# Plugins

Plugins are used extend the way information is displayed, how notifications are sent, alarms are triggered, and more.

The built-in/example plugins that are available by default are listed below. The plugins may still need to be enabled by adding to the ENABLE environment variable.

**Default Plugins**

These can be disabled by adding them to the DISABLE variable, for example DISABLE="direction upbat"

**delta (BG Delta)**

Calculates and displays the change between the last 2 BG values.

**direction (BG Direction)**

Displays the trend direction.

**upbat (Uploader Battery)**

Displays the most recent battery status from the uploader phone. . Use these [extended setting](https://github.com/nightscout/cgm-remote-monitor#extended-settings) to adjust behavior:

* UPBAT\_ENABLE\_ALERTS (false) - Set to true to enable uploader battery alarms via Pushover and IFTTT.
* UPBAT\_WARN (30) - Minimum battery percent to trigger warning.
* UPBAT\_URGENT (20) - Minimum battery percent to trigger urgent alarm.

**timeago (Time Ago)**

Displays the time since last CGM entry. Use these [extended setting](https://github.com/nightscout/cgm-remote-monitor#extended-settings) to adjust behavior:

* TIMEAGO\_ENABLE\_ALERTS (false) - Set to true to enable stale data alarms via Pushover and IFTTT.
* ALARM\_TIMEAGO\_WARN (on) - possible values on or off
* ALARM\_TIMEAGO\_WARN\_MINS (15) - minutes since the last reading to trigger a warning
* ALARM\_TIMEAGO\_URGENT (on) - possible values on or off
* ALARM\_TIMEAGO\_URGENT\_MINS (30) - minutes since the last reading to trigger a urgent alarm

**devicestatus (Device Status)**

Used by upbat and other plugins to display device status info. Supports the DEVICESTATUS\_ADVANCED="true" [extended setting](https://github.com/nightscout/cgm-remote-monitor#extended-settings) to send all device statuses to the client for retrospective use and to support other plugins.

**errorcodes (CGM Error Codes)**

Generates alarms for CGM codes 9 (hourglass) and 10 (???).

* Use [extended settings](https://github.com/nightscout/cgm-remote-monitor#extended-settings) to adjust what errorcodes trigger notifications and alarms:
  + ERRORCODES\_INFO (1 2 3 4 5 6 7 8) - By default the needs calibration (blood drop) and other codes below 9 generate an info level notification, set to a space separate list of number or off to disable
  + ERRORCODES\_WARN (off) - By default there are no warning configured, set to a space separate list of numbers or off to disable
  + ERRORCODES\_URGENT (9 10) - By default the hourglass and ??? generate an urgent alarm, set to a space separate list of numbers or off to disable

**ar2 (AR2 Forecasting)**

Generates alarms based on forecasted values. See [Forecasting using AR2 algorithm](https://github.com/nightscout/nightscout.github.io/wiki/Forecasting)

* Enabled by default if no thresholds are set **OR** ALARM\_TYPES includes predict.
* Use [extended settings](https://github.com/nightscout/cgm-remote-monitor#extended-settings) to adjust AR2 behavior:
  + AR2\_CONE\_FACTOR (2) - to adjust size of cone, use 0 for a single line.

**simplealarms (Simple BG Alarms)**

Uses BG\_HIGH, BG\_TARGET\_TOP, BG\_TARGET\_BOTTOM, BG\_LOW thresholds to generate alarms.

* Enabled by default if 1 of these thresholds is set **OR** ALARM\_TYPES includes simple.

**profile (Treatment Profile)**

Add link to Profile Editor and allow to enter treatment profile settings. Also uses the extended setting:

* PROFILE\_HISTORY (off) - possible values on or off. Enable/disable NS ability to keep history of your profiles (still experimental)
* PROFILE\_MULTIPLE (off) - possible values on or off. Enable/disable NS ability to handle and switch between multiple treatment profiles

**Advanced Plugins:**

**careportal (Careportal)**

An optional form to enter treatments.

**boluscalc (Bolus Wizard)**

**food (Custom Foods)**

An option plugin to enable adding foods from database in Bolus Wizard and enable .

