

NNDAP Assignment

Introduction

Ingrid is a farmer and she's looking to further automate her cowshed. She has over 100 cows and wants to schedule some tasks as well as measure details about every one of them. There are a couple of machines available that can measure the weight and milk production, and there is a feeding robot as well.

All equipment has been set up, but the automation still needs to be programmed. Luckily, you are here to help her out!

Assignment

1. *Set up a RESTful API in Python that returns all cows (when calling GET /cows). The API response with the data model of a single cow can be found under the attachments.*
2. *Create endpoints that machines can call. They must be able to create, read, update, and delete cow "objects" (CRUD). For example: a POST request to /cows must create a new cow in the system, and a PUT request should be able to update one of them. Store the data in such a way that multiple cows can be edited at the same time without corrupting any files or databases.*
3. *Add a method to filter cows. For instance, to allow easy retrieval of the male or female cows.*
4. *The final deliverable is a distributable Python package.*

Remarks

- **Enforce type hints at runtime and data validation.**
- **Apply software best practices** where you can. We prefer quality over quantity, and rather see important engineering aspects covered than lots of extra functionality.
- **Please send the codebase to us the day before the interview takes place.**
- **Think about the scalability of your solution.** What would happen if your system would handle not only this farm, but all farms in Europe?
- Think of **which storage layer** you would use to perform analytics on changing attributes over time.

Attachments

API response of GET /cows when only one cow is stored

```
[
  {
    "name": "Betty",
    "sex": "Male",
    "birthdate": "2019-02-11T03:21:00.000000",
    "condition": "Healthy",
    "weight": {
      "mass_kg": 1100,
      "last_measured": "2022-11-02T11:15:00.000000"
    },
    "feeding": {
      "amount_kg": 5,
      "cron_schedule": "0 */6 * * *",
      "last_measured": "2022-11-02T11:00:00.000000"
    },
    "milk_production": {
      "last_milk": "2022-11-02T09:00:00.000000",
      "cron_schedule": "0 8,12,16,20 * * *",
      "amount_l": 5
    },
    "has_calves": true
  }
]
```