# Dog/Breed Lab and Exercise

## Joshua C. Bartels

## NOTES:

1. *I went with the ViewSet model after having troubles with the snippets*
   1. *Other than the labs we have done in this class, I have zero experience with Python. So I learned a lot in this Exercise. Most of my work is done in Perl or Shell so quick and dirty. However 20+ years ago when I did my CS degree, we used C++, which Python oddly reminds me of very much.*
2. *My breed validations work, but they don’t return useful errors to the user. This caught me when I tried to change one value to a 7 (1-5 allowed) and a breed size from Large to Larger.* 
   1. *The validation is done in the model, and I don’t think the model returns that to the view. Seems like I should have maybe done the validation in the view vs. the model?*
   2. *But using the admin GUI, they work nicely.  
      A screenshot of a computer

      Description automatically generated*
3. *Note the comments inline about the trailing slash requirements and how that it handled. That may be a “feature” or limitation of the runserver webserver more so than it is something in the code.*
4. *I did this in Visual code studio on a Mac and PC, using and env that can run on both machines to accommodate my work travel. Yes, I could have used a VM, cloud, docker, etc. But I like my 4 monitors at home, and my Mac when sitting in an airport. This setup seems to work pretty well.*
   1. *POSTMAN does not seem to have a save-all requests in a collection, which would be awfully nice, vs. saving each one. Its duplicate feature also does not duplicate the BODY or the URL, so I am not sure what the point of duplicating even is.*

## GET/POST /api/dogs/

POST – Create CrazyDog ( created ID:42)

A screenshot of a computer

Description automatically generated

GET – All dogs (now we have our ID:42) A screenshot of a computer

Description automatically generated

## GET/POST /api/breeds/

POST – Create CrazyBreed (ID:23)

A screenshot of a computer

Description automatically generated

GET – All breeds (now we have our ID:23)

A screenshot of a computer

Description automatically generated

## GET/PUT/DELETE to /api/dogs/<id>

GET - our CrazyDog by its ID:42  
  
*NOTE: I did* ***NOT*** *use the trailing slash on the request, which gives me a 301 (Object has moved permanently and POSTMAN follows the request with the trailing slash).*

A screenshot of a computer

Description automatically generated

PUT – Lets update Dog ID:42 (now it eats socks vs. shoes, and plays with shoes vs. socks – This is bad. It also got one year older).

*NOTE: I* ***DID*** *use the trailing slash on the request. If I don’t use it, I get a not helpful error and a 500 server error in POSTMAN. I can’t figure out how the server handles slash vs. no slash on GET vs PUT/DELETE verbs.*

A screenshot of a computer

Description automatically generated

DELETE - CrazyDog (who is now named Dog1) but it still id=26   
*NOTE: It would be nice if my output included the ID of the dog that I deleted, but I could not get that to work.*  
A screenshot of a computer

Description automatically generated

## GET/PUT/DELETE to /api/breeds/<id>

GET – So lets get our Crazy Breed with ID:23  
*NOTE: I used the trailing slash here to avoid the 301 noted in the get dog by ID example.*

A screenshot of a computer program

Description automatically generated

PUT- Let’s update our Breed ID:23 to be crazier

A screenshot of a computer

Description automatically generated

DELETE – No more of this breed ID:13  
*NOTE: Note the 404 in there before the 200. Whoops it was 23* ***NOT*** *33. I guess breed 23 got deleted long ago. The server logged a 404, but POSTMAN got the 417 that I returned in my error catch. Interesting and something to look into…..*A screenshot of a computer

Description automatically generated

So much testing 😊

A screenshot of a computer

Description automatically generated