

ООО «АйТи-Альянс»

<http://www.it-alnc.ru/>

ИНН 7718755096 / КПП 772501001

ОГРН 1097746108745

**Инструкция по установке
модуля ITA DESIGNER в составе
программного обеспечения ITA::FORMS
(«Дизайнер форм и бизнес-процессов»)**

Москва, 2025



Оглавление

1.	Аннотация	3
2.	Термины и определения.....	4
3.	Процедура установки серверной части Приложения	4
3.1	Необходимые для установки сторонние программы и оборудование:	4
3.2	Установка необходимого базового программного обеспечения.....	5
3.2.0	Containerd.....	5
3.2.1	PostgreSQL	5
3.2.2	Apache Kafka.....	5
3.3	Установка и настройка Баз Данных	5
3.3.0	Создание БД приложения.....	5
3.4	Установка приложения.....	6
3.4.1	Расположение дистрибутива ITA DESIGNER	6
3.4.2	Тестовый стенд ITA DESIGNER в сети интернет	6
3.4.3	Документация по ITA DESIGNER для скачивания	6
3.5	Компоненты.....	6
3.5.1	Компонент «keycloak»	8
3.5.2	Компонент «ita-gateway».....	9
3.5.3	Компонент «ita-policy-service».....	9
3.5.4	Компонент «ita-designer».....	9
3.5.5	Компонент «ita-designer-ui»	10
3.5.6	Компонент «ita-user-ui»	10
3.5.7	Компонент «ita-bpm»	10
3.5.8	Компонент «ita-info».....	10
3.5.9	Компонент «ita-objects».....	11
4.	Запуск Приложения на клиентском рабочем месте	11
4.1.	Требование к клиентскому рабочему месту.....	11
4.2.	Подготовка к запуску	11
4.3.	Использование приложения Приложения	12
5.	Требуемые компетенции специалистов	12
5.1.	Для администрирования и поддержки ПО	12
5.2.	Для внутренней разработки	12

1. Аннотация

Документ содержит инструкцию по установке модуля ITA DESIGNER («Дизайнер») в составе программного обеспечения ITA::FORMS.

Модуль обеспечивает:

- редактирование схем бизнес-процессов
- редактирование, хранение и предоставление метаданных пользовательских форм для интерфейса ITA::FORMS
- редактирование, хранение и предоставление метаданных настроек выполнения сервисных задач в бизнес-процессах

Данный документ содержит:

- 1) Перечень необходимого для установки стороннего ПО
- 2) Описание установки
 - а. Базового ПО
 - б. Базы данных
 - в. Приложения и его модулей
- 3) Описание процедуры запуска ПО на клиентском рабочем месте
- 4) Требования к компетенциям специалистов

2. Термины и определения

Термин	Определение
Программное обеспечение / ПО	Программа или множество программ, используемых для управления компьютером
Модуль	Составная часть программного обеспечения, обладающая самостоятельным функционалом
Приложение	Программное обеспечение «Дизайнер форм и бизнес-процессов»
web-приложение, веб-приложение	Прикладное программное обеспечение, которое работает на веб-сервере, в отличие от компьютерных программ, которые запускаются локально в операционной системе (ОС) устройства
БД, база данных	Совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных

3. Процедура установки серверной части Приложения

3.1 Необходимые для установки сторонние программы и оборудование:

Для установки приложения необходимо следующее сторонне ПО, установленное на сервере приложений:

- Containerd – совместимая версия 2.0.0
- Apache Kafka – совместимая версия: 3.1.0
- PostgreSQL – совместимая версия: 14.2

При установке Приложения, сервер БД, сервер очередей сообщений и сервер приложений могут быть размещены на одном физическом сервере (либо виртуальной машине).

Для установки Приложения требуется следующее оборудование для серверной части:

- ЦПУ: не менее 4 ядер ЦПУ
- Память ОЗУ: не менее 16 Гб ОЗУ
- Дисковая память: не менее 50 Гб дискового пространства

3.2 Установка необходимого базового программного обеспечения

3.2.0 Containerd

Компоненты модуля ITA DESIGNER собраны в виде docker образов и разворачиваются в контейнерах Containerd или Docker. Поэтому перед началом работ по установке Приложения на сервере должно быть установлено и доступно для использования базовое ПО – Containerd версии не ниже 2.0.0, распространяемое в формате open-source.

Проверить корректность установки можно следующим образом:

```
$ docker --version  
Docker version 27.5.0, build a187fa5
```

3.2.1 PostgreSQL

Перед началом работ по установке Приложения на сервере должно быть установлено и запущено базовое ПО – PostgreSQL версии 14.2, распространяемое в формате open-source, источник <https://www.postgresql.org/ftp/source/v14.2/>.

