



ООО «Легалтэк»
applegalhelp.ru/help



Модуль цифровых юридических продуктов

Платформа “LEGALHELP” с использованием
базы знаний и интеллектуального чат-бота

Документация, содержащая информацию, необходимую для установки и
эксплуатации экземпляра программного обеспечения,
предоставленного для проведения экспертной проверки

Введение	4
Перечень обозначений и сокращений	6
1. Условия, при соблюдении которых обеспечивается применение экземпляра ПО с установленным Модулем ЦП, предоставленных для проведения экспертной проверки	8
1.1 Требования к технологическому стеку для разработки и поддержки ПО с установленным Модулем	8
1.2 Требования к программному обеспечению Платформы для промышленной эксплуатации	8
1.3 Требования оборудованию и платформе мобильного телефона	9
1.4 Требования к оборудованию для запуска Модуля ЦП, серверной части и интерфейса администрирования	9
1.5 Требования к оборудованию компьютера для запуска стенда и экспертной проверки	10
1.6 Требования к окружению для запуска стенда и экспертной проверки	10
2. Архитектура ПО с установленным Модулем ЦП	16
3. Файловый состав поставки ПО с Модулем ЦП для проведения экспертной проверки	18
4. Порядок установки ПО LEGALHELP с Модулем ЦП	21
4.1. Установка модуля ЦП	21
4.2 Установка Платформы на мобильном телефоне	22
4.3. Установка Платформы в эмуляторе Android Studio	22
4.4. Установка русскоязычной клавиатуры в эмулятор	23
5. Методика комплексного тестирования Модуля ЦП в Платформе	25
5.1. Методика комплексного тестирования Модуля ЦП, серверной части и базы данных	25
5.1.2. Описание API бэкенда сервиса Редактора ЦП, серверной части	28
5.3. Описание фронтенда Редактора	43
5.3.1. Руководство по использованию фронтенда Редактора	43
5.4. Принцип программного подключения к чату со стороны Право-бота	49
5.5. Переписка с ботом	51
5.6. Подключение от имени бота	51
6. Эксплуатация экземпляра ПО, предоставленного для проведения экспертной проверки	53
6.1 Запуск платформы в эмуляторе Android Studio на Ubuntu для запуска стенда и экспертной проверки	53
6.2 Сборка новых версий ПО Платформы	54
6.3 Запуск стенда для экспертной проверки	54

6.4 Инструменты, моделирующие поведение внешних систем	55
6.5 Инструменты, необходимые для администрирование подсистем	56
6.5.1. Администрирование серверной части Модуля ЦП	56
6.5.2. Администрирование состава цифровых юридических продуктов и их свойств	57
6.5.3. Администрирование Мобильного бекенда	59
6.5.2 Администрирование подсистемы авторизации	59
6.5.3 Администрирование базы знаний	61
6.5.4 Администрирование брокера очередей RabbitMQ подсистемы чатов	62
6.5.5 Администрирование с помощью средств мониторинга нагрузки	63
7. Математическое обеспечение Модуля ЦП в рамках платформы	66
7.1. Тренировка нейронных сетей интеллектуального чат-бота для разделов Модуля ЦП	67
8. Установка Ubuntu 68	
8.1. Подготовка к установке	68
8.2. Установка Ubuntu 20.04	71
9. Установка Android Studio на Ubuntu 80	
9.1. Завершение установки Android studio и установка дополнительных пакетов	88
10. Установка Docker и Docker compose на Ubuntu 89	
10.1. Подключение репозитория и обновление списка пакетов в системе	90
10.2. Установка Docker и Docker-compose	90
10.3. Настройка Docker	91
10.4. Проверка установки	91

Введение

Документ содержит описание порядка установки экземпляра программного обеспечения Платформы “LEGALHELP” (далее – Платформа) с использованием базы знаний и интеллектуального чат бота, предоставленного для проведения экспертной проверки, и включает информацию по установке и настройке общего программного обеспечения Платформы.

Также, настоящий документ содержит информацию, необходимую для эксплуатации экземпляра программного обеспечения, предоставленных для проведения экспертной проверки.

Служба технической поддержки

Связаться со службой технической поддержки можно по электронной почте support@applegalhelp.ru.

Страница службы технической поддержки на сайте компании «Легалтэк» <https://applegalhelp.ru/support>.

Сайт в интернете. Информацию о Платформе “LEGALHELP”, продукте компании «Легалтэк» представлена на сайте <https://applegalhelp.ru/help>.

Область применения

Платформа предназначена для оказания юридических консультаций Пользователей (клиентов) с использованием базы знаний и интеллектуального чат-бота для упрощения и повышения оперативности обслуживания в режиме 24/7.

Цель – минимизация участия оператора в оказании юридических консультаций за счет автоматизации процессов обслуживания.

Краткое описание возможностей

Пользователь может использовать Платформу для общения с чат-ботом в качестве цифрового помощника для получения первичной юридической помощи в режиме ведения диалога в чате по широкому кругу вопросов.

Пользователю предоставляется возможность релевантного поиска ответов на его вопросы в актуальной текстовой Базе знаний в соответствующей области юридической экспертизы (юридических кейсов и проблем).

Пользователь для получения юридической помощи может использовать Платформу для общения с юристом в режиме ведения диалога.

Работа пользователей с Платформой осуществляется через интерфейс.

Предоставление доступа осуществляется в офисе компании “Легалтэк”, по электронной почте или на любом информационном носителе: предоставляются авторизационные данные (логин, соответствующий номеру телефона, и пароль).

Важно! В случае экспертной проверки следует ввести в эмуляторе следующие данные для авторизации:

Таблица 1 – Данные для авторизации при экспертной проверке

логин	79999999999
пароль	1234

А также, реализовано разграничение прав пользователей путем назначения флага для ролей.

Доступны следующие роли:

- неавторизованный пользователь – лицо, которое не получило доступ (логин-пароль) к Платформе и не в состоянии использовать его функционал, кроме обращения в службу технической поддержки;

- стандартный пользователь (далее – Пользователь) – лицо, со стандартным набором прав и функций;

Реализованы роли обслуживающего персонала, не взаимодействующие через Платформу, а использующие для этого внешние инструменты:

- администратор – пользователь с расширенным набором прав и привилегий, имеющий возможность управлять доступом Пользователей, конфигурировать подсистемы, проводить рассылки сообщений;

- юрист-консультант – пользователь, взаимодействующий через внешнюю информационную систему “Автоматизированное рабочее место консультанта” для оказания юридических консультаций в чате;

- специалист технической поддержки.

Перечень обозначений и сокращений

В настоящем документе используются следующие сокращения, термины с соответствующими определениями (Таблица 2).

Таблица 2 – Сокращения, термины с соответствующими определениями

Обозначение / Сокращение	
ПО	Программное обеспечение
Клиентская часть	Мобильный клиент, здесь и далее – синоним клиентского приложения LEGALHELP. Клиент в терминах клиент-серверной архитектуры: программное обеспечение Платформы.
Платформа	Автоматизированная платформа оказания юридических консультаций с использованием базы знаний и интеллектуального чат-бота: программное обеспечение Платформы
Модуль ЦП	Модуль цифровых юридических продуктов для Платформы
Подсистема	Множество программных сервисов, модулей, компонент, совокупно выполняющих в результате своего взаимодействия одну или несколько функций одного назначения (достигающих единую цель)
ВИС	Внешняя информационная система
API	Application Programming Interface – программный интерфейс платформы, интерфейс прикладного программирования
GUI, UI	Graphical User Interface – графический интерфейс (платформы)
NLU	Natural Language Understanding – модель машинного обучения, натренированная на понимание естественного языка
Клиент	Заказчик услуг (получатель данных) в терминах клиент-серверной (вычислительной или сетевой) архитектуры, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами
Мобильный бекенд	Подсистема обмена данными с мобильным клиентом (Платформы)

Право-бот	Интеллектуальный чат-бот, оказывающий консультационные услуги юридического характера. Входит в платформенное решение, и предназначен для общения в формате диалоговых чатов с целью значительного упрощения получения оперативной юридической помощи. В Платформе имеет интерфейс общения в виде чата
CRUD	Акроним, обозначающий четыре базовые функции, используемые при работе с данными: создание (англ. create), чтение (read), модификация (update), удаление (delete)
Пользователь	Пользователь, который может использовать стандартный функционал Платформы.
БЗ	База знаний – хранилище структурированной и неструктурированной информации, обеспечивающее возможности релевантного поиска с учетом морфологии русского языка
Юрист	Специалист для оказания юридических консультаций в чате
Интент	Потребность пользователя (намерение, цель), которую он хочет решить с помощью текстового обращения в чате
\$USER	Имя пользователя (и имя домашней папки пользователя) Ubuntu, в которую происходит установка ПО.

