

# Objective

Your task is to implement a REST API for a restaurant.

## Brief

Frogo Baggins, a hobbit from the Shire, has a great idea. He wants to build a restaurant that serves traditional dishes from the world of Middle Earth. The restaurant will be called "**The Orc Shack**" and will have a cozy atmosphere.

Frogo has hired you to build the website for his restaurant. As payment, he has offered you either a chest of gold or a ring. Choose wisely.

## Tasks (Specifications)

### Task 1

Deliver a REST API that meets the following requirements:

- An API user must be able to:
  - Create, View, List, Update, and Delete dishes.
  - Dishes must have a name, description, price, and image.
- Customers must be able to take the following actions:
  - Search, View, and Rate dishes

### Task 2

- Add user, permission, and authentication support.
- Users must be able to register and login.
- All functionality of the API must require a logged in user (except Registration)
- At a minimum, the system should support password based authentication.
- Users must have a name and email address and password.
- Add validation to the data entities in the API.
- An Evil Orc is attempting to brute force passwords for known email addresses. Add functionality to defend against this. (You can use any methodology that you deem suitable)

### Task 3

- The API is running on an old Shire Server that is starting to struggle with the load of the now popular website. Implement a solution to improve the performance of the API on the same hardware.
- Add support for multiple different restaurants to use the product (multi-tenant SaaS)

### Task 4

- The evil Orc has created many sockpuppet accounts and has left many bad reviews. Use an AI/ML solution to provide a sentiment score for each review.
- To prevent abuse, add rate-limiting per logged in customer.
- Allow users to login using OAuth2 based SSO (Google, etc)