**rawbg (Raw BG)**

Calculates BG using sensor and calibration records from and displays an alternate BG values and noise levels. Defaults that can be adjusted with [extended setting](https://github.com/nightscout/cgm-remote-monitor#extended-settings)

* DISPLAY (unsmoothed) - Allows the user to control which algorithm is used to calculate the displayed raw BG values using the most recent calibration record.
  + unfiltered - Raw BG is calculated by applying the calibration to the glucose record's unfiltered value.
  + filtered - Raw BG is calculated by applying the calibration to the glucose record's filtered value. The glucose record's filtered values are generally produced by the CGM by a running average of the unfiltered values to produce a smoothed value when the sensor noise is high.
  + unsmoothed - Raw BG is calculated by first finding the ratio of the calculated filtered value (the same value calculated by the filtered setting) to the reported glucose value. The displayed raw BG value is calculated by dividing the calculated unfiltered value (the same value calculated by the unfiltered setting) by the ratio. The effect is to exagerate changes in trend direction so the trend changes are more noticeable to the user. This is the legacy raw BG calculation algorithm.

**iob (Insulin-on-Board)**

Adds the IOB pill visualization in the client and calculates values that used by other plugins. Uses treatments with insulin doses and the dia and sens fields from the [treatment profile](https://github.com/nightscout/cgm-remote-monitor#treatment-profile).

**cob (Carbs-on-Board)**

Adds the COB pill visualization in the client and calculates values that used by other plugins. Uses treatments with carb doses and the carbs\_hr, carbratio, and sens fields from the [treatment profile](https://github.com/nightscout/cgm-remote-monitor#treatment-profile).

**bwp (Bolus Wizard Preview)**

This plugin in intended for the purpose of automatically snoozing alarms when the CGM indicates high blood sugar but there is also insulin on board (IOB) and secondly, alerting to user that it might be beneficial to measure the blood sugar using a glucometer and dosing insulin as calculated by the pump or instructed by trained medicare professionals. ***The values provided by the plugin are provided as a reference based on CGM data and insulin sensitivity you have configured, and are not intended to be used as a reference for bolus calculation.*** The plugin calculates the bolus amount when above your target, generates alarms when you should consider checking and bolusing, and snoozes alarms when there is enough IOB to cover a high BG. Uses the results of the iob plugin and sens, target\_high, and target\_low fields from the [treatment profile](https://github.com/nightscout/cgm-remote-monitor#treatment-profile). Defaults that can be adjusted with [extended setting](https://github.com/nightscout/cgm-remote-monitor#extended-settings)

* BWP\_WARN (0.50) - If BWP is > BWP\_WARN a warning alarm will be triggered.
* BWP\_URGENT (1.00) - If BWP is > BWP\_URGENT an urgent alarm will be triggered.
* BWP\_SNOOZE\_MINS (10) - minutes to snooze when there is enough IOB to cover a high BG.
* BWP\_SNOOZE - (0.10) If BG is higher then the target\_high and BWP < BWP\_SNOOZE alarms will be snoozed for BWP\_SNOOZE\_MINS.

**cage (Cannula Age)**

Calculates the number of hours since the last Site Change treatment that was recorded.

* CAGE\_ENABLE\_ALERTS (false) - Set to true to enable notifications to remind you of upcoming cannula change.
* CAGE\_INFO (44) - If time since last Site Change matches CAGE\_INFO, user will be warned of upcoming cannula change
* CAGE\_WARN (48) - If time since last Site Change matches CAGE\_WARN, user will be alarmed to to change the cannula
* CAGE\_URGENT (72) - If time since last Site Change matches CAGE\_URGENT, user will be issued a persistent warning of overdue change.
* CAGE\_DISPLAY (hours) - Possible values are 'hours' or 'days'. If 'days' is selected and age of canula is greater than 24h number is displayed in days and hours

**sage (Sensor Age)**

Calculates the number of days and hours since the last Sensor Start and Sensor Change treatment that was recorded.

* SAGE\_ENABLE\_ALERTS (false) - Set to true to enable notifications to remind you of upcoming sensor change.
* SAGE\_INFO (144) - If time since last sensor event matches SAGE\_INFO, user will be warned of upcoming sensor change
* SAGE\_WARN (164) - If time since last sensor event matches SAGE\_WARN, user will be alarmed to to change/restart the sensor
* SAGE\_URGENT (166) - If time since last sensor event matches SAGE\_URGENT, user will be issued a persistent warning of overdue change.