1. Условия, при соблюдении которых обеспечивается применение экземпляра ПО с установленным Модулем ЦП, предоставленных для проведения экспертной проверки

1.1 Требования к технологическому стеку для разработки и поддержки ПО с установленным Модулем

- Контейнеризация: Docker;
- Бекенд сервисов: Nodejs + Typescript, Python 3.8+, Go, PHP 7+;
- Фронтенд: Vue.js / ReactJS + mobx;
- Фронтенд платформы: React Native;
- Бекенд модуля цифровых юридических продуктов: Python 3.8+;
- Фронтенд модуля цифровых юридических продуктов: nginx + Vue.js 3.

1.2 Требования к программному обеспечению Платформы для промышленной эксплуатации

Платформа в промышленном режиме функционирует с использованием следующего программного обеспечения серверов:

- Ноды вычислительного кластера: Ubuntu (Linux);
- Оркестрация контейнеров (для промышленной эксплуатации): Kubernetes;
- Описание инфраструктуры: Terraform, Helm-3;
- Сбор метрик: Grafana + Prometheus;
- Бастион: Ingress-Nginx;
- Веб-серверы: Nginx;
- Хранилище пакетов / docker-образов: Nexus;
- Реляционные базы данных: кластер MySQL CE, кластер PostgreSQL CE;
- Кеш: Redis;
- База знаний и релевантный поиск: Elasticsearch CE;
- Брокер очередей сообщений: кластер RabbitMQ + плагины для MQTT для MQTT поверх веб-сокетов;
- Машинное обучение: Tensorflow / Torch - с поддержкой CUDA;
- Машинное обучение задач NLP: word2vec / doc2vec, gensim, TextBlob, OpenAI GTP2, Transformer, DeepPavlov, BERT, ELMO.

Базой знаний для юридических разделов Модуля ЦП является ElasticSearch с REST-API адаптером, задействованным в подсистеме управления базой знаний.

Модели машинного обучения для автоматизированного чат-бота в рамках Модуля ЦП загружаются в виде бинарных файлов с коэффициентами нейронных сетей и обслуживаются через REST-API сервис, обменивающийся данными через AMQP-адаптер с шиной сообщений.

Обмен информацией между программно-техническими средствами обеспечивается средствами ЛВС, а с Платформой и с пользователями, соответственно, – по каналам сети мобильной связи 4G с использованием унифицированных транспортных протоколов (HTTP / HTTPS), внутри кластера межсервисное взаимодействие с шиной сообщений осуществляется через протокол AMQP, снаружи очередь сообщений обменивается с мобильным клиентом по протоколам WS/WSS и MQTT.

1.3 Требования оборудованию и платформе мобильного телефона

Данный раздел носит исключительно справочный характер для полноты предоставляемых сведений. Экспертиза Платформы производится не на реальном устройстве (телефоне), а в эмуляторе Android Studio на компьютере, являющимся стендом для проведения экспертизы (см. 1.4 Требования к оборудованию компьютера для запуска стенда и экспертной проверки).

Поскольку целевой архитектурой для запуска Платформы является программно-аппаратное обеспечение мобильного телефона и платформа Android, в этом разделе перечислены все необходимые требования к оборудованию и платформе мобильного телефона для запуска Платформы:

- вид и версия операционной системы: Android 6-11 (должна быть ниже 12)
- потребление памяти: 1 Гб;
- объем, Мб (бандл /на диске/ с кэшем): 21 / 50 / 80 Мб;
- обязательно наличие сим-карты в телефоне, выход в интернет по каналу 4G;
- минимальная доступная оперативная память: 1 Гб;
- рекомендуемая доступная оперативная память: 2 Гб;
- рекомендуемая доступная дисковая память (на карте памяти): 10 Гб.

1.4. Требования к оборудованию для запуска Модуля ЦП, серверной части и интерфейса администрирования

Для запуска редактора цифровых продуктов требуется docker версии ≥ 20.10 и docker-compose версии ≥ 1.29

На компьютере должен быть установлен двухъядерный процессор с частотой ≥ 2.0 ГГц, 8 Гб оперативной памяти и жесткий диск объемом ≥ 20 Гб

Должны быть открыты порты: 8070, 80

1.5 Требования к оборудованию компьютера для запуска стенда и экспертной проверки

Системные требования к аппаратной части ПК для проведения экспертизы

- наличие выхода в сеть Интернет, ширина канала не менее 2 Мбит.
- поддержка аппаратной виртуализации для центрального процессора рабочей машины (VT-x или SVM).

Таблица 3 – Системные требования для запуска стенда

Вариант требований	vCPU	RAM, Гб	SSD, Гб
Минимальные требования	4	6	30
Рекомендуемые требования	8	8	30+

1.6 Требования к окружению для запуска стенда и экспертной проверки

Стенд для экспертизы работы Платформы работает на ОС Linux Ubuntu с использованием контейнеров docker. Все необходимые контейнеры оркестрируются через docker-compose и не нуждаются в отдельной установке.

ВАЖНО! На рабочей машине для установки зависимостей у пользователя Ubuntu должен быть sudo-доступ администратора системы.

Обмен информацией между программно-техническими средствами обеспечивается средствами ЛВС внутри локальной машины стенда для проведения экспертизы.

Необходимые пакеты для ОС Linux Ubuntu устанавливаются через инсталлятор, прилагаемый в комплекте к ПО.

Платформа функционирует с использованием следующего программного обеспечения:

- 1) Рабочая машина: Операционная система Ubuntu¹ (Linux) 20.04+ (см. 3.1. Установка Ubuntu).

¹ Лицензия GPL: <https://www.gnu.org/licenses/gpl-3.0.en.html>

- 2) `docker`² и `docker-compose`³ (см. 3.3. Установка Docker и Docker-compose).
- 3) Утилита `cURL`⁴
- 4) Файловая система `ext4` (не `ZFS`).
- 5) Программный эмулятор платформы для сборки и запуска: `Android Studio`⁵ (см. 3.2. Установка Docker и Docker-compose).

ВАЖНО! Поставка предназначена для запуска на локальной машине. Все ссылки в документе ведут на `http://localhost`, мобильный клиент также ожидает наличие API на локальной машине.

ВАЖНО! Запуск ПО на виртуальной машине строго не рекомендуется. Многие виртуальные машины не поддерживают виртуализацию и `docker` не будет на них корректно работать.

ВАЖНО! Для доступа ко всем интерфейсам следующие порты в системе должны быть свободны и открыты файрволом:

² Лицензия: <https://raw.githubusercontent.com/docker/docker-ce/master/components/cli/LICENSE>

³ Лицензия: <https://raw.githubusercontent.com/docker/compose/master/LICENSE>

⁴ Лицензия: <https://raw.githubusercontent.com/cujojs/curl/master/LICENSE.txt>

⁵ Лицензия Apache 2.0: <http://www.apache.org/licenses/LICENSE-2.0>

Таблица 4 – Требования к свободным и открытым портам в системе

Порт	Назначение
8070	Модуль ЦП. Серверная часть
80	Веб-сервер Nginx
3050	Мобильный бекенд (NodeJS)
9230	Порт отладки мобильного бекенда (NodeJS)
8200	Хранилище секретов Vault
6379	Кеш серверных приложений (База данных Redis CE)
3306	База данных MySQL CE
8080	Служба чатов со стороны Пользователя (Golang + JavaScript)
8888	Служба чатов со стороны Специалиста (Golang + JavaScript)
7770	Диалоговый менеджер интеллектуального чат-бота (Python 3) с возможностью голосования моделей
7771	NLU-классификатор намерений пользователя на базе модели ELMO (Python 3)
7772	NLU-классификатор намерений пользователя на базе модели BERT(Python 3)
7773	NLU-классификатор намерений пользователя на базе ядра чат-бота (Python 3, нейронная сеть прямого распространения)
8002	Адаптер предоставления данных Базы знаний (Python 3)
9200	База знаний (База данных Elasticsearch CE) - первая нода кластера
9300	База знаний (База данных Elasticsearch CE) - вторая нода кластера
1883	Брокер очередей RabbitMQ, обслуживающий очереди по протоколу MQTT
5672	Брокер очередей RabbitMQ, обслуживающий очереди по протоколу AMQP
15675	Брокер очередей RabbitMQ, обслуживающий очереди по протоколу MQTT через транспортный уровень Web-Socket
15672	Интерфейс UI админ-панели брокера очередей RabbitMQ
9090	Служба сбора данных о состоянии сервисов платформы (Prometheus)
3000	Админ-панель мониторинга состояния сервисов платформы (Grafana)

