

# BWeb + Bacula Mock Server

## Integration Guide

### Installation, Configuration & Usage

Version 2.0.0 | March 2026

---

How to connect the Bacula Enterprise BWeb Management Suite to the Bacula Mock Server for training, development, and demonstration environments. Covers minimum requirements, step-by-step installation, configuration, the data flow between components, available BWeb pages, and troubleshooting.

<b>BWeb Port</b>	9180 (lighttpd)
<b>Mock Server Port</b>	9101 (configurable)
<b>Catalog</b>	SQLite (auto-synced)
<b>Sync Interval</b>	60 seconds
<b>Installer</b>	install-bweb-mock.sh
<b>License</b>	BSD 2-Clause

Copyright (c) 2026 faaleoleo dev team — License: BSD 2-Clause

*"Bacula" is a registered trademark of Bacula Systems SA. This software is not affiliated with or endorsed by Bacula Systems SA.*

# Table of Contents

---

1. What This Guide Covers
2. Minimum Requirements
3. Architecture Overview
4. Step-by-Step Installation
5. Configuration Reference
6. How the Data Flows
7. Using BWeb with Mock Data
8. Managing Mock Data
9. Systemd Services Reference
10. Network & Firewall
11. Troubleshooting
12. Quick Reference Card

# 1. What This Guide Covers

---

This guide explains how to install and operate the Bacula Enterprise BWeb Management Suite connected to the Bacula Mock Server. The combination provides a fully functional BWeb environment backed by realistic simulated data — no production Bacula Director, Storage Daemon, or File Daemon required.

The mock server simulates the Bacula Enterprise REST API with wire-compatible responses. A synchronization script mirrors the mock server's in-memory catalog to a SQLite database that BWeb reads through its standard Perl DBI layer. The result is a training, development, and demo environment that looks and behaves like a production BWeb installation.

## **Typical use cases:**

- Administrator training without risk to production backups
- Dashboard and reporting development against realistic data
- Sales demonstrations with diverse client profiles
- QA testing of BWeb with configurable error rates
- Evaluation of BWeb before purchasing a license

## 2. Minimum Requirements

### 2.1 Hardware

Resource	Minimum	Recommended
CPU	1 core	2+ cores
Memory	512 MB	1 GB
Disk	500 MB	1 GB
Network	Localhost only	LAN access for browser

### 2.2 Software

Component	Version	Notes
Operating System	Debian 13 (Trixie)	x86_64 only; Ubuntu not tested
Python	3.8 or higher	Used for mock server and sync script
Bacula Enterprise subscription	Active	Required to download BWeb package
Bacula Enterprise version	18.0+	Must match repository version
Internet access	Required	For package downloads during install
Root/sudo access	Required	Installer must run as root

### 2.3 Packages Installed Automatically

The installer automatically installs the following packages. No manual package installation is needed.

Package	Purpose
bacula-enterprise-bweb	BWeb Management Suite (from Bacula repo)
lighttpd	Web server hosting BWeb CGI application
lighttpd-mod-proxy	Reverse proxy for mock API passthrough
libdbd-sqlite3-perl	Perl SQLite driver for BWeb catalog access
python3, python3-venv, python3-pip	Python runtime for mock server
curl, git, ca-certificates, gnupg	Utilities for download and verification
pyyaml (Python)	YAML configuration parser

## 2.4 Information Needed Before Installation

---

The installer prompts for the following values. Have them ready before running the script.

Prompt	Default	Description
Customer ID	(none)	Bacula Systems customer portal ID
BEE Version	(none)	Bacula Enterprise version (e.g. 18.0)
Mock API port	9101	Port for the mock REST API
Mock username	admin	API authentication username
Mock password	bacula2026	API authentication password
Mock clients	100	Number of simulated clients

## 3. Architecture Overview

The system consists of four services running on a single server. All communication happens over localhost.

### 3.1 Components

Component	Port	Technology	Purpose
BWeb	9180	Perl CGI + lighttpd	Web interface for Bacula management
Mock Server	9101	Python HTTP	Simulates Bacula Enterprise REST API
SQLite Catalog	N/A	SQLite file	Catalog database BWeb reads from
Sync Timer	N/A	systemd timer	Mirrors mock data to SQLite every 60s

### 3.2 Request Flow

BWeb has two data paths:

**Catalog queries** (Jobs, Clients, Media, Statistics): BWeb reads the SQLite database directly via `DBI:SQLite`. The sync timer keeps this database updated from the mock server.

