Getting Started

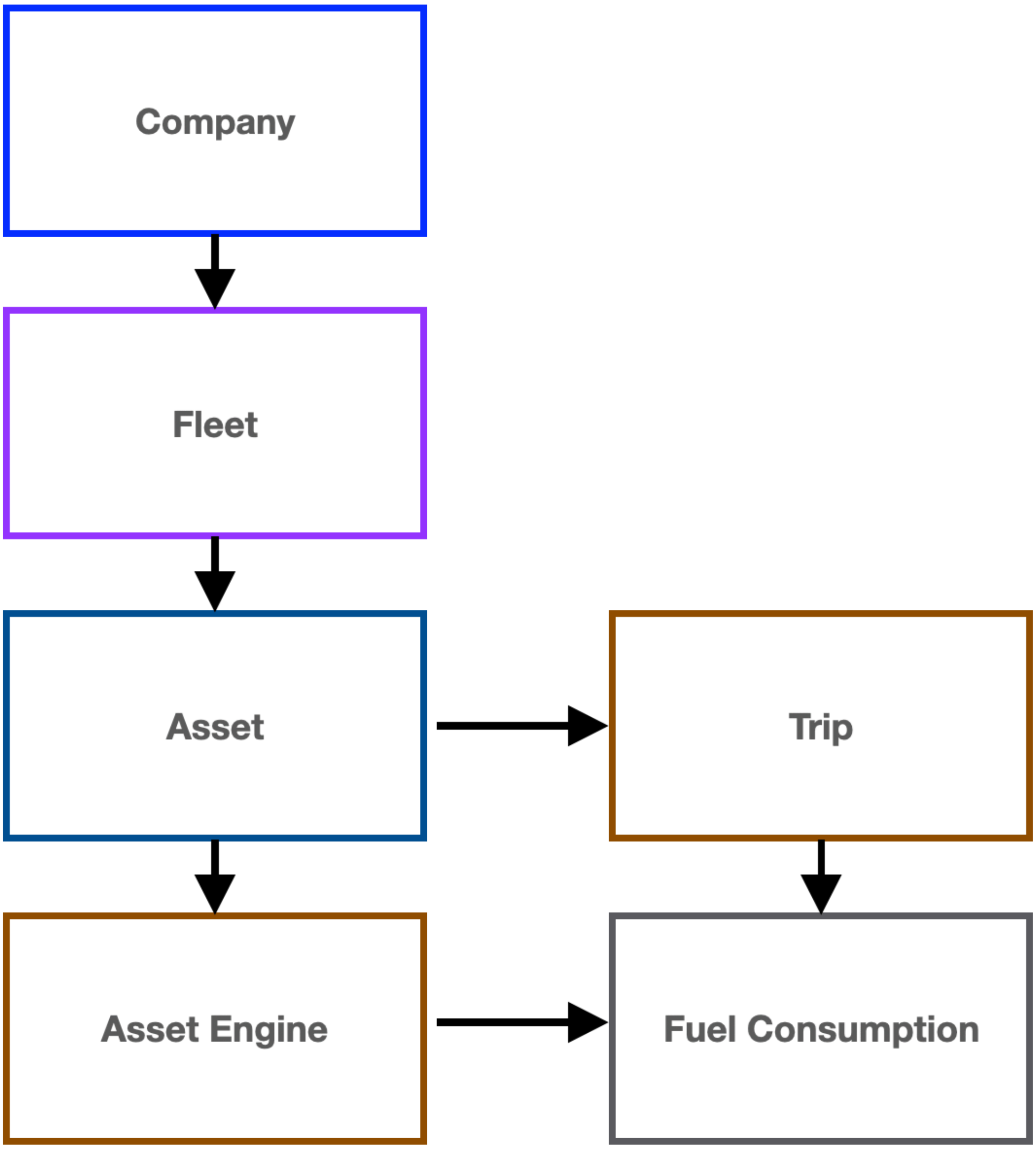
Introduction

This introduction is primary focused on Logistics. We do, however, support many other industries. If you would like to know how to integrate you industry, please feel free to contact us.

The EQM platform performs most of calculations based on a collection of Client assets. These assets can be a truck, generator, or any other object that consumes fuel, uses refrigerants, produces waste, or consumes electricity. Assets also have a variety of attributes that help calculations.

