**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая кибернетика и информационные технологии»

**Отчет по лабораторной работе №3**

по дисциплине «Введение в информационные технологии» на тему:

Телеграм-бот.

Выполнил: студент группы БВТ2108

Чугунова Татьяна Александровна

Проверил:

Мкртчян Грач Маратович

Москва

2021

1. Цель работы:

Цель работы – создание веб-приложения, которое может переводить фразы на русском языке на английский язык, а фразы на английском языке – на русский.

1. Ход работы

Для начала создадим новый проект в IDE PyCharm и назовем его «parse\_corn\_bot» (рис.1):

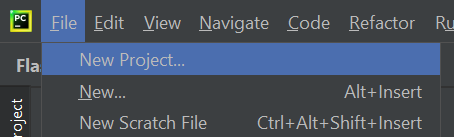


Рисунок 1 Создание проекта

Необходимо создать файл main.py (рис.2):

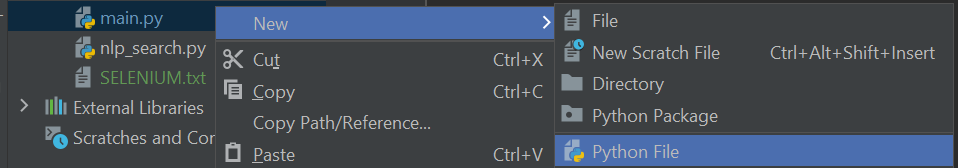


Рисунок 2 Создание файла

Добавляем в него следующий код:

import telebot

from telebot import types

import datetime

import os

from babel.dates import format\_date, format\_datetime

from selenium import webdriver

from selenium.webdriver.chrome.options import Options

from selenium.webdriver.common.by import By

import pprint

import asyncio

from bs4 import BeautifulSoup

import requests

from urllib.request import urlopen

from lxml import etree

import re

TOKEN = '20234556:AAF0Baan6osddgshKLwCfhb0O4utKv4dhdfnclaZEA'

bot = telebot.TeleBot(TOKEN)

x = datetime.datetime.today()

x2 = datetime.datetime.date(x)

date\_today = format\_date(x2, locale='de\_DE')

@bot.message\_handler(commands=['start', 'help'])

def send\_welcome(message):

bot.send\_message(message.chat.id, "Привет новичок! Я новостной бот. :)")

MainMenue(message)

def MainMenue(message):

markup = types.InlineKeyboardMarkup()

button5 = types.InlineKeyboardButton(text="Поиск", callback\_data="FindByWord")

button6 = types.InlineKeyboardButton(text="Добавить синоним", callback\_data="AddSynonim")

markup.add(button5).add(button6)

bot.send\_message(message.chat.id,

"Здесь Вы можете найти новость по ключевому слову. Добавление синонимов улучшает качество поиска!",

reply\_markup=markup)

async def zerno\_ru():

# date\_today = '03.12.2021'

xPATH = '''//\*[text()='%s']//../../span[2]/span/a''' % date\_today

xPATH\_link = '''//\*[text()='%s']//../../span[2]/span/a/@href''' % date\_today

URL = 'https://zerno.ru/news\_list'

# response = urlopen(url)

# htmlparser = etree.HTMLParser()

# tree = etree.parse(response, htmlparser)

# a = tree.xpath(xPATH\_link)

webpage = requests.get(URL)

soup = BeautifulSoup(webpage.content, "html.parser")

dom = etree.HTML(str(soup))

count\_index = dom.xpath(xPATH)

# count\_index = len(count\_index)

# поиск содержимого блоков

cell\_news\_arr = []

for i in range(0, len(count\_index)):

a = dom.xpath(xPATH)[i].text

cell\_news\_arr.append(a)

# print(cell\_news\_arr)

# поиск ссылок на содержимое блоков

news\_link\_arr = []

for i in range(0, len(count\_index)):

a = dom.xpath(xPATH\_link)[i]

news\_link\_arr.append(a)