8090	Служба мониторинга состояния docker-контейнеров (Cadvisor)
9100	Служба мониторинга состояния хостовой машины (Node Exporter)

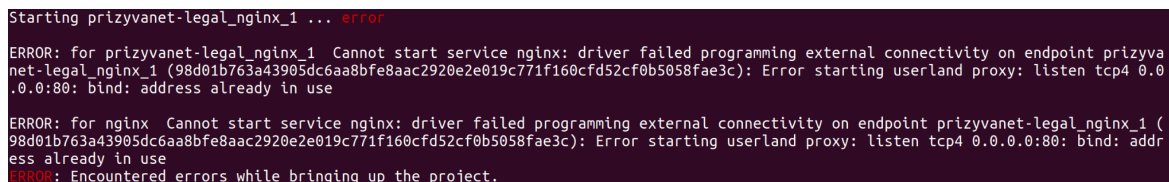
Открытие порта на ОС Ubuntu (Linux)

Открыть порт (например, 3050) в файерволе можно последовательным вводом команд в терминале:

```
sudo ufw allow 3050
sudo service ufw restart
```

Диагностика закрытых портов на ОС Ubuntu (Linux)

Если порт в системе занят каким-либо сервисом, то во время запуска стенда с Платформой (см. 6.3 Запуск стенда для экспертной проверки) можно получить ошибку вида “... bind: address already in use” (Рисунок 1).



```
Starting przyvanet-legal_nginx_1 ... error
ERROR: for przyvanet-legal_nginx_1 Cannot start service nginx: driver failed programming external connectivity on endpoint przyva
net-legal_nginx_1 (98d01b763a43905dc6aa8bfe8aac2920e2e019c771f160cfd52cf0b5058fae3c): Error starting userland proxy: listen tcp4 0.0
.0.0:80: bind: address already in use
ERROR: for nginx Cannot start service nginx: driver failed programming external connectivity on endpoint przyvanet-legal_nginx_1 (
98d01b763a43905dc6aa8bfe8aac2920e2e019c771f160cfd52cf0b5058fae3c): Error starting userland proxy: listen tcp4 0.0.0.0:80: Bind: addr
ess already in use
ERROR: Encountered errors while bringing up the project.
```

Рисунок 1

Первой причиной, почему порт может быть занят, является то, что в системе уже установлено платформа, которое слушает указанный порт (например, 80). Для проверки надо ввести команду (где 80 – номер проверяемого порта):

```
sudo lsof -i:80
```

Результат работы команды отобразится в терминале (Рисунок 2). В данном примере порт 80 занят веб-сервером nginx.

```

sudo lsof -i:80
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE  NAME
nginx    1625 root    6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1625 root    7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
nginx    1626 www-data 6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1626 www-data 7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
nginx    1627 www-data 6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1627 www-data 7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
nginx    1628 www-data 6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1628 www-data 7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
nginx    1629 www-data 6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1629 www-data 7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
nginx    1630 www-data 6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1630 www-data 7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
nginx    1631 www-data 6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1631 www-data 7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
nginx    1632 www-data 6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1632 www-data 7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
nginx    1633 www-data 6u  IPv4  43087      0t0  TCP  *:http (LISTEN)
nginx    1633 www-data 7u  IPv6  43088      0t0  TCP  *:http (LISTEN)
firefox  6178 hor    163u  IPv4  215810     0t0  TCP  WIN-HC1PMOT352S:45416->lm-ln-f94.1e100.net:http (ESTABLISHED)
firefox  6178 hor    164u  IPv4  218931     0t0  TCP  WIN-HC1PMOT352S:45418->lm-ln-f94.1e100.net:http (ESTABLISHED)
firefox  6178 hor    167u  IPv4  220353     0t0  TCP  WIN-HC1PMOT352S:45432->lm-ln-f94.1e100.net:http (ESTABLISHED)
firefox  6178 hor    182u  IPv4  222735     0t0  TCP  WIN-HC1PMOT352S:33858->104.18.31.182:http (ESTABLISHED)
firefox  6178 hor    193u  IPv4  219963     0t0  TCP  WIN-HC1PMOT352S:45434->lm-ln-f94.1e100.net:http (ESTABLISHED)
firefox  6178 hor    251u  IPv4  219038     0t0  TCP  WIN-HC1PMOT352S:43432->server-52-85-114-31.hel50.r.cloudfront.net:http (ESTABLISHED)
firefox  6178 hor    252u  IPv4  219039     0t0  TCP  WIN-HC1PMOT352S:41818->93.184.220.29:http (ESTABLISHED)
firefox  6178 hor    253u  IPv4  219040     0t0  TCP  WIN-HC1PMOT352S:41820->93.184.220.29:http (ESTABLISHED)
firefox  6178 hor    254u  IPv4  219041     0t0  TCP  WIN-HC1PMOT352S:41822->93.184.220.29:http (ESTABLISHED)
firefox  6178 hor    257u  IPv4  219053     0t0  TCP  WIN-HC1PMOT352S:54242->188.170.130.26:http (ESTABLISHED)
firefox  6178 hor    259u  IPv4  223000     0t0  TCP  WIN-HC1PMOT352S:43448->server-52-85-114-31.hel50.r.cloudfront.net:http (ESTABLISHED)
firefox  6178 hor    296u  IPv4  223035     0t0  TCP  WIN-HC1PMOT352S:42472->151.101.86.133:http (ESTABLISHED)
firefox  6178 hor    331u  IPv4  225687     0t0  TCP  WIN-HC1PMOT352S:42008->93.184.220.29:http (ESTABLISHED)
firefox  6178 hor    332u  IPv4  223093     0t0  TCP  WIN-HC1PMOT352S:42014->93.184.220.29:http (ESTABLISHED)
firefox  6178 hor    337u  IPv4  218047     0t0  TCP  WIN-HC1PMOT352S:42020->93.184.220.29:http (ESTABLISHED)

```

Рисунок 2

Сервис, занявший порт, можно отключить командой:

```
sudo kill -9 <PID>
```

Где <PID> – идентификатор процесса, полученный командой `sudo lsof -i:80` (во втором столбце вывода).

Чаще всего конкретно 80 порт может быть занят веб-сервером (Nginx или Apache), а 6379 БД Redis, установленными в системе, для выключения достаточно ввести команды (в зависимости, от того, какой именно сервис установлен):

```

sudo service nginx stop
sudo service apache stop
sudo service apache2 stop
sudo service redis stop

```

Второй причиной, почему занят порт, может быть то, что не все сервисы стенда синхронизировались при включении. Для этого нужно ввести в терминале команды, которые отключат и удалят все контейнеры и подключаемые к ним тома:

```

docker kill $(docker ps -a)
docker rm -f $(docker ps -a -q)
docker network rm $(docker network ls -q)
docker volume rm $(docker volume ls -q)

```

После чего можно перезапустить стенд командой:

```
docker-compose up --build
```

Все вышеперечисленные команды уже указаны в консольной утилите, для запуска достаточно в окне терминала ввести команду с последующим вводом пароля администратора :

```
bash ./run-local.sh
```

2. Архитектура ПО с установленным Модулем ЦП

ПО в целом состоит из Платформы с установленным Модулем ЦП, а также каналов связи между ними и установленных протоколов передачи данных между ними, при этом Платформа представляет собой совокупность следующих компонентов (Рисунок 3):

- модуль цифровых продуктов: редактор цифровых продуктов, серверная часть;
- модуль цифровых продуктов: интерфейс администрирования;
- мобильный бекенд – подсистема обмена данными с Платформой;
- подсистема авторизации;
- подсистема управления очередями и чатами;
- подсистема хранения данных;
- подсистема управления Базой Знаний;
- подсистема интеллектуального чат-бота и ядро интеллектуального чат-бота;
- подсистема мониторинга и логирования;
- подсистема администрирования;
- подсистема информационного обмена с внешними системами;

Модуль цифровых юридических продуктов использует микросервисную архитектуру, включает в себя следующие сервисы и экраны платформы:

1. Интерфейс администрирования, выполненный как веб-приложение SPA для визуального редактирования цифровых продуктов Администратором Платформы - зарегистрирован в Роспатент;
2. Редактор Цифровых продуктов. Серверная часть - для связи через API с платформой клиента и интерфейсом администрирования - зарегистрирован в Роспатент;
3. Экраны цифровых продуктов - для отображения состава Цифровых продуктов в платформе.

