

REST API

09/03/2026 02:45 AM - admin admin

Status:	Closed	Start date:	09/03/2026
Priority:	Normal	Due date:	
Assignee:	salman ifrahim	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
Description // Express.js Route Handling Cross-Subdomain Form Submissions // Target: api.bundini.co.uk/v1/events/log-matrix const express = require('express'); const router = express.Router(); const db = require('../config/database'); // Central Shared MySQL Connection Pool router.post('/log-matrix', async (req, res) => { const { event_id, bed_id, general_actions, plant_matrix } = req.body; // Validate required identifiers if (!event_id !bed_id) { return res.status(400).json({ error: 'Missing core tracking identifiers.' }); } const connection = await db.getConnection(); try { await connection.beginTransaction(); // 1. Process General Bed Infrastructure Action Logs (e.g., Weeded, Watered) if (general_actions && general_actions.length > 0) { const generalLogQueries = general_actions.map(action => { return connection.query(INSERT INTO bed_action_logs (event_id, bed_id, plant_id, action_type, notes) VALUES (?, ?, NULL, ?, 'Logged via public web interface')`, [event_id, bed_id, action]); }); await Promise.all(generalLogQueries); } // 2. Process Individual Plant Lifecycle Checklist Matrix if (plant_matrix && plant_matrix.length > 0) { const matrixQueries = plant_matrix.map(item => { // item contains: { plant_id: 12, action: 'Pruning', comment: 'Removed dead leaves ' } return connection.query(INSERT INTO bed_action_logs (event_id, bed_id, plant_id, action_type, notes) VALUES (?, ?, ?, ?, ?)`, [event_id, bed_id, item.plant_id, item.action, item.comment null]); }); await Promise.all(matrixQueries); } // 3. Automatically Trigger an Internal Governance Cycle Update (recreation subdomain sync) await connection.query(INSERT INTO operational_cycles (bed_id, manager_id, stage, log_details) VALUES (?, 1, 'Do', ?)`,</div><div>[bed_id, `Automated: Public activity matrix processed for Event ID \${event_id}. Subdom ain sync successful.`]);			

```
await connection.commit();
    res.status(201).json({ success: true, message: 'Data successfully synchronized across subdomains.' });

} catch (error) {
    await connection.rollback();
    console.error('Cross-subdomain synchronization error:', error);
    res.status(500).json({ error: 'Database transaction failed during synchronization pipeline.' });
} finally {
    connection.release();
}
});

module.exports = router;
```

History

#1 - 09/03/2026 05:43 PM - admin admin

- Status changed from New to Closed