

Vehicle Tracker API Documentation

Base URL: <http://localhost:5000>

API Version: 1.0

Last Updated: December 2025

Table of Contents

1. [Drivers](#)
 2. [Vehicles](#)
 3. [Live Tracking \(Telemetry\)](#)
 4. [Location History](#)
 5. [Incidents](#)
 6. [Incident Types](#)
 7. [Incident Statuses](#)
 8. [Vehicle Statuses](#)
-

Drivers

Manage driver records.

Get All Drivers

```
GET /api/drivers
```

Response:

```
[  
  {  
    "driverId": 1,  
    "firstName": "John",  
    "lastName": "Doe",  
    "licenseNumber": "DL123456",  
    "phoneNumber": "+966501234567"  
  },  
  {  
    "driverId": 2,  
    "firstName": "Jane",  
    "lastName": "Smith",  
    "licenseNumber": "DL789012",  
    "phoneNumber": "+966509876543"  
  }  
]
```

Get Driver by ID

```
GET /api/drivers/{id}
```

Example: GET /api/drivers/1

Response:

```
{  
  "driverId": 1,  
  "firstName": "John",  
  "lastName": "Doe",  
  "licenseNumber": "DL123456",  
  "phoneNumber": "+966501234567"  
}
```

Create Driver

```
POST /api/drivers  
Content-Type: application/json
```

Request Body:

```
{  
  "firstName": "Ahmed",  
  "lastName": "Al-Hassan",  
  "licenseNumber": "DL555555",  
  "phoneNumber": "+966505555555"  
}
```

Response: 201 Created

```
{  
  "driverId": 3,  
  "firstName": "Ahmed",  
  "lastName": "Al-Hassan",  
  "licenseNumber": "DL555555",  
  "phoneNumber": "+966505555555"  
}
```

Update Driver

```
PUT /api/drivers/{id}
Content-Type: application/json
```

Request Body:

```
{
  "driverId": 1,
  "firstName": "John",
  "lastName": "Doe",
  "licenseNumber": "DL123456",
  "phoneNumber": "+966501111111"
}
```

Response: 204 No Content

Delete Driver

```
DELETE /api/drivers/{id}
```

Response: 204 No Content

Vehicles

Manage vehicle records and driver assignments.

Get All Vehicles

```
GET /api/vehicles
```

Response:

```
[
  {
    "vehicleId": 1,
    "driverId": 1,
    "plateNumber": "ABC-1234",
    "make": "Toyota",
    "model": "Camry",
    "color": "White"
  },
  {
    "vehicleId": 2,
    "driverId": null,
  }
]
```

```
"plateNumber": "XYZ-5678",
"make": "Honda",
"model": "Civic",
"color": "Blue"
}
]
```

Get Vehicle by ID

```
GET /api/vehicles/{id}
```

Example: GET /api/vehicles/1

Response:

```
{
  "vehicleId": 1,
  "driverId": 1,
  "plateNumber": "ABC-1234",
  "make": "Toyota",
  "model": "Camry",
  "color": "White"
}
```

Get Unassigned Vehicles

Returns vehicles that are not assigned to any driver (`driverId` is null).

```
GET /api/vehicles/unassigned
```

Response:

```
[
  {
    "vehicleId": 2,
    "driverId": null,
    "plateNumber": "XYZ-5678",
    "make": "Honda",
    "model": "Civic",
    "color": "Blue"
  }
]
```

Create Vehicle

```
POST /api/vehicles  
Content-Type: application/json
```

Request Body (with driver):

```
{  
    "driverId": 1,  
    "plateNumber": "NEW-0001",  
    "make": "Nissan",  
    "model": "Altima",  
    "color": "Silver"  
}
```

Request Body (unassigned):

```
{  
    "plateNumber": "NEW-0002",  
    "make": "Ford",  
    "model": "Focus",  
    "color": "Red"  
}
```

Response: 201 Created

```
{  
    "vehicleId": 3,  
    "driverId": null,  
    "plateNumber": "NEW-0002",  
    "make": "Ford",  
    "model": "Focus",  
    "color": "Red"  
}
```

Update Vehicle

```
PUT /api/vehicles/{id}  
Content-Type: application/json
```

Request Body:

```
{  
  "vehicleId": 1,  
  "driverId": 1,  
  "plateNumber": "ABC-1234",  
  "make": "Toyota",  
  "model": "Camry",  
  "color": "Black"  
}
```

Response: 204 No Content

Assign Driver to Vehicle

Assigns or unassigns a driver to/from a vehicle.

```
PATCH /api/vehicles/{id}/assign  
Content-Type: application/json
```

Request Body (assign driver):

```
{  
  "driverId": 2  
}
```

Request Body (unassign driver):

```
{  
  "driverId": null  
}
```

Response: 200 OK

```
{  
  "vehicleId": 1,  
  "driverId": 2,  
  "plateNumber": "ABC-1234",  
  "make": "Toyota",  
  "model": "Camry",  
  "color": "White"  
}
```

Delete Vehicle

```
DELETE /api/vehicles/{id}
```

Response: 204 No Content

Live Tracking (Telemetry)

Real-time vehicle location and telemetry data.

