub_product_stores).	42
		test_add_product_store_to_db_already_exists	Test if the 'ProductStore' table (off_sub_product_stores) stays unchanged if the link between product and store is already recorded.	43
	add_product_to_db	test_add_product_to_db_one_new_record	Test if one record has been added to the 'Product' table (off_sub_product).	44
		test_add_product_to_db_product_already_exists	Test if the 'Product' table (off_sub_product) stays unchanged if the product is already recorded.	45
		test_add_product_to_db_same_code	Test if the 'Product' table (off_sub_product) stays unchanged if the (fake) product has the same code as another one already recorded.	46
	get_best_subs	test_get_best_subs_with_good_product	Test if the substitution program returns the best products. In this test case, the selected product SHOULD BE returned.	47
		test_get_best_subs_without_good_product	Test if the substitution program returns the best products. In this test case, the selected product SHOULD NOT BE returned.	48

1.1.6. Application « off_sub » — tests du module « views »

Classe de test : tests.units_tests_off_sub.tests_views.FavoritesPageTestCase

Target function	Test method	Test purpose	Test id
favorites	test_favorites_page_returns_200_with_an_authenticated_user	Test that favorites page returns a 200 code if a user is authenticated.	49
	test_favorites_page_returns_302_with_no_user	Test that favorites page returns a 302 code if no user is logged in.	50

Classe de test : tests.units_tests_off_sub.tests_views.FoodPageTestCase

Target function	Test method	Test purpose	Test id
food	test_food_page_returns_200	Test that food page returns a 200 code.	51
	test_food_page_returns_404	Test that food page returns a 404 code if the item does not exist.	52

Classe de test : tests.units_tests_off_sub.tests_views.IndexPageTestCase

Target function	Test method	Test purpose	Test id
index	test_index_page_returns_200	Test that index page returns a 200 code.	53

Classe de test : tests.units_tests_off_sub.tests_views.LegalPageTestCase

Target function	Test method	Test purpose	Test id
legal	test_legal_page_returns_200	Test that legal page returns a 200 code.	54

Classe de test : tests.units_tests_off_sub.tests_views.ResultsPageTestCase

Target function	Test method	Test purpose	Test id
results	test_results_page_returns_200	Test that results page returns a 200 code.	55
	test_food_page_returns_404	Test that results page returns a 404 code if the item does not exist.	56

1.2. Tests fonctionnels

Les tests fonctionnels de ce projet s'appuient principalement sur les User Stories suivantes :
US01 - En tant que Lily, je veux accéder à la page d'accueil.
US02 - En tant que Lily, je veux chercher un produit pour qu'on me propose des substituts.
US03 - En tant que Lily, je veux sélectionner un produit pour obtenir des informations détaillées.
US04 - En tant que Lily, je veux créer un compte utilisateur.
US05 - En tant que Lily, je veux me connecter avec mon compte utilisateur.
US06 - En tant que Lily, je veux me déconnecter de mon compte utilisateur.
US07 - En tant que Lily, je veux sélectionner un substitut pour le sauvegarder en favori.
US08 - En tant que Lily, je veux accéder à la liste de mes favoris.
US09 - En tant que Lily, je veux sélectionner un produit enregistré pour le retirer de ma liste de favoris.
US10 - En tant que Lily, je veux accéder à mon compte utilisateur.
US11 - En tant que Lily, je veux consulter les mentions légales du site.
US12 - En tant que Lily, je veux me connecter à mon compte depuis la page de résultats, sans avoir à rechercher les mêmes résultats.

Classe de test : tests.functional_tests.tests_functional.TestWithAnonymousUserTestCase

Test method	Test purpose	Test id
test_display_account	Try to display account page, redirect to sign page.	57
test_display_favorites	Try to display favorites page, redirect to sign page.	58
test_display_food	Display food page.	59
test_display_index	Display index (home) page.	60
test_display_legal	Display legal page.	61
test_display_log_out	Try to display log_out page, redirect to sign page.	62
test_display_results	Display results page.	63
test_display_sign	Display sign page.	64
test_look_for_a_product_from_home_with_masthead_form_enter	Test for User Story US02: scenario #1.	65
test_look_for_a_product_from_home_with_masthead_form_click_btn	Test for User Story US02: scenario #2.	66
test_look_for_a_product_from_home_navbar_form_enter	Test for User Story US02: scenario #3.	67

