
MemberMatters

Jaimyn Mayer

Sep 01, 2023

CONTENTS:

1	Getting Started	1
1.1	Quickstart	1
1.2	Deployment Tips	2
1.3	Updating your instance	2
1.4	Kiosk Mode	2
2	Post Installation Steps	5
2.1	Important Notice	5
2.2	Set up a reverse proxy	5
2.3	Customisation	6
3	Indices and tables	13

GETTING STARTED

1.1 Quickstart

Note: MemberMatters only supports running in Docker on Linux. Installing Docker is outside the scope of this guide, please consult your favourite search engine for tips.

To get started, download the latest version from docker hub using the following command:

```
docker pull membermatters/membermatters
```

Create a file to contain all of your environment variables. Place it somewhere you won't forget like `/usr/app/env.list`. This file is where Docker gets the environment variables from. You'll need at least the following:

```
# /usr/app/env.list
MM_ENV=Production
```

Once you've downloaded the docker image and configured your environment variables, you'll need to create a container and mount a volume. Replace `/usr/app/` with the location you'd like to store your database and other data.

```
docker create -p 8000:8000 --name membermatters --restart always --env-file /usr/app/env.
↪list -v /usr/app:/usr/src/data membermatters/membermatters
```

After you've created the container (using the command above) you can start/stop/restart it with:

```
docker start membermatters
docker stop membermatters
docker restart membermatters
```

Once your container is running, you will probably want to load the initial database fixtures so there's an account you can use to login with. To do so, connect to your running docker container:

```
docker exec -it membermatters bash
```

and run the following:

```
python3 manage.py loaddata initial
```

Once you've loaded the initial fixtures, you will be able to login with the default admin account details:

```
Email: default@example.com
Password: MemberMatters!
```

The first thing you should do is change the email address and password of the default admin account.

Once you've done this, your MemberMatters instance is ready for configuration. You should now head to *post installation steps* for important instructions on setting up and customising your instance.

1.2 Deployment Tips

- MemberMatters runs on port 8000 by default. You should run a reverse proxy in front of it so you can protect all traffic with HTTPS. Please consult your favourite search engine on how to setup a reverse proxy. We recommend that you use nginx with let's encrypt.
- MemberMatters is designed to run on site if you have any door, interlock or memberbucks devices. However, some parts need reliable networking and internet to function. Please keep this in mind and make sure you perform thorough testing before relying on it.

1.3 Updating your instance

Docker containers are meant to be disposable, so you'll need to delete it and make a fresh one from the latest image. We suggest writing a bash script like the one below:

```
# update.sh
echo "checking for new docker image"
docker pull membermatters/membermatters
echo "Stopping the docker container"
docker stop membermatters
echo "Removing the old docker container"
docker rm membermatters
echo "Creating new docker container"
docker create -p 8000:8000 --name membermatters --restart always --detach --env-file /
↪usr/app/env.list -v /usr/app:/usr/src/data membermatters/membermatters
echo "Running new docker container"
docker start membermatters
```

1.4 Kiosk Mode

MemberMatters also offers a kiosk mode. You can build this by running `API_BASE_URL=https://portal.example.org npm run build:electron` after reading through the development instructions in the frontend folder. At this time, we do not offer precompiled binaries for download.

This command will compile an electron based application that you can run on a machine set up as a kiosk. We recommend that you run the build command on the machine you intend to use as a kiosk to reduce compatibility problems due to different OS versions and/or architectures. For security reasons, kiosk builds will only have limited profile functionality and are primarily meant to allow members to sign in/out of site and use basic features of MemberMatters. Be sure to specify the full URL to your MemberMatters instance, including the protocol (http or https).

The first time a kiosk connects, you will have to “authorise” it from the MemberMatters dashboard. You should see the kiosk show up with it's serial number. Open the kiosk to edit it, then click on authorise to allow full functionality. This is to prevent “random” kiosks being able to connect to your MemberMatters instance.

1.4.1 Linux

You may need to install ffmpeg and a chromium run time.

POST INSTALLATION STEPS

Once you have completed the *getting started* instructions, you should complete the following steps to setup and customise your instance of MemberMatters.

2.1 Important Notice

Currently, a valid Postmark API key is required for MemberMatters to function correctly. Emails are sent on various tasks like sign ups, MemberBucks actions etc. You will receive errors if you try to use these functions without a correctly configured Postmark API key. They have a free trial (100 emails/mth) which should be more than enough for testing, however we recommend upgrading to a paid tier before use in production.

