

Database Configuration

Because some users had problems with corrupted database files and broken SD cards the process of logging events and messages to the database was reworked.

Memory caching and disk sync

Logged message and events are not immediately written to disk anymore. Instead the changes are cached in memory and are synced periodically to the disk/SD-Card. This heavily reduces disk writes and so should increase the lifetime of your SD-Card.

There are two new configuration options `deleteExpiredInterval` and `diskSyncInterval` :

Setting	Description	Default
<code>deleteExpiredInterval</code>	interval for deleting expired entries from the database	2h
<code>diskSyncInterval</code>	interval for writing logged entries to disk. If this value is smaller then the <code>deleteExpiredInterval</code> setting then the value of it is used instead. Should be a multiple of <code>deleteExpiredInterval</code> .	4h

So with the default configuration, pimatic checks every 2h the database for expired entries and deletes them (in memory) and each second time (every 4h) the changes are synced to disk (deleted and new entries).

You can customize these settings by editing the config section:

```
{
  "settings": {
    [...]
    "database": {
      "client": "sqlite3",
      "connection": {
        "filename": "pimatic-database.sqlite"
      },
      "deleteExpiredInterval": "2h",
      "diskSyncInterval": "4h",
    },
  },
}
```

```
[...]  
}  
[...]  
}
```

Independent logging interval and expire times

Further you can now configure an independent logging interval for events. This means you can still read sensors every 10 seconds to update the value in the gui and trigger rules, but only log a value every 5 minutes. The logging interval and expire time is configuration in the `deviceAttributeLogging` config option. The default config looks like this:

```
{  
  "settings": {  
    [...]  
    "database": {  
      "client": "sqlite3",  
      "connection": {  
        "filename": "pimatic-database.sqlite"  
      },  
    },  
    [...]  
    "deviceAttributeLogging": [  
      {  
        "deviceId": "*", "attributeName": "*", "type": "*",  
        "interval": "0", "expire": "7d"  
      },  
      {  
        "deviceId": "*", "attributeName": "*", "type": "continuous",  
        "interval": "5min", "expire": "7d"  
      },  
      {  
        "deviceId": "*", "attributeName": "temperature", "type": "number",  
        "expire": "1y"  
      },  
      {
```

Database Configuration

```
    "deviceId": "*", "attributeName": "humidity", "type": "number",
    "expire": "1y"
  }
]
[...]
```

The interval and expire time for a device attribute is calculated by checking it against each entry in the config array. The values of the last matching entry is used for the attribute.

So the first entry in the default config says that all attributes are logged in an interval of 0 seconds (so all values) and are deleted after 7 days from the database, if not other entry matches. But there is an exception for continuous, temperature and humidity attributes. Continuous attributes (measured values by sensors) are only logged in an interval of 5 mins and temperature and humidity values are stored for the time of 1 year.

There is a similar table for logged messages:

```
{
  "settings": {
    [...]
    "database": {
      [...]
      "messageLogging": [
        {
          "level": "*", "tags": [],
          "expire": "7d"
        },
        {
          "level": "debug", "tags": [],
          "expire": "0"
        }
      ]
    }
  }
}
```

All logged message are kept for 7 days by default, but message with debug log level not stored to the database at all. All values with `expire` set to `"0"` are not written to the database at all. This also applies to the `deviceAttributeLogging` config.

Database page in mobile-frontend

There is a new database page in the mobile-frontend. To enable it you have to give yourself the permissions to access it:

```
{
  "roles": [
    {
      "name": "admin",
      "permissions": {
        [...]
        "database": "write",
      }
    },
    [...]
  ]
}
```

At this page all device attributes and there settings are listed. Also there is a list of database problems, that can be fixed right there. If devices are changed or removed the info in the database can get out of sync. This is detected there and can be fixed by automatic deleting old attributes form the database or updating the info for changed ones.

General warning

A general problem with SD-Cards is, that all disks (USB-Stick, SSD, SD-Card) based on NAND Flash Chips have a write cycle amount of 100.000 per cell. After this the cell can probably die. The controller of the medium has around 7% extra storage cells to replace the defect ones. If there are no reserved cells left anymore then the storage produces failures.

So a good idea is to move the whole filesystem or the pimatic-database to a external drive and make backups frequently.

Links:

Database Configuration

- [Raspberry Pi: Extending the life of the SD card](#)
- [Install and run Raspbian from a USB Flash Drive](#)
- [SQL malformed probably SD card died due to much write](#)
- [Forum Topic for Discussions](#)