Test method	Test purpose	Test id
test_look_for_a_product_from_home_navbar_form_click_btn	Test for User Story US02: scenario #4. Available on XS, S and M screens only (L and XL screens excluded).	68
test_click_on_product_picture_to_consult_details_results_page	Test for User Story US03: scenario #1. Click on the searched product, i.e. in the masthead section.	69
test_click_on_product_name_to_consult_details_results_page	Test for User Story US03: scenario #2. Click on the searched product, i.e. in the masthead section.	70
test_create_user_account_success	Test for User Story US04: scenario #1.	71
test_create_user_account_failure_already_used_email	Test for User Story US04: scenario #2.	72
test_create_user_account_failure_two_different_passwords	Test for User Story US04: scenario #3.	73
test_login_success	Test for User Story US05: scenario #1.	74
test_login_failure_wrong_email	Test for User Story US05: scenario #2.	75
test_login_failure_wrong_password	Test for User Story US05: scenario #3.	76
test_sign_from_results_page_without_losing_results	Test for User Story US12: unique scenario.	77

Classe de test : tests.functional_tests.tests_functional.TestWithAuthenticatedUserTestCase

Test method	Test purpose	Test id
test_display_account	Display account page.	78
test_display_favorites	Display favorites page.	79
test_display_food	Display food page.	80
test_display_index	Display index (home) page.	81
test_display_legal	Display legal page.	82
test_display_log_out	Display logout page.	83
test_display_results	Display results page.	84
test_display_sign	Display sign page.	85
test_logout_success	Test for User Story US06: unique scenario.	86

Test method	Test purpose	Test id
test_save_a_product_as_favorite	Test for User Story US07: scenario #1. Standard process: 1. click on the button to save the product 2. check that the button state switches 3. check on the favorites page that the product is well saved	87
test_save_a_product_as_favorite_immediate_reverse	Test for User Story US07: scenario #2. Alternative process: 1. click on the button to save the product 2. check that the button state switches 3. reverse clicking to cancel product saving 4. check that the button state switches back to original 5. check on the favorites page that the product is NOT saved	88
test_go_to_favorites_page	Test for User Story US08: unique scenario.	89
test_unsave_product_from_favorites	Test for User Story US09: scenario #1. Standard process: 1. click on the button to unsave the product 2. check that the button state switches 3. refresh the page to check that the product "disappeared"	90
test_unsave_product_from_favorites_immediate_reverse	Test for User Story US09: scenario #2. Alternative process: 1. click on the button to unsave the product 2. check that the button state switches 3. reverse clicking to cancel product unsaving (keep saved) 4. check that the button state switches back to original 5. refresh the page to check that the product is still here	91
test_go_to_account_page	Test for User Story US10: unique scenario.	92
test_go_to_legal_page	Test for User Story US11: unique scenario.	93

1.3. Tests d'intégration

Classe de test : tests.integration_tests.tests_integration.UnitTestCase

Test method	Test purpose	Test id
test_remove_common_favorite_no_change_on_other_user	A product is registered as favorite for two different users. Test if removal of this favorite for one user has no impact on other user's favorites.	94
test_add_a_favorite_no_change_on_other_user	Test if adding a favorite to one user has no impact on other user's favorites.	95

Classe de test : tests.integration_tests.tests_integration.TestWithAnonymousUserTestCase

Test method	Test purpose	Test id
test_look_for_a_product_from_navbar_in_sign_template	Test if searching a product is also possible from 'sign' page.	96

Classe de test : tests.integration_tests.tests_integration.TestWithAuthenticatedUserTestCase

Test method	Test purpose	Test id
test_navbar_click_from_logout_to_sign	Test navigating from 'logout' page to 'sign' page.	97
test_navbar_click_from_account_to_favorites	Test navigating from 'account' page to 'favorites' page.	98
test_navbar_click_from_account_to_logout	Test navigating from 'account' page to 'logout' page.	99
test_navbar_click_from_favorites_to_account	Test navigating from 'favorites' page to 'account' page.	100
test_navbar_click_from_favorites_to_logout	Test navigating from 'favorites' page to 'logout' page.	101
test_look_for_a_product_from_navbar_in_account_template	Test if searching a product is also possible from 'account' page.	102
test_look_for_a_product_from_navbar_in_favorites_template	Test if searching a product is also possible from 'favorites' page.	103
test_look_for_a_product_from_navbar_in_logout_template	Test if searching a product is also possible from 'logout' page.	104

