

## Требования к системе и предустановленное программное обеспечение

Для запуска программы требуется компьютер с поддержкой docker compose и одной из оперативных систем: Ubuntu версии 18.04 или выше, MacOS-X, Windows версии 10 и выше с поддержкой системы WSL.

- Компьютер оператора, включающий в себя:
  - Процессор x86\_64 с тактовой частотой не менее 2 ГГц;
  - Оперативную память объемом не менее 16 ГБ (для запуска ОС и одного программного модуля);
  - Мышь, клавиатура, монитор
  - Доступ к сети Интернет
- Серверный компьютер (в случае размещения для публичного использования), включающий в себя:
  - Процессор x86\_64 с тактовой частотой не менее 2 ГГц;
  - Оперативную память объемом не менее 16 ГБ (для запуска ОС и одного программного модуля);
  - Доступ к сети Интернет

Для функционирования в установленном режиме необходимо обеспечить установку следующих программных средств:

1. операционная система Ubuntu не ниже 16.04;
2. интерпретатор Python версии не ниже 3.7;

Для реализации программного модуля использовались следующие сторонние открытые библиотеки:

- Pandas (библиотека Python для обработки и анализа структурированных данных);
- Docker (инструмент контейнеризации приложений);
- Flask (фреймворк Python для создания веб-приложений);
- Re (модуль Python для поддержки регулярных выражений);
- Requests (библиотека, позволяющая отправлять запросы HTTP в Python);
- Kioskboard (библиотека JavaScript для использования виртуальных клавиатур);

- DevBridge Autocomplete (библиотека для создания полей автоматического заполнения/подсказок для полей ввода текста);
- Bootstrap Icons (библиотека с набором иконок).

## 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. Положить файлы репозитория `framework` в соответствующие пути репозитория `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`.