**REST API calls** (Status, Commands, Configuration): lighttpd reverse-proxies these requests from port 9180 to the mock server on port 9101. The proxy covers `/cat/`, `/cmd/`, `/status/`, `/res/`, `/oauth`, `/configure`, and `/mock/` paths.

### 3.3 File Locations

Path	Description
<code>/opt/bacula-mock/</code>	Mock server installation directory
<code>/opt/bacula-mock/src/</code>	Python source files
<code>/opt/bacula-mock/config.yaml</code>	Mock server configuration
<code>/opt/bacula-mock/venv/</code>	Python virtual environment
<code>/opt/bacula-mock/logs/</code>	Mock server log files
<code>/opt/bweb/</code>	BWeb installation directory
<code>/opt/bweb/etc/bweb.conf</code>	BWeb configuration
<code>/opt/bweb/etc/httpd.conf</code>	lighttpd configuration
<code>/opt/bacula/working/bweb_catalog.db</code>	SQLite catalog database

## 4. Step-by-Step Installation

The entire installation is performed by a single script: `install-bweb-mock.sh`. It handles all nine steps automatically.

### 4.1 Download and Run

```
git clone https://git.faaleoleo.io/faaleoleo-dev-team/BaculaMockServer.git
/opt/bacula-mock && cd /opt/bacula-mock && sudo ./install-bweb-mock.sh
```

### 4.2 Installation Steps

Step	Action	Details
1	System dependencies	Installs curl, git, lighttpd, python3, python3-venv
2	Bacula Enterprise repos	Adds GPG key + main + BWeb apt repositories
3	BWeb package	Installs bacula-enterprise-bweb from repository
4	Mock server	Clones repo, runs install.sh, creates systemd service
5	lighttpd proxy	Adds reverse proxy rules to httpd.conf for API paths
6	bweb.conf	Writes DBI connection string pointing to SQLite catalog
7	SQLite + Perl driver	Installs libdbd-sqlite3-perl, runs initial sync
8	Services	Starts bacula-mock, bweb, enables sync timer
9	Verification	Tests all endpoints, checks BWeb web interface

### 4.3 What the Installer Prompts For

The installer asks six questions interactively. All except Customer ID and BEE Version have defaults shown in square brackets.

```
Bacula Systems Customer ID:          <your-customer-id>
Bacula Enterprise version (e.g. 18.0): 18.0
Mock server API port [9101]:         9101
Mock server username [admin]:         admin
Mock server password [bacula2026]:    bacula2026
Number of mock clients [100]:         100
Proceed with installation? [Y/n]:     Y
```

### 4.4 Post-Installation Checklist

After installation the script runs an automatic checklist. All items should show green checkmarks:

- bacula-mock service is running
- BWeb service is running

- Mock server API responding on port 9101
- BWeb web interface accessible on port 9180
- lighttpd authentication configured
- Firewall: port 9180 allowed
- Mock data initialized (clients present)



## 5. Configuration Reference

### 5.1 bweb.conf

The critical setting is the `dbi` line that tells BWeb where to find the catalog database. The installer writes this automatically.

```
$VAR1 = bless( { 'dbi' => 'DBI:SQLite:dbname=/opt/bacula/working/bweb_catalog.db', 'user' => '', 'password' => '', 'stat_job_table' => 'JobHisto', 'template_dir' => '/opt/bweb/tpl', 'lang' => 'en', 'display_log_time' => 'on', 'fv_write_path' => '/opt/bweb/spool', 'html_dir' => '/opt/bweb/html', 'api_url' => 'http://127.0.0.1:9101', 'api_user' => 'admin', 'api_password' => 'bacula2026', }, 'Bweb::Config' );
```

### 5.2 lighttpd Reverse Proxy

The installer adds proxy rules to `/opt/bweb/etc/httpd.conf` so that API requests arriving at BWeb's port (9180) are forwarded to the mock server (9101).

