НАЦИОНАЛЬНЫЙ ЦЕНТР
КОГНИТИВНЫХ РАЗРАБОТОК
УНИВЕРСИТЕТА ИТМО
НЦКР

Программный комплекс автоматического интеллектуального сбора данных из различных интернет источников

руководство оператора

Санкт-Петербург, 2020 г.

АННОТАЦИЯ

Документ содержит руководство оператора по использованию экспериментального образца программного комплекса автоматического интеллектуального сбора данных из различных интернет источников. Программный комплекс создан в ходе выполнения работ по созданию «Платформы поддержки жизненного цикла интеллектуальных объектов на основе промышленных больших данных», разработанной в рамках реализации программы Национального центра когнитивных разработок и договора о предоставлении гранта на государственную поддержку центров Национальной технологической инициативы на базе образовательных организаций высшего образования и научных организаций № 8/1251/2019 от 15.08.2019.

В документе представлено назначение программного комплекса автоматического интеллектуального сбора данных из различных интернет источников, описаны условия и порядок выполнения, а также предусмотренные сообщения оператору.

ОГЛАВЛЕНИЕ

1. Назначение программного комплекса

Экспериментальный образец программного комплекса «Автоматическая интеллектуальная система сбора данных из различных интернет источников» (ЭО ПК). ЭО ПК обеспечивает функции создания задач сбора данных, тестирования задач, создания сценариев сбора, запуск сценариев локально и в распределенном режиме. Он может быть использован для создания различных систем мониторинга.

ЭО ПК предназначен для решения следующих задач:

- сбор данных из различных источников;
- обеспечение мониторинга в сети интернет.

Интеллектуальность системы заключается в автоматизации процесса сбора, в том числе, идентификации оптимальных траекторий сбора, обогащении и фильтрации массива обходных ссылок.

2. Условия выполнения программы

Работа оператора осуществляется путем взаимодействия с интерфейсом, доступ к которому производится посредством web-браузера, установленного на ПЭВМ (персональную электронно-вычислительную машину) оператора. Для корректного выполнения приложения ПЭВМ оператора должна обладать следующими минимальными характеристиками:

- оперативная память – не менее 1.0 ГБ;
- дисковая подсистема – не менее 1 ГБ.

ПЭВМ оператора должна работать под управлением одной из следующих ОС:

- Windows (версии не ниже Windows 7);
- Linux (на ядре версии не ниже 2.6);
- Mac OS.

На ПЭВМ оператора должно быть установлено программное обеспечение Telegram клиент.

3. Выполнение программы

3.1. Интерфейс управления приложением

Взаимодействие оператора с ЭО ПК осуществляется посредством графического пользовательского интерфейса, загружаемого в приложение Telegram путем манипуляций отображаемыми графическими элементами. С помощью этих манипуляций пользователь может осуществлять работу по созданию и запуску сценариев сбора данных, в рамках которых обеспечивается функциональность ЭО ПК.

Непосредственно интерфейс системы представляет собой telegram-приложение. Рисунок 3.1.1. демонстрирует начало работы с системой.