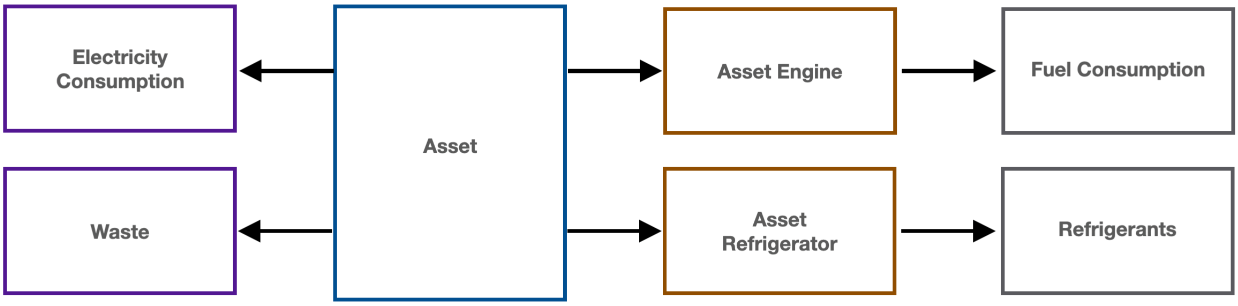
Assets are nested into asset groups. An asset can belong to only one asset group. Asset groups belong to a single company.

In order to assist in calculations, assets have asset engine. Asset engines define the size and characteristics of an asset's engine. Assets can have multiple engines. Some engines can also consume multiple types of fuel (such as marine engines that can use heavy fuel oil or marine diesel oil).



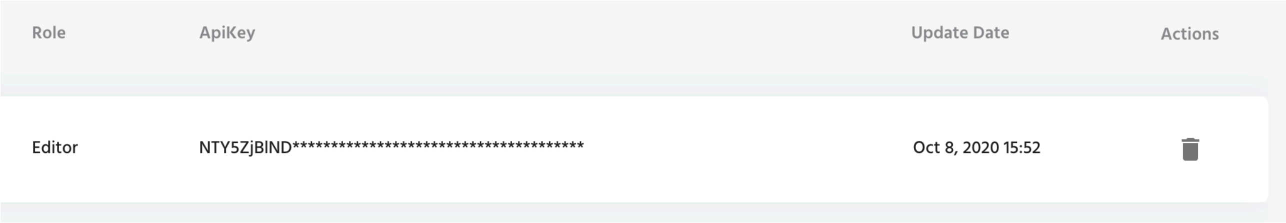
As the above image shows, asset’s can also have trips. Trips can have (but not required to have) fuel consumptions associated with it. If a trip is designated to it will calculate the fuel consumption with the liters per kilometre that the asset’s vehicle type provides. This method of calculation is less accurate when compared to directly measuring fuel consumption.

Assets also produce waste, consume electricity, and use refrigerants. These also cause pollution.



Authorization

The API relies on an API Key. You can define an API Key in your accounts. Each API Key is associated with a permission level. Permission levels for the API are defined as Reader and Editor. A Reader can access data. An Editor can access, create, and modify data.



Role	ApiKey	Update Date	Actions
Editor	NTY5ZjBlND***	Oct 8, 2020 15:52	🗑

In order to use your API Key, place it in the header of each request with

Authorization: X-API-Key {Your API Key}

Now that we have a basic understanding of assets and Authentication, let's look at the APIs that focus on assets.

Getting Started

Creating a Fleet

To quickly get up and running, let's use the API to create an asset group (a.k.a. fleet) and asset, so that we can assign some data to it.

The first step is to create a fleet. Send a POST to /fleet with the API key in your header, and the following body:

{"companyId":{Your Company Id}, "name":"New Fleet", "description":"My new Fleet"}
In order to execute this, let's make a POST with the Auth in the header, and the content in the body:

```
curl -d '{"companyId":{Your Company Id}, "name":"New Fleet", "description":"My new
Fleet"}' -H 'Content-Type: application/json' -H 'Authorization: X-API-Key {Your API Key}' -i
https://api.eqm.world/fleet
```

Be sure to replace {Your Company Id} and {Your API Key} with your provided company ID and API key, respectively.

