

# Überblick

Der air-Q hat einen eigenen Webserver, der die Hauptschnittstellen zur direkten Kommunikation bietet. Darauf kann im Browser direkt unter der IP-Adresse des air-Q im Netzwerk zugegriffen werden. Wenn der air-Q als Hotspot läuft, ist diese Webseite unter `http://192.168.4.1` zu erreichen. Darauf können Grundeinstellungen ohne die *air-Q* App und ohne Programmierkenntnisse vorgenommen werden. Unterseiten, wie `http://192.168.4.1/data`, liefern dann die aktuellen Messdaten in verschlüsselter Form. Ist der air-Q in ein WLAN eingebunden, ist er entsprechend unter der ihm vom Router zugewiesenen IP-Adresse zu erreichen. Zudem gibt es eine mDNS-Namensauflösung, die es ermöglicht, den airQ auch dann schnell im Netzwerk zu finden, wenn man dessen IP-Adresse nicht kennt. Dafür sind die ersten fünf Zeichen der Seriennummer nötig. Ein air-Q mit der Seriennummer `12a34b56c7` wäre so im Netzwerk mit `http://12a34_air-q.local` zu erreichen. Die Endung *.local* wird eventuell durch den Router mit einer spezifischen Endung ersetzt, wie z.B. `http://12a34_air-q.fritz.box`. In vielen Fällen kann die Endung auch komplett gespart werden: `http://12a34_air-q`.

Bei direkter Verbindung mit dem air-Q-Hotspot lautet die URL `http://air-q.local`.

## ! Tipp

air-Q speichert die IP-Adresse, die er bekommen hat, auf der SD-Karte in der Datei `IP-address.txt`. Sollten Sie in Ihrem Netzwerk keine mDNS-Funktionalität haben und auch keinen Zugriff auf den Router, können Sie auf diese Art die richtige IP-Adresse in Erfahrung bringen.

## Konfigurations-Router

Für die einfache Konfiguration sehr vieler Geräte, kann ein Konfigurations-Router eingesetzt werden. Dies kann ein beliebiger Router sein, dessen Netzwerkeinstellungen auf diese Werte gesetzt werden:

```
SSID: airq-setup_nomap
Key:  airqsetup
```

air-Qs, die noch nicht konfiguriert wurden – also bei denen die unteren beiden LEDs noch gelb leuchten (keine interne Konfigurationsdatei und keine `userconfig.json` auf SD-Karte) – verbinden sich mit diesem WLAN. Dann können sie einfach über die hier beschriebenen

Webserver-Befehle konfiguriert werden.

Für bereits konfigurierte Geräte (unterste beide LEDs orange, wenn nicht verbunden), kann dies durch drücken des Buttons unten am Gerät erreicht werden. Dazu müssen die LEDs kurz aufleuchten. Dann sucht der air-Q für 10 Minuten das Netzwerk `airq-setup_nomap`.

## **/average**

Gibt einen gleitenden Mittelwert der aktuellen Messdaten als JSON-Objekt zurück mit dem folgenden Aufbau:

```
{
  "content": "***verschlüsselt(Messdatenpaket in JSON)***"
}
```

Der gleitende Mittelwert wird über 30 Datensätze der unmittelbaren Messdaten, welche über die Route `/data` abgerufen werden können, erstellt. Den Mittelwert abzurufen empfiehlt sich, wenn die Route `/data` nicht aller 2 Sekunden abgerufen werden soll. Hier genügt ein Abruf jede Minute, um keinen Wert zu verpassen.

## **/blink**

Die Unterseite `http://192.168.4.1/blink` lässt das Gerät kurz mit allen LEDs in den Regenbogenfarben aufleuchten und gibt die Geräte-Nummer des entsprechenden air-Qs zurück. Das ist besonders dann hilfreich, wenn mehrere air-Qs im Einsatz sind, um einen speziellen zu identifizieren.

```
{
  "id": "***Gerätenummer als String***"
}
```

## **/calibration**

Erlaubt über HTTP POST das setzen neuer Sensor-Kalibrierwerte.

## **/config**

Gibt die aktuellen Geräteeinstellungen als JSON-Objekt zurück mit dem folgenden Aufbau:

```
{
  "content": "***verschlüsselt(Gerätekonfiguration in JSON)***"
}
```

Erlaubt zudem über HTTP POST das Setzen von Geräteeinstellungen.

## /data

Gibt die aktuellen Messdaten als JSON-Objekt zurück mit dem folgenden Aufbau:

```
{  
  "content": "**verschlüsselt(Messdatenpaket in JSON)**"  
}
```

Die Messdaten werden in der Geschwindigkeit aktualisiert, in der der air-Q einen Messdurchlauf schafft. Dies hängt von der Anzahl der Sensoren ab, die abgefragt werden und deren Messdaten vom air-Q aufbereitet werden. Im Regelfall erhält man ca. alle 2 Sekunden ein neues Messwertpaket unter diesem Aufruf.

### ! Tipp

Eine Abfrage alle 1,5 Sekunden führt dazu, dass kein Messwertpaket verpasst wird.