Send Telemetry Data

Updates vehicle's current location and logs to history.

```
POST /api/livetracking
```

```
Content-Type: application/json
```

Request Body:

```
{
  "vehicleId": 1,
  "latitude": 24.7136,
  "longitude": 46.6753,
  "speed": 65.5,
  "fuel": 75.0,
  "vehicleStatusId": "550e8400-e29b-41d4-a716-446655440000"
}
```

Field	Type	Required	Description
vehicleId	int	✓	Vehicle ID
latitude	double	✓	GPS latitude
longitude	double	✓	GPS longitude
speed	decimal	✗	Speed in km/h
fuel	decimal	✗	Fuel level (0-100%)
vehicleStatusId	GUID	✗	Current vehicle status

Response: 200 OK

```
{
  "vehicleId": 1,
  "vehicleStatusId": "550e8400-e29b-41d4-a716-446655440000",
  "latitude": 24.7136,
```

```
"longitude": 46.6753,  
"currentSpeed": 65.5,  
"currentFuel": 75.0  
}
```

Get All Vehicle Locations

Returns current location of all vehicles with telemetry data.

```
GET /api/livetracking
```

Response:

```
[  
  {  
    "vehicleId": 1,  
    "vehicleStatusId": "550e8400-e29b-41d4-a716-446655440000",  
    "latitude": 24.7136,  
    "longitude": 46.6753,  
    "currentSpeed": 65.5,  
    "currentFuel": 75.0  
  },  
  {  
    "vehicleId": 2,  
    "vehicleStatusId": null,  
    "latitude": 24.6877,  
    "longitude": 46.7219,  
    "currentSpeed": 0,  
    "currentFuel": 90.0  
  }  
]
```

Get Vehicle Location

Returns current location of a specific vehicle.

```
GET /api/livetracking/{vehicleId}
```

Example: GET /api/livetracking/1

Response: 200 OK

```
{  
  "vehicleId": 1,  
  "vehicleStatusId": "550e8400-e29b-41d4-a716-446655440000",  
  "
```

```
"latitude": 24.7136,  
"longitude": 46.6753,  
"currentSpeed": 65.5,  
"currentFuel": 75.0  
}
```

Response (not found): 404 Not Found

```
{  
  "message": "Vehicle not found or no live data available"  
}
```

Location History

Historical location trails for vehicles.

Get Today's Trail

Returns all location points recorded today for a vehicle.

```
GET /api/history/vehicles/{vehicleId}/today
```

Example: GET /api/history/vehicles/1/today

Response:

```
{  
  "vehicleId": 1,  
  "date": "2025-12-08T00:00:00Z",  
  "pointCount": 5,  
  "trail": [  
    {  
      "timestamp": "2025-12-08T08:30:00Z",  
      "latitude": 24.7136,  
      "longitude": 46.6753,  
      "speed": 0,  
      "fuel": 100.0  
    },  
    {  
      "timestamp": "2025-12-08T08:45:00Z",  
      "latitude": 24.7200,  
      "longitude": 46.6800,  
      "speed": 45.5,  
      "fuel": 98.0  
    },  
    {  
      "timestamp": "2025-12-08T09:00:00Z",  
      "latitude": 24.7250,  
      "longitude": 46.6850,  
      "speed": 40.0,  
      "fuel": 95.0  
    }  
  ]  
}
```

```
        "timestamp": "2025-12-08T09:00:00Z",
        "latitude": 24.7300,
        "longitude": 46.6900,
        "speed": 60.0,
        "fuel": 95.0
    }
]
}
```

Get Trail by Date Range

Returns location points within a specified date/time range.

```
GET /api/history/vehicles/{vehicleId}/range?from={datetime}&to={datetime}
```

Example:

```
GET /api/history/vehicles/1/range?from=2025-12-01T00:00:00Z&to=2025-12-08T23:59:59Z
```

Query Parameters:

Parameter	Type	Required	Description
from	datetime	<input checked="" type="checkbox"/>	Start of range (ISO 8601)
to	datetime	<input checked="" type="checkbox"/>	End of range (ISO 8601)

Response:

```
{
  "vehicleId": 1,
  "from": "2025-12-01T00:00:00Z",
  "to": "2025-12-08T23:59:59Z",
  "pointCount": 150,
  "trail": [
    {
      "timestamp": "2025-12-01T08:30:00Z",
      "latitude": 24.7136,
      "longitude": 46.6753,
      "speed": 0,
      "fuel": 100.0
    }
  ]
}
```

Error Response: 400 Bad Request

```
{  
  "message": "'from' must be before 'to'"  
}
```

Incidents

Manage incidents and assign vehicles.

