#!/usr/bin/env python3

import sys

import pickle

import pika

import datetime

import time

import threading

import msvcrt

import npyscreen

import curses

class ChatApp(npyscreen.NPSAppManaged):

def onStart(self):

self.addForm("MAIN", MainForm, name="Chat", color="IMPORTANT",)

self.connection = pika.BlockingConnection(

pika.ConnectionParameters(

host='hub.zaikin.su',

credentials=pika.PlainCredentials('guest', 'surgu2019')))

self.channel = self.connection.channel()

#self.nickname = ''

# self.channel.queue\_declare(queue=self.nickname)

# self.channel.queue\_bind(exchange='amq.fanout', queue=self.nickname)

def onCleanExit(self):

npyscreen.notify\_wait("Goodbye!")

self.connection.close()

class MessageBox(npyscreen.BoxTitle):

\_contained\_widget = npyscreen.TitleMultiLine

class ConnectionButton(npyscreen.ButtonPress):

def whenPressed(self):

#self.parent.parentApp.nickname = self.parent.nickname.value

self.parent.parentApp.channel.queue\_declare(queue=self.parent.nickname.value)

self.parent.parentApp.channel.queue\_bind(exchange='amq.fanout', queue=self.parent.nickname.value)

self.parent.thread.start()

class MainForm(npyscreen.ActionForm):

def Call(self):

self.parentApp.channel.basic\_qos(prefetch\_count=100)

self.parentApp.channel.basic\_consume(queue=self.nickname.value, on\_message\_callback=self.callback)

self.parentApp.channel.start\_consuming()

def callback(self, ch, method, properties, body):

self.messageBox.values += ('InputMessage:',body.decode('utf-8'))

self.messageBox.display()

def create(self):

self.nickname = self.add(npyscreen.TitleText, name = "Nickname:", value= "")

self.connectionButton = self.add(ConnectionButton, name = "Connect", relx=16, rely=4)

self.sentfield = self.add(npyscreen.TitleText, name = "Send:", value="",rely=6)

self.messageBox = self.add(MessageBox, name = "Message", editable = False, max\_height=-1, rely=8, scroll\_exit=True)

self.thread = threading.Thread(target=self.Call, name='Thread')

def on\_ok(self):

message = datetime.datetime.fromtimestamp(time.time()).strftime('%H:%M:%S')+" "+self.nickname.value+": "+self.sentfield.value

self.parentApp.channel.basic\_publish(

exchange='amq.fanout',

routing\_key='',

body=message,

properties=pika.BasicProperties(

delivery\_mode=2,)

)

def on\_cancel(self):

self.parentApp.switchForm(None)

if \_\_name\_\_ == '\_\_main\_\_':

ChatApp().run()

