Nom du Test	Endpoint Testé	Méthode HTTP	Données d'Entrée	Résultat Attendu	Résultat Actuel (selon Postman et unitest)	Problèmes Rencontrés
Test Users						
test_01_create_users	/api/v1/users/	POST	{ "first_name": "...", "last_name": "...", "email": "..." }	Statut `201 CREATED`, la réponse contient un `id`.	Statut `201 CREATED`, Test OK	Oublie des try/except
test_02_create_user_duplicate_email	/api/v1/users/	POST	Email déjà existant.	Statut `400 BAD REQUEST`	Statut `400 BAD REQUEST`, Test OK	Oublie des try/except
test_03_get_user	/api/v1/users/{user_id}	GET	user_id du premier utilisateur créé.	Statut `200 OK`, `email` correspond à 'john.doe@gmail.com'.	Statut `200 OK`, Test OK	Aucun
test_04_get_user_not_found	/api/v1/users/0000...0000	GET	user_id inexistant.	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Aucun
test_05_get_all_users	/api/v1/users/	GET	Aucune	Statut `200 OK`, la réponse est un tableau, contient au moins 3 utilisateurs.	Statut `200 OK`, Test OK	Aucun
test_06_update_user	/api/v1/users/{user_id}	PUT	{ "first_name": "Janet", "last_name": "Doe", "email": "janet.doe@gmail.com" }	Statut `200 OK`, `first_name` et `email` sont mis à jour.	Statut `200 OK`, Test OK	Oublie des try/except
test_07_update_user_not_found	/api/v1/users/0000...0000	PUT	Données de mise à jour.	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Oublie des try/except
test_08_update_user_invalid_data	/api/v1/users/{user_id}	PUT	{ "first_name": "", "last_name": "", "email": "not-an-email" }	Statut `400 BAD REQUEST`	Statut `400 BAD REQUEST`, Test OK	Oublie des try/except
Test Amenities						
test_01_create_amenities	/api/v1/amenities/	POST	{ "name": "WiFi" }	Statut `201 CREATED`, la réponse contient un `id`.	Statut `201 CREATED`, Test OK	Oublie des try/except
test_02_create_amenity_invalid	/api/v1/amenities/	POST	{ "name": "" }	Statut `400 BAD REQUEST`	Statut `400 BAD REQUEST`, Test OK	Oublie des try/except
test_03_get_amenity	/api/v1/amenities/{amenity_id}	GET	amenity_id de la première commodité créée.	Statut `200 OK`, `name` correspond à 'WiFi'.	Statut `200 OK`, Test OK	Aucun
test_04_get_amenity_not_found	/api/v1/amenities/0000...0000	GET	amenity_id inexistant.	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Aucun
test_05_get_all_amenities	/api/v1/amenities/	GET	Aucune	Statut `200 OK`, la réponse est un tableau, contient au moins 3 commodités.	Statut `200 OK`, Test OK	Aucun
test_06_update_amenity	/api/v1/amenities/{amenity_id}	PUT	{ "name": "Climatisation" }	Statut `200 OK`, la réponse contient un message de confirmation.	Statut `200 OK`, Test OK	Oublie des try/except
test_07_update_amenity_not_found	/api/v1/amenities/0000...0000	PUT	{ "name": "Inexistante" }	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Oublie des try/except
test_08_update_amenity_invalid	/api/v1/amenities/{amenity_id}	PUT	{ "name": "" }	Statut `400 BAD REQUEST`	Statut `400 BAD REQUEST`, Test OK	Oublie des try/except

Nom du Test	Endpoint Testé	Méthode HTTP	Données d'Entrée	Résultat Attendu	Résultat Actuel (selon Postman et unitest)	Problèmes Rencontrés
Test Places						
test_01_create_places	/api/v1/places/	POST	{ "title": "...", "description": "...", "price": ..., "latitude": ..., "longitude": ..., "owner_id": "...", "amenities": [...] }	Statut `201 CREATED`, la réponse contient un `id`.	Statut `201 CREATED`, Test OK	Oublie des try/except
