GoLeague Django 2 and Python 3 Developer Questions and Assignment

Questions

(1) Please expand on your experience with Python. Did that include developing apps with Django? What about developing REST APIs with Django, or developing REST APIs with Python in general?

(2) Do you have experience with Docker? Have you created Dockerfiles? Deployed containers to AWS ECS? Deployed with AWS EKS or Kubernetes in general?

(3) Do you have experience with AWS and its various services? If so, which ones?

(4) Do you have experience with the Linux server, especially Ubuntu? This includes skills like: using Linux command-line tools; writing/executing bash scripts; and SSH-ing into a AWS EC2 instance running Ubuntu, and proxying through a bastion server.

(5) Do you have experience with Git? What about GitHub?

(6) Have you created or contributed-to any open-source projects, especially if they are Python-related. If so, which ones? Note, it's not an issue if you haven't.

(7) Would you be willing to create a simple Django app so that we can assess your skills? If so, we will send you further instructions.

Assignment

Instructions

Django TODO App

Create a Python 3 Django app that provides a REST API for a simple TODO app. It should provide the following API:

1. Add one or more new TODOs. Each TODO is composed of:
    a. State: todo, in-progress, done
    b. Due date.

    c. TODO text.

2. Delete one or more TODOs.
3. Update one or more TODOs.
4. List all TODOs.
    a. Able to filter TODOs by state and/or due-date.

Base Software

Use:

- Python 3.6.x
- Django 2.x
    - SQLite as datastore.
- Any other Python packages to help with the assignment.

Use pipenv to setup the virtual environment. I.e., we should be able to run “pipenv install” from the root of your project folder to create the virtual environment and install all the packages into it. If you haven’t used pipenv yet, read up about it here: https://docs.pipenv.org/

Docker

If you said yes to having Docker experience, create a Dockerfile. If you said no, then this is not required.

Test Script

Write a test script to test the API. Options include:

1. Bash script which uses curl. Pipe the curl results through “python -m json.tool” to pretty print the JSON output.
2. Python script. Any dependencies should be installed with pipenv.

If all the tests pass, the test script should exit with status code 0. Otherwise an error status code.

OS

Both the app and test script should run on Ubuntu Server 18.04.

Readme

Add a readme file in Markdown format. Write step-by-step instructions with the terminal commands required to install the Django app, run the Django app, and run the test script.

If you created a Dockerfile, add alternative instructions for building/running the app via Docker.

Delivery

Submit your app as a public repo in GitHub, BitBucket, or GitLab.

Review

We'll examine the repo and run the app/test-script on an Ubuntu 18.04 VM. If you created a Dockerfile, we'll also build and run the app Docker image. Then run the test-script against that.