# parks - Bug #2583

## Unit test

09/03/2026 02:50 AM - admin admin

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 09/03/2026 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Moin Ashraf | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | **Spent time:** | 0.00 hour |

**Description**

/** * PROJECT BUNDINIPARKS - API INTEGRATION UNIT TESTS * Endpoint: POST /v1/events/log-matrix * Target Node: api.bundini.co.uk
*/

```
const chai = require('chai');
const chaiHttp = require('chai-http');
const server = require('../server');
const db = require('../config/database');
const expect = chai.expect;

chai.use(chaiHttp);

describe('Cross-Subdomain Event Matrix Integration Pipeline', () => {

// Clear test mutations from database before running suite
    beforeEach(async () => {
        await db.query("DELETE FROM bed_action_logs WHERE notes LIKE '%TEST_RUN%'");
        await db.query("DELETE FROM operational_cycles WHERE log_details LIKE '%TEST_RUN%'");
    });

// TEST CASE 1: Successful Atomic Commit
    it('SUCCESS: Should process valid payload, execute commit, and trigger governance log entry',
(done) => {
        const validPayload = {
            "event_id": 1,
            "bed_id": 2,    // Maryan Park Bed B2
            "general_actions": ["Weeded", "Watered"],
            "plant_matrix": [
                {
                    "plant_id": 5,
                    "actions": ["Pruning"],
                    "comment": "TEST_RUN: Trimmed upper branches."
                }
            ]
        };

chai.request(server)
            .post('/v1/events/log-matrix')
            .send(validPayload)
            .end((err, res) => {
                expect(res).to.have.status(201);
                expect(res.body).to.be.an('object');
                expect(res.body.success).to.equal(true);
                expect(res.body.message).to.contain('successfully synchronized');
                done();
            });
    });

// TEST CASE 2: Validation Failure and Complete Rollback
    it('ROLLBACK FAILURE: Should abort transaction and register no records if identifiers are miss
ing', (done) => {
        const invalidPayload = {
```

```
            "event_id": null, // Triggers database constraint failure
            "bed_id": 2,
            "general_actions": ["Watered"],
            "plant_matrix": []
        };

chai.request(server)
            .post('/v1/events/log-matrix')
            .send(invalidPayload)
            .end((err, res) => {
                expect(res).to.have.status(400);
                expect(res.body).to.be.an('object');
                expect(res.body).to.have.property('error');

// Confirm atomic integrity: Verification query must return 0 rows
                db.query("SELECT COUNT(*) as count FROM bed_action_logs WHERE notes LIKE '%TEST_RU
N%'", (dbErr, rows) => {
                    expect(rows[0].count).to.equal(0);
                    done();
                });
            });
    });
});
```