import config  
import telebot  
import requests  
from telebot import types  
from bs4 import BeautifulSoup as BS  
  
bot = telebot.TeleBot(config.token)  
  
# Декодировать json  
response = requests.get(config.url).json()  
  
@bot.message\_handler(commands=['start', 'help'])  
def send\_welcome(message):  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True, row\_width=2)  
 itembtn1 = types.KeyboardButton('Погода')  
 itembtn2 = types.KeyboardButton('Курсы валют')  
 markup.add(itembtn1, itembtn2)  
  
 msg = bot.send\_message(message.chat.id, "Выберите:", reply\_markup=markup)  
 bot.register\_next\_step\_handler(msg, process\_select\_step)  
  
def process\_select\_step(message):  
 try:  
 if (message.text == 'Курсы валют'):  
 coins(message)  
 elif (message.text == 'Погода'):  
 weather(message)  
 else:  
 send\_welcome(message)  
  
 except Exception as e:  
 bot.reply\_to(message, 'ooops!')  
  
# Погода  
def weather(message):  
 r = requests.get('https://sinoptik.ua/%D0%BF%D0%BE%D0%B3%D0%BE%D0%B4%D0%B0-%D0%B0%D0%BB%D0%BC%D0%B0%D1%82%D1%8B')  
 html = BS(r.content, 'html.parser')  
  
 for el in html.select('#content'):  
 t\_min = el.select('.temperature .min')[0].text  
 t\_max = el.select('.temperature .max')[0].text  
 text = el.select('.wDescription .description')[0].text  
  
 # убрать клавиатуру  
 markup = types.ReplyKeyboardRemove(selective=False)  
  
 bot.send\_message(message.chat.id, "Привет, погода на сегодня:\n" +  
 t\_min + ', ' + t\_max + '\n' + text, reply\_markup=markup)  
  
# Курсы валют  
def coins(message):  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True, row\_width=2)  
 itembtn1 = types.KeyboardButton('USD')  
 itembtn2 = types.KeyboardButton('EUR')  
 itembtn3 = types.KeyboardButton('RUR')  
 itembtn4 = types.KeyboardButton('BTC')  
 markup.add(itembtn1, itembtn2, itembtn3, itembtn4)  
  
 msg = bot.send\_message(message.chat.id,  
 "Узнать наличный курс ПриватБанка (в отделениях)", reply\_markup=markup)  
 bot.register\_next\_step\_handler(msg, process\_coin\_step)  
  
def process\_coin\_step(message):  
 try:  
 # убрать клавиатуру  
 markup = types.ReplyKeyboardRemove(selective=False)  
  
 for coin in response:  
 if (message.text == coin['ccy']):  
 bot.send\_message(message.chat.id, printCoin(coin['buy'], coin['sale']),  
 reply\_markup=markup, parse\_mode="Markdown")  
  
 except Exception as e:  
 bot.reply\_to(message, 'ooops!')  
  
def printCoin(buy, sale):  
 *'''Вывод курса пользователю'''* return "💰 \*Курс покупки:\* " + str(buy) + "\n💰 \*Курс продажи:\* " + str(sale)  
  
# Enable saving next step handlers to file "./.handlers-saves/step.save".  
# Delay=2 means that after any change in next step handlers (e.g. calling register\_next\_step\_handler())  
# saving will hapen after delay 2 seconds.  
bot.enable\_save\_next\_step\_handlers(delay=2)  
  
# Load next\_step\_handlers from save file (default "./.handlers-saves/step.save")  
# WARNING It will work only if enable\_save\_next\_step\_handlers was called!  
bot.load\_next\_step\_handlers()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 bot.polling(none\_stop=True)