

**Dokumentasi Layanan API**

**Ujian Tengah Semester (UTS)**

Dosen Pengampu:  
**MARDI SISWO UTOMO, S.KOM, M.Cs**



Nama : Muhammad Falah Abdurrafi  
NIM : 20.01.53.0033  
Mata Kuliah / Kelas : Web Service / A1

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI DAN INDUSTRI**  
**UNIVERSITAS STIKUBANK**

**2022**

# API DOCUMENTATION

## ParkingManS (Parking Management System)

ParkingManS is a system to manage parking transactions (booking and payments) for a company that will make their customer's life easier by just processing the booking of parking through the online platform.

### BASE URL (Server)

```
http://falahrafi-webservice.my.id/api/v1
```

### VEHICLE OWNER

---

#### Find all vehicle owners

GET	/vehicle_owners
-----	-----------------

#### Response :

```
{
  "data": [
    {
      "id": 1,
      "username": "alia69",
      "password": "$2y$10$7EM0oro5Y3G3Rz8lePj5TuI6sJXsXrq/q4vhEeIfMo3/MMqU1iFOa",
      "fullname": "Brycen Block III",
      "avatar": "images/avatar/716549.jpg",
      "contact": "087929488672",
      "email": "hane.brooklyn@hotmail.com"
    },
    {
      "id": 2,
      "username": "flatley.amie",
      "password": "$2y$10$vSs3i0PYHKtmTpq/14PQeezpxUbRlir085mY.YKeHRsr7jLmuyU0.",
      "fullname": "Noe Kessler",
      "avatar": "images/avatar/154383.jpg",
      "contact": "087221788168",
      "email": "vlueilwitz@gmail.com"
    }
  ]
}
```

---

### Find vehicle owner by ID

GET     /vehicle\_owners/{id}

- {id} is the ID of the vehicle owner that you want to find. For example: **1**

### Response :

```
{
  "data": {
    "id": 1,
    "username": "alia69",
    "password": "$2y$10$7EM0oro5Y3G3Rz8lePj5TuI6sJXsXrq/q4vhEeIfMo3/MMqU1iFOa",
    "fullname": "Brycen Block III",
    "avatar": "images/avatar/716549.jpg",
    "contact": "087929488672",
    "email": "hane.brooklyn@hotmail.com"
  }
}
```

---

### Filtering vehicle owners

GET     /vehicle\_owners?username[eq]=flatley.amie

- You can replace **username** with other fields, such as **fullname**, **email**, etc.
- You can replace **flatley.amie** with the value that you want to find.
- Supported operators:
  - id : [eq]
  - username : [eq], [s]
  - password : -
  - fullname : [eq], [s]
  - avatar : -
  - contact : [eq], [s]
  - email : [eq], [s]
- Operator map:
  - [eq] : equal (=)
  - [s] : search (LIKE)

### Response :

```
{
  "data": [
    {
      "id": 2,
      "username": "flatley.amie",
      "password": "$2y$10$vSs3i0PYHKtmTpq/14PQeezpxUbRlir085mY.YKeHRsr7jLmuyU0.",
      "fullname": "Noe Kessler",
      "avatar": "images/avatar/154383.jpg",
      "contact": "087221788168",
      "email": "vluailwitz@gmail.com"
    }
  ]
}
```

---

### Add a new vehicle owner

**POST**    **/vehicle\_owners**

### Request body :

```
{
  "username": "mfalahabdurrafi",
  "password": "c0Ba123#",
  "fullname": "Muhammad Falah Abdurrafi",
  "avatar": "/images/avatar/321654.jpg",
  "contact": "085112233445",
  "email": "mfalahabdurrafi@gmail.com"
}
```

### Response :

```
3
```

- It will return the value of the primary key of the newly added vehicle owner.