При установке данного ПО необходимо обеспечить доступность сетевого порта, по которому доступно данное ПО, для остальных модулей Приложения. Так же необходимо зафиксировать логин и пароль для доступа к СУБД. Логин и пароль понадобятся при настройке остальных модулей Приложения.

3.2.2 Apache Kafka

Перед началом работ по установке Приложения на сервер должно быть установлено и запущено базовое ПО – Apache Kafka версии 3.1.0, распространяемое в формате open-source, источник <https://archive.apache.org/dist/kafka/3.1.0/kafka-3.1.0-src.tgz>

При установке данного ПО необходимо обеспечить доступность сетевых портов, по которым доступно данное ПО, для остальных модулей Приложения.

3.3 Установка и настройка Баз Данных

3.3.0 Создание БД приложения

В рамках поставки дистрибутива приложения предоставляются предзаполненные дампы БД, из которых необходимо создать БД, с которыми будет взаимодействовать Приложение.

Список БД, которые необходимо создать для работы Приложения (из предоставленных дампов):

- БД «designer»
- БД «keycloak»
- БД «bpm»

- БД «policy»
- БД «objects_f»
- БД «objects_u»

3.4 Установка приложения

Для установки Приложения необходимо установить следующие компоненты ITA::FORMS:

- keycloak
- ita-gateway
- ita-policy-service
- ita-bpm
- ita-userui
- ita-info
- ita-objects

И компоненты модуля ITA DESIGNER:

- ita-designer
- ita-designer-ui

3.4.1 Расположение дистрибутива ITA DESIGNER

<https://updates.it-alnc.ru>

Логин и пароль предоставляются по запросу Правообладателем

3.4.2 Тестовый стенд ITA DESIGNER в сети интернет

<http://ros.it-alnc.ru:8082/main/>

Логин и пароль предоставляются по запросу Правообладателем

3.4.3 Документация по ITA DESIGNER для скачивания

<https://www.it-alnc.ru/support/docs.php>

3.5 Компоненты

Перед разворачиванием необходимо

1. Скачать архивы образов на сервер
2. Создать (например в /opt/) и заполнить файл с переменными (env.list):

```
AUTH_PUBLIC_KEY=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAAnB7olpa7iukLHJ  
EN8Rx9Tn6829hjmVhr9Zk+2NJKz7YTKGMUSsZz+Ec74gHWL5hRHlt7AVucpsA/5HLnDTrbidDm  
4ixjSrQJcBXCIZTRGfrHQ6Z8RBSQ8P0dXT036hLJGnGCV0nnydS3sTyCN9NSRex3ZuijWkzjsynx1//  
qGf7ZTYVFXLnRdvPFxef4K2LRD0GNFWpqmMdhGPfvC+6BUSUaFI7wSgbUKlNPm1o+awz5tLKd  
BSTctd5IEft3qz1vVx1lrSCFd3SLJ5V8AddVb6UumRgT7eIBDLujMS3YierBitfKyRs1DQi95hgJ9Hqy  
mRKkxzyqQ5k/u+02GG26pwIDAQAB (Публичный ключ keycloak)
```