2.2 Set up a reverse proxy

MemberMatters is designed to run behind some form of reverse proxy, or at the minimum, an SSL terminating CDN like Cloudflare (not recommended). You *should not ever* run MemberMatters in production without some form of HTTPS. The recommended way is with an nginx reverse proxy as explained below. Unfortunately, reverse proxy configurations are highly dependant on your specific environment, so only general guidance can be given. Please consult your favourite search engine if you have any trouble and only open a GitHub issue if you think you've found a bug or way to improve this documentation.

2.2.1 Setting up an nginx reverse proxy on Ubuntu

1. You should first install nginx. On Ubuntu, you can install nginx with `sudo apt install nginx`.
2. Configure your nginx instance to proxy traffic through to the MemberMatters docker container on port **8000**.
3. A sample configuration file is included below, but you should configure this to your needs. You should create this file at `/etc/nginx/sites-available/example.com`, where `example.com` is the name of our domain.

```
server {
    server_name example.com;

    location / {
        proxy_set_header Host          $host;
        proxy_set_header X-Real-IP      $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Host $host;
```

(continues on next page)

(continued from previous page)

```
proxy_set_header X-Forwarded-Port $server_port;
proxy_set_header Upgrade          $http_upgrade;
proxy_set_header Connection       "upgrade";

    proxy_redirect off;
    proxy_pass http://localhost:8000;
}
listen 80 default_server;
listen [::]:80 default_server;
}
```

4. Enable your new configuration file by running this command `sudo ln -s /etc/nginx/sites-available/example.com /etc/nginx/sites-enabled/`.
5. Check the configuration that you added is valid by running this command, there should be no errors: `sudo nginx -t`.
6. Restart nginx to apply your new changes with `sudo systemctl restart nginx`.
7. Note that this process does not include a configuration for HTTPS. We recommend that you use the Let's Encrypt Certbot tool as it will automatically modify your configuration to enable HTTPS and manage certificates for you. [Click here](#) and follow the instructions to install certbot on your system. Once installed, run certbot as per that link and follow the prompts to enable HTTPS for your system.
8. Check that you can access your instance of MemberMatters via HTTPS at the URL that you configured.

2.3 Customisation

The primary way to customise MemberMatters is via the database settings. Once your instance is up and running, navigate to `https://<instance_url>/admin` and login with an admin account. Then click on “Config” under “Constance”. On this page you’ll see a variety of settings. You should customise these settings with your own details.

A summary of the settings is available below. Most settings have a more detailed description and an example of the format required on the settings page itself.

NOTE: You *must* configure the `POSTMARK_API_KEY` setting or else you will have problems processing new signups.

2.3.1 Locale / Language Configuration

MemberMatters has out of the box support for different locales (a combination of language and number/currency/date formatting). Currently, only the following locales are supported. If you want to add or improve support for a new locale then please file an issue, we’d love to help make it available in your language.

MemberMatters will auto-detect the locale setting the user’s browser is set to, and use that translation if available. However, as noted below, currencies will use a hardcoded value set by a configuration option.

NOTE: the `SITE_LOCALE_CURRENCY` option is what determines how the currency is displayed. This is “hardcoded” as a config option to prevent a currency/billing amount being displayed incorrectly. If your locale isn’t directly supported, please open an issue or check below as your currency may already be supported under a different locale.

Locale Options

- **en-AU** - full translation available, currency format \$12.50.
- **en-NZ** - no translation available (fall back to **en-AU**), currency format \$12.50.
- **en-US** - no translation available (fall back to **en-AU**), currency format \$12.50.
- **en-GB** - no translation available (fall back to **en-AU**), currency format £12.50.
- **en-IE** - no translation available (fall back to **en-AU**), currency format €12.50.

2.3.2 General

- **"SITE_NAME"** - Name of the website.
- **"SITE_OWNER"** - Name of the organisation running this website.
- **"GOOGLE_ANALYTICS_PROPERTY_ID"** - Enter your Google Analytics Tracking ID to enable Google Analytics Tracking.
- **"API_SECRET_KEY"** - Secret key used to authenticate some requests from access control devices.