---

### Update some field of a vehicle owner

**PATCH**    **/vehicle\_owners/{id}**

- **{id}** is the ID of the vehicle owner that you want to update. For example: **3**

#### Request body :

```
{
  "email": "mfalahabdurrafi@yahoo.com"
}
```

#### Response :

1

- This adjusts the email of the vehicle owner. And the return value is the number of rows that are set.
- 

#### Update all fields of a vehicle owner

**PUT**     **/vehicle\_owners/{id}**

- **{id}** is the ID of the vehicle owner that you want to update. For example: **3**

#### Request body :

```
{
  "username": "michaelmorgan",
  "password": "c0Ba321#",
  "fullname": "Michael Morgan",
  "avatar": "/images/avatar/654321.jpg",
  "contact": "08111111111",
  "email": "michaelmorgan@protonmail.com"
}
```

#### Response :

1

- This adjusts all fields of the vehicle owner. And the return value is the number of rows that are set.
- 

#### Delete a vehicle owner

**DELETE**     **/vehicle\_owners/{id}**

- **{id}** is the ID of the vehicle owner that you want to delete. For example: **1**

Response :

```
1
```

- It will return the number of deleted rows.
- 

## VEHICLE CATEGORIES

---

Find all vehicle categories

GET /vehicle\_categories

Response :

```
{
  "data": [
    {
      "id": 1,
      "categoryName": "Car"
    },
    {
      "id": 2,
      "categoryName": "Motorcycle"
    },
    {
      "id": 3,
      "categoryName": "Bicycle"
    },
    {
      "id": 4,
      "categoryName": "Truck"
    },
    {
      "id": 5,
      "categoryName": "Bus"
    }
  ]
}
```

---

Find vehicle category by ID

GET /vehicle\_categories/{id}

- **{id}** is the ID of the vehicle category that you want to find. For example: **1**

**Response :**

```
{
  "data": {
    "id": 1,
    "categoryName": "Car"
  }
}
```

---

## VEHICLES

---

### Find all vehicles

GET	/vehicles
-----	-----------

**Response :**

```
{
  "data": [
    {
      "id": 1,
      "vehicleCategoryId": 2,
      "plateNumber": "P 1139 RVO",
      "description": "Et provident aut eos asperiores hic molestiae aspernatur.",
      "image": "images/vehicles/642335.jpg",
      "vehicleOwnerId": 1
    },
    {
      "id": 2,
      "vehicleCategoryId": 1,
      "plateNumber": "K 7026 EFU",
      "description": "Nihil nostrum ea qui a.",
      "image": "images/vehicles/673242.jpg",
      "vehicleOwnerId": 2
    }
  ]
}
```

---

### Find vehicle by ID

GET    /vehicles/{id}

- {id} is the ID of the vehicle that you want to find. For example: **1**

Response :

```
{
  "data": {
    "id": 1,
    "vehicleCategoryId": 2,
    "plateNumber": "P 1139 RVO",
    "description": "Et provident aut eos asperiores hic molestiae aspernatur.",
    "image": "images/vehicles/642335.jpg",
    "vehicleOwnerId": 1
  }
}
```

---

### Filtering vehicle

GET    /vehicles?vehicleCategoryId[eq]=1

- You can replace **vehicleCategoryId** with other fields, such as **plateNumber**, **vehicleOwnerId**, etc.
- You can replace **1** with the value that you want to find.
- Supported operators:
  - id : [eq]
  - vehicleCategoryId : [eq]
  - plateNumber : [eq], [s]
  - description : [eq], [s]
  - image : -
  - vehicleOwnerId : [eq]
- Operator map:
  - [eq] : equal (=)
  - [s] : search (LIKE)

Response :

```
{
```



```
"data": [  
  {  
    "id": 2,  
    "vehicleCategoryId": 1,  
    "plateNumber": "K 7026 EFU",  
    "description": "Nihil nostrum ea qui a.",  
    "image": "images/vehicles/673242.jpg",  
    "vehicleOwnerId": 2  
  }  
]
```

---

### Add a new vehicle

**POST**   **/vehicles**

#### Request body :

```
{  
  "vehicleCategoryId": 1,  
  "plateNumber": "B 9919 XYZ",  
  "description": "Toyota Avanza, warna putih",  
  "image": "/images/vehicles/876241.jpg",  
  "vehicleOwnerId": 3  
}
```

#### Response :

```
3
```

- It will return the value of the primary key of the newly added vehicle.