Once executed, this should respond with a 200 response code and the new Fleet object you have created! If you receive a 401 response code, make sure your company ID and API key are correct.

Take note of the "id" returned. It should be a number. Keep this id on hand as we will need it for the next part.

If you missed the fleet id previously, you can fetch all fleets again using another curl post. It requires your company Id as a parameter

```
curl -H 'Authorization: X-API-Key {Your API Key}' -i
https://api.eqm.world/fleet?companyId={Your Company Id}
```

Creating an Asset

Now that we have created a Fleet, let's create a new asset. Assets are the central object in EQM's calculations. One of the main properties is an asset's engine. An asset can contain multiple engines (for example, a ship with a main and auxiliary engines). In this example, we will create a truck. Since the truck only has a single engine, we will use a helper API to create the truck and asset engine is a single call.

```
curl -d '{"fleetId":{Your Fleet Id},"name":"New Asset", "description":"New Asset I've
made","industryId":1,"fuelTypeId":4, "assetTypeId":2,"countryId":204}' -H 'Content-Type:
application/json' -H 'Authorization: X-API-Key {Your API Key}' -i
https://api.eqm.world/asset/with-engine
```

Let's discuss some of the parameters here. industryId is a parameter for you to define the asset, a VIN number, license plate number, or whatever you want. fuelTypeId is the form of fuel. 4 is Diesel. assetTypeId defines the type of asset. 2 is a light delivery truck. countryId defines the country. 204 is the United States. Each of these ids are an enumeration and can be found in our documentation.

Once you have created an asset, take note of the "id" returned. We can look up your asset with the returned "id".

```
curl -H 'Authorization: X-API-Key {Your API Key}' -i https://api.eqm.world/asset/{Your Asset
Id}?companyId={Your Company Id}
```

Here we can see that we have successfully hooked up your asset to a country, asset type, and fleet. However, since we used a short cut, it also created an asset engine. Let's use the asset engine endpoint to see what engines our asset has.

```
curl -H 'Authorization: X-API-Key {Your API Key}' -i
https://api.eqm.world/assetengine?assetId={Your Asset ID}&companyId={Your Company Id}
```

Take note of the "id". We will need the asset engine "id" later. The response should show that we've hooked up the asset to an asset engine. This engine is of engine type Light Truck and consumes Diesel. Now that we have created an asset and engine, let's add some fuel consumption!

Adding Fuel Consumption

Adding fuel consumption is an easy process. All we need is the asset engine Id, the time the fuel was purchased (in millisecond Unix timestamp) and the liters of fuel purchased.

```
curl -d '{"amount":100, "time":1600000000000, "assetEngineId":{Your Asset Engine Id}}' -H
'Content-Type: application/json' -H 'Authorization: X-API-Key {Your API Key}' -i
https://api.eqm.world/assetfuelconsumption
```

Once we send this request, we should get a 200 response and the asset fuel consumption body. Now we have created fuel consumption, we can call the reporting API and see how much carbon we have created.

Reporting

In order to calculate our pollution, we need to call the reporting API and pass in the timeframe we want to calculate. We also need to tell the API how to group pollutants together. Therefore, we will tell the API to break it up by month.

```
curl -d '{"timeframe": "month", "search": "time>1500000000000 AND time<1700000000000"}'
-H 'Content-Type: application/json' -H 'Authorization: X-API-Key
MGJkN2VhMzItODU0Yi00Y2E3LWEyNmUtYjM1ZDM5NWJmMjdh' -i
https://api.eqm.world/report/pollutants
```

The response will be the carbon produced from your previous fuel consumption. The response object will be nested into the milliseconds for the month the fuel was consumed on. Inside this object will be a breakdown of each pollutant created from burning the diesel consumed. These pollutants include "cox" (carbon), "sox" (sulphur), "nox" (nitrogen), "pm10" (Particulate Matter 10), "pm2_5' (Particulate Matter 2.5).

Conclusion

At this point, we have created a basic asset to record carbon creation. In review, you now know how to create a fleet, asset, and fuel consumptions. The reporting system is very dynamic. For a more detailed brief of our reporting system, please review our documentation.