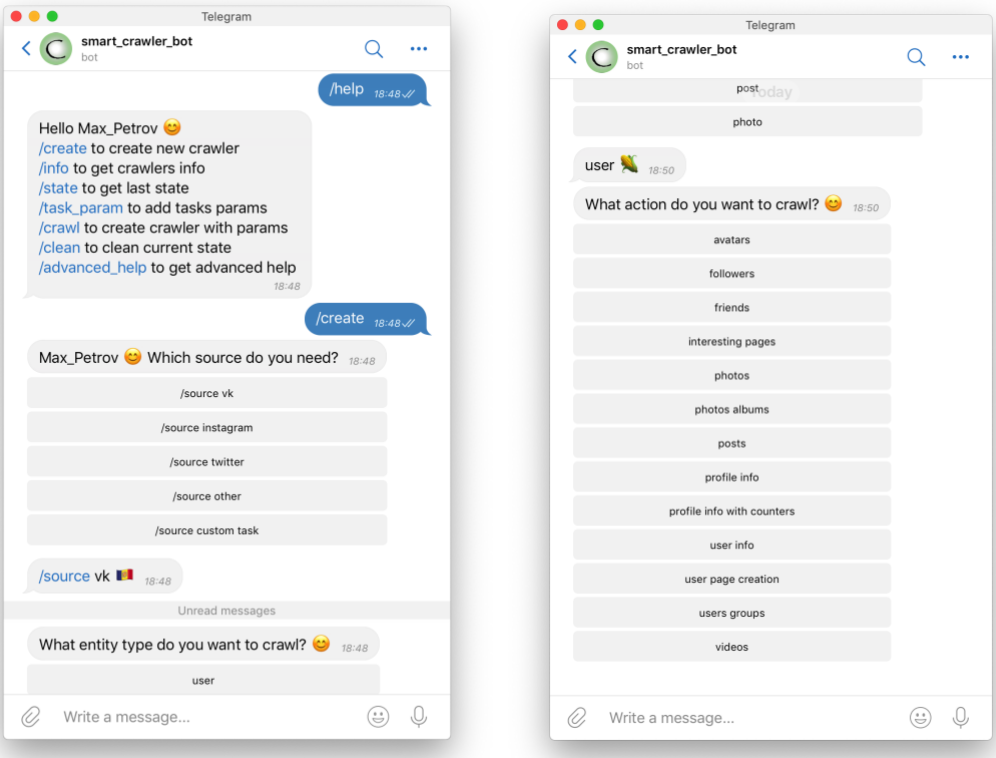
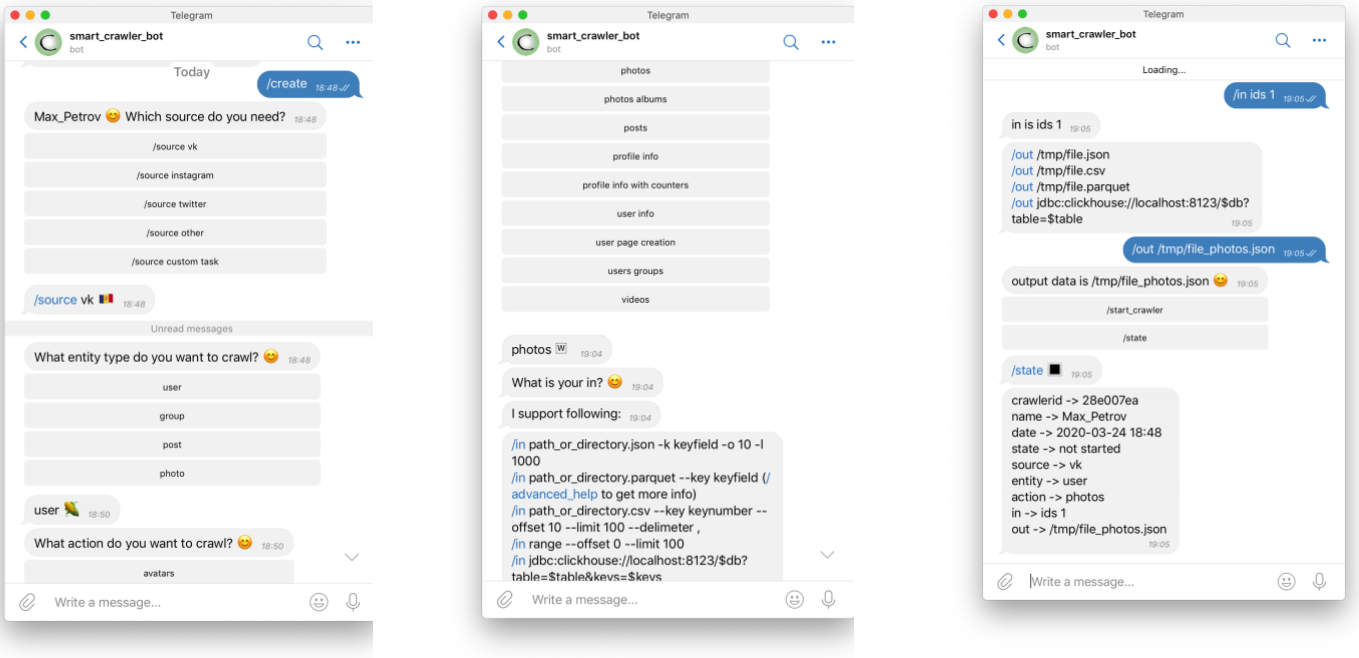


Рисунок 3.1.1. – Начало работы с системой, веб-интерфейс после запуска ЭО ПК

Авторизация в системе происходит с помощью ключа доступа, который пользователь должен получить у администраторов сервиса, отправив запрос по адресу: nasonov@itmo.ru. Для ввода ключа используется команда «/token access_token», после этого пользователь получит доступ и не обязан более вводить токен.