---

### Update some field of a vehicle

**PATCH**   **/vehicles/{id}**

- **{id}** is the ID of the vehicle that you want to update. For example: **1**

#### Request body :

```
{  
  "plateNumber": "AD 1234 EFG"
```

```
}
```

#### Response :

```
1
```

- This adjusts the plate number of the vehicle. And the return value is the number of rows that are set.
- 

#### Update all fields of a vehicle

**PUT**    `/vehicles/{id}`

- `{id}` is the ID of the vehicle that you want to update. For example: **3**

#### Request body :

```
{
  "vehicleCategoryId": 2,
  "plateNumber": "B 9919 XYZ",
  "description": "Honda Vario 125",
  "image": "/images/vehicles/876241.jpg",
  "vehicleOwnerId": 3
}
```

#### Response :

```
1
```

- This adjusts all fields of the vehicle. And the return value is the number of rows that are set.
- 

#### Delete a vehicle

**DELETE**    `/vehicles/{id}`

- `{id}` is the ID of the vehicle that you want to delete. For example: **1**

#### Response :

```
1
```

- It will return the number of deleted rows.

## PARKING SLOT STATUSES

---

### Find all parking slot statuses

GET    /parking\_slot\_statuses

Response :

```
{
  "data": [
    {
      "id": 0,
      "statusName": "Available"
    },
    {
      "id": 1,
      "statusName": "Occupied"
    },
    {
      "id": 2,
      "statusName": "Reserved"
    }
  ]
}
```

---

### Find parking slot status by ID

GET    /parking\_slot\_statuses/{id}

- **{id}** is the ID of the parking slot status that you want to find. For example: **1**

Response :

```
{
  "data": {
    "id": 1,
    "statusName": "Occupied"
  }
}
```

---

## PARKING SLOTS

---

### Find all parking slots

GET /parking\_slots

#### Response :

```
{
  "data": [
    {
      "id": 1,
      "slotNumber": 1,
      "statusId": 2
    },
    {
      "id": 2,
      "slotNumber": 2,
      "statusId": 0
    }
  ]
}
```

---

### Find parking slot by ID

GET /parking\_slots/{id}

- **{id}** is the ID of the parking slot that you want to find. For example: **1**

#### Response :

```
{
  "data": {
    "id": 1,
    "slotNumber": 1,
    "statusId": 2
  }
}
```

---

### Filtering parking slot

GET /parking\_slots?slotNumber[eq]=1

- You can replace **slotNumber** with other fields, such as **statusId**.
- You can replace **1** with the value that you want to find.
- Supported operators:
  - id : [eq]
  - slotNumber : [eq]
  - statusId : [eq]
- Operator map:
  - [eq] : equal (=)
  - [s] : search (LIKE)

Response :

```
{
  "data": [
    {
      "id": 1,
      "slotNumber": 1,
      "statusId": 2
    }
  ]
}
```

### Add a new parking slot

POST /parking\_slots

Request body :

```
{
  "slotNumber": 3,
  "statusId": 0
}
```

Response :

3

- It will return the value of the primary key of the newly added parking slot.

---

### Update some field of a parking slot

**PATCH**    /parking\_slots/{id}

- {id} is the ID of the parking slot that you want to update. For example: **1**

#### **Request body :**

```
{  
  "statusId": 1  
}
```

#### **Response :**

```
1
```

- This adjusts the status ID of the parking slot. And the return value is the number of rows that are set.

---

### Update all fields of a parking slot

**PUT**    /parking\_slots/{id}

- {id} is the ID of the parking slot that you want to update. For example: **3**

#### **Request body :**

```
{  
  "slotNumber": 4,  
  "statusId": 2  
}
```

#### **Response :**

```
1
```

- This adjusts all fields of the parking slot. And the return value is the number of rows that are set.
-

