

Mr-MoneyBags v1.x — Administrator Guide **v1.x**

Document version: 1.x | **Last updated:** August 2025

Audience: System / DevOps administrators responsible for installing, configuring and operating Mr-MoneyBags in production or staging environments.

1 Introduction & Overview

Mr-MoneyBags v1.x is a full-featured, open-source fund-accounting platform for non-profit organisations.

It delivers:

- Multi-entity consolidation
- Double-entry journal engine
- Automated NACHA / ACH vendor payments
- Rich reporting & interactive dashboards

This guide explains **how to install, secure, operate and maintain** the application on an Ubuntu 22.04 LTS (or later) server.

2 System Requirements

Category	Minimum	Recommended
CPU	1 vCPU	2+ vCPU
RAM	2 GiB	4 GiB
Disk	10 GB SSD	30 GB SSD / NVMe
OS	Ubuntu 22.04 LTS (Server)	Ubuntu 24.04 LTS
Node.js	18 LTS	20 LTS
PostgreSQL	14	16
Git	latest from apt	–
PM2	5.x (global)	–
Nginx	1.22+	–

macOS, Debian or RHEL family distros work equally well; adjust package commands as needed.

3 Installation & Setup

3.1 Create Service Account & Directories

```
sudo adduser --system --group fundapp
sudo mkdir -p /opt/mr-moneybags
sudo chown -R fundapp:fundapp /opt/mr-moneybags
```

3.2 Install Prerequisites

```
# Node.js (18 LTS)
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs git nginx postgresql postgresql-contrib

# Process manager
sudo npm i -g pm2
```

3.3 Clone Repository

```
sudo -iu fundapp
cd /opt/mr-moneybags
git clone https://github.com/tpfbill/mr-moneybags-v1.x.git .
```

3.4 Install Dependencies

```
npm ci          # exact, lock-file versions
```

3.5 Database Setup

Run **once** as postgres superuser:

```
sudo -iu postgres

# 1. Create role & DB
psql -c "CREATE ROLE npfadmin WITH LOGIN PASSWORD 'changeme';"
psql -c "CREATE DATABASE fund_accounting_db OWNER npfadmin;"

# 2. Load consolidated schema
psql -U npfadmin -d fund_accounting_db -f /opt/mr-moneybags/database/db-init.sql

# 3. (Optional) Load sample entities, funds, vendors & NACHA data
psql -U npfadmin -d fund_accounting_db -f /opt/mr-moneybags/database/insert-complete-nacha-data.sql
```

3.6 Environment Configuration

```
cp .env.example .env
nano .env          # update PGUSER, PGPASSWORD, etc.
```

Essential vars:

```
PGDATABASE=fund_accounting_db
PGUSER=npfadmin
PGPASSWORD=changeme
PORT=3000
```

3.7 First Start

```
# API (port 3000)
pm2 start "npm start" --name fund-api

# Static UI (port 8080)
pm2 start "npx http-server . -p 8080 --no-cache" --name fund-ui

pm2 save
```

Access:

- API health: <http://localhost:3000/api/health>
- UI: <http://localhost:8080/index.html>

4 Application Architecture

4.1 Backend

- **Express 5 + Node.js 18**
- 13 modular route files in `src/routes/*`
- Single entry - `server-modular.js` (\approx 114 LOC)

4.2 Frontend

- Plain HTML/CSS/JS in `index.html` & `/src/js`
- Charts via Chart.js
- SPA-style navigation handled by `src/js/app.js`

4.3 Database

- **PostgreSQL (15 tables)** – see `database/db-init.sql`
- Primary keys are UUIDv4
- All monetary fields are `NUMERIC(14,2)`

4.4 Removed Docker

Docker artefacts were purged in v1.x; all deployment assumes native OS packages.