Get All Incidents

```
GET /api/incidents  
GET /api/incidents?statusId={statusId}
```

Query Parameters:

Parameter	Type	Required	Description
statusId	int	x	Filter by incident status

Response:

```
[  
  {  
    "incidentId": 1,  
    "type": "Accident",  
    "status": "Pending",  
    "latitude": 24.7500,  
    "longitude": 46.7000,  
    "assignedVehicleId": null,  
    "assignedVehiclePlate": null  
  },  
  {  
    "incidentId": 2,  
    "type": "Breakdown",  
    "status": "In Progress",  
    "latitude": 24.6900,  
    "longitude": 46.6500,  
    "assignedVehicleId": 1,  
    "assignedVehiclePlate": "ABC-1234"  
  }  
]
```

Get Incident by ID

```
GET /api/incidents/{id}
```

Response:

```
{  
  "incidentId": 1,  
  "type": "Accident",  
  "status": "Pending",  
  "latitude": 24.7500,  
  "longitude": 46.7000,  
  "assignedVehicleId": null,  
  "assignedVehiclePlate": null  
}
```

Report New Incident

```
POST /api/incidents  
Content-Type: application/json
```

Request Body:

```
{  
  "incidentTypeId": 1,  
  "incidentStatusId": 1,  
  "latitude": 24.7500,  
  "longitude": 46.7000  
}
```

Response: 201 Created

```
{  
  "incidentId": 3,  
  "message": "Incident reported successfully"  
}
```

Get Nearest Vehicles to Incident

Finds vehicles closest to the incident location (for dispatch).

```
GET /api/incidents/{id}/nearest-vehicles  
GET /api/incidents/{id}/nearest-vehicles?limit={count}
```

Query Parameters:

Parameter	Type	Default	Description
limit	int	5	Number of vehicles to return

Example: GET /api/incidents/1/nearest-vehicles?limit=3

Response:

```
[  
  {  
    "vehicleId": 2,  
    "plateNumber": "XYZ-5678",  
    "make": "Honda",  
    "model": "Civic",  
    "latitude": 24.7400,  
    "longitude": 46.6900,  
    "distanceMeters": 1234.5  
  },  
  {  
    "vehicleId": 1,  
    "plateNumber": "ABC-1234",  
    "make": "Toyota",  
    "model": "Camry",  
    "latitude": 24.7136,  
    "longitude": 46.6753,  
    "distanceMeters": 4567.8  
  }  
]
```

Assign Incident to Vehicle

```
POST /api/incidents/{id}/assign  
Content-Type: application/json
```

Request Body:

```
{  
  "vehicleId": 1,  
  "statusId": 2  
}
```

Field	Type	Required	Description
-------	------	----------	-------------

Field	Type	Required	Description
vehicleId	int	✓	Vehicle to assign
statusId	int	✗	Update incident status

Response:

```
{  
  "message": "Incident assigned successfully",  
  "incidentId": 1,  
  "assignedVehicleId": 1,  
  "assignedVehiclePlate": "ABC-1234",  
  "notificationSent": true  
}
```

Incident Types

Manage types of incidents (e.g., Accident, Breakdown, Emergency).

Get All Incident Types

```
GET /api/incidenttypes
```

Response:

```
[  
  {  
    "incidentTypeId": 1,  
    "typeName": "Accident"  
  },  
  {  
    "incidentTypeId": 2,  
    "typeName": "Breakdown"  
  },  
  {  
    "incidentTypeId": 3,  
    "typeName": "Emergency"  
  }]
```

Get Incident Type by ID

```
GET /api/incidenttypes/{id}
```

Create Incident Type

```
POST /api/incidenttypes  
Content-Type: application/json
```

Request Body:

```
{  
  "typeName": "Traffic Violation"  
}
```

Response: 201 Created

```
{  
  "incidentTypeId": 4,  
  "typeName": "Traffic Violation"  
}
```

Update Incident Type

```
PUT /api/incidenttypes/{id}  
Content-Type: application/json
```

Request Body:

```
{  
  "incidentTypeId": 1,  
  "typeName": "Vehicle Accident"  
}
```

Response: 204 No Content

Delete Incident Type

```
DELETE /api/incidenttypes/{id}
```

Response: 204 No Content

Incident Statuses

Manage incident statuses (e.g., Pending, In Progress, Resolved).

