Example.py

#!/usr/bin/env python3

import emails

import os

import reports

table\_data=[

['Name', 'Amount', 'Value'],

['elderberries', 10, 0.45],

['figs', 5, 3],

['apples', 4, 2.75],

['durians', 1, 25],

['bananas', 5, 1.99],

['cherries', 23, 5.80],

['grapes', 13, 2.48]]

reports.generate("/tmp/report.pdf", "A Complete Inventory of My Fruit", "This is all my fruit.", table\_data)

sender = "sender@example.com"

receiver = "{}@example.com".format(os.environ.get('USER'))

subject = "List of Fruits"

body = "Hi\n\nI'm sending an attachment with all my fruit."

message = emails.generate(sender, receiver, subject, body, "/tmp/report.pdf")

emails.send(message)

cars.py

1. *#!/usr/bin/env python3*
3. **import** collections
4. **import** json
5. **import** locale
6. **import** mimetypes
7. **import** os.path
8. **import** reports
9. **import** emails
10. **import** sys

13. **def** load\_data(filename):
14. """Loads the contents of filename as a JSON file."""
15. **with** open(filename) **as** json\_file:
16. data = json.load(json\_file)
17. **return** data

20. **def** format\_car(car):
21. """Given a car dictionary, returns a nicely formatted name."""
22. **return** "{} {} ({})".format(
23. car["car\_make"], car["car\_model"], car["car\_year"])

26. **def** process\_data(data):
27. """Analyzes the data, looking for maximums.
29. Returns a list of lines that summarize the information.
30. """
31. locale.setlocale(locale.LC\_ALL, 'en\_US.UTF8')
32. max\_sales = {"total\_sales": 0}
33. max\_revenue = {"revenue": 0}
34. car\_year\_sales = collections.defaultdict(int)
35. **for** item **in** data:
36. *# We need to convert "$1234.56" into 1234.56*
37. item\_price = locale.atof(item["price"].strip("$"))
38. item\_revenue = item["total\_sales"] \* item\_price
39. **if** item\_revenue > max\_revenue["revenue"]:
40. item["revenue"] = item\_revenue
41. max\_revenue = item
43. **if** item["total\_sales"] > max\_sales["total\_sales"]:
44. max\_sales = item
45. car\_year\_sales[item["car"]["car\_year"]] += item["total\_sales"]
47. max\_car\_sales\_year = (0,0)
48. **for** year, sales **in** car\_year\_sales.items():
49. **if** sales > max\_car\_sales\_year[1]:
50. max\_car\_sales\_year = (year,sales)
51. summary = []
52. summary.append("The {} generated the most revenue: ${}".format(
53. format\_car(max\_revenue["car"]), max\_revenue["revenue"]))
54. summary.append("The {} had the most sales: {}".format(
55. format\_car(max\_sales["car"]), max\_sales["total\_sales"]))
56. summary.append("The most popular year was {} with {} sales.".format(
57. max\_car\_sales\_year[0], max\_car\_sales\_year[1]))
58. **return** summary

61. **def** cars\_dict\_to\_table(car\_data):
62. """Turns the data in car\_data into a list of lists."""
63. table\_data = [["ID", "Car", "Price", "Total Sales"]]
64. **for** item **in** car\_data:
65. table\_data.append([item["id"], format\_car(item["car"]), item["price"], item["total\_sales"]])
66. **return** table\_data

69. **def** main(argv):
70. data = load\_data(os.path.expanduser('~') + "/car\_sales.json")
71. summary = process\_data(data)
73. *# Generate a paragraph that contains the necessary summary*
74. paragraph = "<br/>".join(summary)
75. *# Generate a table that contains the list of cars*
76. table\_data = cars\_dict\_to\_table(data)
77. *# Generate the PDF report*
78. title = "Sales summary for last month"
79. attachment = "/tmp/cars.pdf"
80. reports.generate(attachment, title, paragraph, table\_data)
82. *# Send the email*
83. sender = "automation@example.com"
84. receiver = "{}@example.com".format(os.environ.get('USER'))
85. body = "**\n**".join(summary)
86. message = emails.generate(sender, receiver, title, body, attachment)
87. emails.send(message)

90. **if** \_\_name\_\_ == "\_\_main\_\_":
91. main(sys.argv)