**Автоматизация сбора документов из интернета: шаг за шагом**

**I. Общая схема процесса**

1. **Определить источники данных**:
   * **Официальные порталы**:
     + [pravo.gov.ru](https://pravo.gov.ru)
     + [sozd.duma.gov.ru](https://sozd.duma.gov.ru)
   * Сайты регуляторов: ФСТЭК, Роскомнадзор, ФСБ.
   * Открытые базы данных (например, ConsultantPlus или Garant).
2. **Выбрать инструмент автоматизации**:
   * Для веб-скрапинга: BeautifulSoup, Selenium (для динамических сайтов).
   * Для API: requests или специализированные SDK.
3. **Собрать данные**:
   * Извлечение названий, ссылок и текста документов.
   * Категоризация и сохранение в базу данных.
4. **Настроить регулярные обновления**:
   * Использовать планировщики задач (cron на Linux или Планировщик заданий на Windows).

**II. Веб-скрапинг: базовый пример**

**1. Установите необходимые библиотеки**

bash

Копировать код

pip install requests beautifulsoup4

**2. Пример кода для сбора данных**

Скрипт для сбора законов с [sozd.duma.gov.ru](https://sozd.duma.gov.ru):

python

Копировать код

import requests

from bs4 import BeautifulSoup

def fetch\_laws():

url = "https://sozd.duma.gov.ru/legislation"

response = requests.get(url)

soup = BeautifulSoup(response.text, "html.parser")

# Найдем таблицу с законами

laws = []

for row in soup.select("tr.document-row"):

title = row.find("td", class\_="document-title").text.strip()

date = row.find("td", class\_="document-date").text.strip()

link = row.find("a", href=True)["href"]

laws.append({

"title": title,

"date": date,

"link": f"https://sozd.duma.gov.ru{link}"

})

return laws

if \_\_name\_\_ == "\_\_main\_\_":

laws = fetch\_laws()

for law in laws:

print(f"{law['date']}: {law['title']} ({law['link']})")

**III. Работа с API**

**1. Использование API, если доступно**

Некоторые сайты (например, право.gov.ru) предоставляют доступ через API. Обычно это документировано на самом портале.

Пример работы с API:

python

Копировать код

import requests

def fetch\_laws\_from\_api():

url = "https://api.pravo.gov.ru/v1/laws"

headers = {"Authorization": "Bearer YOUR\_TOKEN"}

response = requests.get(url, headers=headers)

if response.status\_code == 200:

return response.json()

else:

print(f"Ошибка: {response.status\_code}")

return []

if \_\_name\_\_ == "\_\_main\_\_":

laws = fetch\_laws\_from\_api()

for law in laws:

print(f"{law['name']} - {law['url']}")

**2. Работа с RSS-каналами**

Если сайт предоставляет RSS, можно использовать feedparser:

bash

Копировать код

pip install feedparser

Пример:

python

Копировать код

import feedparser

def fetch\_rss\_updates():

feed\_url = "https://www.pravo.gov.ru/rss/laws"

feed = feedparser.parse(feed\_url)

for entry in feed.entries:

print(f"{entry.title}: {entry.link}")

if \_\_name\_\_ == "\_\_main\_\_":

fetch\_rss\_updates()

**IV. Сохранение данных**

**1. Хранение в файлах**

Сохраняйте законы в формате JSON или TXT:

python

Копировать код

import json

def save\_to\_file(data, filename="laws.json"):

with open(filename, "w", encoding="utf-8") as file:

json.dump(data, file, ensure\_ascii=False, indent=4)

**2. Хранение в базе данных**

Используйте PostgreSQL или SQLite для хранения:

python

Копировать код

import sqlite3

def save\_to\_database(laws):

conn = sqlite3.connect("laws.db")

cursor = conn.cursor()

cursor.execute("""

CREATE TABLE IF NOT EXISTS laws (

id INTEGER PRIMARY KEY,

title TEXT,

date TEXT,

link TEXT

)

""")

for law in laws:

cursor.execute("""

INSERT INTO laws (title, date, link) VALUES (?, ?, ?)

""", (law['title'], law['date'], law['link']))

conn.commit()

conn.close()

**V. Автоматизация запуска**

**1. На Linux (cron)**

Открыть crontab:

bash

Копировать код

crontab -e

Добавить задачу:

bash

Копировать код

0 3 \* \* \* python3 /path/to/your\_script.py

**2. На Windows (Планировщик задач)**

* Создайте новую задачу.
* Укажите путь к Python и скрипту.
* Настройте расписание запуска.

**VI. Настройка уведомлений**

**1. Email-уведомления**

Отправляйте список новых законов на почту:

python

Копировать код

import smtplib

from email.mime.text import MIMEText

def send\_email(updates):

sender = "your\_email@example.com"

recipient = "recipient@example.com"

body = "\n".join([f"{u['date']}: {u['title']} ({u['link']})" for u in updates])

msg = MIMEText(body, "plain")

msg["Subject"] = "Новые законы"

msg["From"] = sender

msg["To"] = recipient

with smtplib.SMTP("smtp.example.com", 587) as server:

server.starttls()

server.login("your\_email@example.com", "your\_password")

server.sendmail(sender, recipient, msg.as\_string())

**2. Telegram-уведомления**

Используйте Telegram API:

python

Копировать код

import requests

def send\_telegram\_message(updates):

bot\_token = "YOUR\_BOT\_TOKEN"

chat\_id = "YOUR\_CHAT\_ID"

message = "\n".join([f"{u['date']}: {u['title']} ({u['link']})" for u in updates])

url = f"https://api.telegram.org/bot{bot\_token}/sendMessage"

requests.post(url, data={"chat\_id": chat\_id, "text": message})

**VII. Итог**

Скрипт автоматически:

1. Скачивает законы с указанных порталов.
2. Сохраняет данные в файл или базу данных.
3. Уведомляет об изменениях через Email или Telegram.

Если нужен готовый код или помощь с настройкой под конкретный портал, сообщите!