

ООО «Преферентум»

Поисково-аналитическая система
«АйТи-Поиск»

Операционная инструкция по установке и настройке

Листов 23

Москва

2022

Оглавление

2. Вводная информация	7
2.1. Назначение документа.....	7
2.2. Базовая информация о ПО «Поисково-аналитическая система «АйТи-Поиск»».....	7
3. Системные требования	9
3.1. Сервер АйТи-Поиск	9
3.2. Сервер БД.....	9
3.3. Требования к предустановленному ПО	9
3.4. Комплект поставки Платформы (состав дистрибутива)	10
4. Установка и настройка системы	10
4.1. Установка и настройка компонентов сервера БД	10
4.1.1. Установка Postgres	11
4.1.2. Настройка БД Postgres.....	11
4.2. Установка и настройка компонентов сервера платформы	12
4.2.1. Установка для Windows Server.....	12
4.2.1.1. Установка .Net Core	12
4.2.1.2. Установка RabbitMQ.....	12
4.2.1.3. Установка дистрибутива Платформы	12
4.2.1.4. Настройка сервисной части компонента «Документорий».....	15
4.2.1.5. Настройка клиентской части компонента «Документорий»	16
4.2.1.6. Настройка модуля машинного обучения	17
4.2.1.7. Запуск Платформы	18
4.2.2. Установка для UNIX-систем	20

4.2.2.1.	Установка .Net Core	20
4.2.2.2.	Установка RabbitMQ	20
4.2.2.3.	Установка дополнительного ПО	21
4.2.2.4.	Установка дистрибутива Платформы	21
4.2.2.5.	Запуск сервиса.....	21
5.	Контроль правильности функционирования системы.....	22

1. Термины и определения

Термин	Описание
Алгоритмическое правило	Жесткие алгоритмы, настройка выполняется разработчиком или администратором
Аннотация	<p>Информационное сообщение определенного формата и структуры, сформированное в виде примечания к тексту документа.</p> <p>Аннотация:</p> <ul style="list-style-type: none"> ● Указывает на место в документе, где сработало (должно было сработать) правило проверки текста либо правило извлечения именованной сущности (факта). ● Несет в себе информацию о том, как сработало (должно было сработать) правило или какой факт был извлечен (должен был быть извлечен). <p>В документе будет применяться терминология в рамках нижеперечисленных видов классификации аннотаций.</p> <p>По природе создания аннотации можно разделить:</p> <ul style="list-style-type: none"> ● На автоматические, созданные ПО при отправке документа на проверку. ● На созданные вручную экспертом при проверке результатов автоматической обработки документа. <p>По отношению к типу документа аннотации можно разделить:</p> <ul style="list-style-type: none"> ● На аннотации извлечения именованных сущностей (фактов). ● На аннотации, созданные по результатам срабатывания интеллектуальных правил проверки. <p>По отношению к проверяемому тексту аннотации можно разделить:</p> <ul style="list-style-type: none"> ● На информационные – аннотации, которые не производят корректировку текста проверяемого документа. ● На корректирующие – аннотации, которые могут автоматически изменить текст проверяемого документа.
Аннотирование	Подтверждение или опровержение Экспертом результатов срабатывания интеллектуальных правил
АРМ	Автоматизированное рабочее место
АС	Автоматизированная система
БД	База данных
ГОСТ	Государственный стандарт
ИС	Информационная система
Интеллектуальные правила	Жесткие алгоритмы, настройки индуцируются системой на основе примеров, задаваемых экспертом
ЛВС	Локальная вычислительная сеть
ЛНА	Локальные нормативные акты
МО, Машинное обучение, Machine Learning	Вероятностные алгоритмы, решение принимается по итогам обработки обучающей выборки
Набор правил	Несколько Правил, объединенных в процесс, исполняемый в модуле «АйТи-Поиск». Набором правил можно управлять (конфигурировать Набор). Например, включать или выключать работу отдельных Правил Набора или устанавливать значения

