FlexiCMMS Deployment Guide on Linux Server

This guide describes the deployment of the FlexiCMMS application on a Linux server. You can choose to deploy with Nginx as a reverse proxy or directly with Kestrel. The deployment process is automated using dedicated scripts.

Prerequisites:

- Linux server (e.g., Ubuntu 20.04+ or Debian 10+).
- SSH access to the server.
- Your published application ZIP archive (`flexicmms.zip`).
- If deploying *without Nginx*, a `.pfx` certificate file (`blazortool.pfx`) for HTTPS is required.

Step 0: Deployment Preparation

1. **Transfer necessary files to the server:**
   Copy the following files to your user's home directory on your server (e.g., `/home/youruser/`):
   - Application ZIP archive: `flexicmms.zip`
   - **Choose one deployment script:**
     - For deployment *with Nginx*: `deploy_flexicmms_nginx.sh`
     - For deployment *without Nginx*: `deploy_flexicmms_NoNginx.sh`
   - If deploying *without Nginx*, also copy the PFX certificate file: `blazortool.pfx` (ensure it is named `blazortool.pfx` and has the password `dtech`, as specified in the script and Kestrel configuration).

   Example using `scp` (adjust based on your chosen deployment method):

   ```
   # For deployment with Nginx
   scp /path/to/local/deploy_flexicmms_nginx.sh
   user@your_server_ip:/home/user/
   scp /path/to/local/FlexiCMMS.zip user@your_server_ip:/home/user/

   # For deployment without Nginx
   scp /path/to/local/deploy_flexicmms_NoNginx.sh
   user@your_server_ip:/home/user/
   scp /path/to/local/FlexiCMMS.zip user@your_server_ip:/home/user/
   scp /path/to/local/blazortool.pfx user@your_server_ip:/home/user/
   ```

2. **Connect to the server via SSH:**

   ```
   ssh user@your_server_ip
   ```

3. **Make the chosen script executable:**

```
# For deployment with Nginx
chmod +x deploy_flexicmms_nginx.sh

# For deployment without Nginx
chmod +x deploy_flexicmms_NoNginx.sh
```

Step 1: Run the Deployment Script

Run the chosen script with superuser privileges. The script will prompt you for the API server address and other necessary information, then perform all required steps to install dependencies, unpack the application, configure `appsettings.json`, and set up the Systemd service.

- **For deployment with Nginx:**

  ```
  sudo ./deploy_flexicmms_nginx.sh
  ```

  During execution, the script will ask you to enter:

  - The actual API server address (e.g., `http://10.1.41.122:1122/api/v1/`). The script will automatically add a trailing slash if it's missing.
  - The domain name or IP address for Nginx (e.g., `example.com` or `192.168.1.100`).

- **For deployment without Nginx:**

  ```
  sudo ./deploy_flexicmms_NoNginx.sh
  ```

  During execution, the script will ask you to enter the actual API server address. Make sure you enter it in the correct format (e.g., `http://10.1.41.122:1122/api/v1/`). The script will automatically add a trailing slash if it's missing.

Step 2: Verify Application Status and Accessibility

After the script completes:

1. **Check the status of the FlexiCMMS service:**

   ```
   sudo systemctl status flexicmms
   ```

   You should see that the service is active (`active (running)`).

2. **Configure Firewall and Access the Application:**

- **If deployed with Nginx:**
  Ensure that port 80 (HTTP) is open in your server's firewall. If you are using `ufw`, you can open it like this:

  ```
  sudo ufw allow 80/tcp
  sudo ufw enable # If the firewall is not active
  ```

  Your FlexiCMMS application should now be accessible via your server's domain name or IP address through Nginx.
  Important Note: If you plan to use HTTPS, you will need to manually configure an SSL certificate in Nginx (e.g., using Certbot) after running this script and open port 443 in your firewall.

- **If deployed without Nginx:**
  Ensure that ports 80 (HTTP) and 443 (HTTPS) are open in your server's firewall. If you are using `ufw`, you can open them like this:

  ```
  sudo ufw allow 80/tcp
  sudo ufw allow 443/tcp
  sudo ufw enable # If the firewall is not active
  ```

  Your FlexiCMMS application should now be accessible via your server's IP address or domain name over HTTP (port 80) and HTTPS (port 443).
  Important Note: If you are using a self-signed certificate (`blazortool.pfx`), browsers will show security warnings. For a production environment, it is recommended to use a certificate issued by a trusted Certificate Authority (e.g., Let's Encrypt).