# print(news\_link\_arr)

full\_links = []

for i in news\_link\_arr:

first\_part\_link = 'https://zerno.ru'

i = first\_part\_link + i

full\_links.append(i)

message\_text = ["%s %02s" % t for t in zip(cell\_news\_arr, full\_links)]

# print(message\_text)

return message\_text

async def asyncFindNews(message):

task1 = asyncio.create\_task(zerno\_ru())

task2 = asyncio.create\_task(zol\_ru())

task3 = asyncio.create\_task(agroinvestor\_ru())

task4 = asyncio.create\_task(agriculture\_com())

await asyncio.wait([task1,task2,task3,task4])

# await asyncio.wait([task1])

message\_words = message.text.lower().split(' ')

print('\nмеседж ловер')

try:

n1 = task1.result()

a = n1

print('ПОЛНЫЙ СПИСОК', a)

except Exception:

print('ОШИБКА ПОИСКА')

for elem\_r\_set in ParseResult:

# bot.send\_message(message.chat.id, elem\_r\_set)

bot.send\_message(message.chat.id, elem\_r\_set, disable\_web\_page\_preview=True, parse\_mode='html')

1. Результат

У нас получился следующий бот. Запустим его:

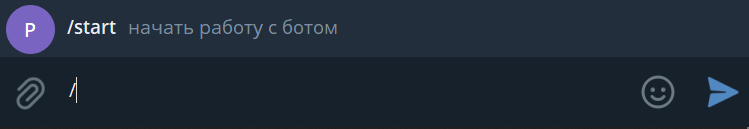


Рисунок Запуск бота

После запуска бот спарсил данные с сайта и передал пользователю данные, отфильтрованные по слову «зерно»:

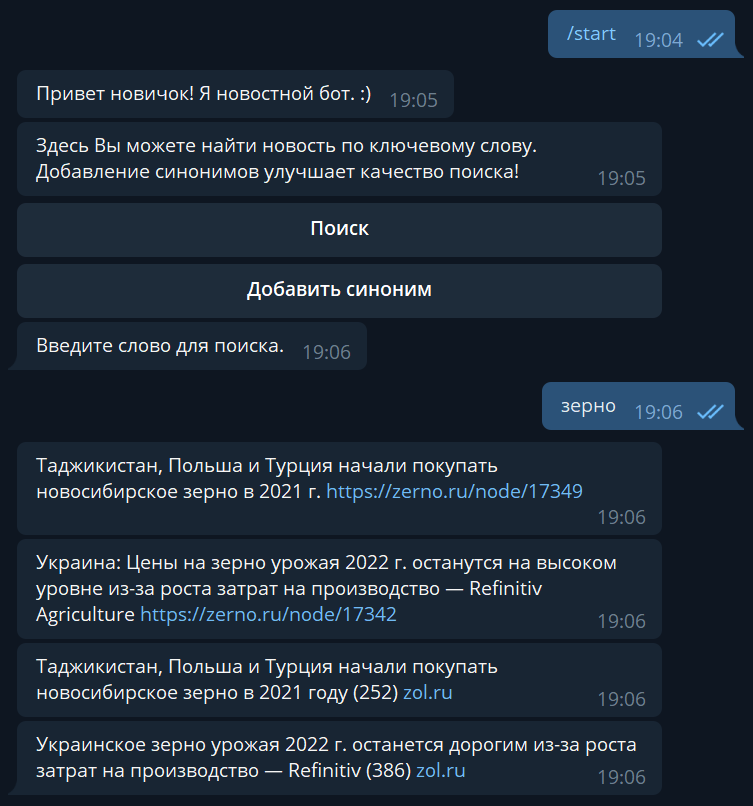


Рисунок Поиск по слову

1. Вывод

Мы научились пользоваться API Telegram, создав на его основе пользовательский интерфейс парсера.