Термин	Описание
	параметров.
НПД	Нормативно-правовые документы
Обратная связь	Информация, внесенная экспертами по результатам проверки задания: решения по срабатыванию правил и предложения новых правил
ОЗУ	Оперативное запоминающее устройство
ООО «Преферентум»	Вендор решения «Поисково-аналитическая система «АйТи-Поиск»»
ОС	Операционная система
ПО	Программное обеспечение
ПО «АйТи-Поиск», «АйТи-Поиск»	Поисково-аналитическая система «АйТи-Поиск»
Правило	Реализованный в виде программного кода интеллектуальный алгоритм, осуществляющий проверку текста на соответствие заданным критериям. Примерами критерия могут быть: наличие или отсутствие определенной информации (Фактов) в документе, соответствие или несоответствие проверяемого документа определенным требованиям: типовой форме или структуре, корректность употребления терминов и выражений, отсутствие ссылок на утратившие силу ГОСТы. Если в результате применения правила выявлены искомые факты или обнаружены ошибки в документе, правило формирует аннотации.
Рейтингование правил	Изменение рейтинга правила для целей проверки допустимости применения правила
СУБД	Система управления базами данных
ТЗ	Документ типа «Техническое задание»
ТКП	Документ типа «Технико-коммерческое предложение»
Факт	Смысловое понятие, выделяемое в тексте документа. Примерами фактов могут быть: наименование, телефон, e-мэйл организации, ФИО руководителя и т.д. Синонимы – атрибут, реквизит, сущность.
AD	Active Directory, службы каталогов корпорации Microsoft для операционных систем семейства Windows Server
API	Программный интерфейс приложения (от англ. application programming interface)
DLL	Dynamic Link Library — динамическая библиотека, позволяющая многократное использование различными программными приложениями
GUI, UI	Графический пользовательский интерфейс (англ. graphical user interface, GUI)
IIS	Internet Information Services, проприетарный набор серверов для нескольких служб Интернета от компании Microsoft
IP-адрес	Уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP
JSON	JavaScript Object Notation — текстовый формат обмена данными, основанный на JavaScript
RabbitMQ	Платформа, реализующая систему обмена сообщениями между компонентами программной системы на основе стандарта AMQP

Термин	Описание
REST	Архитектурный стиль взаимодействия компонентов распределённого приложения в сети, сокращенно от Representational State Transfer.
SMB	Сетевой протокол прикладного уровня для удалённого доступа к файлам и другим сетевым ресурсам (от англ. Server Message Block)
SofA	Source of Analysis, источник анализа. Объект данных сложной структуры, содержащий проверяемые набором правил документы в универсальном структурном представлении и набор аннотаций и реквизитов, выделяемых в процессе обработки документов набором правил.
URL	Единообразный определитель местонахождения ресурса (от англ. Uniform Resource Locator)
Web	Программное обеспечение или оборудование, обеспечивающее взаимодействие информационных систем посредством интернет-протоколов
XML	В данном документе термин используется для обозначения формата файлов обмена данными между информационными системами.

2. Вводная информация

2.1. Назначение документа

Данный документ содержит системные требования и руководство по установке и настройке ПО «АйТи-Поиск», включающего в себя:

- Платформу интеллектуальной обработки документов;
- Автоматизированное рабочее место администратора системы («АРМ администратора»).

Документ предназначен для использования администраторами системы и применяется для ознакомления с основными принципами работы и последовательностью шагов, необходимых для корректной установки и настройки системы.

2.2. Базовая информация о ПО «Поисково-аналитическая система «АйТи-Поиск»»

Программное обеспечение «Поисково-аналитическая система АйТи-Поиск» (далее - АйТи-Поиск или ПО АйТи-Поиск) принадлежит компании ООО «Преферентум» (регистрационный №2009611815 в федеральной службе по интеллектуальной собственности (РОСПАТЕНТ), программа для ЭВМ).

ПО АйТи-Поиск реализует функции семантического анализа, извлечения сущностей из документов и проверки документов на соответствие реализованным в системе правилам.

ПО разрабатывается и поддерживается с учетом возможности работы как на платформе MS Windows, так и с использованием только импортозамещающего ПО.

Данный документ содержит описание процессов, обеспечивающих поддержание жизненного цикла программного обеспечения ПО АйТи-Поиск, в том числе устранение неисправностей, выявленных в ходе эксплуатации ПО, совершенствование ПО, а также информацию о персонале, необходимом для

обеспечения такой поддержки.