**iage (Insulin Age)**

Calculates the number of days and hours since the last Insulin Change treatment that was recorded.

* IAGE\_ENABLE\_ALERTS (false) - Set to true to enable notifications to remind you of upcoming insulin reservoir change.
* IAGE\_INFO (44) - If time since last Insulin Change matches IAGE\_INFO, user will be warned of upcoming insulin reservoir change
* IAGE\_WARN (48) - If time since last Insulin Change matches IAGE\_WARN, user will be alarmed to to change the insulin reservoir
* IAGE\_URGENT (72) - If time since last Insulin Change matches IAGE\_URGENT, user will be issued a persistent warning of overdue change.

**bage (Battery Age)**

Calculates the number of days and hours since the last Pump Battery Change treatment that was recorded.

* BAGE\_ENABLE\_ALERTS (false) - Set to true to enable notifications to remind you of upcoming pump battery change.
* BAGE\_DISPLAY (days) - Set to hours to display time since last Pump Battery Change in hours only.
* BAGE\_INFO (312) - If time since last Pump Battery Change matches BAGE\_INFO hours, user will be warned of upcoming pump battery change (default of 312 hours is 13 days).
* BAGE\_WARN (336) - If time since last Pump Battery Change matches BAGE\_WARN hours, user will be alarmed to to change the pump battery (default of 336 hours is 14 days).
* BAGE\_URGENT (360) - If time since last Pump Battery Change matches BAGE\_URGENT hours, user will be issued a persistent warning of overdue change (default of 360 hours is 15 days).

**treatmentnotify (Treatment Notifications)**

Generates notifications when a treatment has been entered and snoozes alarms minutes after a treatment.

* TREATMENTNOTIFY\_SNOOZE\_MINS (10) - Number of minutes to snooze notifications after a treatment is entered
* TREATMENTNOTIFY\_INCLUDE\_BOLUSES\_OVER (0) - U value over which the bolus will trigger a notification and snooze alarms

**basal (Basal Profile)**

Adds the Basal pill visualization to display the basal rate for the current time. Also enables the bwp plugin to calculate correction temp basal suggestions. Uses the basal field from the [treatment profile](https://github.com/nightscout/cgm-remote-monitor#treatment-profile). Also uses the extended setting:

* BASAL\_RENDER (none) - Possible values are none, default, or icicle (inverted)

**bolus (Bolus Rendering)**

Settings to configure Bolus rendering

* BOLUS\_RENDER\_OVER (0) - U value over which the bolus labels use the format defined in BOLUS\_RENDER\_FORMAT. This value can be an integer or a float, e.g. 0.3, 1.5, 2, etc.
* BOLUS\_RENDER\_FORMAT (default) - Possible values are hidden, default (with leading zero and U), concise (with U, without leading zero), and minimal (without leading zero and U).
* BOLUS\_RENDER\_FORMAT\_SMALL (default) - Possible values are hidden, default (with leading zero and U), concise (with U, without leading zero), and minimal (without leading zero and U).

**bridge (Share2Nightscout bridge)**

Glucose reading directly from the Dexcom Share service, uses these extended settings:

* BRIDGE\_USER\_NAME - Your username for the Share service.
* BRIDGE\_PASSWORD - Your password for the Share service.
* BRIDGE\_INTERVAL (150000 *2.5 minutes*) - The time (in milliseconds) to wait between each update.
* BRIDGE\_MAX\_COUNT (1) - The number of records to attempt to fetch per update.
* BRIDGE\_FIRST\_FETCH\_COUNT (3) - Changes max count during the very first update only.
* BRIDGE\_MAX\_FAILURES (3) - How many failures before giving up.
* BRIDGE\_MINUTES (1400) - The time window to search for new data per update (the default value is one day in minutes).
* BRIDGE\_SERVER (``) - The default blank value is used to fetch data from Dexcom servers in the US. Set to (EU) to fetch from European servers instead.