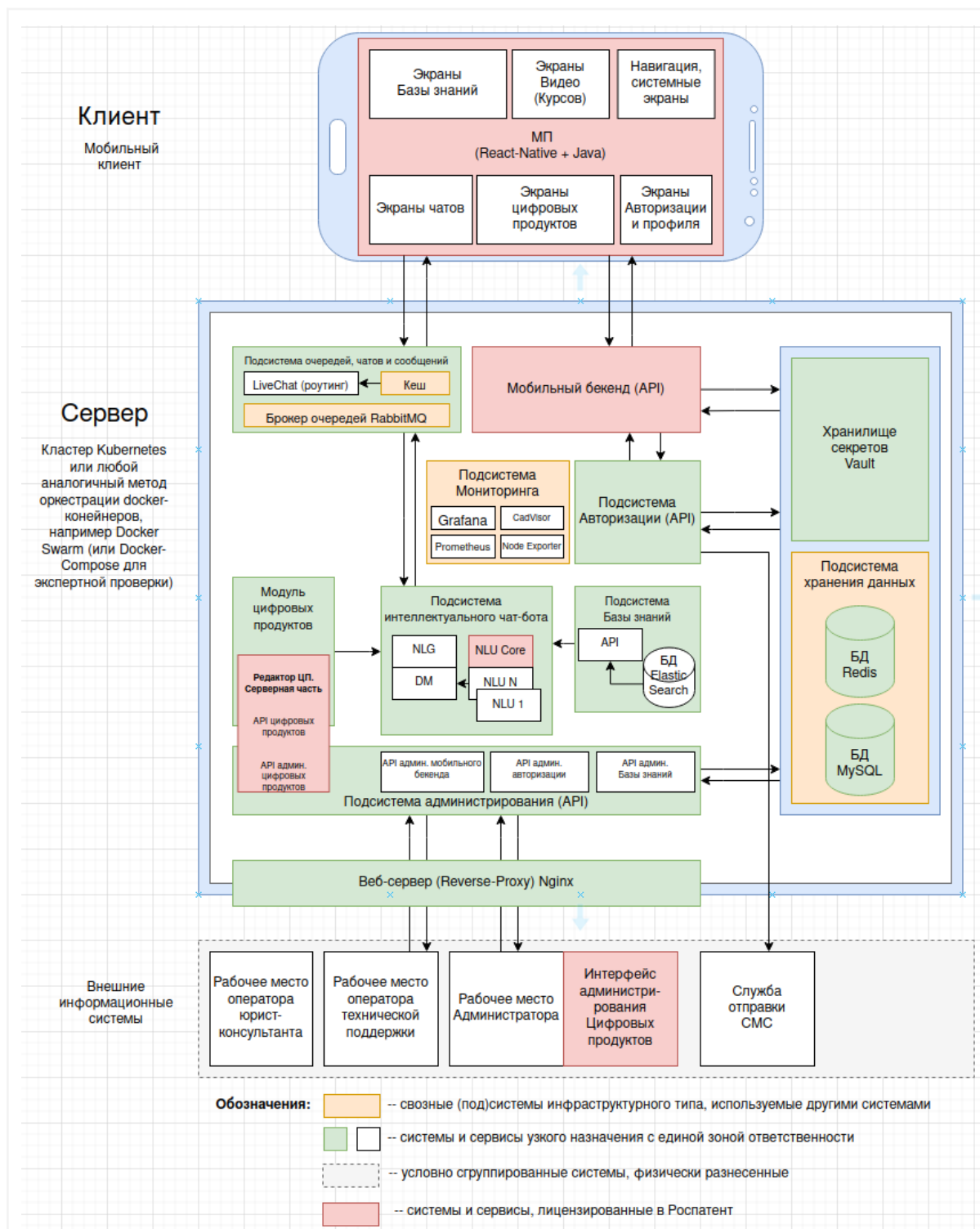


Рисунок 3. Архитектура ПО с установленным Модулем ЦП

3. Файловый состав поставки ПО с Модулем ЦП для проведения экспертной проверки

Все необходимые файлы содержатся внутри архива soft.zip.

После распаковки архива, необходимо перейти в папку ./soft и распаковать в ней архивы backend.zip и src.zip.

В результате этих действий в системе появятся файлы и папки, перечисленные в таблице 5.

Таблица 5 – Файловый состав поставки ПО

Сборки мобильного клиента (директория ./mobile_client/)	
legalhelp-stand.apk	Версия Платформы, запускаемая в эмуляторе Android Studio, настроенная на соединение с Платформой, развернутой на локальной машине для проведения экспертизы
Исходный код модуля ЦП (директория ./digitalproducteditor/)	
backend	Редактор модуля ЦП. Серверная часть
frontend	Интерфейс администрирования Модуля ЦП
Исходные файлы мобильного клиента (директория ./src/legalhelp-mobile/)	
android	Директория с исходными файлами
node_modules	Необходимые модули и зависимости
app	Общие для платформ Android и iOS файлы react-native
__tests__	Модульные тесты Платформы
Установщик зависимостей для ОС Ubuntu (Linux) - корневая директория архива	
./setup_environment.sh	bash-скрипт установщика
Платформа - серверная часть, запускаемая на рабочей машине для проведения экспертизы (директория ./backend)	

.env.example и .env	Переменные окружения для запуска Платформы
docker-compose.yml	Декларативное описание всех сервисов Платформы, необходимое для сборки и запуска docker-контейнеров
LiveChatAPI.md	Документация разработчика по использованию системы чатов Платформы
README.md	Документация разработчика по настройке и запуску Платформы
./config/	конфигурация для получения секретов из Vault во время экспертной проверки
./dependencies/	Подсистема чатов
./dependencies/livechat-backend/	Серверная часть подсистемы чатов
./dependencies/livechat-frontend/	Веб-клиент подсистемы чатов
./docker/	Инфраструктурные зависимости, обеспечивающие работоспособность Платформы
./docker/grafana/	Конфигурация UI сервиса мониторинга за состоянием платформы (Grafana)
./docker/nginx/	Конфигурация веб-сервера (Nginx)
./docker/prometheus/	Конфигурация сбора данных сервиса мониторинга за состоянием платформы (Prometheus)
./docker/scripts/	Конфигурация наблюдения за изменением секретов (Vault)
./docker/vault/	Конфигурация системы хранения секретов (Vault)
./docker/*.dockerfile	Файлы с описанием docker-контейнеров инфраструктурных зависимостей, обеспечивающих работоспособность Платформы
./mob-faq/	Компоненты Базы знаний
./mob-faq/data/	Директория хранения данных для Базы знаний
./mob-faq/svc-app/	Адаптер, обеспечивающий данными чат-бота Платформы при обращении к Базе знаний

./mob-faq/svc-els/Dockerfile	Описание docker-контейнера Elasticsearch с морфологическим поиском для русского языка
./nlu/	Компоненты интеллектуального чат-бота, реализующие обработку естественного языка методами машинного обучения
./nlu/amqp_adapter/	Адаптер интеллектуального чат-бота для подключения к очередям чатов RabbitMQ
./nlu/bert/	NLU, реализующая модель BERT для распознавания намерения пользователя
./nlu/core/	NLU, реализующая модель с нейронной сетью прямого распространения для распознавания намерения пользователя. Ядро интеллектуального чат-бота.
./nlu/elmo/	NLU, реализующая модель ELMO для распознавания намерения пользователя
./nlu/dialog_manager/	Менеджер диалога интеллектуального чат-бота, обеспечивающий голосование NLU-моделей распознавания намерений и извлекающий ответ из Базы знаний
./src/	Мобильный бекенд
./src/api/	Компоненты REST-API мобильного бекенда
./src/entities/	Сущности для передачи и управления данными
./src/services/	Сервисы управления данными
./src/app.ts	Точка входа Мобильного бекенда
./src/swaggerSpec.ts	Спецификация swagger, документирующая API мобильного бекенда

4. Порядок установки ПО LEGALHELP с Модулем ЦП

Для проведения экспертной проверки необходимо выполнить следующие шаги:

- 1) Установить ОС Ubuntu (см. 3.1 Установка Ubuntu).
- 2) Зайти в папку **software-legalhelp** и запустить в терминале инсталлятор **setup_environment.sh**

```
cd ./software-legalhelp/
```

```
/bin/bash ./setup_environment.sh
```

Все необходимые зависимости будут установлены автоматически.

Если зависимости **docker** и **docker-compose** не установились, необходимо их установить вручную (см. 3.3. Установка **docker** и **docker-compose**).

- 3) Установить **Android Studio** (см. 3.2 Установка **Android Studio**).

4.1. Установка модуля ЦП

В корневую папку проекта требуется разместить код Модуля ЦП, который находится в папке **digitalproducteditor** и содержит папки **backend** с серверной частью и **frontend** с интерфейсом администрирования.