AUTH_URI_EXT=*http://ros.it-alc.ru:8080/auth/realms/master/protocol/openid-connect/auth* (внешняя ссылка для авторизации в keycloak)
CAMUNDA_DB_PASSWORD=*postgres* (пароль от БД)
CAMUNDA_DB_URL=*jdbc:postgresql://localhost:5432/bpm* (адрес БД bpm)
CAMUNDA_DB_USER=*postgres* (пользователь БД)
CAMUNDA=*http://ros.it-alc.ru:8089/camunda/app* (адрес сервиса camunda)
CONTEXT_DB_PASSWORD=*postgres* (пароль от БД)
CONTEXT_DB_URL=*jdbc:postgresql://localhost:5432/designer* (адрес сервиса designer)
CONTEXT_DB_USER=*postgres* (пользователь БД)
DESIGNER_UI=*http://ros.it-alc.ru:4201/ita-designer-ui* (адрес фронтенда designer-ui)
DB_ADDR=*ros.it-alc.ru* (адрес БД для keycloak)
DB_DATABASE=*keycloak_docker* (имя БД keycloak)
DB_PASSWORD=*postgres* (пароль от БД)
DB_USER=*postgres* (пользователь БД)
DB_VENDOR=*postgres* (тип БД)
EDITOR=*http://ros.it-alc.ru:8085/editor* (адрес сервиса editor)
EDITOR_URL=*http://ros.it-alc.ru:8085/editor* (адрес сервиса editor)
ELASTIC_URL=*http://elastic:9200* (оставить как есть, если не используется Elastic)
ELASTIC_AUDIT_INDEX=*audit* (оставить так, если не используется Elastic)
GW_CLIENT_EXT=*gateway* (Client в keycloak)
GW_SECRET_EXT=*cf6uLNPHpWSRAr5na22UBbBHUfaKYofx* (secret для client в keycloak)
ITA_BPM_CAMUNDA_URL=*http://ros.it-alc.ru:8089/engine-rest* (адрес сервиса camunda)
ITA_FRONT=*http://ros.it-alc.ru:8082* (URL приложения)
ITA_POLICY=*http://ros.it-alc.ru:8120* (адрес сервиса ita-policy-service)
POLICY=*http://ros.it-alc.ru:8120/aps* (адрес сервиса ita-policy-service)
MAIN_UI=*http://ros.it-alc.ru:4200/main* (адрес фронтенда user-ui (main страницы))
JAVA_OPTS=*-XX:+UseContainerSupport -XX:MaxRAMPercentage=75.0*
XX:+CrashOnOutOfMemoryError -Dkeycloak.profile.feature.upload_scripts=enabled
KAFKA_REPLICATION_FACTOR=*1* (параметры репликации kafka)
KAFKA_SERVERS=*ros.it-alc.ru:9092* (адрес kafka)
KC_DB_DATABASE=*keycloak_docker* (имя БД keycloak)
KC_DB_DRIVER=*org.postgresql.xa.PGXADatasource*
KC_DB_PASSWORD=*postgres* (пароль от БД)
KC_DB_URL=*jdbc:postgresql://localhost:5432/keycloak_docker* (адрес БД keycloak)
KC_DB_USERNAME=*postgres* (пользователь БД)
KC_HEALTH_ENABLED=*true* (включение healthcheck keycloak)
KC_METRICS_ENABLED=*true* (включение метрик keycloak)
KC_HTTP_ENABLED=*true* (включение режима работы http в keycloak)
KEYCLOAK=*http://ros.it-alc.ru:8080/auth/realms/master* (адрес keycloak)
KEYCLOAK_HOSTNAME=*ros.it-alc.ru* (хост keycloak)
KEYCLOAK_PASSWORD=*admin* (пароль от БД)
KEYCLOAK_URL=*http://ros.it-alc.ru:8080/* (url keycloak)
KEYCLOAK_USER=*admin* (пользователь keycloak по умолчанию)
POLICY_DB_PASSWORD=*postgres* (пароль от БД)
POLICY_DB_URL=*jdbc:postgresql://localhost:5432/policy* (адрес бд policy)
POLICY_DB_USER=*postgres* (пользователь БД)
PRODCAT=*http://ros.it-alc.ru:9091/prc* (адрес сервиса prodcats)
PROXY_ADDRESS_FORWARDING=*true*
REALM=*master* (realm в keycloak)
TOKEN_URI_EXT=*http://ros.it-alc.ru:8080/auth/realms/master/protocol/openid-connect/token* (ссылка получения токена в keycloak)
TZ=*Europe/Moscow*

```

URL_GATEWAY=http://ros.it-alc.ru:8082 (адрес сервиса gateway)
USE_EXTERNAL_KEYCLOAK=true (использование внешнего keycloak, оставить true)
USER_UI=http://ros.it-alc.ru:4200/ita-user-ui (адрес сервиса user-ui)
USRF_URI_EXT=http://ros.it-alc.ru:8080/auth/realms/master/protocol/openid-
connect/userinfo (ссылка на получение данных о пользователе из keycloak)
OBJECTS_BO_DB_F_PASSWORD=postgres (пароль от БД)
OBJECTS_BO_DB_F_URL=jdbc:postgresql://ros.it-alc.ru:5432/objects_f (адрес бд objects_f)
OBJECTS_BO_DB_F_USER=postgres (пользователь БД)
OBJECTS_BO_DB_PASSWORD=postgres (пароль от БД)
OBJECTS_BO_DB_U_PASSWORD=postgres (пароль от БД)
OBJECTS_BO_DB_U_URL=jdbc:postgresql://ros.it-alc.ru:5432/objects_u (адрес бд objects_u)
OBJECTS_BO_DB_U_USER=postgres (пользователь БД)
OBJECTS_BO_DB_URL=jdbc:postgresql://ros.it-alc.ru:5432/objects_f (адрес бд objects_f)
OBJECTS_BO_DB_USER=postgres (пользователь БД)
OBJECTS_CACHE_DB_PASSWORD=postgres (пароль от БД)
OBJECTS_CACHE_DB_URL=jdbc:postgresql://ros.it-alc.ru:5432/objects_f (адрес бд objects_f)
OBJECTS_CACHE_DB_USER=postgres (пользователь БД)
OBJECTS_DB_F_PASSWORD=postgres (пароль от БД)
OBJECTS_DB_F_URL=jdbc:postgresql://ros.it-alc.ru:5432/objects_f (адрес бд objects_f)
OBJECTS_DB_F_USER=postgres (пользователь БД)
OBJECTS_DB_PASSWORD=postgres (пароль от БД)
OBJECTS_DB_U_PASSWORD=postgres (пароль от БД)
OBJECTS_DB_U_URL=jdbc:postgresql://ros.it-alc.ru:5432/objects_u (адрес бд objects_u)
OBJECTS_DB_U_USER=postgres (пользователь БД)
OBJECTS_DB_URL=jdbc:postgresql://ros.it-alc.ru:5432/objects_f (адрес бд objects_f)
OBJECTS_DB_USER=postgres (пользователь БД)
OBJECTS=http://ros.it-alc.ru:9191/objects/ (адрес сервиса ita-objects)
INFO_DB_URL=jdbc:postgresql://ros.it-alc.ru:5432/designer (адрес бд designer)
INFO_DB_USER=postgres (пользователь БД)
INFO_DB_PASSWORD=postgres (пароль от БД)
INFO=http://ros.it-alc.ru:8071/info (адрес сервиса ita-info)
ITA_INFO=http://ros.it-alc.ru:8071/info (адрес сервиса ita-info)
INFO_DB_DIALECT=org.hibernate.dialect.PostgreSQLDialect
INFO_DB_DRIVER=org.postgresql.Driver