ПО разработано и развивается сотрудниками ООО «Преферентум».

Исключительные права на программное обеспечение АйТи-Поиск принадлежат ООО «Преферентум».

3. Системные требования

Для работы в высоконагруженном режиме Платформа разворачивается в нижеописанной распределенной конфигурации.

3.1. Сервер АйТи-Поиск

Сервер, на котором будет развернута платформа, с техническими характеристиками не ниже указанных:

Операционная система (OS)	Windows Server 2008, 2012, 2017 Debian 11 Ubuntu 22.04 или выше Centos 7, 8 или выше
Оперативная память (RAM)	64 Гб
Процессор (CPU)	8 ядер по 2,4 ГГц
Жесткий диск (HDD)	100 Гб

3.2. Сервер БД

Сервер СУБД, с техническими характеристиками не ниже указанных:

Операционная система (OS)	ALT Linux (либо другая ОС из списка на https://postgrespro.ru/products/postgrespro/certified)
Оперативная память (RAM)	64 Гб
Процессор (CPU)	8 ядер по 2,4 ГГц
Жесткий диск (HDD)	1000 Гб

3.3. Требования к предустановленному ПО

Для корректной работы каждого из компонентов Платформы, на сервере должно быть предустановлено следующее ПО:

- Сервер ИОИ Преферентум:
 - .NET Core 2.0.9 и выше;
 - RabbitMQ 3.7.4 и выше;
- Сервер БД:
 - Postgres Pro Standard.

3.4. Комплект поставки Платформы (состав дистрибутива)

В комплектацию решения входят:

- Архив с ядром платформы:
 - Web (Веб-сервис, который предоставляет API для взаимодействия платформой);
 - Service (Служба выполнения заданий анализа);
 - Loader (Сервис кэширования правил);
 - Manager (Сервис, распределяющий задания по потокам);
 - Дамп базы данных;
 - Documentory (Компонент, в котором содержится база правовых документов).

Для настройки приложения используются следующие конфигурационные файлы:

Конфигурационный файл компонента «Web»:

- Web\rabbitmq.Production.json

Конфигурационный файл компонента «Worker»:

- Service\appsettings.json

Конфигурационный файл компонента «Service»:

- Loader\appsettings.json

Конфигурационный файл компонента «Loader»:

- Loader\appsettings.json

Конфигурационный файл компонента «Manager»:

- Manager\appsettings.json

Конфигурационный файл компонента «Documentory»:

- Manager\appsettings.json

4. Установка и настройка системы

4.1. Установка и настройка компонентов сервера БД

В данном разделе в качестве основного используемого дистрибутива операционной системы Linux рассматривается Debian 11

Примеры выполнения команд в инструкции представляют собой работу с командной строкой (в нашем случае командной оболочкой является BASH).

4.1.1. Установка Postgres

Для того, чтобы установить Postgres 13, необходимо выполнить в командной оболочке следующие команды:

```
sudo apt update && sudo apt upgrade -y
sudo apt install postgresql postgresql-contrib -y
pg_ctlcluster 13 main start
```

После выполнения команд, в системе будет установлена и проинициализирована база данных Postgres 13

Более подробную информацию о настройке можно найти в официальной документации Postgres.

4.1.2. Настройка БД Postgres

После установки дистрибутива требуется распаковать и перенести папку с дампом базы данных на сервер с Postgres (в примере ниже выбрана директория /home/user/ , в которую был перенесен файл с дампом БД – 1.bak).

После чего требуется создать базу данных (в примере ниже используется название «Preferentum_db»), пользователя (в примере ниже используется логин 'preferentum' и пароль 'preferentum'), которому она будет доступна, загрузить в неё дампы и раздать права на изменение данных, для этого необходимо выполнить в командной оболочке следующие команды:

```
$ su postgres
$      psql
$      CREATE DATABASE preferentum_db;
$      create user preferentum with password 'preferentum';
$      grant all privileges on database preferentum_db to preferentum;
$      \q
$pg_restore -d Preferentum_db /home/user/1.bak
```

4.2. Установка и настройка компонентов сервера платформы

4.2.1. Установка для Windows Server

В данном разделе в качестве основной используемой операционной системы рассматривается Windows Server 2012.

