

#	Test Name / Scenario	Expected Behavior	Actual Behavior	Recommended Fix / Action
1	POST /auth - login with valid credentials	200 OK, returns token	200 OK, returns UUID token	✅ No fix needed, ensure token format & expiration validated
2	GET /booking - retrieve all bookings	200 OK, returns JSON array	200 OK, returns JSON array	✅ OK, consider pagination for large datasets
3	POST /auth - fail with wrong username	401 Unauthorized, reason = "Bad credentials"	200 OK, reason = "Bad credentials"	Change status code to <b>401 Unauthorized</b>
4	GET /booking/:id - retrieve specific booking	200 OK, returns booking object	200 OK, returns booking object	✅ OK
5	POST /auth - fail with wrong password	401 Unauthorized	200 OK, reason = "Bad credentials"	Change status code to <b>401 Unauthorized</b>
6	POST /auth - fail with empty username	400 Bad Request	200 OK, reason = "Bad credentials"	Change status code to <b>400</b> , include validation message
7	GET /booking/:id - non-existent booking	404 Not Found	404 Not Found	✅ OK
8	POST /auth - fail with empty password	400 Bad Request	200 OK, reason = "Bad credentials"	Change status code to <b>400</b> , include validation message
9	POST /auth - fail with missing username field	400 Bad Request	200 OK, reason = "Bad credentials"	Change status code to <b>400</b> , include validation message

10	POST /booking - create booking without authentication	401 Unauthorized	200 OK, reason = "Bad credentials"	Change status code to <b>401</b>
11	POST /auth - fail with missing password field	400 Bad Request	200 OK, reason = "Bad credentials"	Change status code to <b>400</b> , include validation message
12	POST /booking - create booking with valid data	201 Created, returns booking object	200 OK, returns booking object	Change status code to <b>201 Created</b>
13	POST /booking - fail with invalid booking data	400 Bad Request	400 Bad Request	✅ OK, ensure error details are complete
14	POST /booking - request with extra fields	201 Created, ignores extra fields	200 OK, returns token (unexpected)	Fix handling of extra fields, return proper booking response
15	POST /booking - missing required fields	400 Bad Request	200 OK, reason = "Bad credentials"	Change status code to <b>400</b> , include validation details
16	GET /booking - correct Content-Type	Content-Type: application/json	application/json; charset=utf-8	✅ OK
17	PUT /booking/:id - update booking with authentication	200 OK, booking updated	200 OK	✅ OK
18	PUT /booking/:id - reasonable response time	< 2s	119ms	✅ OK
19	PUT /booking/:id - fail without authentication	401 Unauthorized	403 Forbidden	✅ OK (403 acceptable), document in API
20	GET /auth - token format UUID	Valid UUID	Valid UUID	✅ OK


21	GET /booking/:id - consistent response structure	JSON with consistent keys	JSON consistent	✓ OK
22	PUT /booking/:id - fail with invalid token	401/403 Unauthorized	200 OK	Fix to return <b>401/403</b> for invalid token
23	GET /ping - health endpoint	200 OK, status = "OK"	201 Created, body = "Created"	Change status code to <b>200</b> , optional: return JSON {"status": "OK"}
24	GET /ping - response time	< 1s	9ms	✓ OK
25	GET /ping - correct headers	Content-Type: application/json	text/html; charset=utf-8	Change Content-Type to <b>application/json</b>
26	GET /ping - accessible without authentication	200 OK	201 Created	Change status code to <b>200 OK</b>
27	GET /ping - consistent response	Same body across calls	✓ Consistent	✓ OK
28	PATCH /booking/:id - partial update with auth	200 OK, booking updated	200 OK	✓ OK
29	DELETE /booking/:id - with authentication	200/204, booking deleted	201 Created	Change status code to <b>200/204</b> , ensure deletion happens
30	DELETE /booking/:id - fail without authentication	401/403 Unauthorized	403 Forbidden	✓ OK
31	DELETE /booking/:id - non-existent booking	404 Not Found	404 Not Found	✓ OK

32	POST /booking - invalid checkin/check out dates	400 Bad Request	200 OK, booking created	Implement date validation, return <b>400 Bad Request</b>
33	Complete booking lifecycle	All endpoints respond correctly	Delete skipped, other operations OK	Implement delete endpoint fully; standardize response codes

---

## Summary for Developer

### 1. Auth API

- a. Failed login & missing fields currently return 200 → change to 400/401.
- b. Token format validated .

### 2. Booking API

- a. Invalid token, unauthorized, missing fields → fix status codes.
- b. Invalid dates allowed → reject with 400.
- c. DELETE endpoint partially implemented → standardize 200/204 response.

### 3. Health API

- a. Status 201 → change to 200.
- b. Content-Type: return application/json if clients expect JSON.

### 4. General Recommendations

- a. Standardize all error responses with { error: "", details: [] } or { reason: "" }.
- b. Ensure response codes follow **REST best practices**.
- c. Validate all inputs server-side, including data types and logical checks (e.g., dates, totals).