August 26, 2020

Automating updating catalog information

KYLE HOLLANDS | Google IT AUTOMATION WITH PYTHON | Automating Updating catalog INFORMATION

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.

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

* 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 will also:

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

| Requirement | Due Date | Done | Initials |
| --- | --- | --- | --- |
| Modules required:   * Python Image Library (PIL) * Requests (HTTP Client Library) * ReportLab (PDF Creation Library) * email (Email Construction) * psutil (Processes and system utilization) * shutil (File operations) * smtplib (Sending emails) | | | |
| Outline overview script steps:   1. When a supplier has new products, an image and description of that product will be supplied. (Could be any image format.) 2. When the image and description is received, write a script to automatically upload them separately to two different web endpoints.    1. The image will likely need to be formatted and converted over to one format (jpeg.)    2. The description received will be uploaded separately and also require basic formatting to be displayed on the website. 3. Once the website has been updated with the new products from the supplier, a report needs to be sent back to the supplier in PDF format.    1. Using the ReportLab module, a PDF file can be created with the images and descriptions, formatted accordingly and sent back through the smtplib and email modules. 4. Additionally, to ensure the system the script is running on continues to function, a script must be written that actively monitors the overall health of the system. 5. If an issue is detected, an email is sent to the appropriate address informing them of this issue. | | | |
| Create overview outline. | August 26th | Yes | K.T.H. |
| Script to upload images (convert if necessary.) |  |  |  |
| Script to upload descriptions. |  |  |  |
| Create a PDF report of what’s been imported/uploaded and send it to the supplier. |  |  |  |
| Write a script to monitor system health. |  |  |  |
| Write a script that sends an email if specific system health metrics fall below a certain threshold. |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |