

# WebMonitor - User Guide

---

Webmonitor is a Java application for monitoring state of web services. It will periodically check and log a state of configured websites. Optionally it can be configured to send email or Discord webhook notifications.

## Javadoc

---

Javadoc documentation is available in `docs/javadoc` directory.

## Installation

---

**Recommended Java version: 17**

Maven is built tool used for this application.

Run `mvn clean package` to generate JAR file.

This will generate `WebMonitor-1.0.jar` and `lib/` in `target/` directory.

`lib/` **must** be present in the same directory as the JAR file for correct functionality. But both can be moved elsewhere.

```
src/  
pom.xml  
target/  
├─ WebMonitor-1.0.jar  
├─ libs/
```

## Usage

---

`webmonitor` will be used instead of `WebMonitor-1.0.jar`

To display available command line arguments use `webmonitor --help`

## Command line arguments

First argument of the program is configuration file for monitoring.

```
webmonitor config.yaml
```

Additional optional arguments:

argument	function
--help	display available arguments
-gc (--generateConfig)	generate config template
-v (--verbose)	enable more detailed output

To generate a configuration file template in the current directory use `webmonitor -gc`. See Configuration section for more info about configuration.

The application will only log important error messages by default. Use the `-v` option to enable all messages.

## Configuration

YAML formatted is used to specify all settings.

Example:

```
webhook: https://discord.com/api/webhooks/...
email:
  apikey: abc123
  secretkey: cba321
  from: from@email.com
  to: to@email.com

log: log.txt

services:
  - address: https://www.example.com
    status: any
    notify: discord
    interval: 120s
    timeout: 10s
```

More complex example configuration file can be found in `examples/config-example.yaml`.

## Options

webhook (optional)

Discord webhook where notification will be sent. Format:

```
https://discord.com/api/webhooks/abcdef
```

Can be generated by going to a channel of choice, right-clicking it and choosing "Edit channel". In the menu go to "Integrations" tab and select "Create Webhook". This will create a new webhook which you can further customize. To obtain the necessary URL click on "Copy Webhook URL".

### **email (optional)**

[Mailjet](#) service is used to send email notifications.

Create a Mailjet account and complete following steps to enable email notifications:

1. Go to `Account settings > Master API Key & Sub API key management` and copy your `API KEY` and `SECRET KEY` into your configuration.
2. Go to `Account settings > Add a Sender Domain or Address > Add a sender address`, add your **existing** email address and confirm verification email.

You have to add your existing email (or create a new one), otherwise emails will not get sent to you. Copy this email address into `from` option in your config file. `to` can be any email that will receive notifications.

It is very likely that email notifications will arrive into your spam folder at first. Adjust your spam filter.

### **log (optional)**

Specify where the application should log all information. Leave empty for logging into console only. Make sure you are using correct file path.

### **services (required)**

A list of services to be monitored. At least one must be specified.

option	required	description	examples
address	yes	web address (URL)	<a href="https://www.example.com">https://www.example.com</a>
status	yes	status regex	explained below
notify	no	notification level	email , discord or false (default)
interval	yes	monitoring interval	50s , 1h30m , 300s
timeout	no	HTTP request timeout	as above (default: 30s )

## Status regex

Statuses that will be considered as errors can be specified using a regular expression. Default value is "[3-9].." (Meaning all status codes above 300). any is an alias for this value.

To match specific status use 300 , 300|404|500 .

Dot ( . ) can match any single digit. ( 40. , 3.. )