4.2.1.1. Установка .Net Core

Для установки .NetCore необходимо:

1. Скачать установщик .NetCore Runtime & Hosting Bundle с официального сайта Microsoft (<https://www.microsoft.com/net/download>);
2. Запустить установщик, принять лицензионное соглашение и запустить процесс установки;

4.2.1.2. Установка RabbitMQ

Для установки RabbitMQ необходимо:

1. Скачать и установить Erlang OTP с официального сайта (<http://www.erlang.org/downloads>);
2. Скачать и установить RabbitMQ Server с официального сайта (<http://www.rabbitmq.com/download.html>);
3. Запустить командную строку от имени администратора и выполнить следующие команды:

```
$ cd «директория установки RabbitMQ»  
$ cd sbin  
$ rabbitmq-service install  
$ rabbitmq-service start
```

4.2.1.3. Установка дистрибутива Платформы

Для установки платформы необходимо:

1. Распаковать архив с платформой
2. Настроить конфигурационный файл appsettings.json в папке Manager (пример – Рисунок 4.1):
 - В строке "RabbitUri"указать строку подключения к RabbitMQ, если RabbitMQ установлен локально, то строка подключения

будет выглядеть так:"amqp://guest:guest@localhost";

- В строке "ConnectionString" указывается строка подключения к базе данных в формате: *"Server= IP-адрес для подключения; Port=Порт для подключения; Database=Имя базы данных; User Id=Имя пользователя; Password=Пароль"*;
- В строке "ServicePath" указывается путь до файла Service\Service.dll;
- В строке "ServiceConfigPath" указывается путь до файла Service\appsettings.json;
- В строке "ModulePath" указывается путь до папки modules "Путь до папки Web/wwwroot/Modules/";
- В строке "LoaderPath" указывается путь до динамически подключаемой библиотеки компонента «Loader» "путь до папки Loader/Loader.dll";
- В строке "LoaderConfigPath" указывается путь до до конфигурационного файла компонента «Loader» "путь до папки Loader/appsettings.json";
- В строке "UpdatesPath" указывается путь до папки с обновлениями для кэширования "путь до папки Web/wwwroot/Updates".

```
{  
  "RabbitUri": "amqp://guest:guest@localhost",  
  "ConnectionString": "Server=localhost;Port=5432;Database=postgres;User Id=postgres;Password=admin",  
  "ServicePath": "C:\\site\\Service\\Service.dll",  
  "ServiceConfigPath": "C:\\site\\Service\\appsettings.json",  
  "ModulePath": "C:/site/Web/wwwroot/Modules/",  
  "LoaderPath": "C:/site/Loader/Loader.dll",  
  "LoaderConfigPath": "C:/site/Loader/appsettings.json",  
  "UpdatesPath": "C:/site/Web/wwwroot/Updates/"  
}
```

Рисунок 4.1 Конфигурационный файл компонента «Manager»

3. Настроить конфигурационный файл appsettings.json в папке Loader (пример – Рисунок 4.2):

- В строке "RabbitUri"указать строку подключения к RabbitMQ, если RabbitMQ установлен локально, то строка подключения будет выглядеть так:"amqp://guest:guest@localhost";
- В строке "ConnectionString" указывается строка подключения к

базе данных в формате: -"Server= *IP-адрес для подключения*;
Port=*Порт для подключения*; Database=*Имя базы данных*; User
Id=*Имя пользователя*; Password=*Пароль*";

- В строке "ModulePath" указывается путь до папки modules "*Путь до папки Web/wwwroot/Modules/*";
- В строке "LoaderPath" указывается путь до динамически подключаемой библиотеки компонента «Loader» "*путь до папки Loader/Loader.dll*";
- В строке "LoaderConfigPath" указывается путь до конфигурационного файла компонента «Loader» "*путь до папки Loader/appsettings.json*";
- В строке "FilePath" указывается по путь до папки files "*Путь до папки Web/wwwroot/files/*".

```
{  
  "RabbitUri": "amqp://guest:guest@localhost",  
  "ConnectionString": "Server=172.28.0.247;Port=5432;Database=postgres;User Id=postgres;Password=admin",  
  "ModulePath": "C:/site/Web/wwwroot/Modules/",  
  "FilePath": "C:/site/Web/wwwroot/files/"  
}
```