Интерфейс состоит преимущественно из текста и кнопок. Выделяется несколько различных областей и диалогов: диалог создания сценария, диалог сценария, диалог сценариев, диалог помощи, диалог авторизации.

Диалог создания сценария. Здесь пользователь может создать/отредактировать/запустить сценарии, используя советующие кнопки. Для этого необходимо указать источник, сущность, действие, вход и выход. Источник – это сайт, с которого будут собираться данные (например, VK). Сущность – это объект, который находится в процессе сбора (например, пользователь). Действие – это то, что вы можете сделать с сущностью (например, фото). Все это означает: «Для источника VK, для сущности пользователя будут собираться фото» (рисунок 3.1.2. (а) и (б)).



(а) (б) (в)

Рисунок 3.1.2. – Диалог создания сценария

Далее, вы должны предоставить метаданные для генерации задач. Могут быть разные источники: идентификаторы, диапазон, CSV, JSON, PARQUET, RDBS.

Идентификаторы. Например, у вас есть пользователь с идентификатором 1, затем вы можете ввести идентификатор пользователя, например, «/in ids 1». Если идентификаторов много, их можно разделить запятой, например, «/in ids 1,2,3,4,5,6,7».

Диапазон. Если у вас есть диапазон идентификаторов, например от 0 до 1000, вы можете использовать его как «/in range--offset 0 --limit 1000».

Csv. Обычно у вас есть CSV-файл с идентификаторами, и есть возможность создавать задачи на его основе. Строки должны быть разделены «\n». Вы можете перетащить файл в диалог и бот сохранит его в папке «tmp». Файл должен заканчиваться на «.csv». Пример: «/ in /mnt/shdstorage/ids.csv». У csv также есть несколько вариантов. Вы можете указать разделитель с параметром -d, например, «/ in /mnt/shdstorage/ids.csv -d». Вы можете указать номер столбца с -k (--key), например, «/ in /mnt/shdstorage/ids.csv -k 1», он будет использовать второй столбец для генерации задач. По умолчанию используется первый столбец. Если файл содержит заголовки, вы можете пропустить его с примером -o (--offset) «/ in /mnt/shdstorage/ids.csv -o 1». Если вы хотите ограничить ваш файл, то можете использовать его с примером -l (--limit) «/in /mnt/shdstorage/ids.csv -l 100», и он будет читать только 100 строк. Бот поддерживает папки и файлы. Например, папка «/mnt/shdstorage/ids.csv» содержит файлы, бот автоматически прочитает все файлы внутри папки.

Json. Почти то же, что и CSV, только файл должен заканчиваться на «.json». пример «/in /mnt/shdstorage/ids.json». С опцией -k (--key) вы можете указать имя столбца. по умолчанию используется столбец «key». Иногда json файл не валиден, вы можете пропустить строки с помощью --ignore_errors.

