

Vincent Healy

Vincent.healy@raildeliverygroup.com

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

Train Station Asset GUI development

Date

Contents

**No table of contents entries found.**

# Create the Models.py

from flask\_sqlalchemy import SQLAlchemy

from datetime import datetime

db = SQLAlchemy()

class TrainOperator(db.Model):

\_\_tablename\_\_ = 'train\_operator'

operator\_id = db.Column(db.Integer, primary\_key=True, autoincrement=True)

operator\_name = db.Column(db.String(255), unique=True, nullable=False)

contact\_info = db.Column(db.Text)

created\_at = db.Column(db.DateTime, default=datetime.utcnow)

updated\_at = db.Column(db.DateTime, default=datetime.utcnow, onupdate=datetime.utcnow)

class Station(db.Model):

\_\_tablename\_\_ = 'station'

station\_id = db.Column(db.Integer, primary\_key=True, autoincrement=True)

station\_name = db.Column(db.String(255), unique=True, nullable=False)

location = db.Column(db.Text)

operator\_id = db.Column(db.Integer, db.ForeignKey('train\_operator.operator\_id'))

created\_at = db.Column(db.DateTime, default=datetime.utcnow)

updated\_at = db.Column(db.DateTime, default=datetime.utcnow, onupdate=datetime.utcnow)

operator = db.relationship('TrainOperator', backref=db.backref('stations', lazy=True))

class DeviceType(db.Model):

\_\_tablename\_\_ = 'device\_type'

device\_type\_id = db.Column(db.Integer, primary\_key=True, autoincrement=True)

device\_type\_name = db.Column(db.String(255), unique=True, nullable=False)

created\_at = db.Column(db.DateTime, default=datetime.utcnow)

updated\_at = db.Column(db.DateTime, default=datetime.utcnow, onupdate=datetime.utcnow)

class Supplier(db.Model):

\_\_tablename\_\_ = 'supplier'

supplier\_id = db.Column(db.Integer, primary\_key=True, autoincrement=True)

supplier\_name = db.Column(db.String(255), unique=True, nullable=False)

contact\_info = db.Column(db.Text)

created\_at = db.Column(db.DateTime, default=datetime.utcnow)

updated\_at = db.Column(db.DateTime, default=datetime.utcnow, onupdate=datetime.utcnow)

class StationDevice(db.Model):

\_\_tablename\_\_ = 'station\_device'

station\_device\_id = db.Column(db.Integer, primary\_key=True, autoincrement=True)

station\_id = db.Column(db.Integer, db.ForeignKey('station.station\_id'))

device\_type\_id = db.Column(db.Integer, db.ForeignKey('device\_type.device\_type\_id'))

supplier\_id = db.Column(db.Integer, db.ForeignKey('supplier.supplier\_id'))

quantity = db.Column(db.Integer, nullable=False)

created\_at = db.Column(db.DateTime, default=datetime.utcnow)

updated\_at = db.Column(db.DateTime, default=datetime.utcnow, onupdate=datetime.utcnow)

station = db.relationship('Station', backref=db.backref('station\_devices', lazy=True))

device\_type = db.relationship('DeviceType', backref=db.backref('station\_devices', lazy=True))

supplier = db.relationship('Supplier', backref=db.backref('station\_devices', lazy=True))

Explanation

* **Importing SQLAlchemy**: The SQLAlchemy class is imported from flask\_sqlalchemy.
* **Defining Models**: Each table in the SQL schema is represented as a class that inherits from db.Model.
* **Fields**: Columns in the SQL tables are defined as attributes in the model classes using db.Column.
* **Relationships**: Foreign key relationships are represented using db.ForeignKey and db.relationship.
* **Timestamps**: The created\_at and updated\_at fields are set to use the current time by default.

To use these models in your Flask application, make sure to initialise the SQLAlchemy instance with your Flask app:

from flask import Flask

from models import db

app = Flask(\_\_name\_\_)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'mysql://username:password@localhost/train\_device\_management'

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

db.init\_app(app)

with app.app\_context():

db.create\_all()