2.3.3 Signup

- **"INDUCTION_ENROL_LINK"** - URL to enrol in the Canvas LMS induction course.
- **"INDUCTION_COURSE_ID"** - ID of the Canvas LMS induction course (usually found in the course URL on the settings page).
- **"MAX_INDUCTION_DAYS"** - Maximum number of days since they were inducted before they require another induction. Set to 0 to disable induction requirement.
- **"MIN_INDUCTION_SCORE"** - The minimum score considered a "pass" for the induction course.
- **"REQUIRE_ACCESS_CARD"** - Require the member to submit their RFID access card number during signup.
- **"COLLECT_VEHICLE_REGISTRATION_PLATE"** - Allow the portal to collect vehicle registration plate number(s).

2.3.4 Canvas Integration

- **"CANVAS_API_TOKEN"** - the API token for the Canvas LMS integration.

2.3.5 Postmark (Email) Integration

- **"POSTMARK_API_KEY"** - the API token for the Postmark integration. NOTE: required for basic MemberMatters functionality.

2.3.6 Twilio (SMS) Integration

- **SMS_ENABLE** - Enables sending of SMS messages on some events. See below for a current list of events.
- **TWILIO_ACCOUNT_SID** - The **account SID** (*not api key SID*) to use for the twilio integration.
- **TWILIO_AUTH_TOKEN** - The **auth token** (*not an api token*) to use for the twilio integration.
- **SMS_DEFAULT_COUNTRY_CODE** - If the user's number does not start with a +, this default country code will be prepended to it. This allows support for international numbers, while allowing local users to skip specifying one.
- **SMS_SENDER_ID** - The sender ID (either a phone number or alpha numeric sender ID). Some countries support an "alpha numeric" send ID such as a business name. See [this page](#) for more info.
- **SMS_MESSAGES** - The SMS templates / messages to use.
- **SMS_FOOTER** - An optional footer to append to all SMS messages (such as 'from xyz org.'). Please make sure your organisation is identifiable and check the laws around this in your jurisdiction!

List of SMS events currently supported

You cannot currently enable specific events, you either get "all or nothing".

- Swipe Access Enabled
- Swipe Access Disabled
- Swipe Attempt (e.g. swiped at a door) Denied

2.3.7 Contact Information

- "EMAIL_SYSADMIN" - email address used for sysadmin related notifications.
- "EMAIL_ADMIN" - email address used for general notifications.
- "EMAIL_DEFAULT_FROM" - default "from" address that emails from MM will be sent as. NOTE: must be authenticated / approved in Postmark to use.
- "SITE_MAIL_ADDRESS" - the physical address of the organisation for inclusion in the email footer to comply with anti spam requirements.

2.3.8 Discourse SSO Protocol

- "ENABLE_DISCOURSE_SSO_PROTOCOL" - enable the SSO Discourse protocol.
- "DISCOURSE_SSO_PROTOCOL_SECRET_KEY" - secret key for the SSO Discourse protocol.

2.3.9 URLs

- "SITE_URL" - publicly accessible URL this instance of MM is available on.
- "MAIN_SITE_URL" - the main website of the organisation.
- "INDUCTION_URL" - used in the email sent to new members so they can signup for inductions.

2.3.10 Memberbucks

- “MEMBERBUCKS_MAX_TOPUP” - a hard limit on the maximum amount a member can add in one go to their MemberBucks account.
- “MEMBERBUCKS_CURRENCY” - the currency to use when processing MemberBucks top ups.

2.3.11 Images

- “SITE_LOGO” - a link to a logo for use around the site.
- “SITE_FAVICON” - a link to a square favicon style logo for use around the site.
- “STATS_CARD_IMAGE” - a link to an image used as the background on the dashboard’s statistics card.

2.3.12 Group Localisation

- “MEMBERBUCKS_NAME” - [Deprecated]
- “GROUP_NAME” - [Deprecated]
- “ADMIN_NAME” - [Deprecated]
- “WEBCAM_PAGE_URLS” - a JSON array of URLs to be used as the source for each webcam on the webcams page.
 - You should use an array of arrays like to specify the webcam snapshot title & locations like so:

```
[  
  ["Main Room", "https://example.com/mainroom.jpg"],  
  ["Digital Fabrication", "https://example.com/digifab.jpg"],  
]
```

- “HOME_PAGE_CARDS” - a JSON array of cards to be used on the homepage (see below for more info).
- “WELCOME_EMAIL_CARDS” - a JSON array of cards to be used in the welcome email (see below for more info).