В файле **docker-compose.yml** должны быть добавлены и не закомментированы следующие строки (см. Листинг 1).

Листинг 1. Подключение сервисов Модуля ЦП к Платформе

```
dpbackend:
  build: ./digitalproducteditor/backend
  depends_on:
    - mysql
    - nginx
  networks:
    lt-mobile:
      ipv4_address: ${NETWORK_PREFIX:-173.200}.0.36
  aliases:
    - dpbackend

dpfrontend:
  build: ./digitalproducteditor/frontend
  depends_on:
    - mysql
    - dpbackend
    - nginx
    - mobile-api
  networks:
    lt-mobile:
```

```
ipv4_address: ${NETWORK_PREFIX:-173.200}.0.37
aliases:
  - dpfrontend
```

Для проведения экспертной проверки все необходимые файлы уже установлены и сконфигурированы.

4.2 Установка Платформы на мобильном телефоне

Данный раздел носит исключительно справочный характер для полноты предоставляемых сведений.

Важно! Целевой аппаратной архитектурой для Платформы является платформа Android мобильного телефона, **однако для проведения экспертизы запуск Платформы осуществляется в эмуляторе Android Studio** (см. 4.3 Установка Платформы в эмуляторе Android Studio).

Подробное описание установки на платформе мобильного телефона содержится в документе “Руководство пользователя Платформы "LEGALHELP"”.

4.3. Установка Платформы в эмуляторе Android Studio

Важно! Прежде, чем устанавливать Платформу, необходимо выполнить инструкции из раздела “4.4.1. Завершение установки Android Studio и дополнительных пакетов”.

- 1) Запустите Android Studio. Для этого перейдите в папку `bin` сохраненного и распакованного архива Android Studio и запустите `./studio.sh`.



- 2) Нажмите на иконку AVD Manager и запустите эмулятор (клик по зеленому треугольнику "play" - Рисунок 40)

⁶ <https://www.applegalhelp.ru/3.pdf>

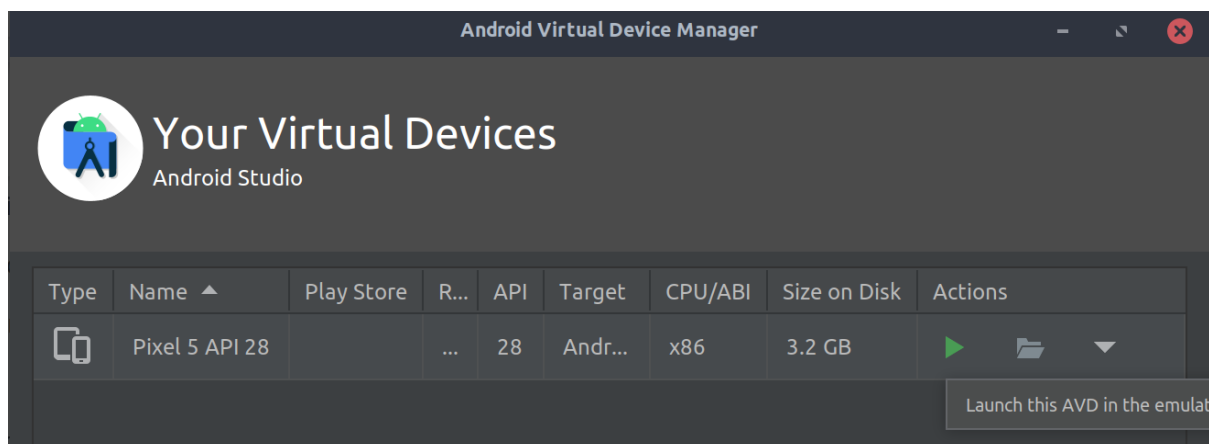


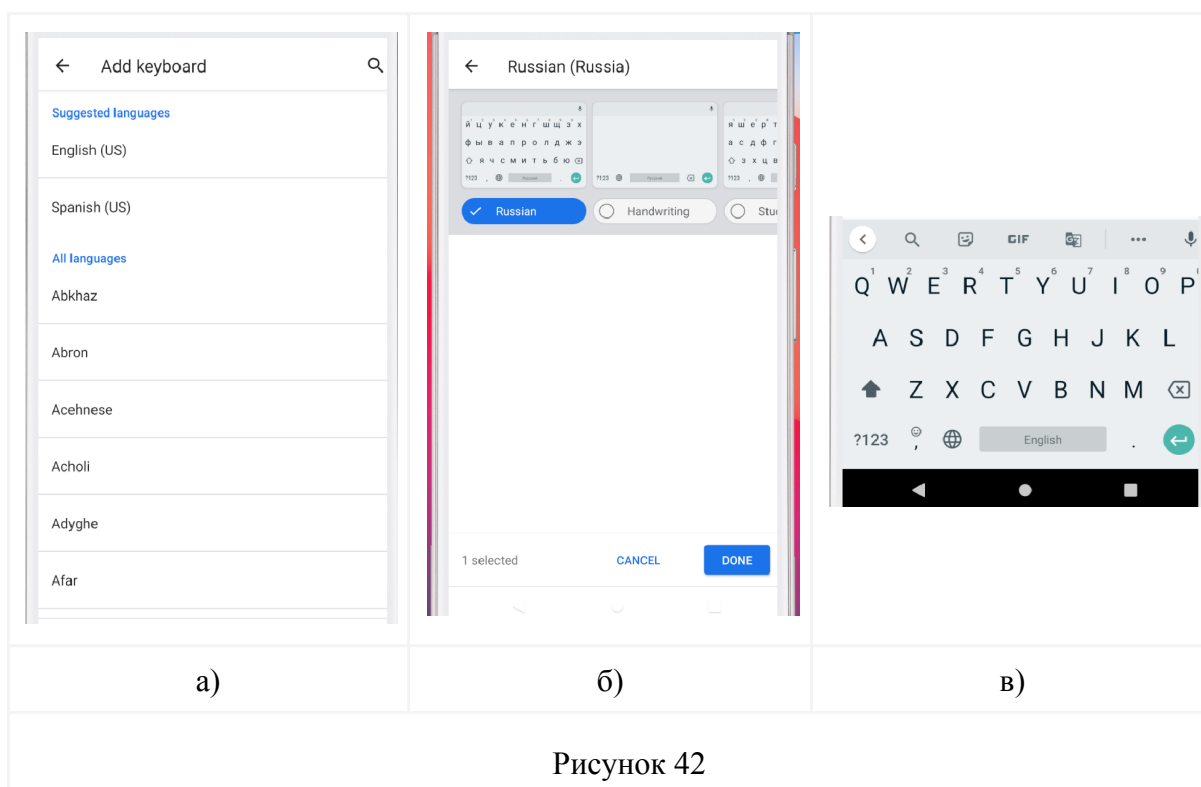
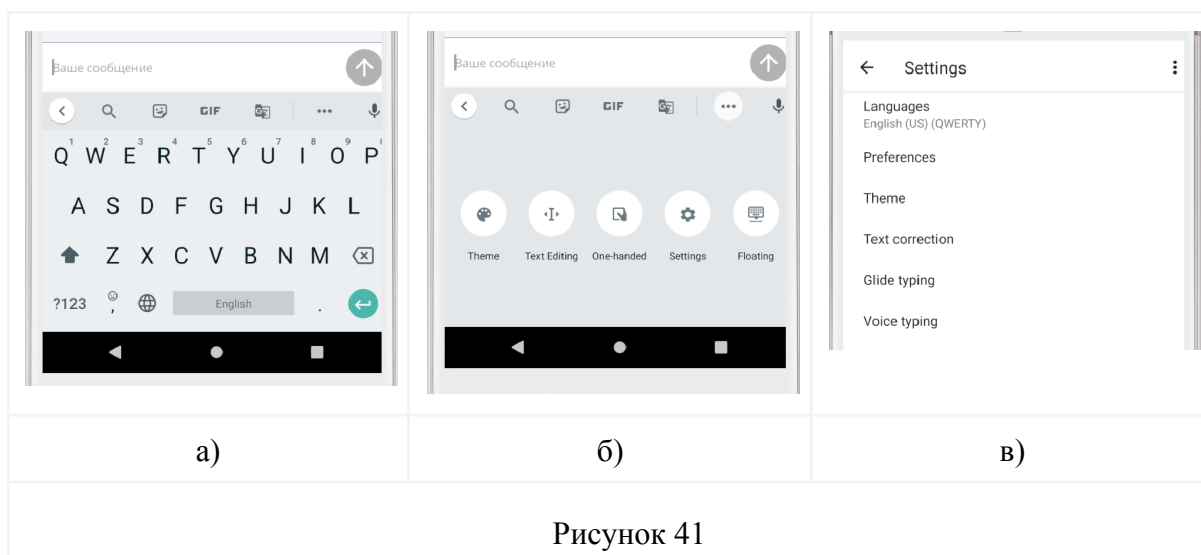
Рисунок 40. Запуск Android в эмуляторе