Рисунок 4.2 Конфигурационный файл компонента «Loader»

4. Настроить конфигурационный файл appsettings.json в папке Service (пример – Рисунок 4.3):
 - В строке "RabbitUri" указать строку подключения к RabbitMQ, если RabbitMQ установлен локально, то строка подключения будет выглядеть так: "amqp://guest:guest@localhost";
 - В строке "ConnectionString" указывается строка подключения к базе данных в формате: -"Server= *IP-адрес для подключения*;
Port=*Порт для подключения*; Database=*Имя базы данных*; User
Id=*Имя пользователя*; Password=*Пароль*";
 - В строке "ModulePath" указывается путь до папки modules "*Путь до папки Web/wwwroot/Modules/*";
 - В строке "FilePath" указывается по путь до папки files "*Путь до*

nanuki Web/wwwroot/files/”;

- В строке “LicensePath” указывается по путь до папки с файлом-лицензией.

```
{  
  "RabbitUri": "amqp://guest:guest@localhost",  
  "ConnectionString": "Server=localhost;Port=5432;Database=preferentum_db;User Id=postgres;Password=postgres",  
  "ModulePath": "C:/Distr/Preferentum/Web/wwwroot/Modules/",  
  "FilePath": "C:/Distr/Preferentum/Web/wwwroot/files/",  
  "LicensePath": "C:/Distr/Preferentum/Web/wwwroot/"  
}
```

Рисунок 4.3 Конфигурационный файл компонента “Service”

5. Настроить конфигурационный файл rabbitmq.Production.json в папке Web (пример файла - Рисунок 4.4):

- В строке “Uri” указать строку подключения к RabbitMQ, если RabbitMQ установлен локально, то строка подключения будет выглядеть так: "amqp://guest:guest@localhost";
- В строке "ConnectionString" указывается строка подключения к базе данных в формате: -"Server= IP-адрес для подключения; Port=Порт для подключения; Database=Имя базы данных; User Id=Имя пользователя; Password=Пароль";
- В строке "ModulePath" указывается путь до папки modules “Путь до папки Web/wwwroot/Modules/”.

```
{  
  "Uri": "amqp://guest:guest@localhost",  
  "ConnectionString": "Server=localhost;Port=5432;Database=preferentum_db;User Id=postgres;Password=postgres",  
  "ModulePath": "C:/Distr/Preferentum/Web/wwwroot/Modules/",  
}
```

Рисунок 4.4 Конфигурационный файл компонента “Web”

4.2.1.4. Настройка сервисной части компонента «Документорий»

Для того, чтобы настроить сервисную часть компонента «Документорий», необходимо:

1. Распаковать содержимое архивов “Index1” и “Index2” в любую директорию, в распакованных папках хранится первичная база данных нормативных документов, которая позже может пополняться;

2. Перейти в папку «Documentory», которая находится в папке, куда был распакован архив с платформой;
3. Настроить конфигурационный файл «appsettings.json» (пример файла - Рисунок 4.5):
 - “PathIndex” – путь к папке с основным индексом документория;
 - “PathSource” – путь к папке, из которой будет производится загрузка новых документов;
 - “PathSuccess” – путь, где будут храниться уже загруженные документы из папки Source;
 - “PathError” – путь к папке с ошибками, возникшими при загрузке;
 - “Documentories” – путь к другим индексам, которые необходимо загружать;
 - “ConnectionString” – строка подключения к базе данных;
 - “Time” – время опроса папки Source, указанное в миллисекундах.

```
{  
  "PathIndex": "test/",  
  "PathSource": "source/",  
  "PathSuccess": "success/",  
  "PathError": "error/",  
  "Documentories": "C:/Distr/1/Index1; C:/Distr/2/Index2",  
  "ConnectionString": "Server=localhost;Port=5432;Database=preferentum_db;User Id=postgres;Password=postgres",  
  "Time": 1000  
}
```

Рисунок 4.5 Конфигурационный файл компонента “Documentory”

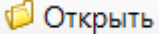
4. Перенести в папку Source все файлы, которые необходимо загрузить в документорий.

