

Размещение экземпляра ПО, развернутого для экспертизы

Для экспертизы экземпляр ПО предоставляется в виде дистрибутива на GitHub. Он доступен по адресу: https://github.com/tsa-korpus-parsing/TSA_Aggregator_Compose

Состав и структура дистрибутива

- Репозиторий **framework**:
 - Директория **static**: Содержит ресурсы, необходимые для работы пользовательского интерфейса.
 - Поддиректории:
 - **autocomplete_devbridge**: Файлы для автозаполнения.
 - **bootstrap-5.0.0-beta1-dist**: Файлы Bootstrap для стилизации.
 - **css**: Каскадные таблицы стилей.
 - **d3**: Библиотека D3.js для визуализации данных.
 - **icons-1.4.1**: Иконки для интерфейса.
 - **img**: Изображения.
 - **jquery-3.5.1-dist**: Библиотека jQuery.
 - **js**: Скрипты JavaScript.
 - **keyboards**: Виртуальные клавиатуры.
 - **kioskboard-1.4.0**: Файлы для реализации киоск-интерфейсов.
 - **vendor**: Дополнительные сторонние библиотеки и плагины.
 - Директория **templates**: Хранит HTML-шаблоны для различных частей веб-интерфейса.
 - Поддиректории:
 - **admin**: Шаблоны для административной панели.
 - **`index`**: Главная страница.
 - **`modals`**: Модальные окна.
 - **`search_results`**: Страница с результатами поиска.
 - Файлы:
 - **footer.html**: Шаблон подвала страницы.
 - **fulltext.html**: Шаблон для отображения полного текста.

- [head_add.html](#): Дополнительные элементы в ``<head>``.
- [header.html](#): Шаблон заголовка страницы.
- [index.html](#): Шаблон главной страницы.
- [query_area.html](#): Шаблон области ввода запросов.
- [app.py](#): Основной файл приложения на Python, запускающий сервер и обрабатывающий запросы.
- Корневые файлы:
 - [.gitmodules](#): Конфигурационный файл для Git submodules.
 - [Dockerfile](#): Файл для контейнеризации приложения.
 - [README.md](#): Документация по установке и использованию приложения.
 - [docker-compose.yml](#): Файл для развертывания приложения с помощью Docker Compose.

Настройка программы

Для настройки программы следует:

1. Клонировать репозиторий с подмодулями: `git clone --recurse-submodules https://github.com/tsa-korpus-parsing/TSA_Aggregator_Compose`
2. Перейти в директорию проекта: `cd TSA_Aggregator_Compose`
3. Запустить Docker Compose для сборки и запуска контейнеров: `docker compose up --build -d`
4. После успешного запуска, проверить свою консоль на наличие сообщения о том, что веб-приложение запущено и доступно по определенному URL-адресу. Открыть этот URL в своем браузере.

ПО запускается либо на компьютере оператора, либо удаленно на сервере. В последнем случае должна существовать устойчивая связь по сети между компьютером пользователя и кластером, на котором размещены эти программные модули, а так же должен быть доступен для внешних пользователей соответствующий порт веб-приложения. Веб-приложение может быть размещено в составе уже развёрнутого сайта при помощи веб-сервера, подобного Nginx.

Настройка корпусов, подлежащих агрегации, осуществляется при помощи редактирования файла [app.py](#) из репозитория [framework](#).

Алгоритм, лежащий в основе программы

1. подгрузка перечня корпусных ресурсов, работающих в системе TSAKorpus;
2. проверка доступности данных ресурсов;
3. получение уникальных токенов сессии для каждого работающего корпуса;
4. Извлечение информации о доступных поисковых модулях на ресурсах;
5. Определение общего подмножества поисковых модулей, пригодных для интегрированной работы;
6. Создание графического интерфейса для осуществления поиска;
7. Обработка пользовательского поискового запроса, сформулированного в интерфейсе, путём выполнения серии запросов к соответствующим корпусным менеджерам для получения данных, отвечающих первоначальному запросу пользователя;
8. Интеграция полученной информации из различных источников в единый интерфейс выдачи с поддержкой пагинации;
9. Формирование файлов для скачивания результатов в оригинальном формате платформы TSAKorpus;
10. Создание файлов JSON для скачивания полных результатов в машиночитаемом формате.

Логин и пароль для экспертов

Логин и пароль для экспертов отсутствует.