Скрипт-скрапер решений по рекламным делам с сайта ФАС России

Последовательность выполнения скрипта

Этап 1: Ввод временного периода. Пользователь вводит даты в формате dd.mm.yyyy-dd.mm.yyyy, например "01.01.2024-31.01.2024". Система валидирует формат через datetime.strptime() и создает уникальный SESSION_ID, объединяя текущую дату-время с датами периода: "20240315_143022_01012024_31012024".
SESSION_ID нужен на следующих шагах пайплайна (сборка таблицы релевантных для RAG дел из сгенерированных нейросетью json-ов).

Этап 2: Загрузка существующего лога. Функция load_existing_log() читает Excel-файл с ранее собранными данными и извлекает множество case_ids (UUID, присвоенные ФАС каждой карточке с делом), а также создает словарь doc_counts с количеством документов по каждому делу. Эта информация используется для определения новых документов при повторных проверках дел.

Этап 3: Поиск новых дел по заданному периоду. Система выполняет четыре различных поисковых запроса к базовому URL с параметрами *divisions* (Управление – Управление контроля рекламы и недобросовестной конкуренции), *procedure* (Процедура – Реклама), *market_type* (Сфера - Рынок рекламы) и *tag* (Дополнительные критерии – реклама). Каждый запрос формируется через params={'start_date': start_date, 'finish_date': finish_date, 'page': page_num} и отправляется GET-запросом к корневому адресу сайта ФАС (https://br.fas.gov.ru/).

Этап 4: Извлечение ссылок на дела. Из HTML-ответа библиотека BeautifulSoup извлекает элементы по селектору 'div.grey-card > div.row > div.col-sm-10 > b > a', получая href в формате '/cases/UUID/'. Из этого href система извлекает case_id как последний сегмент после разделения по слешу, затем формирует полную ссылку через urljoin(BASE_URL, case_path).

Этап 5: Обход страниц дел и поиск документов решений. Для каждого case_url система выполняет GET-запрос и парсит страницу дела. BeautifulSoup ищет все div.grey-card, затем внутри каждого проверяет наличие div с inline-стилем font-weight: 600, содержащего слово "*решение*" в тексте.

Этап 6: Формирование ссылок на документы. Внутри найденных карточек документов система ищет теги 'a[href*="/to/"]' и извлекает href. Из полученной ссылки последний сегмент после разделения становится doc_id, а полный URL формируется через urljoin() с базовым адресом сайта.

Этап 7: Проверка обновлений в существующих делах Функция check_case_updates() берет известный case_id, формирует URL как f"{BASE_URL}cases/{case_id}/" и сравнивает текущее количество документов решений с ранее сохраненным в doc_counts. Документы с порядковыми номерами выше known_doc_count помечаются как новые.

Этап 8: Скачивание текстов документов Для каждого doc_url система выполняет GET-запрос и ищет элемент с id='document_text_container'. Извлеченный через get_text(separator='\n', strip=True) текст сохраняется в файл с именем "{doc_id}.txt" в директории TXT_DIR.

Этап 9: Формирование итогового лога Все найденные документы объединяются в список словарей с ключами case_id, case_name, doc_id, doc_name, doc_url. Pandas создает DataFrame из этих данных, объединяет с существующим логом через concat(), удаляет дубликаты по Doc ID и сохраняет в Excel-файл.

Ключевые технические особенности

Интеллектуальная проверка обновлений Система запоминает количество документов решений для каждого дела через groupby('Case ID').size().to_dict() и при повторных проверках сравнивает текущее количество с сохраненным. Новые документы идентифицируются по превышению счетчика, что позволяет отслеживать появление решений в уже известных делах без повторного анализа всех документов.

Механизм кеширования документов Функция download_document_text() проверяет существование файла через os.path.exists() перед скачиванием. Если файл "{doc_id}.txt" уже существует в TXT_DIR, загрузка пропускается, что экономит время и трафик при повторных запусках скрипта.