```

3.5.1 Компонент «keycloak»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)
2. Выполнить загрузку образа командой:
`docker image load --input keycloak.tar`, где keycloak.tar – архив или путь к архиву на сервере.
3. Выполнить запуск контейнера командой:
`docker run -d -p 8080:8080 -p 8443:8443 --env-file env.list --name keycloak keycloak:latest`
4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

3.5.2 Компонент «ita-gateway»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)
2. Выполнить загрузку образа командой:
`docker image load --input ita-gateway.tar`, где ita-gateway.tar– архив или путь к архиву на сервере.
3. Выполнить запуск контейнера командой:
`docker run -d -p 8082:8443 --env-file env.list --name gateway ita-gateway:latest`
4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

3.5.3 Компонент «ita-policy-service»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)
2. Выполнить загрузку образа командой:
`docker image load --input ita-policy-service.tar`, где ita-policy-service.tar– архив или путь к архиву на сервере.
3. Выполнить запуск контейнера командой:
`docker run -d -p 8120:8120 --env-file env.list --name ita-policy-service ita-policy-service:latest`
4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

3.5.4 Компонент «ita-designer»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)
2. Выполнить загрузку образа командой:
`docker image load --input ita-designer.tar`, где ita-designer.tar– архив или путь к архиву на сервере.
3. Выполнить запуск контейнера командой:
`docker run -d -p 8085:8085 --env-file env.list --name ita-designer ita-designer:latest`
4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

3.5.5 Компонент «ita-designer-ui»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)
2. Выполнить загрузку образа командой:
`docker image load --input ita-designer-ui.tar`, где ita-designer-ui.tar – архив или путь к архиву на сервере.
3. Выполнить запуск контейнера командой:
`docker run -d -p 4201:4200 --env-file env.list --name designer-ui ita-designer-ui:latest`
4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

3.5.6 Компонент «ita-user-ui»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)
2. Выполнить загрузку образа командой:
`docker image load --input ita-user-ui.tar`, где ita-user-ui.tar – архив или путь к архиву на сервере.
3. Выполнить запуск контейнера командой:
`docker run -d -p 4200:4200 --env-file env.list --name user-ui ita-user-ui:latest`
4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

3.5.7 Компонент «ita-bpm»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)
2. Выполнить загрузку образа командой:
`docker image load --input camunda.tar`, где camunda.tar – архив или путь к архиву на сервере.
3. Выполнить запуск контейнера командой:
`docker run -d -p 8089:8089 --env-file env.list --name camunda camunda:latest`
4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

3.5.8 Компонент «ita-info»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)
2. Выполнить загрузку образа командой:

`docker image load --input ita-info.tar`, где `ita-info.tar`– архив или путь к архиву на сервере.