### Delete a parking slot

**DELETE** /parking\_slots/{id}

- {id} is the ID of the parking slot that you want to delete. For example: 1

**Response :**

```
1
```

- It will return the number of deleted rows.
- 

## BOOKING STATUSES

---

### Find all booking statuses

**GET** /booking\_statuses

**Response :**

```
{
  "data": [
    {
      "id": 0,
      "statusName": "Processing"
    },
    {
      "id": 1,
      "statusName": "Approved"
    },
    {
      "id": 2,
      "statusName": "Rejected"
    },
    {
      "id": 3,
      "statusName": "Cancelled"
    },
    {
      "id": 4,
      "statusName": "Completed"
    }
  ]
}
```

```
}
```

---

### Find booking status by ID

GET    /booking\_statuses/{id}

- {id} is the ID of the booking status that you want to find. For example: 1

### Response :

```
{
  "data": {
    "id": 1,
    "statusName": "Approved"
  }
}
```

---

## BOOKINGS

---

### Find all bookings

GET    /bookings

### Response :

```
{
  "data": [
    {
      "id": 1,
      "vehicleOwnerId": 1,
      "vehicleId": 1,
      "parkingSlotId": 1,
      "duration": 5,
      "statusId": 4
    },
    {
      "id": 2,
      "vehicleOwnerId": 2,
      "vehicleId": 2,

```



```
    "parkingSlotId": 2,  
    "duration": 48,  
    "statusId": 1  
  }  
]  
}
```

---

### Find booking by ID

GET    /bookings/{id}

- {id} is the ID of the booking that you want to find. For example: **1**

### **Response :**

```
{  
  "data": {  
    "id": 1,  
    "vehicleOwnerId": 1,  
    "vehicleId": 1,  
    "parkingSlotId": 1,  
    "duration": 5,  
    "statusId": 4  
  }  
}
```

---

### Filtering booking

GET    /bookings?duration[gt]=24

- You can replace **duration** with other fields, such as **vehicleOwnerId**, **statusId**, etc.
- You can replace **24** with the value that you want to find.
- Supported operators:
  - id : [eq]
  - vehicleOwnerId : [eq]
  - vehicleId : [eq]
  - parkingSlotId : [eq]
  - duration : [eq], [lt], [lte], [gt], [gte]
  - statusId : [eq]

- Operators map:
  - **[eq]** : equal (=)
  - **[lt]** : lower than (<)
  - **[lte]** : lower than or equal (<=)
  - **[gt]** : greater than (>)
  - **[gte]** : greater than or equal (>=)

**Response :**

```
{
  "data": [
    {
      "id": 2,
      "vehicleOwnerId": 2,
      "vehicleId": 2,
      "parkingSlotId": 2,
      "duration": 48,
      "statusId": 1
    }
  ]
}
```

**Add a new booking**

**POST** /bookings

**Request body :**

```
{
  "vehicleOwnerId": 3,
  "vehicleId": 3,
  "parkingSlotId": 3,
  "duration": 72,
  "statusId": 0
}
```

**Response :**

3

- It will return the value of the primary key of the newly added booking.

### Update some field of a booking

**PATCH**    **/bookings/{id}**

- **{id}** is the ID of the booking that you want to update. For example: **1**

#### **Request body :**

```
{  
  "statusId": 1  
}
```

#### **Response :**

1

- This adjusts the status ID of the booking. And the return value is the number of rows that are set.
- 

### Update all fields of a booking

**PUT**    **/bookings/{id}**

- **{id}** is the ID of the booking that you want to update. For example: **3**

#### **Request body :**

```
{  
  "vehicleOwnerId": 3,  
  "vehicleId": 3,  
  "parkingSlotId": 3,  
  "duration": 24,  
  "statusId": 1  
}
```

#### **Response :**

1

- This adjusts all fields of the booking. And the return value is the number of rows that are set.
-

## Delete a booking

**DELETE** /bookings/{id}

- {id} is the ID of the booking that you want to delete. For example: 1

**Response :**

1

- It will return the number of deleted rows.
- 

## **PAYMENT STATUSES**

---

### Find all payment statuses

**GET** /payment\_statuses

**Response :**

```
{
  "data": [
    {
      "id": 0,
      "statusName": "Pending"
    },
    {
      "id": 1,
      "statusName": "Completed"
    },
    {
      "id": 2,
      "statusName": "Expired"
    },
    {
      "id": 3,
      "statusName": "Cancelled"
    }
  ]
}
```