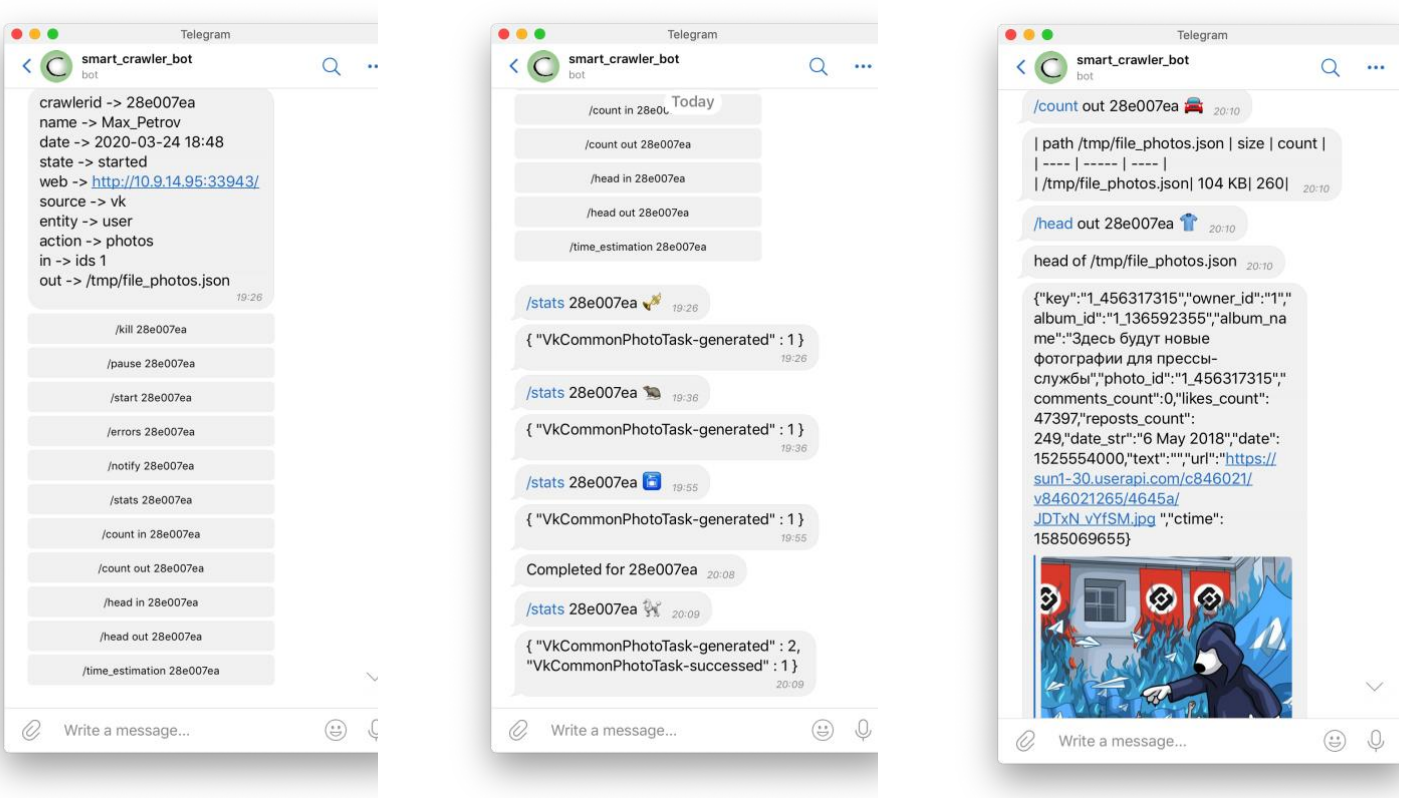
Parquet. Почти так же, как JSON только файл должен заканчиваться на «.parquet». Пример: «/ in /mnt/shdstorage/ids.parquet».

JDBC. Экспериментальный. Пример «/ in jdbc: clickhouse: // localhost: 8123 / $ db? Table = $ table & keys = $ keys». Вам нужно указать host, db, table и key для чтения. На основе значения ключа он будет генерировать задачи.

Далее, вы должны предоставить метаданные для вывода задач – out. Это может быть CSV, JSON, паркет, JDBC. Если вы не знаете формат данных – используйте json. пример «/out /mnt/shdstorage/followers.json». Если он статический, вы можете использовать CSV и parquet. Если у вас есть реляционная база данных – JDBC «/out jdbc:clickhouse://localhost:8123/$db?table=$table».

Далее, бот предложит вам запустить сценарий или посмотреть его состояние (рисунок 3.1.2. (в)).

Диалог сценария. После запуска сценария бот предоставляет новый диалог. Здесь можно остановить, приостановить сценарий, посмотреть его ошибки, статистику задач. Можно посчитать количество входных и выходных записей, а также примеры данных. В качестве экспериментальной возможности можно получить приблизительное время сбора, нажав на кнопку time_estimation (рисунок 3.1.3.).



(а) (б) (в)

Рисунок 3.1.3. – Диалог сценария

Диалог помощи вызывается с помощью команды «/advanced_help» и отображает расширенную информацию о работе бота (рисунок 3.1.4.).