- 3) Перенесите [app-dev-release-local.apk](#) файл в открывшийся эмулятор - перетаскиванием и отпускаем в окне (drag and drop функционал) - файл лежит в папке `~/soft/mobile_client/`
- 4) Откройте платформа в эмуляторе

4.4. Установка русскоязычной клавиатуры в эмулятор

При первой необходимости в русской раскладке следует:

- 1) Над буквами “QWERTY” клавиатуры нужно нажать на иконку “стрелка вправо”, затем нажать на иконку “три точки” с правой стороны (Рисунок 41а).
- 2) Выбрать Settings (Рисунок 41б).
- 3) Выбрать Languages (Рисунок 41в).
- 4) Выбрать ADD KEYBOARD.
- 5) Найти в разделе All languages Russian и нажать (Рисунок 42а).
- 6) Выбрать Russian(Russia).
- 7) Выбрать Done (Рисунок 42б).
- 8) Добавлена русская клавиатура.
- 9) Слева вверху в меню нажать на иконку “Стрелка назад”, чтобы выйти из добавления клавиатур.
- 10) Еще раз нажать в меню на иконку “Стрелка назад”, чтобы выйти из раздела Settings.
- 11) При следующем показе клавиатуры в нижнем ряду появится символ “Глобус” для переключения языка (Рисунок 40в).



5. Методика комплексного тестирования Модуля ЦП в Платформе

Для того, чтобы провести комплексное тестирование ПО Платформы без использования мобильного клиента, необходимо перейти в папку `~/soft/backend/` и запустить `./run-local.sh`. Загрузка в первый раз может быть долгой (см. “6.3 Запуск стенда для экспертной проверки”).

Ради удобства тестирования воспользуйтесь утилитами из раздела “6.4. Инструменты, моделирующие поведение внешних систем”.

5.1. Методика комплексного тестирования Модуля ЦП, серверной части и базы данных

Описание структуры базы данных

База данных Редактора содержит 4 таблицы:

- Цифровые продукты (digital_products)
- Базовые свойства цифровых продуктов (feature_base)
- Свойства цифровых продуктов (digital_product_feature)
- Настройки для платформы (app_settings)

Ниже представлено описание каждой из таблиц:

Цифровые продукты

Название поля	Тип	Описание
name	Строка (не более 255 символов)	Название ЦП
description	Строка (не более 5000 символов)	Описание ЦП
icon_id	Целое число	ID изображения ЦП для платформы
is_visible	Булево значение	Отображать ли продукт в платформе.
enabled	Булево значение	Доступен ли цифровой

		продукт для использования
--	--	---------------------------

Базовые свойства ЦП

Название поля	Тип	Описание
feature_id	Строка (не более 255 символов)	Идентификатор базового свойства
name_base	Строка (не более 255 символов)	Название базового свойства
tooltip	Строка (не более 5000 символов)	Описание базового свойства

Свойства ЦП

Название поля	Тип	Описание
id	Целое число	Идентификатор свойства
digital_product_id	Целое число	Идентификатор ЦП
feature_id	Строка (не более 255 символов)	Идентификатор базового свойства
name_override	Строка (не более 255 символов)	Название базового свойства
tooltip	Строка (не более 5000 символов)	Описание базового свойства

Настройки платформы

Название поля	Тип	Описание
vk_auth_enabled	Булево значение	Зарезервировано для интеграции с ВКонтакте
app_only_available_for	Строка (не более 255 символов)	Указывается для целей тестирования
profile_edit_enabled	Булево значение	Флаг редактирования профиля
profile_phone_change_enabled	Булево значение	Флаг редактирования номера телефона

survey_page	Строка (не более 255 символов)	Страница продукта в интернете
escort_chat_enabled	Булево значение	Зарезервировано для будущих версий
agent_actions_show	Булево значение	Зарезервировано для будущих версий
user_can_write_messages_to_specs	Булево значение	Зарезервировано для будущих версий
manager_chat_enabled	Булево значение	Зарезервировано для будущих версий
sos_enabled	Булево значение	Зарезервировано для будущих версий
pin_enabled	Булево значение	Установка ПИН-кода вместо пароля
welcome_video_id	Строка (не более 255 символов)	Указывается для просмотра видео при запуске платформы
archived_chats_enabled	Булево значение	Флаг для архивации чатов
faq_enabled	Булево значение	Флаг управления базой знаний (по всем ЦП)
payments_split_enabled	Булево значение	Зарезервировано для внедрения платежей
mlm_enabled	Булево значение	Зарезервировано для будущих версий

5.1.2. Описание API бэкенда сервиса Редактора ЦП, серверной части

Протестировать API Редактора можно через свagger по адресу <http://localhost:8070/docs> (при условии запуска на локальном компьютере). Ниже показан интерфейс сваггера

Digitalproduct		^
GET	/digitalproduct Get All	▼
POST	/digitalproduct Create One	▼
GET	/digitalproduct/{item_id} Get One	▼
PUT	/digitalproduct/{item_id} Update One	▼
DELETE	/digitalproduct/{item_id} Delete One	▼
Featurebase		^
GET	/featurebase Get All	▼
POST	/featurebase Create One	▼
GET	/featurebase/{item_id} Get One	▼
PUT	/featurebase/{item_id} Update One	▼
DELETE	/featurebase/{item_id} Delete One	▼
Featuredigitalproduct		^
GET	/featuredigitalproduct/{digital_product_id} Read Digital Product Features	▼
POST	/featuredigitalproduct/{digital_product_id} Create Digital Product Feature	▼
GET	/featuredigitalproduct/{digital_product_id}/{feature_digital_product_id} Read Digital Product Feature	▼
PUT	/featuredigitalproduct/{digital_product_id}/{feature_digital_product_id} Update Digital Product Feature	▼
DELETE	/featuredigitalproduct/{digital_product_id}/{feature_digital_product_id} Delete Digital Product Feature	▼
Appsettings		^
GET	/appsettings Read Current Settings	▼
PUT	/appsettings Update App Settings	▼
Appupdateparams		^
GET	/api/v1/notifications/updateParams Read App Updates	▼

В API Редактора цифровых продуктов доступны следующие методы:

1. Получить список всех цифровых продуктов, имеющих в базе
 - a. тип запроса: “**GET**”
 - b. постфикс url-адреса: “**/digitalproduct**”
 - c. Пример запроса:


```
curl -X 'GET' 'http://localhost:8070/digitalproduct' -H 'accept: application/json'
```

d. Пример ответа:

```
Код ответа: 200
[
  {
    "name": "Трудовое право",
    "description": "Цифровой продукт по трудовому праву",
    "icon_id": 0,
    "is_visible": false,
    "enabled": true,
    "url": "",
    "digital_product_id": 1
  },
  {
    "name": "Социальные вопросы",
    "description": "Цифровой продукт по социальной тематике",
    "icon_id": 1,
    "is_visible": false,
    "enabled": true,
    "url": "",
    "digital_product_id": 2
  }
]
```

2. Создать новый цифровой продукт в базе данных Редактора

- a. тип запроса: **“POST”**
- b. постфикс url-адреса: **“/digitalproduct”**
- c. Тело запроса:

```
{
  "name": строка длиной не более 255 символов,
  "description": строка длиной не более 5000 символов,
  "icon_id": целое число,
  "is_visible": булево значение,
  "enabled": булево значение,
  "url": строка
}
```

d. Пример запроса:

```
curl -X 'POST' 'http://localhost:8070/digitalproduct' -H \
'accept: application/json' -H 'Content-Type: application/json' \
-d '{
  "name": "Новый продукт",
  "description": "Test new product",
  "icon_id": 0,
  "is_visible": false,
  "enabled": true,
  "url": ""
}'
```

```
"icon_id": 0,  
"is_visible": true,  
"enabled": true,  
"url": "http://newproduct.test"  
'}
```

е. Пример ответа:

Код ответа: 200

```
{  
  "name": "Новый продукт",  
  "description": "Test new product",  
  "icon_id": 0,  
  "is_visible": true,  
  "enabled": true,  
  "url": "http://newproduct.test",  
  "digital_product_id": 4  
}
```

3. Получить конкретный цифровой продукт по его digital_product_id

а. тип запроса: **“GET”**

б. постфикс url-адреса: **“/digitalproduct/{digital_product_id}”**, где digital_product_id - номер искомого ЦП в базе данных