4.2.1.5. Настройка клиентской части компонента «Документорий»

Для того, что настроить компонент «Документорий», необходимо:

1. Распаковать содержимое архива «Documentory_TestDesk»;
2. Перейти в папку «Documentory_TestDesk»;
3. Запустить от имени администратора файл

«EP.Documentory.TestDesk.exe»;

4. Нажать на кнопку  в левом верхнем углу, выбрать флаг «удаленный» (см. Рисунок 4.6), после чего в поле ввода указать адрес сервера и порт, на котором работает сервис документория, нажать кнопку «ОК»;

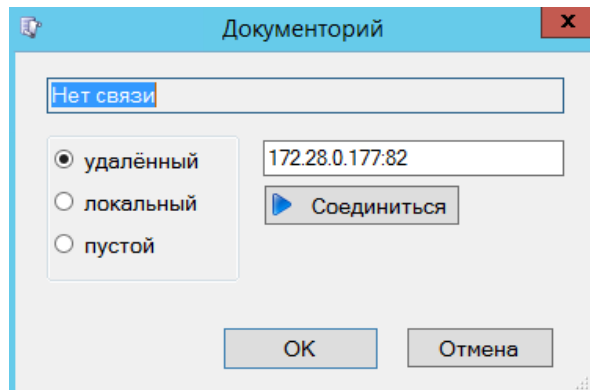


Рисунок 4.6 Подключение к сервису документория

4.2.1.6. Настройка модуля машинного обучения

Модуль машинного обучения состоит из двух веб-сервисов, написанных на языке Python 3:

1. UI_Service – реализует веб-интерфейс рабочего места специалиста по машинному обучению, с помощью данного сервиса возможно обучать систему, делить исходные размеченные файлы на выборки и измерять метрики качества работы системы;
2. REST_Service – реализует REST-API для взаимодействия с платформой и веб-интерфейсом специалиста по машинному обучению.

Для того, чтобы настроить модуль машинного обучения, необходимо:

- Распаковать архив ML.zip на диске C;
- Перейти в папку «ML\ui_service»;
- Настроить конфигурационный файл «settings.json»:
 - db\user – имя пользователя для подключения к БД;
 - db\password – пароль пользователя для подключения к БД;
 - db\host – хост для подключения к БД;
 - db\port – порт для подключения к БД;

- db\database – база данных, к которой необходимо подключиться по умолчанию;
 - paths\input_path – папка, в которой размещаются все размеченные данные;
 - paths\train_path – отдельная папка для обучающей выборки;
 - paths\test_path – отдельная папка для тестовой выборки;
 - paths\models_path – папка с моделями машинного обучения;
 - attribute_extractors – модуль извлечения атрибутов;
 - enable_auth – включение\отключение аутентификации в интерфейсе;
 - learning\folder_level – уровень вложенности папок, если исходные данные размещены по подпапкам;
 - learning\test_size – размер тестовой выборки из общего количества файлов.
- Перейти в папку «C:\ML\ui_service\cli_scripts\»;
 - Запустить следующие файлы от имени администратора:
 - instal-rest-service.bat;
 - instal-ui-service.bat.
 - По умолчанию rest-сервис будет запущен на 5000 порту для того, чтобы это изменить необходимо поменять порты в файлах:
 - C:\ML\Apache24_rest\conf\httpd.conf;
 - C:\ML\Apache24_rest\conf\extra\httpd-vhosts.conf.
 - В правиле «Правило МО» в системе необходимо изменить параметры подключения к REST-сервису, указав ip и порт, на котором этот сервис запущен.

