python -m venv venv

Linux

$ source venv/bin/activate

(venv) $ \_

Microsoft Windows

$ venv\Scripts\activate

(venv) $ \_

(venv) $ pip install -r requirements/base.txt

(venv) $ flask run

app.veosat.routes

@module.route('/db\_init/')  
def db\_init():  
 *"""Create a data base tables".  
 Example of the route:  
 http://127.0.0.1:5000/veosat/db\_init/  
 """*

@module.route('/check\_velocity/')  
def check\_velocity():  
 *"""Check vehicle velocity status".  
 Example of the route:  
 http://127.0.0.1:5000/veosat/check\_velocity/  
 """*

app.veosat\_rest\_x.routes

@rest\_x.route('/get\_id\_state/<int:id\_state>')  
class GetIdState(Resource):  
 *"""Seleccionar los vehíclos con un parámetro para filtrar el id\_estado  
 Example of the route:  
 http://127.0.0.1:5000/veosat\_rest\_x/get\_id\_state/1  
 """*

*.flaskenv* : environment variables for the command flask

FLASK\_APP=main.py

FLASK\_ENV=development  
FLASK\_RUN\_PORT=5000  
FLASK\_RUN\_HOST=127.0.0.1

*.env*: variables to config flask.

SECRET\_KEY=a-really-long-and-unique-key-that-nobody-knows

*config.py* : flask config

import os

from dotenv import load\_dotenv

BASEDIR = os.path.abspath(os.path.dirname(\_\_file\_\_))

load\_dotenv(os.path.join(basedir, '.env'))

class Config(object):

SECRET\_KEY = os.environ.get('SECRET\_KEY') or 'you-will-never-guess'

# for mysql use in .env:  
 # DATABASE\_URL = mysql+pymysql://root:pass@localhost/mydb  
 # If DATABASE\_URL don't set, by default will be used sqlite

SQLALCHEMY\_DATABASE\_URI = os.environ.get('DATABASE\_URL') or \  
 'sqlite:///' + os.path.join(BASEDIR, 'app.db')

SQLALCHEMY\_TRACK\_MODIFICATIONS = False

*app / \_\_ init\_\_.py* : flask fabric, *initialization of*  Flask-SQLAlchemy

from flask import Flask  
# flask\_restplus is dead now, move to flask-restx. Or downgrading to Werkzeug==0.16.1 to solves this  
from app.rest\_x import api  
from config import Config  
from app.database import db  
import app.veosat.routes as veosat  
import app.veosat\_rest\_x.routes as veosat\_rest\_x  
  
  
  
def create\_app(config\_class=Config):  
 """Function to create Flask instance"""  
  
 app = Flask(\_\_name\_\_)  
  
 # applying applications config  
 app.config.from\_object(config\_class)  
  
 # data base initialization  
 db.init\_app(app)  
 # flask\_restx initialization  
 api.init\_app(app)  
  
 # application module register in application instance  
 app.register\_blueprint(veosat.module)  
 # app.register\_blueprint(veosat\_rest\_x.module)  
  
 @app.route('/')  
 @app.route('/index')  
 def index():  
 return 'Hello, World'  
  
 return app

*app/veosat/models.py* : data base models

from app import db

class Vehiculo(db.Model):  
 id = db.Column(db.Integer, primary\_key=True)  
 matricula = db.Column(db.String(10))  
 id\_estado = db.Column(db.Integer)

posiciones = db.relationship('Posiciones', backref='vehiculo', lazy='dynamic')

def \_\_repr\_\_(self):  
 return '<Vehiculo {}>'.format(self.matricula)

class Posiciones(db.Model):  
 id = db.Column(db.Integer, primary\_key=True)  
 latitud = db.Column(db.Float)  
 longitud = db.Column(db.Float)  
 velocidad = db.Column(db.Float)  
 id\_vehiculo = db.Column(db.Integer, db.ForeignKey('Vehiculo.id'))  
  
 def \_\_repr\_\_(self):  
 return '<Posiciones {}>'.format(  
 "(latitud=" + self.latitud + ", longitud=" + self.longitud + ")")

*main.py : import models to the project.*

from app import create\_app  
  
app = create\_app()