Module 4 Introduction

Great job, you've made it to the final module! You’ve come so far! Can you believe how much you’ve learned?

In the past few modules, you've seen how you can modify images using Python Imaging Library; how you can interact with web services using the Python requests module, sending data in JSON format; how you can generate PDF files with the content you want; and how you can send emails with those PDFs as an attachment.

For the final project in this course, you’ll use the techniques and concepts you've seen to build a solution to a complex IT task. This can seem a bit daunting at first, but don't worry, you already have all the tools to solve this task!

In the next couple of readings, we're going to get into what to expect, and some things you should keep in mind when writing your solution.

Okay, here's the scenario:

You work for an online fruit store, and you need to develop a system that will update the catalog information with data provided by your suppliers. When each supplier has new products for your store, they give you an image and a description of each product.

Given a bunch of images and descriptions of each of the new products, you'll:

- Upload the new products to your online store. Images and descriptions should be uploaded separately, using two different web endpoints.
- Send a report back to the supplier, letting them know what you imported.

Since this process is key to your business's success, you need to make sure that it keeps running! So, you'll also:

- Run a script on your web server to monitor system health.
- Send an email with an alert if the server is ever unhealthy.

Hopefully this summary has helped you start thinking about how you'll approach this task. In case you're feeling a little scared, don't worry, you can definitely do this! You have all the necessary tools, and the lab description will go into a lot more detail of what you need to do.

Up next, we'll give you a few tips that can help you along the way.

We're giving you a pretty big project to do at the end of this course -- but you can totally complete it with what you've learned until now! Take your time, and be methodical. Use these tips to help you:

Break the problem down into smaller pieces. If you're not sure how to solve a piece of the puzzle, look for an even smaller piece that you *can* solve. Build up those smaller pieces into a larger solution!

Make one change at a time. Write unit tests to make sure that each new part of the solution works the way you think it does. Run your unit tests frequently to make sure that each part of your solution **keeps working** as you make changes.

Use version control. Check each part of your solution into version control as you complete it, so you can always roll back to a known version of your code if you make a mistake.

Review module documentation! You are going to need to use these modules to complete the final project. Reading the documentation takes time, but as you become more familiar with the APIs provided by these modules, it could save you from writing a bunch of custom code that could have just been a call to a module function! Remember, we've covered these modules in previous lessons too, so feel free to go back and review them if you need a refresher!

- Python Image Library (PIL) - Tutorial
- Requests (HTTP client library) - Quickstart
- ReportLab (PDF creation library)
- email (constructing email)

- psutil (processes and system utilization)
- shutil (file operations)
- smtplib (sending email)

Read the lab instructions carefully! Following the instructions and implementing your solution to the specifications that you’re given are critical to completing the task, and to being accurately graded!

Qwiklabs Assessment: Automate updating catalog information

Introduction

You work for an online fruits store, and you need to develop a system that will update the catalog information with data provided by your suppliers. The suppliers send the data as large images with an associated description of the products in two files (.TIF for the image and .txt for the description). The images need to be converted to smaller jpeg images and the text needs to be turned into an HTML file that shows the image and the product description. The contents of the HTML file need to be uploaded to a web service that is already running using Django. You also need to gather the name and weight of all fruits from the .txt files and use a Python request to upload it to your Django server.

You will create a Python script that will process the images and descriptions and then update your company's online website to add the new products.

Once the task is complete, the supplier should be notified with an email that indicates the total weight of fruit (in lbs) that were uploaded. The email should have a PDF attached with the name of the fruit and its total weight (in lbs).

Finally, in parallel to the automation running, we want to check the health of the system and send an email if something goes wrong.

What you'll do

- Write a script that summarizes and processes sales data into different categories
- Generate a PDF using Python
- Automatically send a PDF by email
- Write a script to check the health status of the system

You'll have 90 minutes to complete this lab.

Passed 80%