```python
"""entry_detect.py | Robin Forestier | 28.03.2022

[WARN] The camera is placed on top of the door.

This file is used to detect entry and exit. It uses PersonneDetect & PersonneTracking.
"""

# Imports
from datetime import datetime
import cv2
import requests

# import personally module
from sample.personne_detect import PersonneDetect
from sample.personne_tracking import PersonneTracking


def send(info):
    """Send the data to the server

    :param info: the information to send
    :type info: int
    """

    # It's getting the current time.
    t = datetime.now()
    current_time = t.strftime("%H:%M")

    # It's opening the file /sys/class/thermal/thermal_zone0/temp and reading the temperature.
    try:
        with open('/sys/class/thermal/thermal_zone0/temp', 'r') as ftemp:
            temp = int(int(ftemp.read()) / 1000)
    except OSError:
        temp = 0

    # It's creating a string that will be sent to the server.
    data = "{}{:03d}{}".format(current_time, temp, info)
    data = {'data': '$,RPWCSD,{:03d},{},0*'.format(len(data), data)}

    try:
        # It's sending the data to the server.
        # Change the url to your own server.
        r = requests.post("http://172.16.32.133/camera", data=data, timeout=0.5)

        # This is checking if the status code is bigger than 299. If it is, it's printing an error message.
        if r.status_code > 299:
            print("[Error] Communication error, code : ", r.status_code)
        else:
            # It's getting the data from the server.
            data = r.text
            # if the data is a correct trame ($,...,*)
            if data[0] == "$" and data[::-1][0] == "*":
                data = data.split(',')

                # Communication OK
                if data[1] == "RPWCOK":
                    print("ok")
                # Communication Error
                if data[1] == "RPWCER":
                    print("[ERROR] The cam had send a bad trame.")

    # It's catching the error if the server is not available.
    except requests.exceptions.RequestException as e:
        print(e)


if __name__ == '__main__':
    # It's opening the webcam.
    cap = cv2.VideoCapture(0)

```

```
71      # It's creating a PersonneDetect object and storing it in the variable `detect`.
72      detect = PersonneDetect()
73      # It's creating a PersonneTracking object and storing it in the variable
        `tracking`.
74      tracking = PersonneTracking()
75
76      while True:
77          # It's getting the frame from the webcam.
78          ret, frame = cap.read()
79
80          # It's resizing the frame to 640x480.
81          frame = cv2.resize(frame, (640, 480), interpolation=cv2.INTER_AREA)
82
83          # It's detecting people in the frame.
84          frame = detect.personne_detect(frame)
85          # It's calculating the centroid of the detected people.
86          tracking.calc_centroide(frame, detect.detected)
87
88          inouts = tracking.inout
89
90          for inout in inouts:
91              if inout == 1:
92                  print("[INFO] Entrée")
93                  send(inout)
94              else:
95                  print("[INFO] Sortie")
96                  send(inout)
97
98          tracking.inout.clear()
99
100         # It's showing the image in a window.
101         cv2.imshow("image", frame)
102
103         t = datetime.now()
104         print(t.strftime("%H:%M"))
105
106         # It's checking if the user press the key "q". If it is, it's breaking the
            loop.
107         if cv2.waitKey(1) == ord("q"):
108             break
109
110     # It's closing the webcam and the window.
111     cv2.destroyAllWindows()
112     cap.release()
113
```