test_02_create_place_invalid	/api/v1/places/	POST	{ "title": "", "description": "", "price": -10, ... }	Statut `400 BAD REQUEST`	Statut `400 BAD REQUEST`, Test OK	Oublie des try/except
test_03_get_place	/api/v1/places/{place_id}	GET	place_id du premier lieu créé.	Statut `200 OK`, `title` correspond à "Chez Timi", contient `owner` et `amenities`.	Statut `200 OK`, Test OK	Aucun
test_04_get_place_not_found	/api/v1/places/0000...0000	GET	place_id inexistant.	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Aucun
test_05_get_all_places	/api/v1/places/	GET	Aucune	Statut `200 OK`, la réponse est un tableau, contient au moins 3 lieux.	Statut `200 OK`, Test OK	Aucun
test_06_update_place	/api/v1/places/{place_id}	PUT	{ "title": "Appartement rénové", "description": "Tout neuf et confortable", "price": 80.0 }	Statut `200 OK`, la réponse contient un message de confirmation.	Statut `200 OK`, Test OK	Oublie des try/except
test_07_update_place_not_found	/api/v1/places/0000...0000	PUT	Données de mise à jour.	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Oublie des try/except
test_08_update_place_invalid	/api/v1/places/{place_id}	PUT	{ "price": -100 }	Statut `400 BAD REQUEST`	Statut `400 BAD REQUEST`, Test OK	Oublie des try/except
Test Reviews						
test_01_create_reviews	/api/v1/reviews/	POST	{ "text": "...", "rating": ..., "user_id": "...", "place_id": "..." }	Statut `201 CREATED`, la réponse contient un `id`.	Statut `201 CREATED`, Test OK	Oublie des try/except
test_02_create_review_invalid	/api/v1/reviews/	POST	{ "text": "", "rating": 10, "user_id": "invalid", "place_id": "invalid" }	Statut `400 BAD REQUEST`	Statut `400 BAD REQUEST`, Test OK	Oublie des try/except
test_03_get_review	/api/v1/reviews/{review_id}	GET	review_id du premier avis créé.	Statut `200 OK`, `text` correspond à "Excellent!".	Statut `200 OK`, Test OK	Aucun
test_04_get_review_not_found	/api/v1/reviews/0000...0000	GET	review_id inexistant.	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Aucun
test_05_get_all_reviews	/api/v1/reviews/	GET	Aucune	Statut `200 OK`, la réponse est un tableau, contient au moins 3 avis.	Statut `200 OK`, Test OK	Aucun
test_06_update_review	/api/v1/reviews/{review_id}	PUT	{ "text": "Updated review", "rating": 4 }	Statut `200 OK`, la réponse contient un message de confirmation.	Statut `200 OK`, Test OK	Oublie des try/except
test_07_update_review_not_found	/api/v1/reviews/0000...0000	PUT	{ "text": "Ghost", "rating": 1 }	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Oublie des try/except
test_08_update_review_invalid	/api/v1/reviews/{review_id}	PUT	{ "text": "", "rating": 100 }	Statut `400 BAD REQUEST`	Statut `400 BAD REQUEST`, Test OK	Oublie des try/except
test_09_delete_review	/api/v1/reviews/{review_id}	DELETE	review_id du dernier avis créé.	Statut `200 OK`	Statut `200 OK`, Test OK	Aucun
test_10_delete_review_not_found	/api/v1/reviews/0000...0000	DELETE	review_id inexistant.	Statut `404 NOT FOUND`	Statut `404 NOT FOUND`, Test OK	Aucun
test_11_get_reviews_for_place	/api/v1/places/{place_id}/reviews	GET	place_id du lieu créé.	Statut `200 OK`, la réponse est un tableau, contient au moins 1 avis.	Statut `200 OK`, Test OK	Aucun