I made a creature CRUD using Springboot, it's similar to a customer or students CRUD, except it's more abstract and just general living beings. This assignment was built on top of the Springboot base/template.

Create

http://localhost:8081/api/v1/creatures/create/5/Roddy Piper creates a creature with ID of 5 and name "Roddy Piper" adding a creature with an ID that already exists returns a temporary 404 creature.

Update

To update (update the name, by id, in this case) change mode to POST and go http://localhost:8081/api/v1/creatures/update/1/rodney,

Read

http://localhost:8081/api/v1/creatures/find/1

Displays data for creature of id "1". Trying to find a non-existing creature returns a temporary 404 creature. If input is invalid, the 404 creature is temporary and doesn't actually go anywhere.

In postman, set the mode to GET and enter http://localhost:8081/api/v1/creatures/all to retrieve a list of current creatures.

Delete

http://localhost:8081/api/v1/creatures/delete/1

deletes a creature that has an id of "1" from the list, deleting a non-existent creature returns a failure message string.

The penultimate part of the address is the id of an existing creature, the last part is the name you want to update the creature to have. updating a non-existing creature returns a temporary 404 creature.

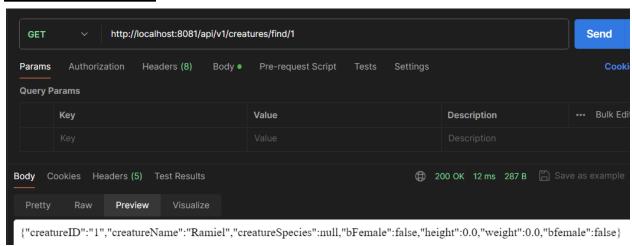
The assignment utilizes a custom controller (WebController) and CreatureService and CreatureRepository, as well as a Creature model

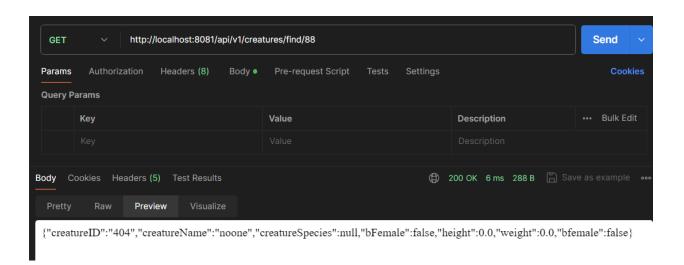
Problems Encountered and How They Were Fixed:

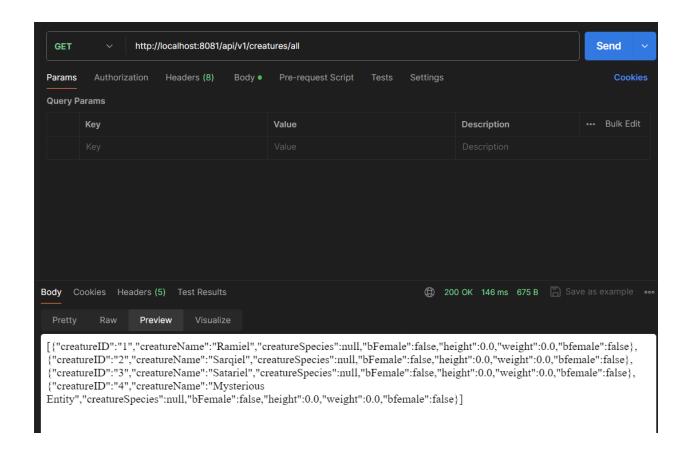
For some reason, the service was not binding to the Controller automatically, so I had to specify a constructor using @Autowired with the service as a parameter, in order to force SpringBoot to use that constructor, and thus force it to create a CreatureService and bind it to the constructing WebController.