**mmconnect (MiniMed Connect bridge)**

Transfer real-time MiniMed Connect data from the Medtronic CareLink server into Nightscout ([read more](https://github.com/mddub/minimed-connect-to-nightscout))

* MMCONNECT\_USER\_NAME - Your user name for CareLink Connect.
* MMCONNECT\_PASSWORD - Your password for CareLink Connect.
* MMCONNECT\_INTERVAL (60000 *1 minute*) - Number of milliseconds to wait between requests to the CareLink server.
* MMCONNECT\_MAX\_RETRY\_DURATION (32) - Maximum number of total seconds to spend retrying failed requests before giving up.
* MMCONNECT\_SGV\_LIMIT (24) - Maximum number of recent sensor glucose values to send to Nightscout on each request.
* MMCONNECT\_VERBOSE - Set this to "true" to log CareLink request information to the console.
* MMCONNECT\_STORE\_RAW\_DATA - Set this to "true" to store raw data returned from CareLink as type: "carelink\_raw" database entries (useful for development).
* MMCONNECT\_SERVER - Set this to EU if you're using the European Medtronic services

**pump (Pump Monitoring)**

Generic Pump Monitoring for OpenAPS, MiniMed Connect, RileyLink, t:slim, with more on the way

* Requires DEVICESTATUS\_ADVANCED="true" to be set
* PUMP\_ENABLE\_ALERTS (false) - Set to true to enable notifications for Pump battery and reservoir.
* PUMP\_WARN\_ON\_SUSPEND (false) - Set to true to get an alarm when the pump is suspended.
* PUMP\_FIELDS (reservoir battery) - The fields to display by default. Any of the following fields: reservoir, battery, clock, status, and device
* PUMP\_RETRO\_FIELDS (reservoir battery clock) - The fields to display in retro mode. Any of the above fields.
* PUMP\_WARN\_CLOCK (30) - The number of minutes ago that needs to be exceed before an alert is triggered.
* PUMP\_URGENT\_CLOCK (60) - The number of minutes ago that needs to be exceed before an urgent alarm is triggered.
* PUMP\_WARN\_RES (10) - The number of units remaining, a warning will be triggered when dropping below this threshold.
* PUMP\_URGENT\_RES (5) - The number of units remaining, an urgent alarm will be triggered when dropping below this threshold.
* PUMP\_WARN\_BATT\_P (30) - The % of the pump battery remaining, a warning will be triggered when dropping below this threshold.
* PUMP\_URGENT\_BATT\_P (20) - The % of the pump battery remaining, an urgent alarm will be triggered when dropping below this threshold.
* PUMP\_WARN\_BATT\_V (1.35) - The voltage (if percent isn't available) of the pump battery, a warning will be triggered when dropping below this threshold.
* PUMP\_URGENT\_BATT\_V (1.30) - The voltage (if percent isn't available) of the pump battery, an urgent alarm will be triggered when dropping below this threshold.
* PUMP\_WARN\_BATT\_QUIET\_NIGHT (false) - Do not generate battery alarms at night.

**openaps (OpenAPS)**

Integrated OpenAPS loop monitoring, uses these extended settings:

* Requires DEVICESTATUS\_ADVANCED="true" to be set
* OPENAPS\_ENABLE\_ALERTS (false) - Set to true to enable notifications when OpenAPS isn't looping. If OpenAPS is going to offline for a period of time, you can add an OpenAPS Offline event for the expected duration from Careportal to avoid getting alerts.
* OPENAPS\_WARN (30) - The number of minutes since the last loop that needs to be exceed before an alert is triggered
* OPENAPS\_URGENT (60) - The number of minutes since the last loop that needs to be exceed before an urgent alarm is triggered
* OPENAPS\_FIELDS (status-symbol status-label iob meal-assist rssi) - The fields to display by default. Any of the following fields: status-symbol, status-label, iob, meal-assist, freq, and rssi
* OPENAPS\_RETRO\_FIELDS (status-symbol status-label iob meal-assist rssi) - The fields to display in retro mode. Any of the above fields.
* OPENAPS\_PRED\_IOB\_COLOR (#1e88e5) - The color to use for IOB prediction lines. Colors can be in #RRGGBB format, but [other CSS color units](https://www.w3.org/TR/css-color-3/#colorunits) may be used as well.
* OPENAPS\_PRED\_COB\_COLOR (#FB8C00) - The color to use for COB prediction lines. Same format as above.
* OPENAPS\_PRED\_ACOB\_COLOR (#FB8C00) - The color to use for ACOB prediction lines. Same format as above.
* OPENAPS\_PRED\_ZT\_COLOR (#00d2d2) - The color to use for ZT prediction lines. Same format as above.
* OPENAPS\_PRED\_UAM\_COLOR (#c9bd60) - The color to use for UAM prediction lines. Same format as above.
* OPENAPS\_COLOR\_PREDICTION\_LINES (true) - Enables / disables the colored lines vs the classic purple color.