4.2.1.7. Запуск Платформы

Для запуска Платформы в ручном режиме необходимо:

1. Запустить и настроить «Документорий», повторив шаги 3-6 из пункта 4.2.1.4 данного руководства;
2. Запустить 2 окна командной строки от имени администратора;
3. В первом окне ввести:

```
$ cd "путь до папки с компонентом Manager"  
$ dotnet manager.dll
```
4. Во втором окне необходимо ввести:

```
$ cd "путь до папки с компонентом Web"
$ dotnet Web.dll --server.urls="http://IP-адрес сервера:5000"
```

- Открыть главную страницу приложения по адресу `http://IP-адрес сервера:5000` (см. Рисунок 4.7);

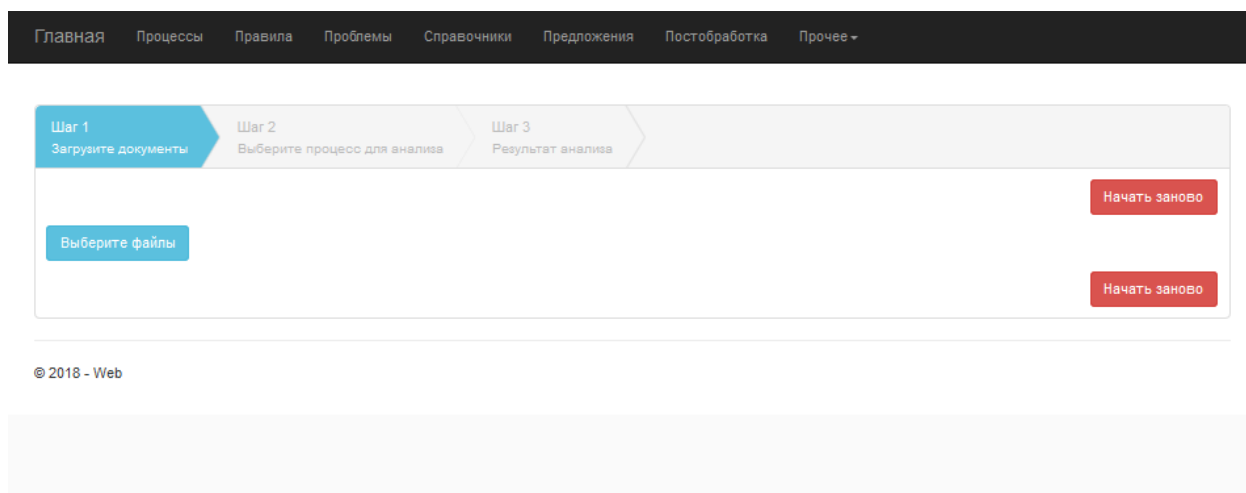


Рисунок 4.7 Главная страница «АРМ администратора»

- Перейти на вкладку «Процессы» веб-интерфейса Платформы (см. Рисунок 4.8);

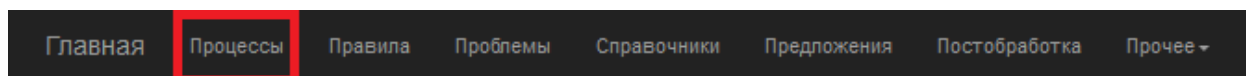


Рисунок 4.8 Меню «АРМ администратора»

- Убедиться, что в строках с процессами ТЗ и ТКП стоят ненулевые числа «Workers», и в случае необходимости добавить обработчики заданий, нажав на иконку «плюс» (+) около процессов «ТКП» и «ТЗ» (см. Рисунок 4.9);

Название	Запущенные процессы	Действия
ТКП	- 0 +	изменить архивировать
ТЗ	- 1 +	изменить архивировать

Рисунок 4.9 Вкладка «Процессы»

Для запуска Платформы в автоматическом режиме необходимо:

- Запустить и настроить «Документорий», повторив шаги 3-6 из пункта 4.2.1.4 данного руководства;
- Запустить файл `Preferentum.bat`, который находится в директории, в которую была распакована платформа.

4.2.2. Установка для UNIX-систем

В данном разделе в качестве основного используемого дистрибутива операционной системы Linux рассматривается Debian 11 В качестве пакетного менеджера используется утилита apt.

Примеры выполнения команд в инструкции представляют собой работу с командной строкой (в нашем случае командной оболочкой является BASH). Во всех примерах символы '#' и '\$' в начале строки является приглашением командной строки для пользователя 'root' и для обычного пользователя соответственно.

Установка производится из предоставляемого дистрибутива ИС.

