# Test assessment

## Spy Cat

### Overview

This task involves building of a a simple yet comprehensive CRUD application. The goal is to create a simple yet comprehensive system that demonstrates your understanding in building RESTful APIs, interacting with SQL-like databases, integrating third-party services, and optionally creating user interfaces. The test assessment is expected to be done under 4 hours.

### Business Task

Spy Cat Agency (SCA) asked you to create a management application, so that it simplifies their spying work processes. SCA needs a system to manage their cats, missions they undertake, and targets they are assigned to.

From cats perspective, a mission consists of spying on targets and collecting data. One cat can only have one mission at a time, and a mission assumes a range of targets (minimum: 1, maximum: 3). While spying, cats should be able to share the collected data into the system by writing notes on a specific target. Cats will be updating their notes from time to time and eventually mark the target as complete. If the target is complete, notes should be frozen, i.e. cats should not be able to update them in any way. After completing all of the targets, the mission is marked as completed.

From agency perspective, they regularly hire new spy cats and so should be able to add them to and visualize in the system. SCA should be able to create new missions and then assign them to cats that are available. Targets are created in place along with a mission, meaning that there will be no page to see/create all/individual targets. A target can be added to or deleted from a mission.

**Backend Requirements**

* **Spy Cats**
  + Ability to create a spy cat in the system
    - A cat is described as Name, Years of Experience, Breed, and Salary.
    - Breed must be validated, see General
  + Ability to remove spy cats from the system
  + Ability to update spy cats’ information (Salary)
  + Ability to list spy cats
  + Ability to get a single spy cat
* **Missions / Targets**
  + Ability to create a mission in the system along with targets
    - A mission contains information about Cat, Targets and Complete state
    - Each target is unique to a mission, so the endpoint should accept an object describing targets
    - A target is described as Name, Country, Notes and Complete state
  + Ability to delete a mission
    - A mission cannot be deleted if it is already assigned to a cat
  + Ability to update mission
    - Ability to mark it as completed
  + Ability to update mission targets
    - Ability to mark it as completed
    - Ability to update Notes
      * Notes cannot be updated if either the target or the mission is completed
  + Ability to delete targets from an existing mission
    - A target cannot be deleted if it is already completed
  + Ability to add targets to an existing mission
    - A target cannot be added if the mission is already completed
  + Ability to assign a cat to a mission
  + Ability to list missions
  + Ability to get a single mission
* **General**
  + Framework
    - Use any modern framework, e.g. Gin, Echo, Fiber, etc.
  + Database
    - You can use any SQL-like database
  + Logger middleware
    - Integrate a logging middleware to log HTTP requests and responses for debugging and monitoring
  + Validations
    - Make sure endpoints validate the request body and return an adequate status code if it’s not valid
    - Validate cat’s breed with [TheCatAPI](<https://api.thecatapi.com/v1/breeds>)

### Submission

* Create a simple README file explaining how to build/start the application
  + Include any information you think will be useful for us
* Define all of the endpoints in a Postman collection and provide us with a file
* Make sure to provide us a link to the repository with the source code
* Don't hesitate to ask any questions