Also see [Pushover](https://github.com/nightscout/cgm-remote-monitor#pushover) and [IFTTT Maker](https://github.com/nightscout/cgm-remote-monitor#ifttt-maker).

**loop (Loop)**

iOS Loop app monitoring, uses these extended settings:

* Requires DEVICESTATUS\_ADVANCED="true" to be set
* LOOP\_ENABLE\_ALERTS (false) - Set to true to enable notifications when Loop isn't looping.
* LOOP\_WARN (30) - The number of minutes since the last loop that needs to be exceeded before an alert is triggered
* LOOP\_URGENT (60) - The number of minutes since the last loop that needs to be exceeded before an urgent alarm is triggered
* Add loop to SHOW\_FORECAST to show forecasted BG.

For remote overrides, the following extended settings must be configured:

* LOOP\_APNS\_KEY - Apple Push Notifications service (APNs) Key, created in the Apple Developer website.
* LOOP\_APNS\_KEY\_ID - The Key ID for the above key.
* LOOP\_DEVELOPER\_TEAM\_ID - Your Apple developer team ID.
* LOOP\_PUSH\_SERVER\_ENVIRONMENT - (optional) Set this to production if you are using a provisioning profile that specifies production aps-environment, such as when distributing builds via TestFlight.

**override (Override Mode)**

Additional monitoring for DIY automated insulin delivery systems to display real-time overrides such as Eating Soon or Exercise Mode:

* Requires DEVICESTATUS\_ADVANCED="true" to be set

**xdripjs (xDrip-js)**

Integrated xDrip-js monitoring, uses these extended settings:

* Requires DEVICESTATUS\_ADVANCED="true" to be set
* XDRIPJS\_ENABLE\_ALERTS (false) - Set to true to enable notifications when CGM state is not OK or battery voltages fall below threshold.
* XDRIPJS\_STATE\_NOTIFY\_INTRVL (0.5) - Set to number of hours between CGM state notifications
* XDRIPJS\_WARN\_BAT\_V (300) - The voltage of either transmitter battery, a warning will be triggered when dropping below this threshold.

**alexa (Amazon Alexa)**

Integration with Amazon Alexa, [detailed setup instructions](https://github.com/nightscout/cgm-remote-monitor/blob/master/docs/plugins/alexa-plugin.md)

**googlehome (Google Home/DialogFLow)**

Integration with Google Home (via DialogFlow), [detailed setup instructions](https://github.com/nightscout/cgm-remote-monitor/blob/master/docs/plugins/googlehome-plugin.md)

**speech (Speech)**

Speech synthesis plugin. When enabled, speaks out the blood glucose values, IOB and alarms. Note you have to set the LANGUAGE setting on the server to get all translated alarms.

**cors (CORS)**

Enabled [CORS](https://en.wikipedia.org/wiki/Cross-origin_resource_sharing) so other websites can make request to your Nightscout site, uses these extended settings:

* CORS\_ALLOW\_ORIGIN (\*) - The list of sites that are allow to make requests

**dbsize (Database Size)**

Show size of Nightscout Database, as a percentage of declared available space or in MiB.

Many deployments of Nightscout use free tier of MongoDB Atlas on Heroku, which is limited in size. After some time, as volume of stored data grows, it may happen that this limit is reached and system is unable to store new data. This plugin provides pill that indicates size of Database and shows (when configured) alarms regarding reaching space limit.

**IMPORTANT:** This plugin can only check how much space database already takes, *but cannot infer* max size available on server for it. To have correct alarms and realistic percentage, DBSIZE\_MAX need to be properly set - according to your mongoDB hosting configuration.