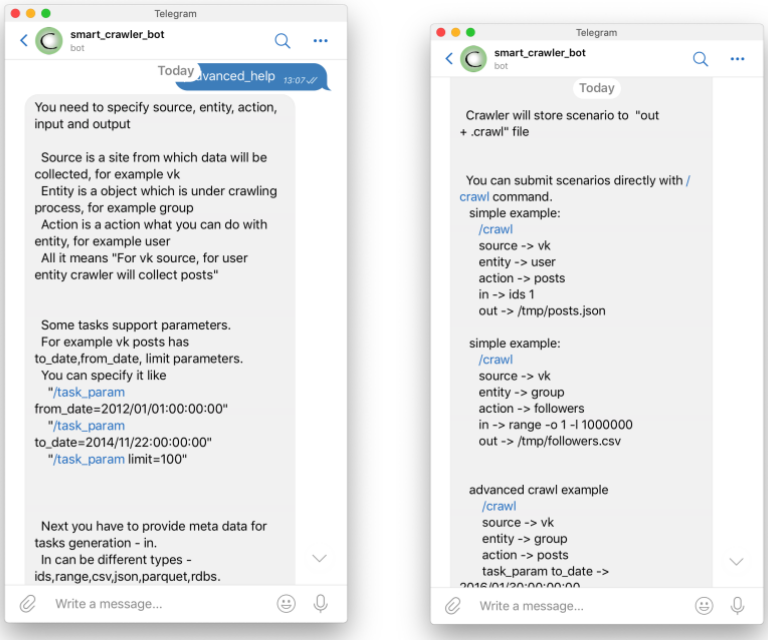


Рисунок 3.1.4. – Область построения задачи сбора

Бот способен так же принимать готовые сценарии, для этого его необходимо начать с команды /crawl, а затем вставить тело сценария. Например, на рис 3.1.5. показан запуск готового сценария сбора постов.

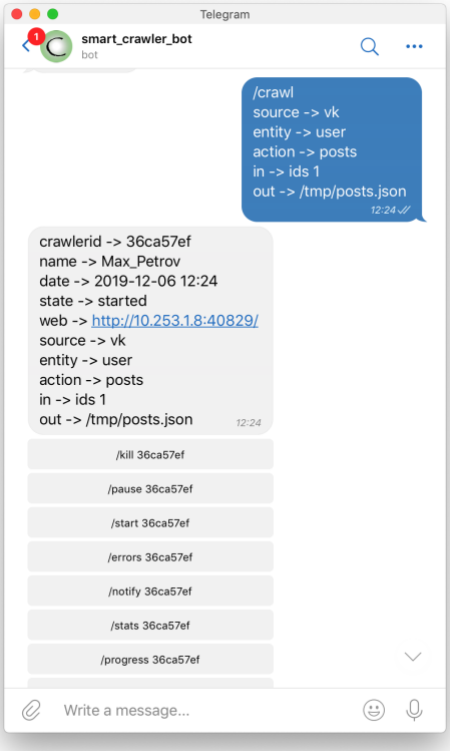


Рисунок 3.1.5. – Запуск готового сценария сбора постов

3.2 Интерфейс доступа через WEB и HTTP API

Система также поддерживает запуск сценариев через WEP и HTTP. На странице <IP crawler>/submit_scenario доступен веб интерфейс (рисунок 3.2.1.), который представляет собой набор виджетов для построения сценария аналогичному выше. Выбор сущности и действия осуществляется нажатием на соответствующие шары. По нажатию на шары in и out появляется контекстное меню ввода in и out (рисунок 3.2.2.). По нажатию на шар Scenario появляется меню показа сценария (рисунок 3.2.3.). По нажатию на кнопку Submit App, происходит оправка сценария на сервер и его запуск, в ответ приходит статус сценария и ссылка на веб интерфейс приложения (рисунок 3.2.4.).