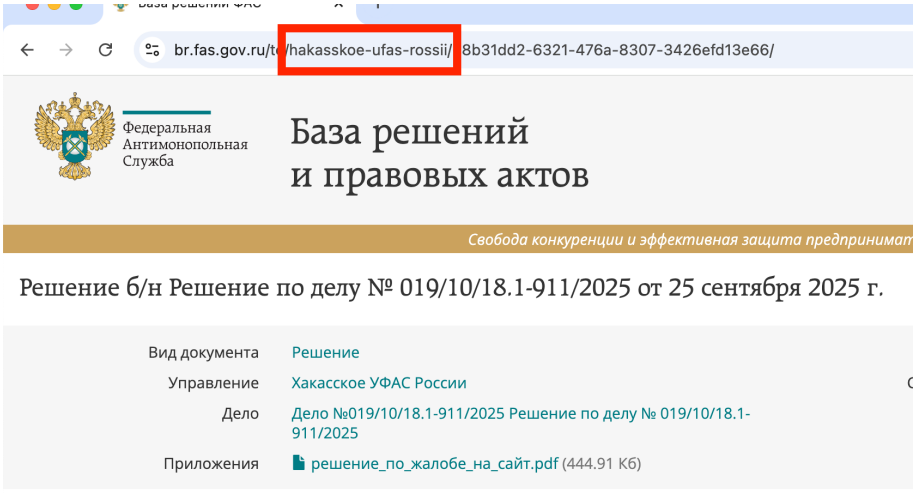
Вежливое обращение с сервером Между запросами устанавливаются паузы через time.sleep(0.5) для основных запросов и time.sleep(0.3) при проверке обновлений. Все HTTP-запросы выполняются с заголовками HEADERS, имитирующими браузерное поведение для снижения вероятности блокировки.

Обработка множественных поисковых критериев Массив SEARCH_QUERIES содержит четыре различных набора параметров поиска, что обеспечивает максимальный охват дел через разные фильтры. Каждый набор параметров объединяется с базовыми параметрами периода через params.update(), позволяя находить дела по подразделениям, процедурам, рыночным типам и тегам.

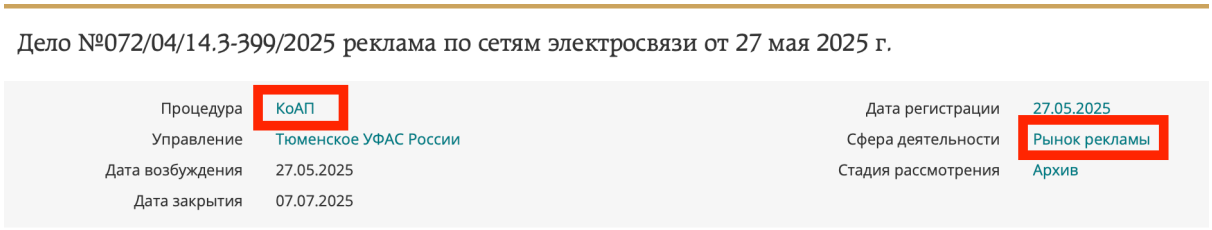
Постраничная навигация с автоматической остановкой Система увеличивает page_num в цикле while True и прекращает обход при получении HTTP 404 или отсутствии элементов case_cards на странице. Это обеспечивает полный сбор данных без заранее известного количества страниц результатов.

