

---

# Lektion 07: MongoDB einrichten und in Rallye Anwendung integrieren

---

von: Max Holzmann

---

In der Rallye Anwendung wird als Datenbank für die Zeitstempel die MongoDB verwendet.

## Ziele

- Einrichten der Datenbank MongoDB auf dem Raspberry Pi
- Befüllen der Datenbank über Node Red
- Anzeigen von Datenbankeinträgen in einer einfachen Python QT5 GUI

Hinweis: Auf dem Raspberry Pi sind nur MongoDB Server bis zur Version 2.4 verfügbar. Grund hierfür ist, dass ab höheren Versionen MongoDB eine 64 bit Plattform benötigt.

## Voraussetzungen und erforderliches Equipment

- Die Tutorial Lektionen ...
  - 01: Grundlegende Einstellungen und Installationen [link](#)
  - 04: Node Red installieren [link](#)
  - ... müssen erledigt sein
- Hardware
  - 1x Raspberry Pi 3 B

## Lösungsschritte Installation

### 1. Installation von MongoDB über den Paketmanager apt

```
sudo apt update
sudo apt upgrade
sudo apt install mongodb
```

### 2. MongoDB als Service starten

```
sudo systemctl enable mongod
sudo systemctl start mongod
```

### 3. Test und kommandieren der Datenbank durch ...

```
mongo
```

Die MongoDB Konsole sollte starten und die Version ausgeben. Im Falle des Tutorials version 2.4.14.

#### 4. Weitere Tests können sein

- check if the service is activ

```
sudo systemctl status mongod
```

- check the mongoDB connection status by

```
mongo --eval 'db.runCommand({ connectionStatus: 1 })'
```

## Lösungsschritte Datenbank einrichten

### 1. MongoDB Konsole öffnen

```
mongo
```

### 2. Datenbank erstellen (TIPP: eine bereits erstellte Datenbank auswählen, erfolgt über den selben Befehl)

```
use rallyDB
```

### 3. Collection mit auto Indexierung für Durchfahrt erstellen

```
db.createCollection("driveThroughs", { autoIndexID :true })
```

### 4. Test Dokument in Collection einfügen, mit dem Befehl

```
db.driveThroughs.insert({ nr: "1", car: "car_x", time: "12:15:35,12  
2021.01.07", diff: "1234" })
```

### 5. Testen, ob Dokument in Collection eingefügt wurde

```
db.driveThroughs.find()
```

## Datenbank in Node Red einbinden

1. Installieren einer Bibliothek, falls das noch nicht in der Lektion 04: Node red erfolgt ist.
2. Einfügen eines "mongodb2 in" Knoten in den flow.

Einstellungen des mongodb2 nodes:

- Service: External Service
- Server:
- URI: mongodb://localhost:27017/rallyDB
- Name: rallyDB
- Collection: driveThroughs
- Operation: Dynamic(msg.operation)
- Name: rallyDB

Nun sollte es möglich sein msg Objekte an den Node zu senden. Das Objekt muss das Attribut *operation* besitzen. Zum Beispiel 'insert' als DB Kommando, um ein Dokument in die Collection einzufügen. Eingefügt wird die Payload der msg. Wenn das *\_id* Attribut der msg auf *null* gesetzt wird, dann wird automatisch eine neue ID für den Datenbankeintrag beim Einfügen generiert.

In der Lektion 04: Node Red ist der Code für Node Red enthalten sowie eine detaillierte Beschreibung der Node ketten

## Darstellen der Datenbankeinträge in einer QT5 Python Anwendung

Eine einfache Python3 Anwendung stellen die letzten 10 Einträge aus dem Raspberry Pi Display dar. Hierzu wird die Python Bibliothek QT5 und pymongo(version 2.9.5) verwendet. Es ist wichtig keine neuere pymongo version zu verwenden, da sonst die MongoDB 2.4 nicht mehr unterstützt wird.

1. Installation der Binlitheken

```
pip3 install pymongo == pymongo==2.9.5
sudo apt-get install python3-pyqt5
```

2. Das Programm liegt im Ordner [00\\_Code](#).
3. Hier ein Bild der GUI

car	mNbr	Difference	timespamp
car_4	4	-24.882	09:48:35.118
car_4	3	26.361	09:48:26.361
car_4	2	20.275	09:48:20.275
car_4	1	8.433	09:48:08.433
car_3	7	-16.033	09:47:43.967
car_3	6	-17.375	09:47:42.625
car_3	5	-19.397	09:47:40.603
car_3	4	-23.547	09:47:36.453
car_3	3	-25.809	09:47:34.191
car_3	2	-29.432	09:47:30.568

Close

## Quellen

[1] [pimylifeup.com](http://pimylifeup.com)

[2] [www.ionos.de](http://www.ionos.de)