GREEN DRIVE

Table of Contents

Green Drive consists of 6 EndPoint:

- login into Ecosystem
- Register New Company
- Upload Company Fleet
- Retrieve Company Emission
- Retrieve Employee Emission
- Retrieve Replaceable Vehicles

1. Login

At very first step you need to login to ecosystem and get your authorization token.

1.1. Address

```
POST /api/getToken
```

1.2. HTTP headers

Name	Description
`Content-Type`	Must be `application/json`

1.3. Request Body

Name	**Type**	**Required**	**Description**
`email`	`String`	Yes	The field represents the user email.
`password`	`String`	Yes	The field represents user password.

1.4. Response fields

Name	**Type**	**Description**
`token`	`String`	This field is user token for accessing APIs.
`result.status`	`Integer`	Numeric code of the result.
`result.message`	`String`	A descriptive message for the result.

Name	Type	Description
`result.level`	`String`	Result level.

1.5. HTTP request example

```
curl --location 'http://localhost:8090/api/getToken'
--header 'Content-Type: application/json'
--header 'Accept-Encoding: UTF8'
--data-raw '{
    "email": "test@gmail.com",
    "password": "Test@123123"
}'
```

1.6. HTTP response example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
    "result": {
        "title": "SUCCESS",
        "status": 0,
        "message": "success",
        "level": "INFO"
    },
    "token":
"eyJhbGciOiJIUzI1NiJ9.eyJtYW5hZ2VtZW50OnJlYWQiOiJtYW5hZ2VtZW50OnJlYWQiLCJtYW5hZ2VtZW50
OmRlbGV0ZSI6Im1hbmFnZW1lbnQ6ZGVsZXRlIiwiYWRtaW46dXBkYXRlIjoiYWRtaW46dXBkYXRlIiwibWFuYW
dlbWVudDpjcmVhdGUiOiJtYW5hZ2VtZW50OmNyZWF0ZSIsImFkbWluOnJlYWQiOiJhZG1pbjpyZWFkIiwibWFu
YWdlbWVudDp1cGRhdGUiOiJtYW5hZ2VtZW50OnVwZGF0ZSIsIlJPTEVfQURNSU4iOiJST0xFX0FETUlOIiwiYW
RtaW46ZGVsZXRlIjoiYWRtaW46ZGVsZXRlIiwiYWRtaW46Y3JlYXRlIjoiYWRtaW46Y3JlYXRlIiwic3ViIjoi
dGVzdEBnbWFpbC5jb20iLCJpYXQiOjE2OTUxOTgxNDEsImV4cCI6MTY5NTI4NDU0MX0.vESqrz2inlltdXiLve
hgAPvJ_JBnUIUna_pIRT3tJdM"
}
```

1.7. Address

```
POST /company/register
```

1.8. HTTP headers

Name	Description
`Authorization`	Auth access token issued to bearer

Name	Description
`Content-Type`	Must be `application/json;charset=UTF-8`

1.9. Request Body

Name	Type	Required	Description
`name`	`String`	Yes	The field represents the Company Name.

1.10. Response fields

Name	Type	Description
`name`	`String`	This field is Company Name.
`companyCode`	`String`	This field represents company uniqId.
`result.status`	`Integer`	Numeric code of the result.
`result.message`	`String`	A descriptive message for the result.
`result.level`	`String`	Result level.

1.11. HTTP request example

```
curl --location 'localhost:8090/company/register'
--header 'Content-Type: application/json'
--header 'Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJtYW5hZ2VtZW50OnJlYWQiOiJtYW5hZ2VtZW50OnJlYWQiLCJtYW5hZ2VtZW50O
mRlbGV0ZSI6Im1hbmFnZW1lbnQ6ZGVsZXRlIiwiYWRtaW46dXBkYXRlIjoiYWRtaW46dXBkYXRlIiwibWFuYWd
lbWVudDpjcmVhdGUiOiJtYW5hZ2VtZW50OmNyZWF0ZSIsImFkbWluOnJlYWQiOiJhZG1pbjpyZWFkIiwibWFuY
WdlbWVudDp1cGRhdGUiOiJtYW5hZ2VtZW50OnVwZGF0ZSIsIlJPTEVfQURNSU4iOiJST0xFX0FETUlOIiwiYWR
taW46ZGVsZXRlIjoiYWRtaW46ZGVsZXRlIiwiYWRtaW46Y3JlYXRlIjoiYWRtaW46Y3JlYXRlIiwic3ViIjoid
GVzdEBnbWFpbC5jb20iLCJpYXQiOjE2OTUxOTgxNDEsImV4cCI6MTY5NTI4NDU0MX0.vESqrz2inlltdXiLveh
gAPvJ_JBnUIUna_pIRT3tJdM'
--data '{
    "name":"test-name"
}'
```

1.12. HTTP response example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
    "result": {
        "title": "SUCCESS",
        "status": 0,
        "message": "success",
        "level": "INFO"
    },
    "companyCode": "1695198740040-7691",
    "name": "test-name"
}
```

2. Upload Company Fleet

This endpoint receives a valid csv file and after validation store them in database.