---

### Find payment status by ID

GET /payment\_statuses/{id}

- {id} is the ID of the payment status that you want to find. For example: 1

Response :

```
{
  "data": {
    "id": 1,
    "statusName": "Completed"
  }
}
```

---

## PAYMENTS

### Find all payments

GET /payments

Response :

```
{
  "data": [
    {
      "id": 1,
      "bookingId": 1,
      "amount": 10000.00,
      "method": "Bank Transfers",
      "statusId": 0,
      "notes": "Rem qui reiciendis qui autem beatae sed ex.",
      "paidBy": "John Doe",
      "paidDate": "2022-10-28 00:00:00"
    },
    {
      "id": 2,
      "bookingId": 2,
      "amount": 50000.00,
      "method": "Credit Card",
      "statusId": 0,
      "notes": "Ipsum necessitatibus cum fugiat facere.",
      "paidBy": "William Morgan",

```

```
    "paidDate": "2022-10-28 00:00:00"
  }
[
}
```

---

### Find payment by ID

**GET**    **/payments/{id}**

- **{id}** is the ID of the payment that you want to find. For example: **1**

### **Response :**

```
{
  "data": {
    "id": 1,
    "bookingId": 1,
    "amount": 10000.00,
    "method": "Bank Transfers",
    "statusId": 0,
    "notes": "Rem qui reiciendis qui autem beatae sed ex.",
    "paidBy": "John Doe",
    "paidDate": "2022-10-28 00:00:00"
  }
}
```

---

### Filtering payment

**GET**    **/payments?amount[gt]=20000**

- You can replace **amount** with other fields, such as **method**, **statusId**, etc.
- You can replace **20000** with the value that you want to find.
- Supported operators:
  - id : [eq]
  - bookingId : [eq]
  - amount : [eq], [lt], [lte], [gt], [gte]
  - method : [eq], [s]
  - statusId : [eq]
  - notes : [eq], [s]
  - paidBy : [eq], [s]

- paidDate : [eq], [lt], [lte], [gt], [gte]
- Operators map:
  - [eq] : equal (=)
  - [lt] : lower than (<)
  - [lte] : lower than or equal (<=)
  - [gt] : greater than (>)
  - [gte] : greater than or equal (>=)

**Response :**

```
{
  "data": [
    {
      "id": 2,
      "bookingId": 2,
      "amount": 50000.00,
      "method": "Credit Card",
      "statusId": 0,
      "notes": "Ipsum necessitatibus cum fugiat facere.",
      "paidBy": "William Morgan",
      "paidDate": "2022-10-28 00:00:00"
    }
  ]
}
```

**Add a new payment**

**POST** /payments

**Request body :**

```
{
  "bookingId": 3,
  "amount": 75000.00,
  "method": "Credit Card",
  "statusId": 0,
  "notes": "Lorem ipsum dolor sit amet consectetur.",
  "paidBy": "Michael Smith",
  "paidDate": "2022-10-29 00:00:00"
}
```

**Response :**

3

- It will return the value of the primary key of the newly added payment.
- 

### Update some field of a payment

**PATCH**    `/bookings/{id}`

- `{id}` is the ID of the payment that you want to update. For example: **3**

#### Request body :

```
{
  "statusId": 1
}
```

#### Response :

1

- This adjusts the status ID of the payment. And the return value is the number of rows that are set.
- 

### Update all fields of a payment

**PUT**    `/payments/{id}`

- `{id}` is the ID of the payment that you want to update. For example: **3**

#### Request body :

```
{
  "bookingId": 3,
  "amount": 75000.00,
  "method": "Bank Transfers",
  "statusId": 2,
  "notes": "Lorem ipsum dolor sit amet consectetur.",
  "paidBy": "Michael Smith",
  "paidDate": "2022-10-29 00:00:00"
}
```

#### Response :

1



- This adjusts all fields of the payment. And the return value is the number of rows that are set.