## /dir

Gibt den Inhalt eines Daten-Verzeichnisses auf der SD-Karte wieder. Mit dem Aufruf `http://192.168.4.1/dir?request=**verschlüsselt("2020/6/3")**` erhält man eine Liste der Dateien des Ordners für den 3. Juni 2020. Die Dateinamen sind Unix-UTC-Zeitstempel des ersten Messwertes der Dateien. Die Dateinamen haben keine .csv oder ähnliche Endung.

## /dirbuff

Während für `/dir` der Verzeichnisinhalt der SD-Karte geladen wird, holt `/dirbuff` diesen aus dem Arbeitsspeicher. Statt eines einzelnen Verzeichnisinhalts, wird die komplette Verzeichnisstruktur für die Ordner des laufenden und des vergangenen Monats zurückgegeben.

## /file

Gibt eine Messdaten-Datei zurück, die auf der SD-Karte gespeicherte Mittelwerte enthält.

Der Aufruf `http://192.168.4.1/file?request=**verschlüsselt("2020/6/3/1591176905")**` gibt den Inhalt der Datei `1591176905` im Verzeichnis `/sd/2020/6/3/` zurück. Der Name der Datei wurde zuvor mit `/dirbuff` oder `/dir` ermittelt.

Die Messwerte sind zeilenweise verschlüsselt gespeichert. Der air-Q verschlüsselt jede Zeile beim Befüllen der Datei mit dem aktuell gesetzten Passwort. Die Rückgabe ist damit so strukturiert:

```
**verschlüsselt(Messdatenpaket 1 als JSON)**\n**verschlüsselt(Messdatenpaket 2 als JSON)**\n**verschlüsselt(Messdatenpaket 3 als JSON)**\n**verschlüsselt(Messdatenpaket 4 als JSON)**\n**verschlüsselt(Messdatenpaket 5 als JSON)**\n...
```

## /file\_recrypt

Gibt das gleiche zurück, wie `/file`, jedoch mit dem aktuellen Passwort neu verschlüsselt. Während `/file` die bereits vorverschlüsselten Daten aus dem SD-Kartenverzeichnis des aktuellen Jahres, also im Beispiel `/sd/2020/`, holt, lädt `/file_recrypt` die noch unverschlüsselten Dateien aus dem Verzeichnis des Jahres mit dem Anhang `.uncrypt`, also im Beispiel aus dem Verzeichnis `/sd/2020.uncrypt/`. Diese Abfrage dauert deutlich länger als über `/file`, erlaubt es aber, gespeicherte Messdaten der SD-Karte abzurufen, die mit einem alten, in Vergessenheit geratenen Passwort im verschlüsselten Verzeichnis gespeichert sind.

### ! Tipp

Sollte kein triftiger Grund bestehen `/file_recrypt` zu nutzen, ist `/file` zu bevorzugen.

## /health

Gibt auf GET `200 OK` zurück.

## /log

Gibt die letzten zehn Log-Einträge (falls vorhanden) seit dem Systemstart des air-Q als JSON-Objekt zurück mit dem folgenden Aufbau:

```
{
  "content": "**verschlüsselt(Bis zu zehn Log-Einträge im HTML-Format)**"
}
```

Genutzt zum Debugging.

## /ping

Gibt eine schnelle Kurzzusammenfassung als JSON-Objekt zurück mit dem folgenden Aufbau:

```
{
  "id": "**Gerätenummer als String**",
  "content": "**verschlüsselt(Kurzzusammenfassung als JSON)**"
}
```

Dies ist nur für die schnelle Verarbeitung in der *air-Q* App gedacht.

## /playsound

Gibt dem air-Q per `POST` die Anweisung einen Ton über den eingebauten Pieper-Lautsprecher auszugeben.

Genutzt für die App, um eine Vorschau auf einen zu setzenden Warnton bei Über- oder Unterschreitung eines bestimmten Wertes zu geben.

## /ssid

Gibt eine Liste der WLAN-SSIDs wieder, die der air-Q beim Scannen im Hotspot-Modus findet. Die Liste ist absteigend nach der Signalstärke geordnet. Das Rückgabe-JSON-Objekt hat den folgenden Aufbau:

```
{
  "content": "**verschlüsselt(Liste von gefundenen Netzwerken nach Signalstärke absteigend geordnet)**"
}
```

Genutzt zur Einrichtung des air-Q über die Webseite, die auf dem air-Q läuft, sowie für die air-Q App.

## /standardpass

Gibt *true* oder *false* als String zurück, je nachdem, ob das Standard-Passwort *airqsetup*, welches das Gerät direkt nach der Auslieferung hat, noch gesetzt ist. Dies ist für die *air-Q* App und die Webseite auf dem Gerät relevant.

## **/usercalibration**

Erlaubt über HTTP POST das setzen neuer Sensor-Kalibrierwerte, welche mittels eines Zurücksetzen in der Werkzustand wieder entfernt werden können.

## **/version**

Gibt die API-Version als Zahl zurück.