URL Pattern	Proxy Target
/cat/*	localhost:9101
/cmd/*	localhost:9101
/status/*	localhost:9101
/res/*	localhost:9101
/oauth*	localhost:9101
/configure*	localhost:9101
/mock/*	localhost:9101

### 5.3 Mock Server config.yaml

The mock server reads `/opt/bacula-mock/config.yaml`. Key settings for BWeb integration:

Setting	Default	Description
server.port	9101	Must match proxy rules in httpd.conf
authentication.username	admin	Must match api_user in bweb.conf
authentication.password	bacula2026	Must match api_password in bweb.conf
simulation.default_clients	0	Set >0 to auto-populate on startup
simulation.error_rate	0.25	Percentage of jobs with error status

## 6. How the Data Flows

---

### 6.1 Initialization

When the mock server starts (or when `/mock/initialize` is called), the `DataGenerator` creates realistic catalog entries in memory: clients across 30+ profiles, jobs with configurable error rates, volumes with diverse statuses and media types, pools, file sets, storage devices, locations, snapshots, events, and tags. No data is written to disk at this point.

### 6.2 Synchronization

The `sync_mock_to_sqlite.py` script connects to the mock API over HTTP, fetches every table, and writes the data into the SQLite database. The systemd timer `bweb-sync.timer` runs this every 60 seconds automatically.

The sync covers 19 catalog tables plus BWeb-specific tables (`client_group`, `bweb_user`, `bweb_role`). The SQLite schema matches the official Bacula Enterprise Edition schema version 1027.

### 6.3 BWeb Reads the Catalog

BWeb connects to the SQLite database using the Perl `DBD::SQLite` driver. The connection string in `bweb.conf` points to the file. BWeb reads `Job`, `Client`, `Media`, `Pool`, and other tables exactly as it would read a real PostgreSQL or MySQL Bacula catalog.

**NOTE:** BWeb pages that require a live connection to a Storage Daemon or Director (such as the Storage Daemon status page or running a real backup) will show empty or produce errors. This is expected — no real daemons are running. Use the `Jobs`, `Clients`, `Media`, `Pool`, and `Statistics` pages instead.

## 7. Using BWeb with Mock Data

### 7.1 Accessing BWeb

Open a web browser and navigate to:

```
http://<server-ip>:9180/bweb/
```

If lighttpd authentication is configured, enter the credentials set during installation. The default BWeb page shows an overview dashboard.

### 7.2 Pages That Work

BWeb Page	URL Action	What It Shows
Jobs Overview	action=job	All backup/restore jobs with status, size, duration
Job Details	action=job_zoom;jobid=N	Single job detail with log messages
Clients	action=client	All backup clients with retention settings
Client Status	action=client_status	Per-client job history and statistics
Media / Volumes	action=media	All volumes with status, pool, bytes, retention
Pools	action=pool	Storage pools with volume counts and settings
Statistics / Graphs	action=graph	Job statistics over time from JobHisto table
Job Logs	action=log;jobid=N	Detailed log messages for a specific job
Client Groups	action=client_group	Groups auto-created from client name prefixes

### 7.3 Pages That Show Empty (Expected)

BWeb Page	Reason
Storage Daemon Status	Requires live bacula-sd process (not simulated)
Running Jobs	Mock jobs complete instantly; no persistent running state
Restore Wizard (full)	Needs live Director for actual file restore
bconsole Output	No real bconsole binary connected

## 8. Managing Mock Data

---

### 8.1 Initialize Data

---

```
# Create 100 clients with 25% error rate
curl -u admin:bacula2026 -X POST \
  http://localhost:9101/mock/initialize \
  -H 'Content-Type: application/json' \
  -d '{"clients": 100, "error_rate": 0.25}'
```

### 8.2 Reset All Data

---

```
curl -u admin:bacula2026 -X POST \
  http://localhost:9101/mock/reset
```

### 8.3 Manual Sync to SQLite

---

```
cd /opt/bacula-mock
source venv/bin/activate
python3 src/sync_mock_to_sqlite.py \
  --api-url http://127.0.0.1:9101 \
  --user admin --password bacula2026 \
  --db /opt/bacula/working/bweb_catalog.db
```

### 8.4 Verify Data in BWeb

---

After initializing and syncing, restart BWeb and refresh:

```
systemctl restart bweb
```