---

### Delete a payment

**DELETE** /payments/{id}

- {id} is the ID of the payment that you want to delete. For example: 1

### Response :

1

- It will return the number of deleted rows.

---

## USERS

### Find all users

**GET** /users

### Response :

```
{
  "data": [
    {
      "id": 1,
      "username": "admin_x",
      "password":
"$2y$10$KLi3VLZQ9Ug2.qDZ/N9KRuviqFIzs509JnEFq4Ha72r1ut643f284",
      "fullname": "Admin X",
      "avatar": "/images/avatar/654321.jpg",
      "contact": "085123456789",
      "email": "adminx@gmail.com",
      "user_level": 2
    },
    {
      "id": 2,
      "username": "admin_y",
      "password":
```

```
"$2y$10$KHu2GCZQ9Ug2.qDZ/Y4TYaviqFIzs509JnEFq4Ha72r1ut643e672",
  "fullname": "Admin X",
  "avatar": "/images/avatar/543210.jpg",
  "contact": "085987654321",
  "email": "adminy@gmail.com",
  "user_level": 2
}
[
}
}
```

---

### Find user by ID

GET	/users/{id}
-----	-------------

- **{id}** is the ID of the user that you want to find. For example: **1**

### Response :

```
{
  "data": {
    "id": 1,
    "username": "admin_x",
    "password": "$2y$10$KLi3VLZQ9Ug2.qDZ/N9KRuviqFIzs509JnEFq4Ha72r1ut643f284",
    "fullname": "Admin X",
    "avatar": "/images/avatar/654321.jpg",
    "contact": "085123456789",
    "email": "adminx@gmail.com",
    "user_level": 2
  }
}
```

---

### Filtering user

GET	/users?username[eq]=admin_y
-----	-----------------------------

- You can replace **username** with other fields, such as **fullname**, **email**, etc.
- You can replace **admin\_y** with the value that you want to find.
- Supported operators:
  - id : **[eq]**
  - username : **[eq], [s]**
  - password : -

- fullname : [eq], [s]
  - avatar : -
  - contact : [eq], [s]
  - email : [eq], [s]
  - user\_level : [eq]
- Operator map:
    - [eq] : equal (=)
    - [s] : search (LIKE)

**Response :**

```
{
  "data": [
    {
      "id": 2,
      "username": "admin_y",
      "password": "$2y$10$KHu2GCZQ9Ug2.qDZ/Y4TYaviqFIs509JnEFq4Ha72r1ut643e672",
      "fullname": "Admin X",
      "avatar": "/images/avatar/543210.jpg",
      "contact": "085987654321",
      "email": "adminy@gmail.com",
      "user_level": 2
    }
  ]
}
```

**Add a new user**

**POST** /users

**Request body :**

```
{
  "username": "admin_falah",
  "password": "raHasia123$",
  "fullname": "Admin Falah",
  "avatar": "/images/avatar/543210.jpg",
  "contact": "085111222333",
  "email": "adminfalah@gmail.com",
  "user_level": 2
}
```

**Response :**

3

- It will return the value of the primary key of the newly added user.
- 

#### Update some field of a user

**PATCH**    `/users/{id}`

- `{id}` is the ID of the user that you want to update. For example: **3**

#### **Request body :**

```
{
  "email": "adminfalah@yahoo.com"
}
```

#### **Response :**

1

- This adjusts the email of the user. And the return value is the number of rows that are set.
- 

#### Update all fields of a user

**PUT**    `/users/{id}`

- `{id}` is the ID of the user that you want to update. For example: **3**

#### **Request body :**

```
{
  "username": "admin_falah",
  "password": "raHasia123$",
  "fullname": "Admin Falah",
  "avatar": "/images/avatar/543210.jpg",
  "contact": "085333222111",
  "email": "adminfalah@protonmail.com",
  "user_level": 2
}
```

#### **Response :**

1

- This adjusts all fields of the user. And the return value is the number of rows that are set.
- 

#### Delete a user

**DELETE** /users/{id}

- **{id}** is the ID of the user that you want to delete. For example: **1**

#### **Response :**

1

- It will return the number of deleted rows.
-