2.3.13 “Stripe Integration”

- “STRIPE_PUBLISHABLE_KEY” - the publishable Stripe key.
- “STRIPE_SECRET_KEY” - the secret Stripe key.
- “STRIPE_WEBHOOK_SECRET” - the webhook secret to authenticate webhook requests are really from Stripe.
- “ENABLE_STRIPE_MEMBERSHIP_PAYMENTS” - enable the “Membership Plan” menu page on the front end so members can sign up with the Stripe billing integration. NOTE: make sure you configure these first from the “Admin Tools” > “Membership Plans” page.
- “STRIPE_MEMBERBUCKS_TOPUP_OPTIONS” - the options a member can see when on the MemberBucks top up page (in cents).

2.3.14 Trello Integration

- “ENABLE_TRELLO_INTEGRATION” - [Deprecated]
- “TRELLO_API_KEY” - [Deprecated]
- “TRELLO_API_TOKEN” - [Deprecated]
- “TRELLO_ID_LIST” - [Deprecated]

2.3.15 Space Directory

- “ENABLE_SPACE_DIRECTORY” - enable a space directory compliant API. The various configuration options in this section should be self explanatory.

2.3.16 Theme Swipe Integration

- “THEME_SWIPE_URL” - a URL to hit on each door/interlock swipe that can trigger a theme song played over your intercom system, or something else.
- “ENABLE_THEME_SWIPE” - enable the theme song swipe webhook.

2.3.17 Door Bump API

- “ENABLE_DOOR_BUMP_API” - Enable an API endpoint that ‘bumps’ (temporarily unlocks) a door for third party integration.
- “DOOR_BUMP_API_KEY” - The API key used to authenticate requests to the door bump API endpoint. MUST be set or the endpoint is automatically disabled.

To use the door bump API, you can send a POST request in the format below: With Authorization header:

```
Authorization: Bearer <DOOR_BUMP_API_KEY>
...
POST /api/access/doors/DOOR_ID/bump/
```

With Query parameter:

```
POST /api/access/doors/DOOR_ID/bump/?secret=DOOR_BUMP_API_KEY
```

Where DOOR_ID is the ID of the door you wish to bump (can be found in the URL of the door’s admin page). The request must be authenticated with an Authorization header set to the value of DOOR_BUMP_API_KEY or as a query parameter as above (NOT recommended for security).

2.3.18 Discord Integration

- “ENABLE_DISCORD_INTEGRATION” - enable the post to Discord channel feature when an interlock or door swipe is recorded.
- “DISCORD_DOOR_WEBHOOK” - URL for the door webhook.
- “DISCORD_INTERLOCK_WEBHOOK” - URL for the interlock webhook.

2.3.19 Home Page and Welcome Email Cards

The settings called “HOME_PAGE_CARDS” and “WELCOME_EMAIL_CARDS” control the content that is displayed on the MemberMatters home page, and the content in the welcome email each user receives when they are converted to a member. These options are configured with a JSON object specifying the content. You can add as many cards as you want, but we recommend 6 as a maximum for the homepage, and 4 for the email. You can find the icon names on [this page](#). Absolute and relative URLs, and vue routes are supported.

An example with 3 cards is below:

```
[
  {
    "title": "Brisbane Makerspace Wiki",
    "description": "Our wiki is like the rule book for BMS. It contains all the
    ↪ information about our tools, processes and other helpful tips.",
    "icon": "class",
    "url": "https://bms.wiki",
    "btn_text": "Read Wiki"
  },
  {
    "title": "Member Bucks",
    "description": "If you need to make a payment for tool usage or something else, tap
    ↪ here.",
    "icon": "mdi-alert",
    "routerLink": {
      "name": "memberbucks"
    },
    "btn_text": "Member Bucks"
  },
  {
    "title": "Discord Server",
    "description": "Discord is an instant messaging platform that allows you to
    ↪ connect with other BMS members to share you skills, knowledge and projects. You can
    ↪ also get realtime support from staff/other members.",
    "icon": "mdi-chat",
    "links": [
      {
        "url": "https://s.bms.wiki/discord",
        "btn_text": "Join Discord",
        "newLine": true
      },
      {
        "url": "https://s.bms.wiki/discord",
        "btn_text": "Join Discord 2"
      }
    ]
  }
]
```


INDICES AND TABLES

- `genindex`
- `modindex`
- `search`