Complete RESTful API’s

PyDantic V1 vs PyDanticv2

* New information in this will include
  + FULL SQL DB
  + Authentication
  + Authorization
  + Hashing Passwords
* Creating a TODO Table
  + We will create new todo table model for our application.
  + We will be using these todos to save records throughout this project

Section 8: Setup Database

What is Database?

Organized collection of structured information of data, which is stored in a computer system.

1. The data can be easily accessed
2. The data can be modified
3. The data can be controlled and organized
4. Many databases use a structured query language (SQL) to modify or write data
5. Data can be related to about any object
6. For example, a user on application may have
   1. Name
   2. Image
   3. Email
   4. Password

* A DB is a collection of data since data, on its own, its just data a database allows management of this data
* Databases are organized in how data can be retrieved, stored and modified.
* There are many types of database management system’s
  + Sqlite
  + SQL
  + PosgreSQL

What is SQL?

* Pronounced as either as S-Q-L or See Quel
* Structured language for dealing with RDBMS
* SQL can be used to do different things with database records such as
  + - Create
    - Read
    - Update
    - Delete

DB Connection with ORM SQLAlchemy

* We need to pip install sqlalchemy for dealing with databases.

from sqlalchemy import create\_engine

from sqlalchemy.orm import sessionmaker

from sqlalchemy.ext.declarative import declarative\_base

SQLALCHEMY\_DB\_URL = 'sqlite:///./bikes.db'

engine = create\_engine(SQLALCHEMY\_DB\_URL, connect\_args={'check\_same\_thread' : False})

SessionLocal = sessionmaker(autocommit = False, autoflush=False, bind=engine)

Base = declarative\_base()

The above code is the boiler plate code for defining a Database (name the file database.py).

Database tables (Models)

from database import Base

from sqlalchemy import Column, Integer, String, Boolean

class Todos(Base):

    \_\_table\_\_ = 'bikes'

    id = Column(Integer, primary\_key=True, index=True)

    title = Column(String)

    description= Column(String)

    priority= Column(Integer)

    complete= Column(Boolean, default=False)

Create DB Connection For API

Create a file named main.py

from fastapi import FastAPI

import models

from database import engine

app = FastAPI()

models.Base.metadata.create\_all(bind=engine)

Installation of SQLite 3

* Download Zip file of precompiled binaries for windows from sqlite.org
* Extract the files.
  + Add to the system variable from environment variable
  + Run sqlite3 in cmd to check the DB.
  + Sqlite3 DB\_NAME.DB

Now the lesson is about SQL queries so we will skip that in this document.

To view formatted data in sqlite3 use the below commands.

* .mode Column
* .mode markdown
* .mode box
* .mode table

API Request Methods.

we have to create DB dependency inside main.py

def get\_db():

    db = SessionLocal() #from database.py

    try:

        yield db

    finally:

        db.close()