2. Exécution des tests

Avec ce découpage en trois catégories, on recense 113 tests :

- 65 tests unitaires, répartis ainsi :
  - 7 pour les modèles l'application « auth »
  - 3 pour les processeurs de contexte de l'application « off_sub »
  - 22 pour la commande personnalisée d'initialisation de la base de données (application « off_sub »)
  - 9 pour la commande personnalisée de mise à jour de la base de données (application « off_sub »)
  - 16 pour les modèles de l'application « off_sub »
  - 8 pour les vues de l'application « off_sub »
- 37 tests fonctionnels, répartis en deux classes :
  - 21 pour les fonctionnalités sans utilisateur initialement connecté
  - 16 pour les fonctionnalités avec un utilisateur initialement connecté

  11 tests d'intégration, répartis en trois classes :
  - 2 tests unitaires
  - 5 pour les interactions entre fonctionnalités sans utilisateur initialement connecté
  - 4 pour les interactions entre fonctionnalités avec un utilisateur initialement connecté

En exécutant la commande suivante à la racine du projet

```
python -m manage test
```

on obtient le résultat suivant :

```
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
............Initialisation de la base de données, veuillez patienter SVP...
Initialisation terminée !
.....................Mise à jour de la base de données, veuillez patienter SVP...
Tâche terminée : 1 produit(s) mis à jour !
.....................................................................................
----------------------------------------------------------------------
Ran 113 tests in 110.790s

OK
Destroying test database for alias 'default'...
```

Le message **en jaune** s'affiche au début du chargement (commande *db_init*) ou la mise à jour (commande *db_update*) de la base de données.
Le message **en vert** s'affiche à la fin du chargement ou de la mise à jour de la base de données. Ici, il s'agit de la base de données provisoire utilisée pour les tests par Django.

Résultat de l'exécution des tests :
L'ensemble des tests passe avec succès.

3. Rapport de l'exécution des tests

On utilise le module *coverage* pour vérifier la couverture de tests du projet. Afin d'obtenir un résultat consistant, certains répertoires importés depuis des libraires externes sont exclues du rapport. Pour cela, on utilise le fichier de configuration *.coveragerc* dont voici le contenu :

```
# .coveragerc to control coverage.py
[run]
command_line = "manage.py" test
source = .

[report]
# Regexes for lines to exclude from consideration
omit =
    tests*
    *migrations*
    runtests.py
    *wsgi*
    *asgi*
    *site-packages*

ignore_errors = True

[html]
directory = coverage_html_report
```

Ainsi, dans la console, l'exécution successive des commandes « coverage run », « coverage report » puis « coverage html » permet notamment d'obtenir le rapport suivant :

Coverage report: 92%

Module ↓	statements	missing	excluded	coverage
auth/__init__.py	1	0	0	100%
auth/admin.py	79	4	0	95%
auth/apps.py	4	0	0	100%
auth/forms.py	5	0	0	100%
auth/models.py	32	0	0	100%
auth/urls.py	4	0	0	100%
auth/views.py	43	0	0	100%
manage.py	12	2	0	83%
off_sub/__init__.py	0	0	0	100%
off_sub/ajax_views.py	31	1	0	97%
off_sub/apps.py	3	3	0	0%
off_sub/context_processors.py	9	0	0	100%
off_sub/management/commands/db_init.py	67	0	0	100%
off_sub/management/commands/db_update.py	25	2	0	92%
off_sub/models.py	100	14	0	86%
off_sub/urls.py	4	0	0	100%
off_sub/views.py	40	0	0	100%
pur_beurre/__init__.py	0	0	0	100%
pur_beurre/settings/__init__.py	23	0	0	100%
pur_beurre/settings/production.py	12	12	0	0%
pur_beurre/settings/travis.py	2	2	0	0%
pur_beurre/urls.py	4	0	0	100%
Total	**500**	**40**	**0**	**92%**

coverage.py v5.0.4, created at 2020-04-27 15:02

Conclusion : avec un résultat de **92%**, le taux de couverture de tests du projet est **satisfaisant**.