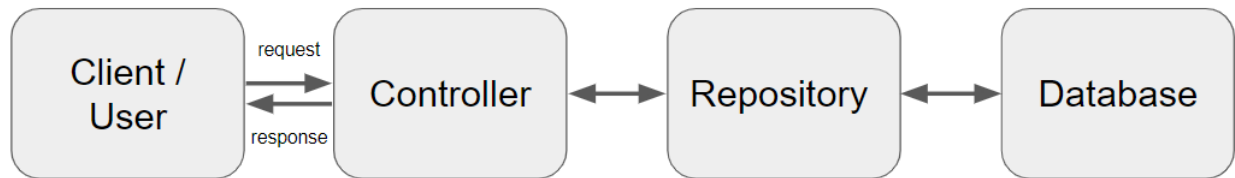


Virtual Pets API

In this assignment, we will build out an REST API for some virtual pet data.

This could become the foundation if we were to build out the virtual pet app further for more users and potentially a GUI.



Required Tasks

- You are creating a Virtual Pet API, it's advised that you don't build on top of the Virtual Pet Amok assignment. Instead, create a new project using the Spring Initializer
- You DO NOT have to code functionality that deals with all or individual pets ex. `feedAllPets()`, `tick()`, etc. We are just building out an API for this assignment.
- No game play interface is required for this assignment.
- Create a project using the Spring Initializer
 - With JPA, Web, and H2 as the dependencies
- Create an entity "Virtual Pet"
 - Add 4-5 fields that were used to describe the pet in the previous assignments (name, description, hunger, thirst, boredom, etc)
- Create a repository for Virtual Pet
- Create a controller for for Virtual Pet
 - In the controller create a method to retrieve all pets
 - In the controller create a method to retrieve a single pet by ID
 - In the controller create a method to post (add) a new pet to the database
 - In the controller create a method to delete a pet with the ID
- Create a `LoadDatabase` class that will auto load 4-5 mock virtual pet data into the

database

- Create an error handling message when a pet requested by ID is not found
- Using Javascript, create an app that will demonstrate the following features of the API and outputs the results in the console (`console.log()`) using `fetch()`
 - Retrieve all pets in the database
 - Retrieve a single pet by ID
 - Add a new pet and retrieve that pet
 - Delete a pet and check that the pet is no longer present by getting the error handling message