

Certainly, let's provide the complete JSON objects for both the request and response bodies for each endpoint:

To create a ride:

Method: POST

URL: /rides

Request Body:

json

Copy code

```
{
  "pickup_location": "pickup_location_value",
  "destination": "destination_value",
  "price": "price_value",
  "duration": "duration_value",
  "distance": "distance_value",
  "payment": {
    "status": "Pending"
  }
}
```

Response Codes:

201 Created (on successful creation)

400 Bad Request (if the request is invalid)

Response Body:

json

Copy code

```
{
  "id": "ride_id_value",
  "pickup_location": "pickup_location_value",
  "destination": "destination_value",
  "price": "price_value",
}
```

```
"duration": "duration_value",  
"distance": "distance_value",  
"status": "Called",  
"payment": {  
  "id": "payment_id_value",  
  "status": "Pending"  
}  
}
```

To edit a ride:

Method: PUT

URL: /rides/{ride_id}

Request Body:

json

Copy code

```
{  
  "pickup_location": "updated_pickup_location_value",  
  "destination": "updated_destination_value",  
  "price": "updated_price_value",  
  "duration": "updated_duration_value",  
  "distance": "updated_distance_value"  
}
```

Response Codes:

200 OK (on successful update)

400 Bad Request (if the request is invalid)

404 Not Found (if the ride with the specified ID does not exist or if the ride status is not "Called" for clients)

Response Body:

json

Copy code

```
{
```

```
"id": "ride_id_value",
"pickup_location": "updated_pickup_location_value",
"destination": "updated_destination_value",
"price": "updated_price_value",
"duration": "updated_duration_value",
"distance": "updated_distance_value",
"status": "Called",
"payment": {
  "id": "payment_id_value",
  "status": "Pending"
}
}
```

To list the rides of a user:

Method: GET

URL: /users/{user_id}/rides

Request Body: None

Response Codes:

200 OK (on successful retrieval)

404 Not Found (if the user does not exist)

Response Body:

json

Copy code

```
[
{
  "id": "ride_id_1",
  "pickup_location": "pickup_location_1",
  "destination": "destination_1",
  "price": "price_1",
  "duration": "duration_1",
  "distance": "distance_1",
```

```
"status": "Called",
"payment": {
  "id": "payment_id_1",
  "status": "Pending"
},
{
  "id": "ride_id_2",
  "pickup_location": "pickup_location_2",
  "destination": "destination_2",
  "price": "price_2",
  "duration": "duration_2",
  "distance": "distance_2",
  "status": "Finished",
  "payment": {
    "id": "payment_id_2",
    "status": "Paid"
  }
}
// More ride objects if applicable
]
```

To list the details of a ride:

Method: GET

URL: /rides/{ride_id}

Request Body: None

Response Codes:

200 OK (on successful retrieval)

404 Not Found (if the ride with the specified ID does not exist)

Response Body:

json

Copy code

```
{
  "id": "ride_id_value",
  "pickup_location": "pickup_location_value",
  "destination": "destination_value",
  "price": "price_value",
  "duration": "duration_value",
  "distance": "distance_value",
  "status": "Called",
  "payment": {
    "id": "payment_id_value",
    "status": "Pending"
  }
}
```

To list the rides of a driver:

Method: GET

URL: /drivers/{driver_id}/rides

Request Body: None

Response Codes:

200 OK (on successful retrieval)

404 Not Found (if the driver does not exist)

Response Body:

json

Copy code

```
[
  {
    "id": "ride_id_1",
    "pickup_location": "pickup_location_1",
    "destination": "destination_1",
    "price": "price_1",
```

```
"duration": "duration_1",
"distance": "distance_1",
"status": "Started",
"payment": {
  "id": "payment_id_1",
  "status": "Paid"
}
},
{
  "id": "ride_id_2",
  "pickup_location": "pickup_location_2",
  "destination": "destination_2",
  "price": "price_2",
  "duration": "duration_2",
  "distance": "distance_2",
  "status": "Finished",
  "payment": {
    "id": "payment_id_2",
    "status": "Paid"
  }
}
// More ride objects if applicable
]
```

These JSON objects provide a complete representation of both the request and response bodies for each endpoint, including all the attributes specified in the initial question.