2.1. Address

```
POST /company/upload
```

2.2. Query parameters

Name	Description
`companyCode`	this code generates in previous step.

2.3. HTTP headers

Name	Description
`Authorization`	Auth access token issued to bearer
`Content-Type`	Must be `application/json;charset=UTF-8`

2.4. Request Body

Name	Type	Required	Description
`file`	`form-data(File)`	Yes	The file must be a valid csv.

2.5. Response fields

Name	Type	Description
`message`	`String`	This field represents message of upload result.
`result.status`	`Integer`	Numeric code of the result.
`result.message`	`String`	A descriptive message for the result.
`result.level`	`String`	result level.

2.6. HTTP request example

```
curl --location 'http://localhost:8090/company/upload?companyCode=1695198740040-7691'
\
--header 'Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJtYW5hZ2VtZW50OnJlYWQiOiJtYW5hZ2VtZW50OnJlYWQiLCJtYW5hZ2VtZW50O
mRlbGV0ZSI6Im1hbmFnZW1lbnQ6ZGVsZXRlIiwiYWRtaW46dXBkYXRlIjoiYWRtaW46dXBkYXRlIiwibWFuYWd
lbWVudDpjcmVhdGUiOiJtYW5hZ2VtZW50OmNyZWF0ZSIsImFkbWluOnJlYWQiOiJhZG1pbjpyZWFkIiwibWFuY
WdlbWVudDp1cGRhdGUiOiJtYW5hZ2VtZW50OnVwZGF0ZSIsIlJPTEVfQURNSU4iOiJST0xFX0FETUlOIiwiYWR
taW46ZGVsZXRlIjoiYWRtaW46ZGVsZXRlIiwiYWRtaW46Y3JlYXRlIjoiYWRtaW46Y3JlYXRlIiwic3ViIjoid
GVzdEBnbWFpbC5jb20iLCJpYXQiOjE2OTUxOTgxNDEsImV4cCI6MTY5NTI4NDU0MX0.vESqrz2inlltdXiLveh
gAPvJ_JBnUIUna_pIRT3tJdM' \
--form 'file=@"./valid_company_fleet.csv"'
```

2.7. HTTP response example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
    "result": {
        "title": "SUCCESS",
        "status": 0,
        "message": "success",
        "level": "INFO"
    },
    "message": "Uploaded the file successfully: valid_company_fleet.csv"
}
```

3. Company Emission

Getting company fleet emissions

3.1. Address

```
GET  /company/emission/{companyCode}
```

3.2. Request path variable

Name	Type	Description
`companyCode`	`String`	The field represents company uniqId that generate in register time.

3.3. Response fields

Name	Type	Description
`result.status`	`Integer`	Numeric code of the result.
`result.message`	`String`	A descriptive message for the result.
`result.level`	`String`	result level.
`vehicleType`	`String`	This field represents vehicleType.
`averageEmission`	`String`	This field represents average emission of vehicleType from last week.
`totalMileages`	`Long`	This field represents total mileages of that specific vehicleType from last week.
`totalEmissions`	`Long`	This field represents total emissions of vehicleType from last week.
`totalVehicles`	`Long`	This field represents total vehicles of specific vehicleType.

3.4. HTTP request example

```
curl --location 'http://localhost:8090/company/emissions/1695200555482-0989' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJtYW5hZ2VtZW50OnJlYWQiOiJtYW5hZ2VtZW50OnJlYWQiLCJtYW5hZ2VtZW50O
mRlbGV0ZSI6Im1hbmFnZW1lbnQ6ZGVsZXRlIiwiYWRtaW46dXBkYXRlIjoiYWRtaW46dXBkYXRlIiwibWFuYWd
lbWVudDpjcmVhdGUiOiJtYW5hZ2VtZW50OmNyZWF0ZSIsImFkbWluOnJlYWQiOiJhZG1pbjpyZWFkIiwibWFuY
WdlbWVudDp1cGRhdGUiOiJtYW5hZ2VtZW50OnVwZGF0ZSIsIlJPTEVfQURNSU4iOiJST0xFX0FETUlOIiwiYWR
taW46ZGVsZXRlIjoiYWRtaW46ZGVsZXRlIiwiYWRtaW46Y3JlYXRlIjoiYWRtaW46Y3JlYXRlIiwic3ViIjoid
GVzdEBnbWFpbC5jb20iLCJpYXQiOjE2OTUyMDA0MjIsImV4cCI6MTY5NTI4NjgyMn0.JHKuLuRtTwggvgfKdr_
4WXmZgOoMFemEBFsRoNCrBbE'
```

3.5. HTTP response example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
    "result": {
        "title": "SUCCESS",
        "status": 0,
        "message": "success",
        "level": "INFO"
    },
    "vehicleAverageEmissions": [
        {
            "vehicleType": "DUMP_TRUCK",
            "averageEmission": 40000,
            "totalMileages": 300,
            "totalEmissions": 120000,
            "totalVehicles": 3
        },
        {
            "vehicleType": "AMBULANCE",
            "averageEmission": 22000,
            "totalMileages": 900,
            "totalEmissions": 198000,
            "totalVehicles": 9
        }
    ]
}
```

