

Mr. MoneyBags v1.x

macOS Clean Installation Guide

Version 1.x | August 2025

Table of Contents

1. Introduction
2. Prerequisites
3. Quick Start
4. Step-by-Step Installation
 - 4.1. Clone the Repository
 - 4.2. Install Dependencies
 - 4.3. Bootstrap Database
 - 4.4. Start the Application
5. Verify Installation
6. Troubleshooting
7. Appendix: Common Commands

1. Introduction

This guide provides step-by-step instructions for installing Mr. MoneyBags v1.x on macOS, including setup of The Principle Foundation (TPF) sample data. Mr. MoneyBags is a comprehensive fund accounting system designed for nonprofit organizations.

2. Prerequisites

Before beginning the installation, ensure your system meets the following requirements:

Requirement	Minimum Version	Notes
Operating System	macOS 12+	Monterey or newer recommended
Homebrew	Latest	Package manager for macOS
Node.js	20.x or newer	LTS version recommended
Git	Latest	For repository cloning
PostgreSQL	15.x or newer	Will be installed by bootstrap script if not present
Google Chrome	Latest	Optional, for PDF generation

If you don't have these prerequisites installed, here are the commands to install them:

Install Homebrew:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Install Node.js:

```
brew install node@20
echo 'export PATH="/usr/local/opt/node@20/bin:$PATH"' >> ~/.zshrc
source ~/.zshrc
```

Install Git:

```
brew install git
```

3. Quick Start

For experienced users who want to get up and running quickly, here's the complete installation process in a single block:

```
# Clone the repository
git clone https://github.com/tpfbill/mr-moneybags-v1.x.git
cd mr-moneybags-v1.x
```

```
# Install dependencies
npm install

# Bootstrap the database (PostgreSQL will be installed if needed)
npm run bootstrap:mac

# Start the application (API server and frontend)
npm run dev
```

After running these commands, access the application at <http://localhost:8080> and log in with:

- **Username:** admin
- **Password:** admin

Note: If you encounter issues with the TPF data loader, you can set `LOAD_TPF_DATA=false` before running the bootstrap command:

```
LOAD_TPF_DATA=false npm run bootstrap:mac
```

This is usually not necessary as the `db-init.sql` script already includes TPF sample data.

4. Step-by-Step Installation

This section provides detailed step-by-step instructions for installing Mr. MoneyBags v1.x.

4.1. Clone the Repository

First, clone the repository from GitHub:

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

4.2. Install Dependencies

Install the required Node.js dependencies:

```
npm install
```

This will install all dependencies defined in the `package.json` file, including:

- Express.js (API server)
- PostgreSQL client libraries
- HTTP server for frontend
- Other required packages

4.3. Bootstrap Database

The bootstrap script will:

- Install PostgreSQL via Homebrew if not already installed
- Create the necessary database user and permissions
- Create and initialize the database with schema and sample data
- Configure environment variables

Run the bootstrap script:

```
npm run bootstrap:mac
```

If you encounter issues with the TPF data loader, you can disable it:

```
LOAD_TPF_DATA=false npm run bootstrap:mac
```

Note: The bootstrap script uses the following default database settings:

- Host: localhost
- Port: 5432
- User: postgres
- Password: npfa123
- Database: fund_accounting_db

These can be overridden by setting environment variables (PGHOST, PGPORT, PGUSER, PGPASSWORD, PGDATABASE) before running the script.

4.4. Start the Application

Start both the API server and frontend development server with a single command:

```
npm run dev
```

This will start:

- The API server on port 3000 (<http://localhost:3000>)
- The frontend server on port 8080 (<http://localhost:8080>)

You should see output similar to:

```
Starting servers...
[0] Server running on port 3000 (bound to 0.0.0.0)
[0] Local   : http://localhost:3000
[0] Remote  : http://<this-host-IP-or-Tailscale-hostname>:3000
[0] Server is ready to accept connections
[1] Starting up http-server, serving .
[1] Available on:
[1]   http://127.0.0.1:8080
[1]   http://192.168.1.x:8080
[1] Hit CTRL-C to stop the server
```

5. Verify Installation

To verify that the installation was successful:

1. Open your web browser and navigate to <http://localhost:8080>
2. You should see the Mr. MoneyBags login screen
3. Log in with the default credentials:
 - **Username:** admin
 - **Password:** admin
4. After logging in, you should see the dashboard with The Principle Foundation sample data
5. Verify that you can navigate to different sections of the application:
 - Chart of Accounts
 - Funds
 - Journal Entries
 - Fund Reports

Success: If you can log in and see the sample data, your installation is complete and working correctly!

6. Troubleshooting

Here are solutions to common issues you might encounter during installation:

TPF Data Loader Issues

Problem: The bootstrap process warns about TPF loader failures.

Solution: This is usually not critical as the `db-init.sql` script already includes TPF sample data. You can safely ignore these warnings or disable the loader:

```
LOAD_TPF_DATA=false npm run bootstrap:mac
```

Port Conflicts

Problem: The application fails to start because ports 3000 or 8080 are already in use.

Solution: Change the ports by setting environment variables:

```
# For API server (default: 3000)
PORT=3001 npm run start

# For frontend server (default: 8080)
# Edit package.json and change the "client" script port
```

PostgreSQL Connection Issues

Problem: The application cannot connect to PostgreSQL.

Solution: Verify PostgreSQL is running and check connection settings in `.env` file:

```
# Check if PostgreSQL is running
brew services list | grep postgres

# Start PostgreSQL if needed
brew services start postgresql@16
```

```
# Verify connection
psql -h localhost -U postgres -d fund_accounting_db
```

Node.js Version Issues

Problem: Errors related to Node.js version compatibility.

Solution: Ensure you're using Node.js 20 or newer:

```
# Check Node.js version
node --version

# Install correct version if needed
brew install node@20
echo 'export PATH="/usr/local/opt/node@20/bin:$PATH"' >> ~/.zshrc
source ~/.zshrc
```

7. Appendix: Common Commands

Here are some common commands you might need during development:

Database Operations

```
# Recreate the database (drops and recreates)
npm run db:recreate

# Seed the database with sample data
npm run db:seed

# Verify database data
npm run db:verify
```

Application Management

```
# Start API server only
npm run start

# Start frontend server only
npm run client

# Start both (development mode)
npm run dev
```

PostgreSQL Direct Access

```
# Connect to the database
psql -h localhost -U postgres -d fund_accounting_db

# Common psql commands:
# \dt - List tables
# \d tablename - Describe table
# \q - Quit psql
```

Note: For production deployments, make sure to change default passwords and configure proper security settings in the `.env` file.