3. Выполнить запуск контейнера командой:

```
docker run -d -p 8071:8071 --env-file env.list --name ita-info  
ita-info:latest
```

4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

3.5.9 Компонент «ita-objects»

Порядок действий для установки компонента

1. Перейти в директорию, где находится заполненный файл с переменными (например в /opt/)

2. Выполнить загрузку образа командой:

`docker image load --input ita-objects.tar`, где `ita-objects.tar`– архив или путь к архиву на сервере.

3. Выполнить запуск контейнера командой:

```
docker run -d -p 9091:9091 --env-file env.list --name ita-objects  
ita-objects:latest
```

4. Проверить, что контейнер успешно запущен (status Running) командой
`docker ps -a`

4. Запуск Приложения на клиентском рабочем месте

4.1. Требование к клиентскому рабочему месту

Для использования приложения требуется следующее программное обеспечение на клиентском рабочем месте:

- Операционная система:
 - Windows
 - Linux
 - Mac OS X
- Интернет-браузер:
 - Google Chrome 23 и выше (рекомендуется)
 - Mozilla Firefox 18 и выше
 - Safari 6 и выше
 - Mozilla Firefox 17 ESR, 24 ESR, 31 ESR, 38 ESR, 45 ESR
 - Internet Explorer 10 и выше, Edge
 - Opera 16 и выше

4.2. Подготовка к запуску

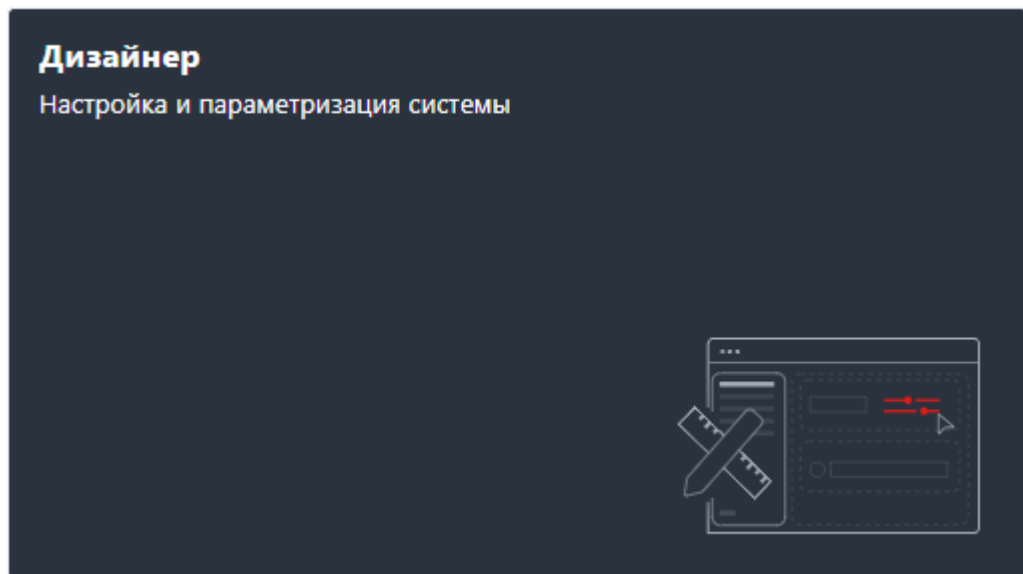
Приложение преднастроено таким образом, что оно должно быть доступно для пользователя по адресу <http://ros.it-alnc.ru:8082/main/>

Поэтому необходимо на клиентском рабочем месте установить соответствие имени **ros.it-alnc.ru** и ip-адреса сервера.

- в ОС Linux в файле `/etc/hosts`
- в ОС Windows в файле `c:\windows\system32\drivers\etc\hosts`

4.3. Использование приложения Приложения

1. Открыть в браузере адрес <http://ros.it-alnc.ru:8082/main/>
2. Ввести логин и пароль:
логин – admin
пароль – admin
3. Открыть модуль «Дизайнер»



5. Требуемые компетенции специалистов

5.1. Для администрирования и поддержки ПО

- Знания по администрированию БД PostgreSQL
- Знания по администрированию Containerd

5.2. Для внутренней разработки

- Знание Java Spring - опыт разработки не менее 1 года
- Знание SQL - опыт работы с SQL не менее 1 года
- Знание Angular - опыт разработки не менее 1 года
- Базовые знания ООП
- Знание базовых классов и принципов разработки в ITA DESIGNER