

Deploying a Masonite Application on aaPanel Using Gunicorn and tmux

Deploying a Masonite application on a VPS managed with **aaPanel** is straightforward when done step by step. This guide walks you through setting up your environment, running Masonite with **Gunicorn**, and keeping the application alive using **tmux**, along with configuring a reverse proxy in aaPanel.

Step 1: Create a Static Website (MySQL Optional)

Start by creating a **static website** in aaPanel.

If your Masonite project requires a database, you may also create a **MySQL database** at this stage. Otherwise, MySQL is optional.

This step mainly helps aaPanel manage the domain and web root.

Step 2: Remove the Default

```
index.html
```

After the site is created, navigate to the website's root directory and remove the default `index.html` file created by aaPanel.

This ensures it does not conflict with your Masonite application.

Step 3: Upload Your Masonite Project Files

Upload all your **Masonite project files** into the website directory using FTP, SFTP, or the aaPanel file manager.

Make sure your project structure is intact, including the `wsgi.py` (or equivalent) entry file.

Step 4: Install tmux from aaPanel Terminal

Open the **aaPanel Terminal** and switch to the **Super User (root) shell**.

Install `tmux`, which will help keep Gunicorn running even after you close the terminal.

```
yum install tmux -y  
# or (for Ubuntu/Debian)  
apt install tmux -y
```

Step 5: Navigate to Your Project Folder

Still in the super user shell, move to your Masonite project directory:

```
cd /www/wwwroot/your-domain.com
```

Step 6: Initialize Python 3.11.4 Environment

Initialize and activate a **Python 3.11.4 virtual environment** for your project.

“ Refer to previously [Creating new masonite project using python 3.8.5](#) for setting up Python 3.11.4 virtual environment.

Once activated, ensure Python and pip are pointing to the correct environment.

Step 7: Install Gunicorn

With the virtual environment active, install Gunicorn:

```
pip install gunicorn
```

Gunicorn will act as the WSGI server for your Masonite application.

Step 8: Run Gunicorn

Start your Masonite application using the following command:

```
gunicorn wsgi:application --bind your-domain.com:8010
```

- `wsgi:application` refers to your Masonite WSGI entry point
 - `8010` is the internal port Gunicorn will listen on
-

Step 9: Verify the Application

At this point, check if your Masonite application is running correctly.

Yes, it is working — because **you know your code better than anyone else** ☑

If there are issues, check logs and fix them before proceeding.

Step 10: Run Gunicorn in a Detached tmux Session

To keep Gunicorn running in the background, create a new tmux session:

```
tmux new -s gunicorn_session
```

Run Gunicorn again inside the tmux session:

```
gunicorn wsgi:application --bind your-domain.com:8010
```

Detach from the session without stopping Gunicorn:

Press: `Ctrl + B`, then `D`

Your application will continue running in the background.

Step 11: Configure Reverse Proxy in aaPanel

Now go back to **aaPanel** → **Website Settings** → **Reverse Proxy** and add a new proxy rule:

- **Target URL:** `http://127.0.0.1:8010`
- **Important:** Do **not** add a trailing slash (`/`)

Save the configuration.

Step 12: Access Your Masonite Application

Open your domain in the browser.

You will now see your **Masonite application running successfully on aaPanel**, served through Gunicorn and managed via reverse proxy.

Revision #2

Created 28 January 2026 12:48:34 by AI Channel

Updated 28 January 2026 12:55:40 by AI Channel