4.2.2.1. Установка .Net Core

Для того, чтобы установить .Net Core, введем в командную оболочку следующие команды:

```
sudo apt install wget
wget https://packages.microsoft.com/config/debian/11/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
rm packages-microsoft-prod.deb
sudo apt update -y
sudo apt install -y apt-transport-https
apt install dotnet-sdk-2.1

$ dotnet --version
```

После чего в командном приложении должна отобразиться текущая версия .Net Core.

4.2.2.2. Установка RabbitMQ

Для того, чтобы установить RabbitMQ, введем в командную оболочку следующие команды:

```
apt install gnupg2 curl wget apt-transport-https software-properties-common -y
wget https://packages.erlang-solutions.com/erlang/debian/pool/esl-erlang_23.1.5-1~debian~stretch_amd64.deb
dpkg -i esl-erlang_23.1.5-1~debian~stretch_amd64.deb
```

```
apt install -f -y
apt install erlang erlang-nox
add-apt-repository 'deb http://www.rabbitmq.com/debian/ testing main'
wget -O- https://www.rabbitmq.com/rabbitmq-release-signing-key.asc | apt-key add
-
apt install rabbitmq-server
sudo systemctl start rabbitmq-server
sudo systemctl enable rabbitmq-server
sudo systemctl status rabbitmq-server
rabbitmq-plugins enable rabbitmq_management
```

Проверить <http://localhost:15672/>

логин guest

пароль guest

4.2.2.3. Установка дополнительного ПО

Для того, чтобы установить дополнительное необходимое ПО (unzip, screen), введем в командную оболочку следующие команды:

```
apt install unzip -y
```

```
apt install screen -y
```

4.2.2.4. Установка дистрибутива Платформы

Для установки дистрибутива необходимо распаковать архив, в котором он содержится, с помощью следующих команд:

```
$ mkdir /usr/share/rbt
$ chmod 777 /usr/share/rbt
$ cd /usr/share/rbt/
$ unzip /usr/share/rbt/«название архива».zip
$ mkdir /usr/share/rbt/Web/wwwroot/files
$ mkdir /usr/share/rbt/Web/wwwroot/modules
$ chmod 777 /usr/share/rbt/Web/wwwroot/files
$ chmod 777 /usr/share/rbt/Web/wwwroot/modules
```

4.2.2.5. Запуск сервиса

Для запуска Платформы в ручном режиме необходимо:

- Запустить 4 окна командной строки от имени администратора;
- В первом окне ввести:

```
$ cd "путь до папки с компонентом Manager"
$ dotnet manager.dll
```
- Во втором окне ввести:

```
$ cd "путь до папки с компонентом NR_console"
$ dotnet NR.console.dll
```
- В третьем окне необходимо ввести:

```
$ cd "путь до папки с компонентом Web"
$ dotnet Web.dll --server.urls=http://IP-адрес сервера:5000
```
- В четвертом окне ввести:

```
$ cd "путь до папки с компонентом Documentory"
$ dotnet DocumentoryService.dll --server.urls=http://IP-адрес сервера:82
```
- Запустить сервисы «TVEL_rest» и «TVEL_UI» через меню «Службы»;
- Открыть главную страницу приложения по адресу <http://IP-адрес сервера:5000> (Рисунок 4.1);

5. Контроль правильности функционирования системы

Для контроля правильности функционирования системы необходимо зайти в веб-интерфейс приложения по адресу, указанному при запуске Web.dll.

Если веб-интерфейс работает, необходимо перейти в раздел «Процессы» и включить обработчики заданий, нажав на кнопку «+» для процесса «ТКП» и «ТЗ».

Далее, необходимо нажать на кнопку «Главная» в верхней части страницы и на открывшейся странице загрузить файл любого типа для анализа.

Если в результате система отобразила дерево объектов (см. Рисунок 4.10) – система работает исправно.

Шаг 1
Загрузите документы

Шаг 2
Выберите процесс для анализа

Шаг 3
Результат анализа

Начать заново

▼ Object

- ▶ Log: Array [28]
- ▶ Documents: Array [1]
- ▶ Entities: Array [50]
- ▶ Groups: Object
- ▶ Errors: Object

Начать заново

Рисунок 4.10 Результат анализа документа

Ошибки и другая информация о работе платформы отображается в консоли.