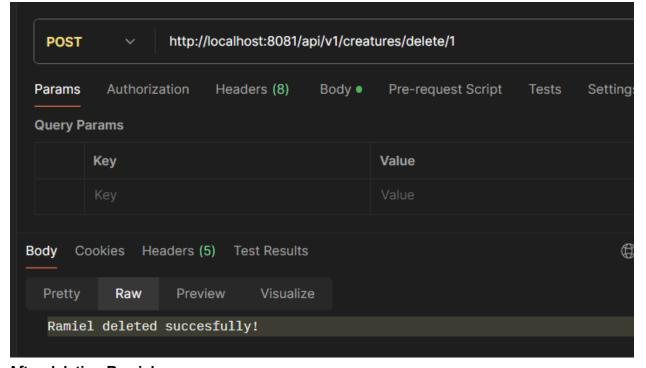
```
@Autowired /*Ensure the service loads*/
public WebController(CreatureService creatureService) {
    this.creatureService = creatureService;
}/*So Spring because it sees the @Autowired, it will know to use this constructor
when creating an instance of the WebController, and thus be forced to create a CreatureService,
because it's a constructor with that as a required parameter.*/
```

Screenshots:

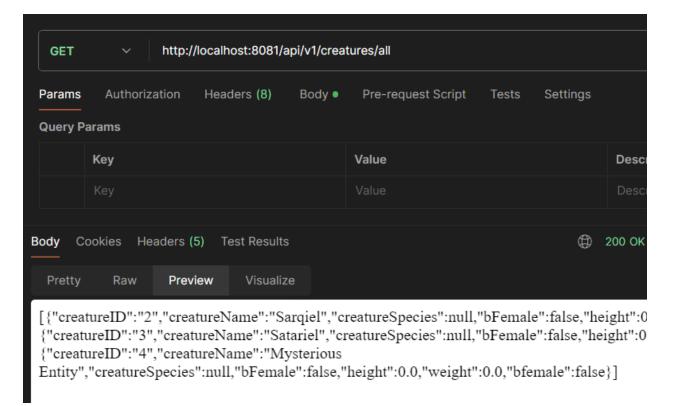




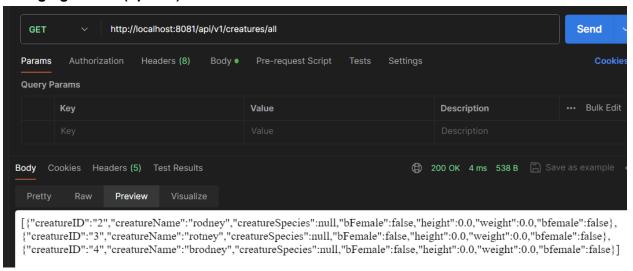




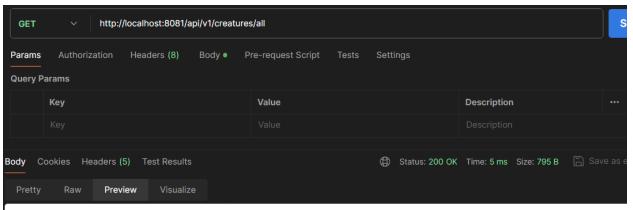
After deleting Ramiel



Changing names (update)



Adding new creatures



[{"creatureID":"2","creatureName":"rodney","creatureSpecies":null,"bFemale":false,"height":0.0,"weight":0.0,"bfemale":false}, {"creatureID":"3","creatureName":"rotney","creatureSpecies":null,"bFemale":false,"height":0.0,"weight":0.0,"bfemale":false}, {"creatureID":"4","creatureName":"brodney","creatureSpecies":null,"bFemale":false,"height":0.0,"weight":0.0,"bfemale":false}, {"creatureID":"5","creatureName":"Roddy Piper","creatureSpecies":null,"bFemale":false,"height":0.0,"weight":0.0,"bfemale":false}, {"creatureID":"6","creatureName":"Zhong Xina","creatureSpecies":null,"bFemale":false,"height":0.0,"weight":0.0,"bfemale":false}]