5 Database Administration

5.1 Recommended Settings

Edit `/etc/postgresql/14/main/postgresql.conf`:

```
shared_buffers = 512MB
work_mem       = 8MB
wal_level      = replica
```

5.2 Back-up

```
pg_dump -U npfadmin -Fc fund_accounting_db \
> /backups/$(date +%F)_fund_accounting_db.dump
```

5.3 Restore

```
pg_restore -U npfadmin -d fund_accounting_db \
/backups/2025-08-06_fund_accounting_db.dump
```

5.4 Schema Updates

When `database/db-init.sql` changes:

```
git pull
psql -U npfadmin -d fund_accounting_db -f database/db-init.sql
```

6 User Management

Action	Location
Add / Edit / Disable	Settings → Users
Roles	Administrator / Accountant / Viewer
Password rules	Min 12 chars, 1 uppercase, 1 digit

Forgotten password = reset link emailed (SMTP settings in `.env`).

7 Entity & Fund Management

- **Settings** → **Entities** – Build org hierarchy
 - Exactly **one** root entity marked *Consolidated*.
 - **Funds** are created under each entity.
 - **Inter-Entity Transfer Wizard** (Utilities) automates due-to/due-from journals.
-

8 Security Considerations

1. Database

- Use strong password for `npfadmin`
- Restrict `pg_hba.conf` to local / trusted subnets

2. Application

- Set `NODE_ENV=production`
- Use HTTPS only (force via Nginx)

3. Network

- Enable `ufw` ; allow 22, 80, 443 only
 - Fail2Ban for SSH
-

9 Production Deployment

9.1 PM2 Service Autostart

```
pm2 startup systemd -u fundapp --hp /home/fundapp
pm2 save
```

9.2 Nginx Reverse Proxy

/etc/nginx/sites-available/mrmoneybags.conf:

```
server {
    listen 80;
    server_name accounting.example.org;

    location /api/ {
        proxy_pass http://127.0.0.1:3000/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location / {
        proxy_pass http://127.0.0.1:8080/;
        try_files $uri $uri/ /index.html;
    }
}
```

Enable & reload:

```
sudo ln -s /etc/nginx/sites-available/mrmoneybags.conf /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl reload nginx
```

9.3 SSL/TLS (Let's Encrypt)

```
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d accounting.example.org
```

9.4 Monitoring & Logging

- `pm2 monit` – live stats
 - `pm2-logrotate` – auto-rotation
 - `journalctl -u nginx -f` – web logs
-

10 Maintenance & Troubleshooting

Task	Frequency	Command
OS security updates	weekly	<code>sudo unattended-upgrade -d</code>
DB vacuum & analyse	weekly	<code>VACUUM (VERBOSE, ANALYZE);</code>
Log review	daily	<code>pm2 logs --lines 100</code>
Rebuild indexes	quarterly	<code>REINDEX DATABASE fund_accounting_db;</code>

Common Issues

Symptom	Resolution
<code>DB Offline</code> badge	Check <code>systemctl status postgresql</code>
502 via Nginx	API process down → <code>pm2 restart fund-api</code>
<code>psql: FATAL password authentication failed</code>	Verify <code>.env</code> matches PG user

11 Backup & Recovery

1. Hot backup (daily cron)

```
pg_dump -U npfadmin -Fc fund_accounting_db > /backups/db_$(date +%F).dump
```

2. File backup – tar /opt/mr-moneybags (code + uploads)

3. Off-site sync – rclone to S3 / Backblaze

4. Disaster Recovery Test (quarterly)

Spin up fresh VM, restore dump, checkout same git tag, point `.env` to restored DB, verify UI.

Appendix A — Quick Commands

```
# Pull latest release
cd /opt/mr-moneybags
git fetch --tags
git checkout v1.0.1
npm ci && pm2 restart fund-api fund-ui

# Check API health
curl -s http://localhost:3000/api/health | jq

# Create read-only DB user
psql -U postgres -d fund_accounting_db -c \
  "CREATE ROLE reporter LOGIN PASSWORD 'xxxx'; GRANT SELECT ON ALL TABLES IN
  SCHEMA public TO reporter;"
```
