CREATING AN API

Create a blueprint in our app. Works from our database whereby the use connects to an API route, we query our database for relevant information, organize that information into json data and send back to the user as json data

Things to consider when creating an API.

1. What data are we providing?
2. Do we have an existing data set?
3. What is the main format of our data?
4. What functionality do we want to interact with this data?

Our Answers For Class Exercise

1. What data are we providing DB Model- designing a model for the animals in an exotic pet shop
2. Do we have an existing data set? No, we’ll make some stuff by creating a database
3. What is the main format of our data? Main format- Database Model-design this first

.DB Model-designing a model for the animals in an exotic pet shop

-Id uuid= string

-Name = string

-Scientific Name= string

-size= string (needs to specify units)

-weight= integer

-diet= string

habitat = string

-lifespan = interger

-realease date= Numeric

-description = string

-price = numeric(float w/2 decimals)

A.In the models file start laying out the new database model, the columns that are going to go into this model.

B.Now we can think about how to structure the init method. Since we are going to be allowing users to input new things into the api the init has to be a dictionary the init for this is going to be different than the other table for the user because we

#are going to let some users create new data in our database and create a new animal(the user will be sending

# the 'POST' request in Json formatt to our system which translates into a python dictionary). Therefore,

#our init method for creating a new animal has to accept a single dictionary.

C. Now we can update my database to have this model

#1activate my virtual environment

#2update database flask db migrate

#3 flask db upgrade

#test init method by adding some data by giving flask shell access to my database model,over to run.py

#import the Animal model next to the user model and in the context processor

#in terminal type flask shell (enter), then db(enter): to access my db, then the name of the model:i.e user, animal, whatever your table name is (enter)

# to create a test of animals and the scenario of required info and unrequired info need to create a dictionary since that is the format of my init

# type in terminal ad = {}, then add key value pairs to it by saying ad['name']='Fennec Fox'(enter), ad['sci\_name']='Vulpes zerda'

#now type ad to print the dict.(enter), then type fox = Animal(ad)"ad is the name of the dictionary (enter), then type fox(enter), then fox\_\_.dict\_\_(enter)

#check everything is right then type db.session.add(fox), db.session.commit(), then check if it worked by querying the database. type Animal.query.all() or go check elephant database

D. Now we can start creating our API. Over to number 4

1. What functionality do we want to interact with this data?

Create a new api blueprint inside of the app folder if there is not one already and call the new folder api. Make another file inside of the api folder and call it routes.py

A.)

#intial blueprint setup

from flask import Blueprint

#instantiate blueprint api and then create a connection between the blueprint api and my flask app.

api = Blueprint('api', \_\_name\_\_, url\_prefix='/api')

B.) #Head over to init.py

C.) head over to the init.py under import blueprints type: from .api.routes import api

Then under register blueprints type:app.register\_blueprint(api)

-read all data from database

-read single data from database

-create new data

the init for this is going to be different than the other table for the user because we are going to let some users create new data in our database and create a new animal(the user will be sending the 'POST' request in Json formatt to our system which translates into a python dictionary). Therefore, our init method for creating a new animal has to accept a single dictionary.

-updata data

-delete data

Answers for the homework assignment- on my own practice

1. What data are we providing? DB Model- designing a model for the horror movies available to buy from shop of horrors video store.
2. Do we have an existing data set? No make some stuff up
3. What is the main format of our data? in format- Database Model-design this first

-Id uuid= string

-Name = string

-Rating= string

-Box office= integer

-Director= string

-category = string

-price = numeric(float w/2 decimals)

A.In the models file start laying out the new database model, the columns that are going to go into this model.

B.Now we can think about how to structure the init method. Since we are going to be allowing users to input new things into the api the init has to be a dictionary the init for this is going to be different than the other table for the user because we

#are going to let some users create new data in our database and create a new animal(the user will be sending

# the 'POST' request in Json formatt to our system which translates into a python dictionary). Therefore,

#our init method for creating a new animal has to accept a single dictionary.

C. Now we can update my database to have this model

#1activate my virtual environment

#2update database flask db migrate

#3 flask db upgrade

#test init method by adding some data by giving flask shell access to my database model,over to run.py

#import the Animal model next to the user model and in the context processor

#in terminal type flask shell (enter), then db(enter): to access my db, then the name of the model:i.e user, animal, whatever your table name is (enter)

# to create a test of animals and the scenario of required info and unrequired info need to create a dictionary since that is the format of my init

# type in terminal ad = {}, then add key value pairs to it by saying ad['name']='Fennec Fox'(enter), ad['sci\_name']='Vulpes zerda'

#now type ad to print the dict.(enter), then type fox = Animal(ad)"ad is the name of the dictionary (enter), then type fox(enter), then fox\_\_.dict\_\_(enter)

#check everything is right then type db.session.add(fox), db.session.commit(), then check if it worked by querying the database. type Animal.query.all() or go check elephant database

D. Now we can start creating our API. Over to number 4

1. What functionality do we want to interact with this data?

Create a new api blueprint inside of the app folder if there is not one already and call the new folder api. Make another file inside of the api folder and call it routes.py

A.)

#intial blueprint setup

from flask import Blueprint

#instantiate blueprint api and then create a connection between the blueprint api and my flask app.

api = Blueprint('api', \_\_name\_\_, url\_prefix='/api')

B.) #Head over to init.py

C.) head over to the init.py under import blueprints type: from .api.routes import api

Then under register blueprints type:app.register\_blueprint(api)

-read all data from database

-read single data from database

-a dynamic route for getting a subset of data (ex. Film with a particular rating)

-create new data(create a new movie that’s not on the list)

the init for this is going to be different than the other table for the user because we are going to let some users create new data in our database and create a new animal(the user will be sending the 'POST' request in Json formatt to our system which translates into a python dictionary). Therefore, our init method for creating a new animal has to accept a single dictionary.