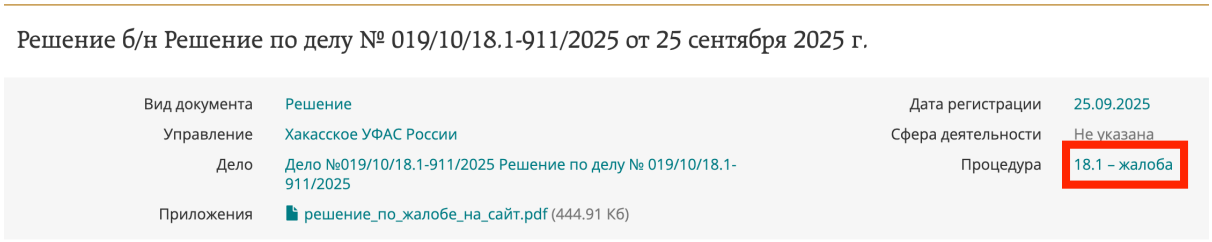
«Бутылочные горлышки» базы данных ФАС

UUID для caseID указываются в стабильном формате ссылки (https://br.fas.gov.ru/cases/[caseID], а для docID – после элемента с адресом УФАСа.

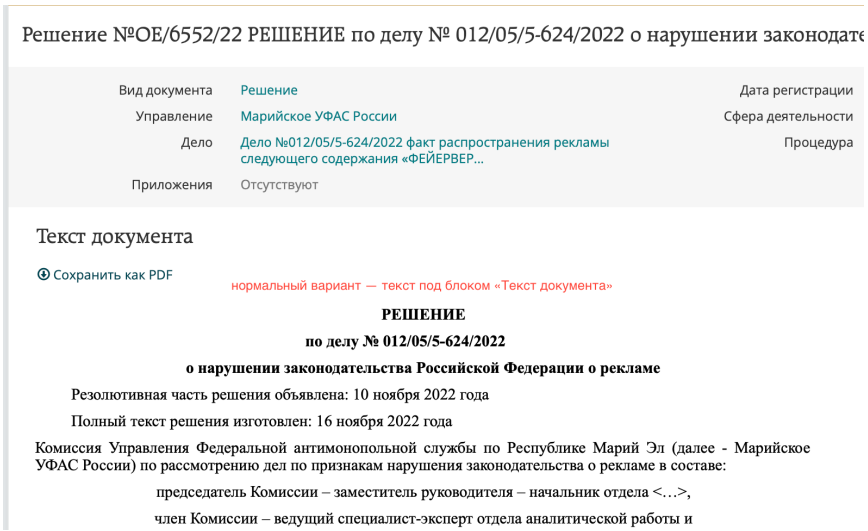


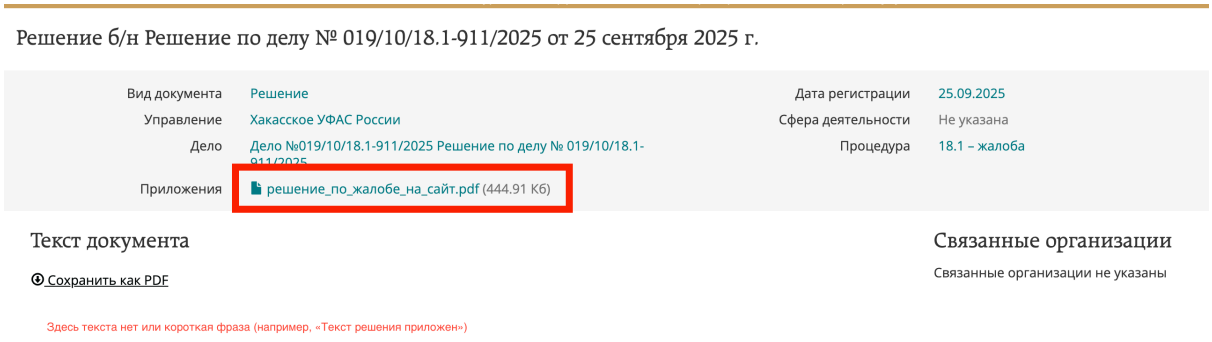
Разметка дел в базе неточна – рекламное дело может прятаться под разными фильтрами