**NOTE:** This plugin rely on db.stats() for reporting *logical* size of database, which may be different than *physical* size of database on server. It may work for free tier of MongoDB on Atlas, since it calculate quota according to logical size too, but may fail for other hostings or self-hosted database with quota based on physical size.

**NOTE:** MongoDB Atlas quota is for **all** databases in cluster, while each instance will get only size of **its own database only**. It is ok when you only have **one** database in cluster (most common scenario) but will not work for multiple parallel databases. In such case, spliting known quota equally beetween databases and setting DBSIZE\_MAX to that fraction may help, but wont be precise.

All sizes are expressed as integers, in *Mebibytes* 1 MiB == 1024 KiB == 1024\*1024 B

* DBSIZE\_MAX (496) - Maximal allowed size of database on your mongoDB server, in MiB. You need to adjust that value to match your database hosting limits - default value is for standard Heroku mongoDB free tier.
* DBSIZE\_WARN\_PERCENTAGE (60) - Threshold to show first warning about database size. When database reach this percentage of DBSIZE\_MAX size - pill will show size in yellow.
* DBSIZE\_URGENT\_PERCENTAGE (75) - Threshold to show urgent warning about database size. When database reach this percentage of DBSIZE\_MAX size, it is urgent to do backup and clean up old data. At this percentage info pill turns red.
* DBSIZE\_ENABLE\_ALERTS (false) - Set to true to enable notifications about database size.
* DBSIZE\_IN\_MIB (false) - Set to true to display size of database in MiB-s instead of default percentage.

This plugin should be enabled by default, if needed can be diasabled by adding dbsize to the list of disabled plugins, for example: DISABLE="dbsize".

**Extended Settings**

Some plugins support additional configuration using extra environment variables. These are prefixed with the name of the plugin and a \_. For example setting MYPLUGIN\_EXAMPLE\_VALUE=1234 would make extendedSettings.exampleValue available to the MYPLUGIN plugin.

Plugins only have access to their own extended settings, all the extended settings of client plugins will be sent to the browser.

* DEVICESTATUS\_ADVANCED (true) - Defaults to true. Users who only have a single device uploading data to Nightscout can set this to false to reduce the data use of the site.
* DEVICESTATUS\_DAYS (1) - Defaults to 1, can optionally be set to 2. Users can use this to show 48 hours of device status data for in retro mode, rather than the default 24 hours. Setting this value to 2 will roughly double the bandwidth usage of nightscout, so users with a data cap may not want to update this setting.

**Pushover**

In addition to the normal web based alarms, there is also support for [Pushover](https://pushover.net/) based alarms and notifications.

To get started install the Pushover application on your iOS or Android device and create an account.

Using that account login to [Pushover](https://pushover.net/), in the top left you’ll see your User Key, you’ll need this plus an application API Token/Key to complete this setup.

You’ll need to [Create a Pushover Application](https://pushover.net/apps/build). You only need to set the Application name, you can ignore all the other settings, but setting an Icon is a nice touch. Maybe you'd like to use [this one](https://raw.githubusercontent.com/nightscout/cgm-remote-monitor/master/static/images/large.png)?

Pushover is configured using the following Environment Variables:

\* `ENABLE` - `pushover` should be added to the list of plugin, for example: `ENABLE="pushover"`.

\* `PUSHOVER\_API\_TOKEN` - Used to enable pushover notifications, this token is specific to the application you create from in [Pushover](https://pushover.net/), \*\*\*[additional pushover information](#pushover)\*\*\* below.

\* `PUSHOVER\_USER\_KEY` - Your Pushover user key, can be found in the top left of the [Pushover](https://pushover.net/) site, this can also be a pushover delivery group key to send to a group rather than just a single user. This also supports a space delimited list of keys. To disable `INFO` level pushes set this to `off`.

\* `PUSHOVER\_ALARM\_KEY` - An optional Pushover user/group key, will be used for system wide alarms (level > `WARN`). If not defined this will fallback to `PUSHOVER\_USER\_KEY`. A possible use for this is sending important messages and alarms to a CWD that you don't want to send all notification too. This also support a space delimited list of keys. To disable Alarm pushes set this to `off`.

\* `PUSHOVER\_ANNOUNCEMENT\_KEY` - An optional Pushover user/group key, will be used for system wide user generated announcements. If not defined this will fallback to `PUSHOVER\_USER\_KEY` or `PUSHOVER\_ALARM\_KEY`. This also support a space delimited list of keys. To disable Announcement pushes set this to `off`.