4. Employee Emission

Getting specific employee fleet emissions

4.1. Address

```
GET  /vehicle/emissions/{employeeId}
```

4.2. Request path variable

Name	Type	Description
`employeeId`	`String`	The field represents employeeId.

4.3. Response fields

Name	Type	Description
`result.status`	`Integer`	Numeric code of the result.
`result.message`	`String`	A descriptive message for the result.
`result.level`	`String`	result level.
`vehicleType`	`String`	This field represents vehicleType.
`averageEmission`	`String`	This field represents average emission of vehicleType from last week.
`totalMileages`	`Long`	This field represents total mileages of that specific vehicleType from last week.
`totalEmissions`	`Long`	This field represents total emissions of vehicleType from last week.
`totalVehicles`	`Long`	This field represents total vehicles of specific vehicleType.

4.4. HTTP request example

```
curl --location 'http://localhost:8090/api/vehicle/emissions/E1' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJtYW5hZ2VtZW50OnJlYWQiOiJtYW5hZ2VtZW50OnJlYWQiLCJtYW5hZ2VtZW50O
mRlbGV0ZSI6Im1hbmFnZW1lbnQ6ZGVsZXRlIiwiYWRtaW46dXBkYXRlIjoiYWRtaW46dXBkYXRlIiwibWFuYWd
lbWVudDpjcmVhdGUiOiJtYW5hZ2VtZW50OmNyZWF0ZSIsImFkbWluOnJlYWQiOiJhZG1pbjpyZWFkIiwibWFuY
WdlbWVudDp1cGRhdGUiOiJtYW5hZ2VtZW50OnVwZGF0ZSIsIlJPTEVfQURNSU4iOiJST0xFX0FETUlOIiwiYWR
taW46ZGVsZXRlIjoiYWRtaW46ZGVsZXRlIiwiYWRtaW46Y3JlYXRlIjoiYWRtaW46Y3JlYXRlIiwic3ViIjoid
GVzdEBnbWFpbC5jb20iLCJpYXQiOjE2OTUyMDA0MjIsImV4cCI6MTY5NTI4NjgyMn0.JHKuLuRtTwggvgfKdr_
4WXmZgOoMFemEBFsRoNCrBbE'
```

4.5. HTTP response example

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
    "result": {
        "title": "SUCCESS",
        "status": 0,
        "message": "success",
        "level": "INFO"
    },
    "vehicleEmissionsReports": [
        {
            "vehicleType": "DUMP_TRUCK",
            "averageEmission": 40000,
            "totalMileages": 300,
            "totalEmissions": 120000,
            "totalVehicles": 3
        },
        {
            "vehicleType": "AMBULANCE",
            "averageEmission": 22000,
            "totalMileages": 900,
            "totalEmissions": 198000,
            "totalVehicles": 9
        }
    ]
}
```

5. Vehicle Suggestion

This endpoint represents this vehicle that would be replaced with Electric Vehicles

5.1. Address

```
GET /vehicle/suggestion
```

5.2. HTTP headers

Name	Description
`Authorization`	Auth access token issued to bearer
`Content-Type`	Must be `application/json;charset=UTF-8`

5.3. Response fields

Name	Type	Description
`vehicleCode`	`String`	This field represents specific id.
`employeeId`	`String`	This field represents employeeId.
`mileage`	`Long`	This field represents vehicle mileage.
`vehicleType`	`String`	This field represents vehicle Type.
`result.status`	`Integer`	Numeric code of the result.
`result.message`	`String`	A descriptive message for the result.
`result.level`	`String`	result level.

6. HTTP response codes

NOTE We use conventional HTTP response codes to indicate the success or failure of an API request.

- HTTP code 200 indicates success.
- HTTP code 400 indicates invalid params.
- HTTP code 401 and 403 indicates unauthenticated user and unauthorized access respectively.
- HTTP code 422 (Unprocessable Entity) indicates business errors.
- HTTP code 500 indicates internal errors.

7. Version history

Version	Date	Author	Description
1.0	09/20/2023	Mehdi Qanbarzade	