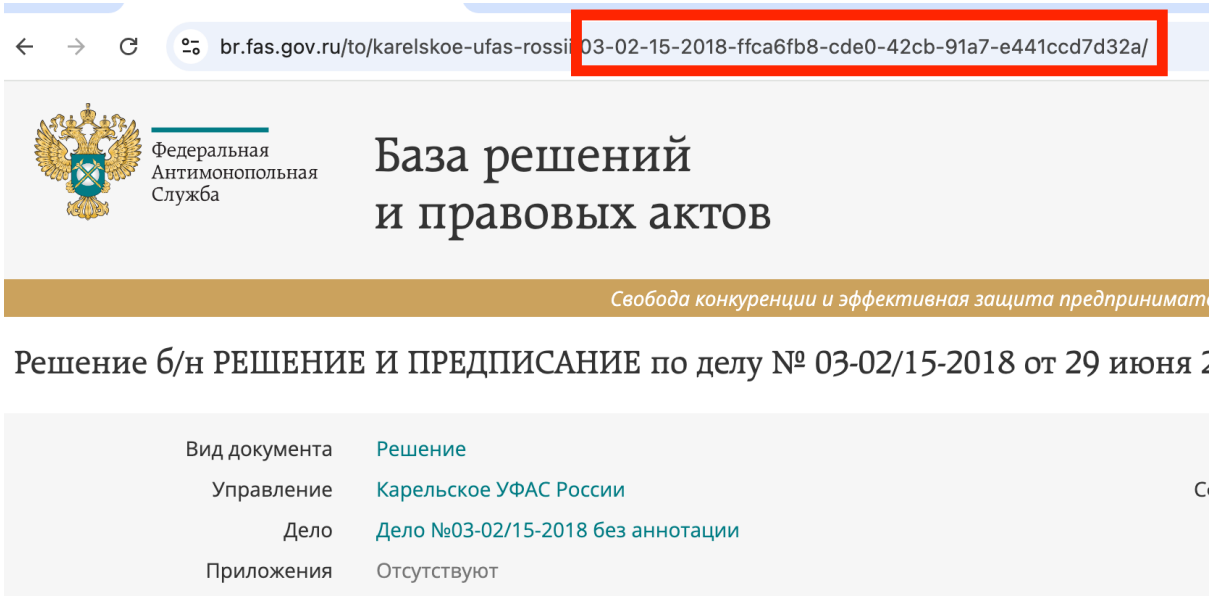


Текст решений вынесен в отдельный документ-аттач





С 2018-го года (в прошлое) docID оформляется не в формате UUID



Результаты работы скрипта

Ввод периода для скрапинга

```
Введите период для проверки/скачивания дел
Формат: dd.mm.yyyy-dd.mm.yyyy
Пример: 01.01.2024-31.01.2024
Или нажмите Enter для проверки только обновлений в существующих делах
Период: 01.06.2018-31.12.2018
```

```
Введите период для проверки/скачивания дел
Формат: dd.mm.yyyy-dd.mm.yyyy
Пример: 01.01.2024-31.01.2024
Или нажмите Enter для проверки только обновлений в существующих делах
Период:  01.06.2018-31.12.2018
✅ Период установлен: 01.06.2018 - 31.12.2018
🆔 ID сессии: 20250925_164700_01062018_31122018
```

Результат работы скрапера

```
================================================
ЭТАП 1: СКРАПИНГ И ПРОВЕРКА ОБНОВЛЕНИЙ
================================================
📊 Загружено 6081 дел из лога

🔍 Поиск дел за период 01.01.2020 - 31.08.2020
  Поиск by Division
  Поиск by Procedure
  Поиск by Sphere
  Поиск by Tag
  ✅ Найдено 1278 новых документов

🔄 Проверка обновлений в существующих делах...
Error displaying widget: model not found

📊 Всего найдено документов для обработки: 1278

📥 Скачивание текстов документов...
Error displaying widget: model not found
✅ Скачано текстов: 1102
✅ Лог обновлен: /Users/e                                    /scraped_cases_log.xlsx
```