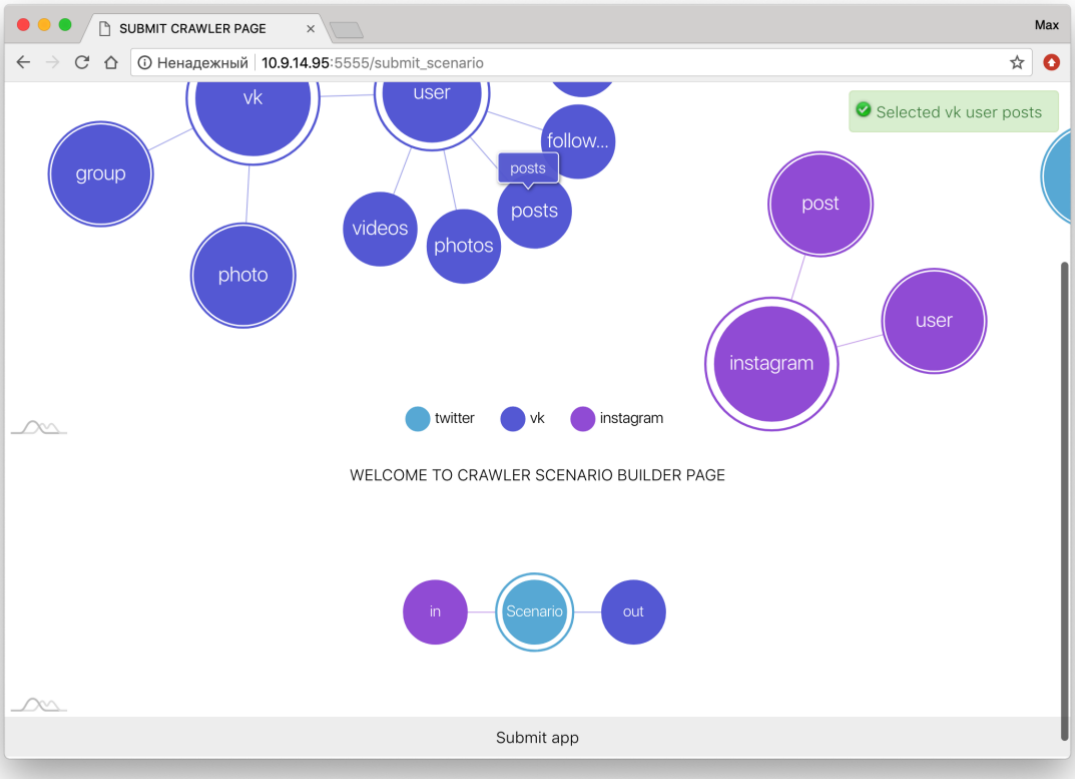


Рисунок 3.2.1. – Веб интерфейс построения сценария; выбор задачи

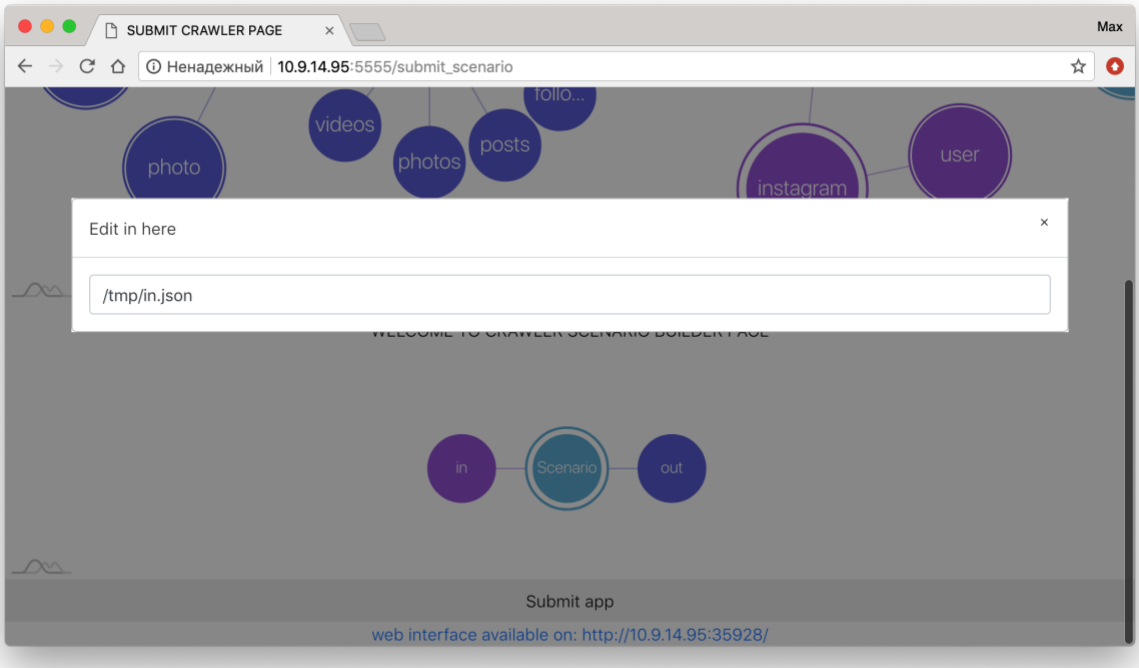


Рисунок 3.2.2. – Веб интерфейс построения сценария; ввод in

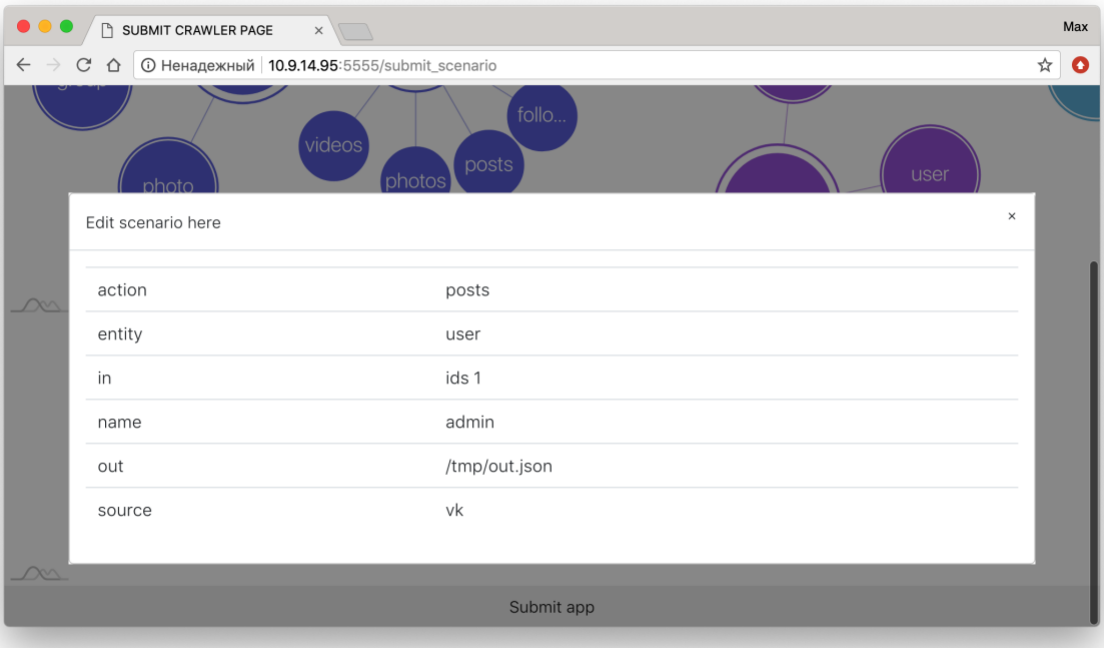


Рисунок 3.2.3. – Веб интерфейс построения сценария; содержание сценария

Рисунок 3.2.4. – Веб интерфейс построения сценария;
запуск сценария и ссылка на веб интерфейс приложения

Система поддерживает http post запросы, тело которого составляет сценарий сбора (рисунки 3.2.5. и 3.2.6.). В результате система запустит сбор данных и в ответ выдаст статус сценария и ссылку на веб интерфейс.