\* `BASE\_URL` - Used for pushover callbacks, usually the URL of your Nightscout site, use https when possible.

\* `API\_SECRET` - Used for signing the pushover callback request for acknowledgments.

If you never want to get info level notifications (treatments) use `PUSHOVER\_USER\_KEY="off"`

If you never want to get an alarm via pushover use `PUSHOVER\_ALARM\_KEY="off"`

If you never want to get an announcement via pushover use `PUSHOVER\_ANNOUNCEMENT\_KEY="off"`

If only `PUSHOVER\_USER\_KEY` is set it will be used for all info notifications, alarms, and announcements

For testing/development try [localtunnel](http://localtunnel.me/).

**IFTTT Maker**

In addition to the normal web based alarms, and pushover, there is also integration for [IFTTT Webhooks](https://ifttt.com/maker_webhooks).

With Maker you are able to integrate with all the other [IFTTT Services](https://ifttt.com/services). For example you can send a tweet when there is an alarm, change the color of hue light, send an email, send and sms, and so much more.

1. Setup IFTTT account: [login](https://ifttt.com/login) or [create an account](https://ifttt.com/join)
2. Follow the [Detailed IFTTT setup Instructions](https://github.com/nightscout/cgm-remote-monitor/blob/master/docs/plugins/maker-setup.md)
3. Configure Nightscout by setting these webpage environment variables:

* ENABLE - maker should be added to the list of plugins, for example: ENABLE="maker".
* MAKER\_KEY - Set this to your secret key (see [[Detailed Instructions](https://github.com/nightscout/cgm-remote-monitor/blob/master/docs/plugins/maker-setup.md) ) MAKER\_KEY="abcMyExampleabc123defjt1DeNSiftttmak-XQb69p" This also supports a space delimited list of keys.
* MAKER\_ANNOUNCEMENT\_KEY - An optional Maker key, will be used for system wide user generated announcements. If not defined this will fallback to MAKER\_KEY. A possible use for this is sending important messages and alarms to another device that you don't want to send all notification too. This also support a space delimited list of keys.

Plugins can create custom events, but all events sent to IFTTT webhooks will be prefixed with ns-. The core events are:

* ns-event - This event is sent to the maker service for all alarms and notifications. This is good catch all event for general logging.
* ns-allclear - This event is sent to the maker service when an alarm has been ack'd or when the server starts up without triggering any alarms. For example, you could use this event to turn a light to green.
* ns-info - Plugins that generate notifications at the info level will cause this event to also be triggered. It will be sent in addition to ns-event.
* ns-warning - Alarms at the warning level with cause this event to also be triggered. It will be sent in addition to ns-event.
* ns-urgent - Alarms at the urgent level with cause this event to also be triggered. It will be sent in addition to ns-event.
* see the [full list of events](https://github.com/nightscout/cgm-remote-monitor/blob/master/lib/plugins/maker-setup.md#events)

**Treatment Profile**

Some of the [plugins](https://github.com/nightscout/cgm-remote-monitor#plugins) make use of a treatment profile that can be edited using the Profile Editor, see the link in the Settings drawer on your site.

Treatment Profile Fields:

* timezone (Time Zone) - time zone local to the patient. *Should be set.*
* units (Profile Units) - blood glucose units used in the profile, either "mg/dl" or "mmol"
* dia (Insulin duration) - value should be the duration of insulin action to use in calculating how much insulin is left active. Defaults to 3 hours.
* carbs\_hr (Carbs per Hour) - The number of carbs that are processed per hour, for more information see [#DIYPS](http://diyps.org/2014/05/29/determining-your-carbohydrate-absorption-rate-diyps-lessons-learned/).
* carbratio (Carb Ratio) - grams per unit of insulin.
* sens (Insulin sensitivity) How much one unit of insulin will normally lower blood glucose.
* basal The basal rate set on the pump.
* target\_high - Upper target for correction boluses.
* target\_low - Lower target for correction boluses.

Some example profiles are [here](https://github.com/nightscout/cgm-remote-monitor/blob/master/docs/plugins/example-profiles.md).