User Login

URL: /api/auth/users/login

Method: POST

Parameters:

1. username
2. password

Result:

If Success:

json array of user

If Unsuccess:

Status Code: 403(incorrect login details)

Validate Session

URL: /api/auth/users/{userId}/validate-session

Method: GET

Headers: auth_token

Result:

If Success:

Status Code: 200

If Unsuccess:

Status Code: 401(session invalid)

Status Code: 404(user not found)

Send password reset request (this will send a mail to given email address with pw reset token. also that token only valid for 24 hours since sent)

URL: /api/auth/users/password-reset-request

Method: POST

Parameters:

1. email(required,email)

Result:

If Success:

Status Code: 200

If Unsuccess:

Status Code: 404(there are no user with that email)

Status Code: 400(validation errors.errors will show on console)

Password Reset

URL: /api/auth/users/password-reset

Method: POST

Parameters:

1. password(required,min:5)
2. password_confirmation(required,min:5)
3. token(required)

Result:

If Success:

Status Code: 200

If Unsuccess:

Status Code: 404(there are no user for combined with that token)

Status Code: 400(validation errors.errors will show on console)

List all trips

URL: /api/trips

Method: GET

Parameters:

1. start(required,numeric)
2. count(required,numeric)

Headers: auth_token

Result:

If Success:

-trips list on json format

-also it receive header called ‘App-Content-Full-Count’. It has full count of trips.able to use for pagination.

If Unsuccess:

Status Code: 400(validation errors.errors will show on console)

Create a new trip

URL: /api/trips

Method: POST

Parameters:

1. start_date(required,date)
2. start_time(required)
3. start_latitude(required,min:0.000001,max:999.999999)
4. start_longitude(required,min:0.000001,max:999.999999)
5. start_odometer(required,min:0.001,max:999999.999)

Headers: auth_token

Result:

If Success:

Trip object on json format

If Unsuccess:

Status Code: 400(validation errors.errors will show on console)

Update a trip

URL: /api/trips/{id}

Method: PUT

Parameters:

1. start_date(date)
2. start_time
3. start_latitude(min:0.000001,max:999.999999)
4. start_longitude(min:0.000001,max:999.999999)
5. start_odometer(min:0.001,max:999999.999)
6. end_date(date)
7. end_time
8. end_latitude(min:0.000001,max:999.999999)
9. end_longitude(min:0.000001,max:999.999999)
10. end_odometer(min:0.001,max:999999.999)
11. client_name
12. purpose

Headers: auth_token

Result:

If Success:

updated trip object on json format

If Unsuccess:

Status Code: 404(there are no trip found)

Status Code: 400(validation errors.errors will show on console)

Delete a trip

URL: /api/trips/{id}

Method:DELETE

Headers: auth_token

Result:

If Success:

Status Code: 200

If Unsuccess:

Status Code: 404(there are no trip found)

View single trip

URL: /api/trips/{id}

Method: GET

Headers: auth_token

Result:

If Success:

json array of trip.

If Unsuccess:

Status Code: 404(there are no trip found)