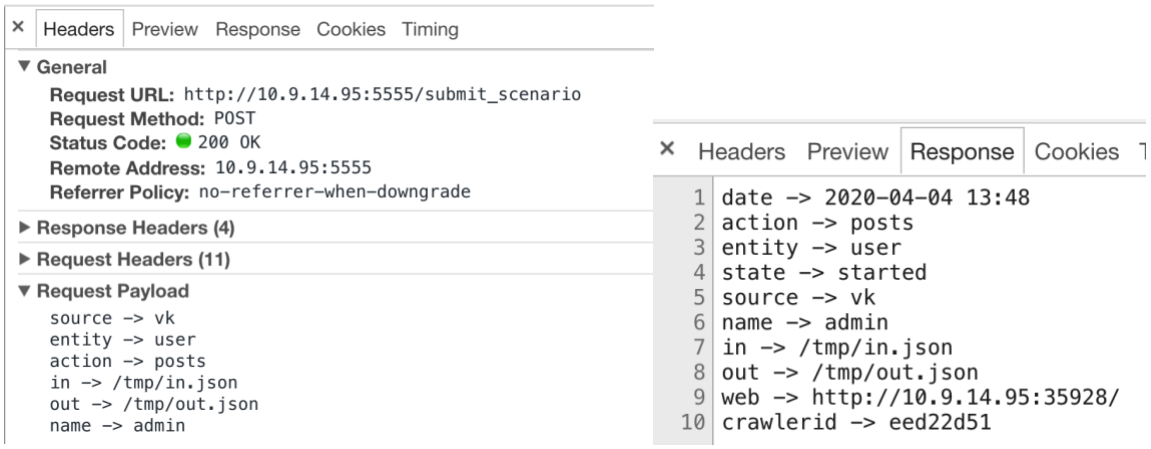


Рисунок 3.2.5. – HTTP API для запуска сценария; пример запроса и ответа

```
curl 'http://10.9.14.95:5555/submit_scenario' -H 'Origin: http://10.9.14.95:5555' -H 'Accept-Encoding: gzip, deflate' -H 'Accept-Language: ru-RU,ru;q=0.9' -H 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.167 Safari/537.36' -H 'Content-Type: application/json' -H 'Accept: application/json, text/plain, */*' -H 'Referer: http://10.9.14.95:5555/submit_scenario' -H 'Cookie: _ga=GA1.1.1348710922.1585913443; _gid=GA1.1.1733713799.1585913443' -H 'Connection: keep-alive' --data-binary $'source -> vk\nentity -> user\naction -> photos\nin -> /tmp/in.json\nout -> /tmp/out.json\nname -> admin\n' --compressed
```

Рисунок 3.2.6. – HTTP API запуска сценария через curl; пример запроса

Запрос метода API submit_scenario для выполнения HTTP сценария имеет сигнатуру аргумента, такую же, как и перечень параметров, описанных в разделе 3.1.

3.3. Решение прикладных задач

ЭО ПК позволяет решать различные задачи по созданию и запуску сценариев сбора данных. Ниже рассматриваются пример построения таких задач и сценариев в соответствии с назначением ЭО ПК.

3.3.1. Задача построения задачи сбора фотографий в vk.com

а) Описание задачи. Программный комплекс, имея информацию о идентификаторах пользователей, должен произвести сбор информации об их фотографиях в соцсети VK.

б) Условия применения.

в) Входные данные для решения задачи представляют собой набор идентификаторов пользователей.

г) Порядок запуска.

Для начала работы необходимо открыть клиент telegram и создать новый сценарий - /create. Выбрать соцсеть VK, затем выбрать user, после этого выбрать photos. Далее вводится список идентификаторов «/in ids 1,2,3» и выходной файл – «/out /tmp/file_photos.json», после этого можно запустить сценарий кнопкой /start_crawler. Бот запустит сценарий и уведомит, когда он будет выполнен

4. Сообщения оператору

Для контроля ошибок ЭО ПК используется стандартная система журналирования Nodejs. Наиболее важные ошибки и их интерпретация представлены в таблице:

Основные сообщения оператору

Сообщение	Типовые действия по выявлению и устранению ошибки
File not found	Проверить наличие файла по указанному пути
Сервер возвратил ошибку: NotFound	Обратиться в службу поддержки комплекса за устранением ошибки

Нет соединения с сетью	Проверить наличие соединения с Интернетом и попробовать снова выполнить операцию
Истекло время ожидания ответа от сервера	Проверить наличие соединения с Интернетом и попробовать снова выполнить операцию. В случае повторения обратиться в службу поддержки комплекса
Соединение принудительно прервано сервером	Попробовать снова выполнить операцию. В случае повторения обратиться в службу поддержки комплекса
Не все входные данные задачи указаны	Указать все требуемые для задачи файлы и повторить операции
Невозможно запустить задачу: недостаточно свободных ресурсов	Подождать 3–5 минут и попробовать выполнить операцию снова. В случае повторения обратиться в службу поддержки комплекса
Completed	Сценарий успешно выполнен

5. Перечень сокращений

ЭО ПК Экспериментальный образец комплекса программных средств

ПК Программный комплекс

JSON JavaScript Object Notation

URL Uniform Resource Locator

ПЭВМ Персональная электронно-вычислительная машина