

1. Thông Tin Kết Nối MQTT Broker

Thông số	Giá trị
MQTT Host	<code>mqtt://116.118.95.187:1883</code>
Username	<code>ducchien0612</code>
Password	<code>123456</code>
QoS	<code>1</code>

2. Xe gửi dữ liệu lên MQTT

Publish: `car/{carId}/telemetry`

Dữ liệu gửi lên

```
{
  "orders": [
    {"lat": 1.0, "lng": 2.0, "orderStatus": done},
    {"lat": 3.0, "lng": 4.0, "orderStatus": going},
    {"lat": 5.0, "lng": 6.0, "orderStatus": routing},
    {"lat": 7.0, "lng": 8.0, "orderStatus": routing},
    ...
  ],
  "status": available,
  "charge": 85.4,
  "lat": 9.0,
  "lng": 10.0,
  "speed": 45.2
}
```

- orders:** list các điểm đến và trạng thái. Khi mới khởi động xe hoặc xử lí xong các điểm đến trong 1 chuyến đi thì list này sẽ rỗng.

(lưu ý: autopilot không cần phân biệt giữa điểm đến này là đón khách hay trả khách, thông tin này chỉ nên để high-level (server, app, tablet) xử lí)

- **status**: trạng thái hiện tại của xe / hệ thống autopilot. Server sẽ dựa vào trạng thái này để assign xe / bắt đầu chuyển đi / cảnh báo có sự cố.
- **charge**: phần trăm xăng / pin hiện tại của xe.
- **lat, long**: vị trí hiện tại của xe
- **speed**: vận tốc hiện tại của xe (km/h).

Các trạng thái của orderStatus (trạng thái của điểm đến) gồm:

Enum	Trạng thái	Mô tả chi tiết
0	routing	Xe đã nhận điểm đến, đang tính toán đường đi.
1	routed	Xe đã tính toán đường đi, đang đợi lệnh bắt đầu đi từ server. (ở status routed, xe sẽ đợi action startTrip từ server gửi xuống mới bắt đầu đi (going), để cho thời gian hành khách lên / xuống xe và xác nhận qua điện thoại / tablet)
2	going	Xe đang đi tới điểm đến.
3	stopping	Xe đang dừng lại giữa chuyến đi (dừng vì emergency, không phải vì autopilot có vận tốc bằng 0).
4	done	Xe đã tới điểm đến, đang đợi khách lên / xuống xe.

Các trạng thái của status (trạng thái của xe / hệ thống autopilot) gồm:

Enum	Trạng thái	Ý nghĩa
0	available	Xe sẵn sàng nhận chuyển mới.
1	waitRouting	Autopilot đang tính toán đường đi.
2	waitConfirming	Autopilot đang đợi lệnh bắt đầu đi từ hành khách / server. (ở status waitConfirming, xe sẽ đợi action startTrip từ server gửi xuống mới bắt đầu đi (onRoad), để cho thời gian hành khách lên / xuống xe và xác nhận qua điện thoại / tablet)
3	onRoad	Xe đang thực hiện chuyến đi.
4	charging	Xe đang đi đổ xăng / sạc.
5	emergency	Xe / Autopilot đang gặp sự cố, cần manual / remote intervention.

3. Nhận lệnh

Subscribe: `car/{carId}/assign`

Dữ liệu nhận được

```
{
  "rideId": "637ab12",
  "action": assignRide,
  "assignedAt": "1754449358",
  "orders": [
    {"lat": 1.0, "lng": 2.0},
    {"lat": 3.0, "lng": 4.0},
    {"lat": 5.0, "lng": 6.0},
    {"lat": 7.0, "lng": 8.0},
    ...
  ]
}
```

- **rideId**: text, phía autopilot chỉ dùng để log.
- **action**: lệnh từ server.

(lưu ý: server nên kiểm tra action có bị conflict hay không, ví dụ assignRide thì list điểm đến không rỗng, updateRoute thì chỉ đổi các điểm đến có orderStatus là 0 (chưa thực hiện điểm đến đó)...)

(bởi vì nếu thực hiện check ở autopilot thì có lỗi action phải trao đổi lại với server, tăng phức tạp truyền nhận)

- **assignedAt**: thời gian UTC tính từ năm 1970 (giây).
- **orders**: list các điểm đến.

Các trạng thái của action (lệnh từ server cho autopilot) bao gồm:

Enum	Action	Ý nghĩa
0	assignRide	Giao chuyến cho xe, di chuyển tới điểm đón.
1	cancelRide	Hủy chuyến.
2	updateRoute	Cập nhật lộ trình.
3	startTrip	Khách đã lên / xuống xe, bắt đầu di chuyển.
4	completeRide	Kết thúc chuyến đi.

=====

*****Tài liệu API danh cho ứng dụng tablet*****

URL : <https://dolphin-api.phenikaax.com>

1. Đăng ký Tablet với Vehicle

POST /api/v1/devices/register

Mỗi tablet khi khởi động sẽ gửi deviceId (cấu hình sẵn trên tablet) lên server để map với vehicleId

Body (JSON)

```
{
  "deviceId": "ABC-DEVICE-0001",
  "vehicleId": "6809eca097fb852fb7f306e1"
}
```

2. Lấy danh sách xe (xem khi chưa biết tablet gắn xe nào)

GET /api/v1/vehicles?searchTerm=&page=1&limit=10&order=DESC

3. Lấy đơn (Ride) mới nhất đã phân công cho Vehicle

GET vehicles/6809ab6d6161aea702b10695/assigned-ride

4. Kết nối socket

- Kết nối : `const socket = io(`${SERVER_URL}${NAMESPACE}`, { transports: ['websocket'] });` `SERVER_URL="http://116.118.95.187:4000"` , `NAMESPACE= "/rides"`
- Join Room: `socket.emit('joinVehicle', vehicleId);`
- Lắng nghe sự kiện: `ride.assigned.toVehicle`

No labels

