|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**Отчет по лабораторной работе № 11 по курсу**

**«Информационные системы и технологии»**

" Telegram бот и XML "

Вариант № <6>

Кобзева Елизавета

СГН3-61Б

2025 г.

**Задание**

Телеграм бот должен выдавать курс валюты по отношению к рублю. Валюта

выбирается преподавателем. Дата вводится пользователем с клавиатуры

**Дополнительное задание**

Бот должен иметь клавиатуру с выбором из не менее чем 2 разных валют

**Порядок выполнения работы**

1. **Напишем скрипт программы следующим образом:**

import telebot  
from telebot import types  
import requests  
import xml.etree.ElementTree as ET  
import datetime  
from datetime import date  
  
bot = telebot.TeleBot("7687972992:AAGJNdajKIHbcYNGPm7fPX1aWKYrpAAILlk") *# 🔁 Замените на токен своего бота  
  
# Список поддерживаемых валют (код валюты: код ЦБ РФ)*CURRENCIES = {  
 "USD": "R01235",  
 "EUR": "R01239"  
}  
  
*# Словарь для хранения выбранной валюты по chat\_id*user\_state = {}  
  
  
def validate\_date(input\_date\_str):  
 try:  
 input\_date = datetime.datetime.strptime(input\_date\_str, "%d.%m.%Y").date()  
 if input\_date > date.today():  
 return None  
 return input\_date  
 except ValueError:  
 return None  
  
  
def get\_currency\_rate(currency\_id, date\_req):  
 try:  
 url = f"https://www.cbr.ru/scripts/XML\_daily.asp?date\_req={date\_req.strftime('%d/%m/%Y')}"  
 response = requests.get(url, timeout=10)  
 response.raise\_for\_status()  
  
 root = ET.fromstring(response.text)  
  
 *# Получаем дату курса* actual\_date = root.attrib.get("Date", date\_req.strftime("%d.%m.%Y"))  
  
 *# Ищем нужную валюту* for valute in root.findall("Valute"):  
 if valute.attrib["ID"] == currency\_id:  
 nominal = valute.find("Nominal").text  
 name = valute.find("Name").text  
 value = valute.find("Value").text.replace(",", ".")  
 return {  
 "date": actual\_date,  
 "nominal": nominal,  
 "name": name,  
 "value": value  
 }  
 return None  
 except (requests.RequestException, ET.ParseError) as e:  
 print(f"Error fetching currency rate: {e}")  
 return None  
  
  
@bot.message\_handler(commands=['start', 'help'])  
def start(message):  
 markup = types.ReplyKeyboardMarkup(resize\_keyboard=True, one\_time\_keyboard=True)  
 markup.add(\*CURRENCIES.keys())  
 bot.send\_message(message.chat.id, "Выберите валюту:", reply\_markup=markup)  
  
  
@bot.message\_handler(func=lambda message: message.text in CURRENCIES)  
def choose\_currency(message):  
 user\_state[message.chat.id] = {"currency": message.text}  
 bot.send\_message(message.chat.id, "Теперь введите дату в формате ДД.ММ.ГГГГ:")  
  
  
@bot.message\_handler(func=lambda message: True)  
def handle\_date(message):  
 chat\_id = message.chat.id  
 if chat\_id not in user\_state or "currency" not in user\_state[chat\_id]:  
 bot.send\_message(chat\_id, "Сначала выберите валюту, нажав /start.")  
 return  
  
 *# Проверяем дату* input\_date = validate\_date(message.text)  
 if not input\_date:  
 bot.send\_message(chat\_id, "Неверная дата. Введите дату в формате ДД.ММ.ГГГГ (не будущую дату).")  
 return  
  
 *# Получаем данные о валюте* currency\_name = user\_state[chat\_id]["currency"]  
 currency\_data = get\_currency\_rate(CURRENCIES[currency\_name], input\_date)  
  
 if not currency\_data:  
 bot.send\_message(chat\_id, f"Не удалось получить курс {currency\_name} на указанную дату.")  
 return  
  
 *# Форматируем ответ* response = (f"Курс ЦБ РФ на {currency\_data['date']}:\n"  
 f"{currency\_data['nominal']} {currency\_data['name']} = {currency\_data['value']} руб.")  
  
 bot.send\_message(chat\_id, response)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("Бот запущен...")  
 bot.polling()

1. **Пример работающей программы:**

