

Reviews: The Testing Process

#	Endpoint	Purpose	Input Data	Expected Output	Actual Output	Result
1	POST /api/v1/reviews/	Create a new review	{ "user_id": "u1", "place_id": "p1", "rating": 5, "comment": "Excellent stay!", "upload_image": ["https://img.com/1.jpg"] }	Status: 201 Response: Review fields match input	Status: 201 Response matches expected	Passed
2	POST /api/v1/reviews/	Validate rating below 1	{ "user_id": "u1", "place_id": "p1", "rating": 0, "comment": "Too low" }	Status: 400 "Rating must be between 1 and 5"	Status: 400 Error message returned	Passed
3	POST /api/v1/reviews/	Validate rating above 5	{ "user_id": "u1", "place_id": "p1", "rating": 6, "comment": "Too high" }	Status: 400 "Rating must be between 1 and 5"	Status: 400 Error message returned	Passed
4	POST /api/v1/reviews/	Validate rating is not a number	{ "user_id": "u1", "place_id": "p1", "rating": "five", "comment": "String rating" }	Status: 400 "Rating must be a number"	Status: 400 Error message returned	Passed
5	POST /api/v1/reviews/	Validate comment too long (>300 chars)	{ "user_id": "u1", "place_id": "p1", "rating": 4, "comment": "a"*301 }	Status: 400 "Comment cannot exceed 300 characters"	Status: 400 Error message returned	Passed
6	POST /api/v1/reviews/	Validate invalid upload_image type	{ "user_id": "u1", "place_id": "p1", "rating": 4, "comment": "Nice!", "upload_image": 123 }	Status: 400 "upload_image must be a list of image URLs or files"	Status: 400 Error message returned	Passed
7	POST /api/v1/reviews/	Validate invalid image content	{ "user_id": "u1", "place_id": "p1", "rating": 4, "comment": "Nice", "upload_image": [("fake.jpg", b"notanimage")] }	Status: 400 "Invalid image data or unsupported format"	Status: 400 Error message returned	Passed
8	POST /api/v1/reviews/	Validate user not found	{ "user_id": "invalid", "place_id": "p1", "rating": 4, "comment": "Test" }	Status: 400 "User not found"	Status: 400 Error message returned	Passed
9	POST	Validate	{ "user_id": "u1",	Status: 400	Status: 400	Passed

	GET <code>/api/v1/reviews/</code>	place not found	<code>"place_id": "invalid", "rating": 4, "comment": "Test" }</code>	"Place not found"	Error message returned	
10	GET <code>/api/v1/reviews/<review_id></code>	Retrieve review by ID	Valid review_id from created review	Status: 200 Returns review JSON	Status: 200 Response matches expected	Passed
11	GET <code>/api/v1/reviews/<invalid_id></code>	Handle invalid review ID	<code>"invalid_id"</code>	Status: 404 "Review not found"	Status: 404 Error returned	Passed
12	PUT <code>/api/v1/reviews/<review_id></code>	Update review details (by author)	<code>{ "rating": 3, "comment": "Updated comment", "current_user_id": "u1" }</code>	Status: 200 Updated fields match input	Status: 200 Response matches expected	Passed
13	PUT <code>/api/v1/reviews/<review_id></code>	Update review with invalid rating	<code>{ "rating": 6, "current_user_id": "u1" }</code>	Status: 400 "Rating must be between 1 and 5"	Status: 400 Error message returned	Passed
14	PUT <code>/api/v1/reviews/<review_id></code>	Update review by non-author	<code>{ "rating": 4, "current_user_id": "u2" }</code>	Status: 403 "You are not allowed to update this review"	Status: 403 Error returned	Passed
15	DELETE <code>/api/v1/reviews/<review_id></code>	Delete a review by ID	Valid review_id	Status: 200 "Review deleted successfully"	Status: 200 Message returned	Passed
16	DELETE <code>/api/v1/reviews/<invalid_id></code>	Handle delete with invalid ID	<code>"invalid_id"</code>	Status: 404 "Review not found"	Status: 404 Error returned	Passed
17	GET <code>/api/v1/reviews?place_id=p1</code>	List all reviews for a place	Valid place_id with reviews	Status: 200 Array of review objects	Status: 200 Matches expected	Passed
18	GET <code>/api/v1/reviews?place_id=invalid</code>	No reviews found for place	Invalid place_id	Status: 404 "No reviews found"	Status: 404 Message returned	Passed
19	GET <code>/api/v1/reviews/place/<place_id></code>	List all reviews for a specific place	Valid place_id	Status: 200 List of reviews	Status: 200 Matches expected	Passed
20	GET <code>/api/v1/reviews/user/<user_id></code>	List all reviews by a user	Valid user_id	Status: 200 List of reviews	Status: 200 Matches expected	Passed
21	GET <code>/api/v1/reviews/user/<invalid_user_id></code>	Handle user with no reviews	Invalid or inactive user_id	Status: 404 "No reviews found for user_id 'invalid_user'"	Status: 404 Message returned	Passed
22	GET	Compute	place_id with multiple	Status: 200	Status: 200	Passed

	/api/v1/reviews/place/<place_id>/average (optional)	average rating for a place	reviews	{ "average_rating" : <float> }	Value computed correctly	
--	--	----------------------------	---------	--------------------------------------	--------------------------	--