

Backend Software Engineer Coding Exercise

SOFTWARE ENGINEER I SOFTWARE ENGINEER II SENIOR SOFTWARE ENGINEER

While this challenge can be presented to any level of software engineer, an SEI may struggle to complete and deliver all requirements (especially with the object detection API). Each candidate's code should be evaluated based on their level. A Senior should absolutely be capable of completing this challenge in its entirety.

Overview

Build a HTTP REST API in **{Java|Node.js|Python}** for a service that ingests user images, analyzes them for object detection, and returns the enhanced content. It should implement the following specification::

API Specification

GET /images GET <http://localhost:8080/images>

- Returns HTTP 200 OK with a JSON response containing all image metadata.

GET /images?objects="dog,cat" GET [http://localhost:8080/images?objects="tree,lake"](http://localhost:8080/images?objects='tree,lake')

- Returns a HTTP 200 OK with a JSON response body containing only images that have the detected objects specified in the query parameter.

GET /images/{imageId} GET <http://localhost:8080/images/67e829570dc6644c265d3909>

- Returns HTTP 200 OK with a JSON response containing image metadata for the specified image.

POST /images POST <http://localhost:8080/images?enableObjectDetection=true&imageLabel=file three&filename=three.jpg>

- Send a JSON request body including an image file or URL, an optional label for the image, and an optional field to enable object detection.
- Returns a HTTP 200 OK with a JSON response body including the image data, its label (generate one if the user did not provide it), its identifier provided by the persistent data store, and any objects detected (if object detection was enabled).

POST http://localhost:8080/images?enableObjectDetection=true&imageLabel=GolfBall&image_url=https://www.golfcartreport.com/wp-content/uploads/2020/08/golfballongrassontee.jpg

Object detection instructions

Image object detection can be performed using any API offering of your choosing (such as [Google](#), IBM, [Imagga](#), etc), or with a process managed by your backend. The only requirement is that it must return a list of object names detected within that image.

That is the extent of the API contract. HTTP error codes should be used to indicate the proper level of system failure (i.e. client versus server).

Database

A persistent data store is required, any variant of SQL is encouraged.

Expectations

No frontend is required, but you may create one to demo the API. Regardless, a user of the API should be able to:

- Upload an optionally labelled image and run image object detection on it
- Retrieve all images and any metadata obtained from their analyses
- Search for images based on detected objects