Get All Incident Statuses

```
GET /api/incidentstatuses
```

Response:

```
[  
  {  
    "incidentStatusId": 1,  
    "statusName": "Pending"  
  },  
  {  
    "incidentStatusId": 2,  
    "statusName": "In Progress"  
  },  
  {  
    "incidentStatusId": 3,  
    "statusName": "Resolved"  
  }]
```

Get Incident Status by ID

```
GET /api/incidentstatuses/{id}
```

Create Incident Status

```
POST /api/incidentstatuses  
Content-Type: application/json
```

Request Body:

```
{  
  "statusName": "Cancelled"  
}
```

Response: 201 Created

Update Incident Status

```
PUT /api/incidentstatuses/{id}  
Content-Type: application/json
```

Delete Incident Status

```
DELETE /api/incidentstatuses/{id}
```

Vehicle Statuses

Manage vehicle statuses (e.g., Available, On Route, Maintenance).

Get All Vehicle Statuses

```
GET /api/vehiclestatuses
```

Response:

```
[  
  {  
    "vehicleStatusId": "550e8400-e29b-41d4-a716-446655440000",  
    "status": "Available"  
  },  
  {  
    "vehicleStatusId": "550e8400-e29b-41d4-a716-446655440001",  
    "status": "On Route"  
  },  
  {  
    "vehicleStatusId": "550e8400-e29b-41d4-a716-446655440002",  
    "status": "Maintenance"  
  }  
]
```

Get Vehicle Status by ID

```
GET /api/vehiclestatuses/{id}
```

Example: GET /api/vehiclestatuses/550e8400-e29b-41d4-a716-446655440000

Create Vehicle Status

```
POST /api/vehiclestatuses  
Content-Type: application/json
```

Request Body:

```
{  
    "status": "Off Duty"  
}
```

Response: 201 Created

```
{  
    "vehicleStatusId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",  
    "status": "Off Duty"  
}
```

Update Vehicle Status

```
PUT /api/vehiclestatuses/{id}  
Content-Type: application/json
```

Request Body:

```
{  
    "vehicleStatusId": "550e8400-e29b-41d4-a716-446655440000",  
    "status": "Ready"  
}
```

Response: 204 No Content

Delete Vehicle Status

```
DELETE /api/vehiclestatuses/{id}
```

Response: 204 No Content

Error Responses

All endpoints may return the following error responses:

400 Bad Request

```
{  
  "message": "Validation error description"  
}
```

404 Not Found

```
{  
  "message": "Resource not found"  
}
```

500 Internal Server Error

```
{  
  "message": "An error occurred while processing your request"  
}
```

CORS

The API allows cross-origin requests from any origin:

- `Access-Control-Allow-Origin: *`
 - `Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE`
 - `Access-Control-Allow-Headers: *`
-

Testing with cURL

Create a driver

```
curl -X POST http://localhost:5000/api/drivers \  
  -H "Content-Type: application/json" \  
  -d  
'{"firstName":"Test","lastName":"Driver","licenseNumber":"TEST123","phoneNumber":  
 "+966500000000"}'
```

Create an unassigned vehicle

```
curl -X POST http://localhost:5000/api/vehicles \  
  -H "Content-Type: application/json" \  
  /
```

```
-d '{"plateNumber":"TEST-001","make":"Toyota","model":"Corolla","color":"White"}'
```

Assign a driver to a vehicle

```
curl -X PATCH http://localhost:5000/api/vehicles/1/assign \
-H "Content-Type: application/json" \
-d '{"driverId":1}'
```

Send telemetry data

```
curl -X POST http://localhost:5000/api/livetracking \
-H "Content-Type: application/json" \
-d
'{"vehicleId":1,"latitude":24.7136,"longitude":46.6753,"speed":50.0,"fuel":80.0
}'
```

Get today's vehicle trail

```
curl http://localhost:5000/api/history/vehicles/1/today
```

Report an incident

```
curl -X POST http://localhost:5000/api/incidents \
-H "Content-Type: application/json" \
-d
'{"incidentTypeId":1,"incidentStatusId":1,"latitude":24.75,"longitude":46.70}'
```

Find nearest vehicles to incident

```
curl http://localhost:5000/api/incidents/1/nearest-vehicles?limit=3
```