## 9. Systemd Services Reference

Service	Unit Name	Description
Mock Server	bacula-mock.service	REST API server on port 9101
BWeb	bweb.service	lighttpd web server on port 9180
Catalog Sync	bweb-sync.timer	Runs sync script every 60 seconds

```
# Check all services
systemctl status bacula-mock bweb bweb-sync.timer

# Restart everything
systemctl restart bacula-mock
systemctl restart bweb

# View mock server logs
journalctl -u bacula-mock -f

# Force immediate sync
systemctl start bweb-sync.service
```

## 10. Network & Firewall

Port	Service	Access
9180	BWeb (lighttpd)	Open to network (browser access)
9101	Mock Server	Localhost only (proxied through 9180)

Only port 9180 needs to be accessible from the network. Port 9101 is used internally by the lighttpd proxy and the sync script, both running on localhost.

```
# UFW
sudo ufw allow 9180/tcp

# iptables
sudo iptables -I INPUT -p tcp --dport 9180 -j ACCEPT
```

## 11. Troubleshooting

---

### BWeb shows 'Bad user string format'

The dbi line is missing from bweb.conf. Re-run `install-bweb-mock.sh` or manually add: 'dbi' => 'DBI:SQLite:dbname=/opt/bacula/working/bweb\_catalog.db'

### BWeb shows 'unable to open database file'

The SQLite database does not exist. Run the sync script: `python3 src/sync_mock_to_sqlite.py --db /opt/bacula/working/bweb_catalog.db`. Then set permissions: `chmod 666 /opt/bacula/working/bweb_catalog.db`

### BWeb shows empty pages (no jobs, no clients)

The mock server has no data. Initialize it: `curl -u admin:bacula2026 -X POST http://localhost:9101/mock/initialize -H 'Content-Type: application/json' -d '{"clients":100}'`. Then wait 60 seconds for the sync timer, or run the sync manually.

### Storage Daemon status page is empty

This is expected. No real Storage Daemon is running. The mock server simulates the REST API, not the actual daemon processes. Use the Jobs, Clients, and Media pages.

### Mock server won't start: 'Address already in use'

Port 9101 is occupied. Find the process: `ss -tlnp | grep 9101`. Kill it or change the port in config.yaml.

### Mock server: 'No module named yaml'

The Python virtual environment is broken. Recreate it: `cd /opt/bacula-mock; rm -rf venv; python3 -m venv venv; source venv/bin/activate; pip install -r requirements.txt; systemctl restart bacula-mock`

### Sync script returns 404 errors

The mock server is running old code without the new endpoints. Restart the service: `systemctl restart bacula-mock`. Wait 3 seconds, then re-run the sync.

### BWeb shows 'DBD::SQLite' not found

Install the Perl driver: `apt-get install -y libdbd-sqlite3-perl`. Then restart BWeb.

## 12. Quick Reference Card

### URLs

What	URL
BWeb Web Interface	http://<server>:9180/bweb/
Mock API Health	http://localhost:9101/mock/health
Mock API Docs	http://localhost:9101/docs

### Default Credentials

Setting	Value
Mock API Username	admin
Mock API Password	bacula2026

### Key Commands

```
# Service management
systemctl status bacula-mock bweb bweb-sync.timer
systemctl restart bacula-mock && systemctl restart bweb

# Initialize mock data
curl -u admin:bacula2026 -X POST http://localhost:9101/mock/initialize -H 'Content-Type: application/json'

# Manual sync
cd /opt/bacula-mock && source venv/bin/activate
python3 src/sync_mock_to_sqlite.py --api-url http://127.0.0.1:9101 --db /opt/bacula/working/bweb_catalog.db

# View logs
journalctl -u bacula-mock -f
tail -f /opt/bweb/log/error.log
```

### Key Files

File	Purpose
/opt/bweb/etc/bweb.conf	BWeb config (DBI string, API credentials)
/opt/bweb/etc/httpd.conf	lighttpd config (proxy rules, auth)
/opt/bacula-mock/config.yaml	Mock server config (port, auth, profiles)
/opt/bacula/working/bweb_catalog.db	SQLite catalog database
/opt/bacula-mock/src/sync_mock_to_sqlite.py	Catalog sync script

*"Bacula" is a registered trademark of Bacula Systems SA. This software is not affiliated with or endorsed by Bacula Systems SA.*