с. Пример запроса:

```
curl -X 'GET' 'http://localhost:8070/digitalproduct/10' -H 'accept:  
application/json'
```

д. Пример ответа в случае если продукт с указанным номером найден:

Код ответа: 200

```
{  
  "name": "Новый продукт",  
  "description": "Test new product",  
  "icon_id": 0,  
  "is_visible": true,  
  "enabled": true,  
  "url": "http://newproduct.test",  
  "digital_product_id": 4  
}
```

е. Пример ответа в случае, когда продукт с указанным номером не найден:

Код ответа: 404

```
{  
  "detail": "Item not found"  
}
```

4. Изменить конкретный цифровой продукт по его digital_product_id

а. тип запроса: **“PUT”**

- b. постфикс url-адреса: “/digitalproduct/{digital_product_id}”, где digital_product_id - номер искомого ЦП в базе данных
- c. Тело запроса:

```
{
  "name": строка длиной не более 255 символов,
  "description": строка длиной не более 5000 символов,
  "icon_id": целое число,
  "is_visible": булево значение,
  "enabled": булево значение,
  "url": строка длиной не более 255 символов
}
```

- d. Пример запроса:

```
curl -X 'PUT' \
'http://localhost:8070/digitalproduct/4' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "name": "Измененный цифровой продукт",
  "description": "New product was changed",
  "icon_id": 0,
  "is_visible": true,
  "enabled": true,
  "url": "newurlfordigitalproduct.test"
}'
```

- e. Пример ответа в случае если продукт с указанным номером найден:
Код ответа: 200

```
{
  "name": "Измененный цифровой продукт",
  "description": "New product was changed",
  "icon_id": 0,
  "is_visible": true,
  "enabled": true,
  "url": "newurlfordigitalproduct.test",
  "digital_product_id": 4
}
```

- f. Пример ответа в случае если продукт с указанным номером не найден:
Код ответа: 404

```
{
  "detail": "Item not found"
}
```

- g. Пример ответа в случае если в теле запроса невалидный объект:
Код ответа: 422

```
{
  "detail": [
    {
      "loc": [
        "body",
        "name"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

5. Удалить конкретный цифровой продукт по его digital_product_id

- тип запроса: **“DELETE”**
- постфикс url-адреса: **“/digitalproduct/{digital_product_id}”**, где digital_product_id - номер искомого ЦП в базе данных
- Пример запроса:

```
curl -X 'DELETE' 'http://localhost:8070/digitalproduct/4' -H 'accept: application/json'
```

- Пример ответа в случае если продукт с указанным номером найден:
Код ответа: 200

```
{
  "name": "Измененный цифровой продукт",
  "description": "New product was changed",
  "icon_id": 0,
  "is_visible": true,
  "enabled": true,
  "url": "newurlfordigitalproduct.test",
  "digital_product_id": 4
}
```

- Пример ответа в случае если продукт с указанным номером не найден:
Код ответа: 404

```
{
  "detail": "Item not found"
}
```

6. Получить список всех базовых свойств цифровых продуктов

- тип запроса: **“GET”**
- постфикс url-адреса: **“/featurebase”**
- Пример запроса:

```
curl -X 'GET' 'http://localhost:8070/featurebase' -H 'accept: application/json'
```

- Пример ответа в случае если продукт с указанным номером найден:
Код ответа: 200

```
[
  {
    "feature_id": "id-func-video",
    "name_base": "Video",
    "tooltip": "video tooltip"
  }
]
```

7. Создать новое базовое свойство цифрового продукта

- a. тип запроса: **“POST”**
- b. постфикс url-адреса: **“/featurebase”**
- c. Тело запроса:

```
{
  "feature_id": строка длиной не более 255 символов,
  "name_base": строка длиной не более 255 символов,
  "tooltip": строка длиной не более 5000 символов
}
```

- d. Пример запроса:

```
curl -X 'POST' \
'http://localhost:8070/featurebase' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "feature_id": "id-func-new",
  "name_base": "NewBaseFeature",
  "tooltip": "Test featurebase post"
}'
```

- e. Пример ответа:

Код ответа: 200

```
{
  "feature_id": "id-func-new",
  "name_base": "NewBaseFeature",
  "tooltip": "Test featurebase post"
}
```

- f. Пример ответа:

Код ответа: 422

```
{
  "detail": [
    {
      "loc": [
        "body",
        "feature_id"
      ]
    }
  ]
}
```

```

    ],
    "msg": "field required",
    "type": "value_error.missing"
  }
]
}

```

8. Получить конкретное базовое свойство ЦП по его feature_id

- тип запроса: **“GET”**
- постфикс url-адреса: **“/featurebase/{feature_id}”**, где feature_id - feature_id искомого базового свойства ЦП
- Пример запроса:

```

curl -X 'GET' 'http://localhost:8070/featurebase/id-func-new' \
-H 'accept: application/json'

```

- Пример ответа в случае если продукт с указанным номером найден:
Код ответа: 200

```

{
  "feature_id": "id-func-new",
  "name_base": "NewBaseFeature",
  "tooltip": "Test featurebase post"
}

```

- Пример ответа в случае если продукт с указанным номером не найден:
Код ответа: 404

```

{
  "detail": "Item not found"
}

```

9. Изменить конкретное базовое свойство ЦП по его feature_id

- тип запроса: **“PUT”**
- постфикс url-адреса: **“/featurebase/{feature_id}”**, где feature_id - id базового свойства
- Тело запроса:


```

{
  "name_base": строка длиной не более 255 символов,
  "tooltip": строка длиной не более 5000 символов,
}

```
- Пример запроса:

```

curl -X 'PUT' \
'http://localhost:8070/featurebase/id-func-new' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "name_base": "New Name",
  "tooltip": "New Tooltip"
}'

```

```
}'
```

- e. Пример ответа в случае если продукт с указанным номером найден:
Код ответа: 200

```
{
  "feature_id": "id-func-new",
  "name_base": "New Name",
  "tooltip": "New Tooltip"
}
```

- f. Пример ответа в случае если продукт с указанным номером не найден:
Код ответа: 404

```
{
  "detail": "Item not found"
}
```

- g. Пример ответа в случае если в теле запроса невалидный объект:
Код ответа: 422

```
{
  "detail": [
    {
      "loc": [
        "body",
        "name"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

10. Удалить конкретное базовое свойство ЦП по его feature_id

- a. тип запроса: **“DELETE”**
b. постфикс url-адреса: **“/featurebase/{feature_id}”**
c. Пример запроса:

```
curl -X 'DELETE' 'http://localhost:8070/featurebase/id-func-new' \
-H 'accept: application/json'
```

- d. Пример ответа в случае если продукт с указанным номером найден:
Код ответа: 200

```
{
  "feature_id": "id-func-new",
  "name_base": "New Name",
  "tooltip": "New Tooltip"
}
```

```
}

```

- е. Пример ответа в случае если продукт с указанным номером не найден:
Код ответа: 404

```
{
  "detail": "Item not found"
}

```

11. Получить список всех свойств данного цифрового продукта

- а. тип запроса: **“GET”**
б. постфикс url-адреса: **“/featuredigitalproduct/{digital_product_id}”**,
digital_product_id - id цифрового продукта
в. Пример запроса:

```
curl -X 'GET' 'http://localhost:8070/featuredigitalproduct/1' \
-H 'accept: application/json'

```

- д. Пример ответа в случае если продукт с указанным номером найден:
Код ответа: 200

```
[
  {
    "feature_id": "id-func-video",
    "name_override": "Видео по трудовому праву",
    "tooltip_override": "видео-консультации по трудовому праву",
    "id": 1
  }
]

```

- е. Пример ответа в случае если продукт с указанным номером не найден:
Код ответа: 404

```
{
  "detail": "Digital product not found"
}

```

12. Создать новое свойство у цифрового продукта

- а. тип запроса: **“POST”**
б. постфикс url-адреса: **“/featuredigitalproduct/{digital_product_id}”**
в. Тело запроса:

```
{
  "feature_id": строка длиной не более 255 символов (должно
    совпадать с feature_id базового свойства),
  "name_override": строка длиной не более 255 символов,
  "tooltip_override": строка длиной не более 5000 символов
}

```

- д. Пример запроса:

```
curl -X 'POST' \

```



```
'http://localhost:8070/featuredigitalproduct/1' \  
-H 'accept: application/json' \  
-H 'Content-Type: application/json' \  
-d '{  
    "feature_id": "id-func-video",  
    "name_override": "New dp feature",  
    "tooltip_override": "just for test"  
}'
```

е. Пример ответа:

Код ответа: 200

```
{  
    "feature_id": "id-func-video",  
    "name_override": "New dp feature",  
    "tooltip_override": "just for test",  
    "id": 2  
}
```

ф. Примеры ответа:

Код ответа: 404

```
{  
    "detail": "Digital product not found"  
}
```

```
{  
    "detail": "Digital product feature not found"  
}
```

г. Пример ответа:

Код ответа: 422

```
{  
    "detail": [  
        {  
            "loc": [  
                "body",  
                "feature_id"  
            ],  
            "msg": "field required",  
            "type": "value_error.missing"  
        }  
    ]  
}
```

13. Получить конкретное свойство ЦП по id продукта и id свойства

а. тип запроса: “**GET**”

б. постфикс url-адреса:

“/featuredigitalproduct{digital_product_id}/{feature_digital_product_id},

где digital_product_id - id цифрового продукта, feature_digital_product_id - id свойства цифрового продукта

- c. Пример запроса:

```
curl -X 'GET' \
'http://localhost:8070/featuredigitalproduct/1/2' -H 'accept: application/json'
```

- d. Пример ответа в случае если продукт с указанным номером найден:
Код ответа: 200

```
{
  "feature_id": "id-func-video",
  "name_override": "New dp feature",
  "tooltip_override": "just for test",
  "id": 2
}
```

- e. Пример ответа в случае если продукт с указанным номером не найден:
Код ответа: 404

```
{
  "detail": "Digital product not found"
}
```

```
{
  "detail": "Digital product feature not found"
}
```

14. Изменить конкретное свойство цифрового продукта по digital_product_id и feature_digital_product_id

- a. тип запроса: **“PUT”**
b. постфикс url-адреса:
“featuredigitalproduct/{digital_product_id}/{feature_digital_product_id}”
c. Тело запроса:

```
{
  "feature_id": "id-func-video",
  "name_override": "changed name",
  "tooltip_override": "changed tooltip"
}
```

- d. Пример запроса:

```
curl -X 'PUT' \
'http://localhost:8070/featuredigitalproduct/1/2' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "feature_id": "id-func-video",
  "name_override": "changed name",
  "tooltip_override": "changed tooltip"
}'
```

```
}'
```

- е. Пример ответа в случае если продукт с указанным номером и свойство с указанным номером найдены:

Код ответа: 200

```
null
```

- ф. Пример ответа в случае если продукт с указанным номером или свойство продукта с указанным номером не найдены:

Код ответа: 404

```
{
  "detail": "Digital product not found"
}
```

```
{
  "detail": "Digital product feature not found"
}
```

- г. Пример ответа в случае если в теле запроса невалидный объект:

Код ответа: 422

```
{
  "detail": [
    {
      "loc": [
        "body",
        "name"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

15. Удалить конкретное базовое свойство ЦП по его feature_id

- а. тип запроса: **“DELETE”**

- б. постфикс url-адреса:

“featuredigitalproduct/{digital_product_id}/{feature_digital_product_id}”

- с. Пример запроса:

```
curl -X 'DELETE' \
'http://localhost:8070/featuredigitalproduct/1/2' -H 'accept: application/json'
```

- д. Пример ответа в случае если продукт с указанным номером найден:

Код ответа: 200

```
{
  "feature_id": "id-func-video",
  "name_override": "changed name",
}
```

```
"tooltip_override": "changed tooltip",
"id": 2
}
```

- е. Пример ответа в случае если продукт с указанным номером или свойство продукта с указанным номером не найдены:

Код ответа: 404

```
{
  "detail": "Digital product not found"
}
```

```
{
  "detail": "Digital product feature not found"
}
```

16. Получить список настроек для платформы (для редактора цифровых продуктов)

- а. тип запроса: **“GET”**
- б. постфикс url-адреса: **“/appsettings”**
- в. Пример запроса:

```
curl -X 'GET' 'http://localhost:8070/appsettings' -H 'accept: application/json'
```

- д. Пример ответ:

Код ответа: 200

```
{
  "vk_auth_enabled": true,
  "app_only_available_for": "",
  "profile_edit_enabled": false,
  "profile_phone_change_enabled": false,
  "survey_page": "https://forms.yandex.ru/u/601d3c4638eaf64b900176ad/",
  "escort_chats_enabled": true,
  "agent_actions_show": true,
  "user_can_write_messages_to_specs": false,
  "manager_chat_enabled": true,
  "sos_enabled": true,
  "pin_enabled": true,
  "welcome_video_id": "IOBrvvPkIA4",
  "archived_chats_enabled": false,
  "faq_enabled": false,
  "payments_split_enabled": true,
  "mlm_enabled": true
}
```

17. Изменить настройки для платформы

- а. тип запроса: **“PUT”**
- б. постфикс url-адреса: **“/appsettings”**
- в. Тело запроса:

```
{
```

```
"vk_auth_enabled": true,
"app_only_available_for": "string",
"profile_edit_enabled": true,
"profile_phone_change_enabled": true,
"survey_page": "string",
"escort_chats_enabled": true,
"agent_actions_show": true,
"user_can_write_messages_to_specs": true,
"manager_chat_enabled": true,
"sos_enabled": true,
"pin_enabled": true,
"welcome_video_id": "string",
"archived_chats_enabled": true,
"faq_enabled": true,
"payments_split_enabled": true,
"mlm_enabled": true
}
```

d. Пример запроса:

```
curl -X 'PUT' \
'http://localhost:8070/appsettings' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "vk_auth_enabled": true,
  "app_only_available_for": "string",
  "profile_edit_enabled": true,
  "profile_phone_change_enabled": true,
  "survey_page": "string",
  "escort_chats_enabled": true,
  "agent_actions_show": true,
  "user_can_write_messages_to_specs": true,
  "manager_chat_enabled": true,
  "sos_enabled": true,
  "pin_enabled": true,
  "welcome_video_id": "string",
  "archived_chats_enabled": true,
  "faq_enabled": true,
  "payments_split_enabled": true,
  "mlm_enabled": true
}'
```

e. Пример ответа:

Код ответа: 200

```
null
```

- f. Пример ответа в случае если в теле запроса невалидный объект:
Код ответа: 422

```
{
  "detail": [
    {
      "loc": [
        "body",
        "name"
      ],
      "msg": "field required",
      "type": "value_error.missing"
    }
  ]
}
```

18. Получить список настроек для платформы (для платформы)

- a. тип запроса: **“GET”**
b. постфикс url-адреса: **“/api/v1/notifications/updateParams”**
c. Пример запроса:

```
curl -X 'GET' \
'http://localhost:8070/api/v1/notifications/updateParams' \
-H 'accept: application/json'
```

- d. Пример ответа:
Код ответа: 200

```
{
  "currentRevision": 2,
  "storeLink": "https://vk.cc/bYgJZP",
  "force": true,
  "appSettings": {
    "profile_edit_enabled": false,
    "survey_page":
"https://forms.yandex.ru/u/601d3c4638eaf64b900176ad/",
    "vk_auth_enabled": true,
    "agent_actions_show": true,
    "user_can_write_messages_to_specs": false,
    "sos_enabled": true,
    "welcome_video_id": "IOBrvvPkIA4",
    "faq_enabled": false,
    "mlm_enabled": true,
    "app_only_available_for": "",
    "profile_phone_change_enabled": false,
    "escort_chats_enabled": true,
    "manager_chat_enabled": true,
  }
}
```

```
    "pin_enabled": true,
    "archived_chats_enabled": false,
    "payments_split_enabled": true,
    "digitalproducts": [
      {
        "digital_product_id": 1,
        "description": "Цифровой продукт по трудовому праву",
        "is_visible": false,
        "url": "",
        "icon_id": 0,
        "name": "Трудовое право",
        "enabled": true
      },
      {
        "digital_product_id": 2,
        "description": "Цифровой продукт по социальной тематике",
        "is_visible": false,
        "url": "",
        "icon_id": 1,
        "name": "Социальные вопросы",
        "enabled": true
      }
    ]
  }
}
```

5.3. Описание фронтенда Редактора

Фронтенд редактора цифровых продуктов предназначен для управления цифровыми продуктами, их составом и свойствами

5.3.1. Руководство по использованию фронтенда Редактора

Фронтенд доступен для использования по адресу <http://localhost:8080>

Главная страница предназначена для управления непосредственно набором цифровых продуктов

Управление цифровыми продуктами

Создать продукт

Список цифровых продуктов

	Трудовое право	Редактировать	Состав	Удалить
	Социальные вопросы	Редактировать	Состав	Удалить

Интерфейс страницы позволяет:

- просмотреть список имеющихся цифровых продуктов. При нажатии на поле цифрового продукта появляется/скрывается его описание.

Управление цифровыми продуктами

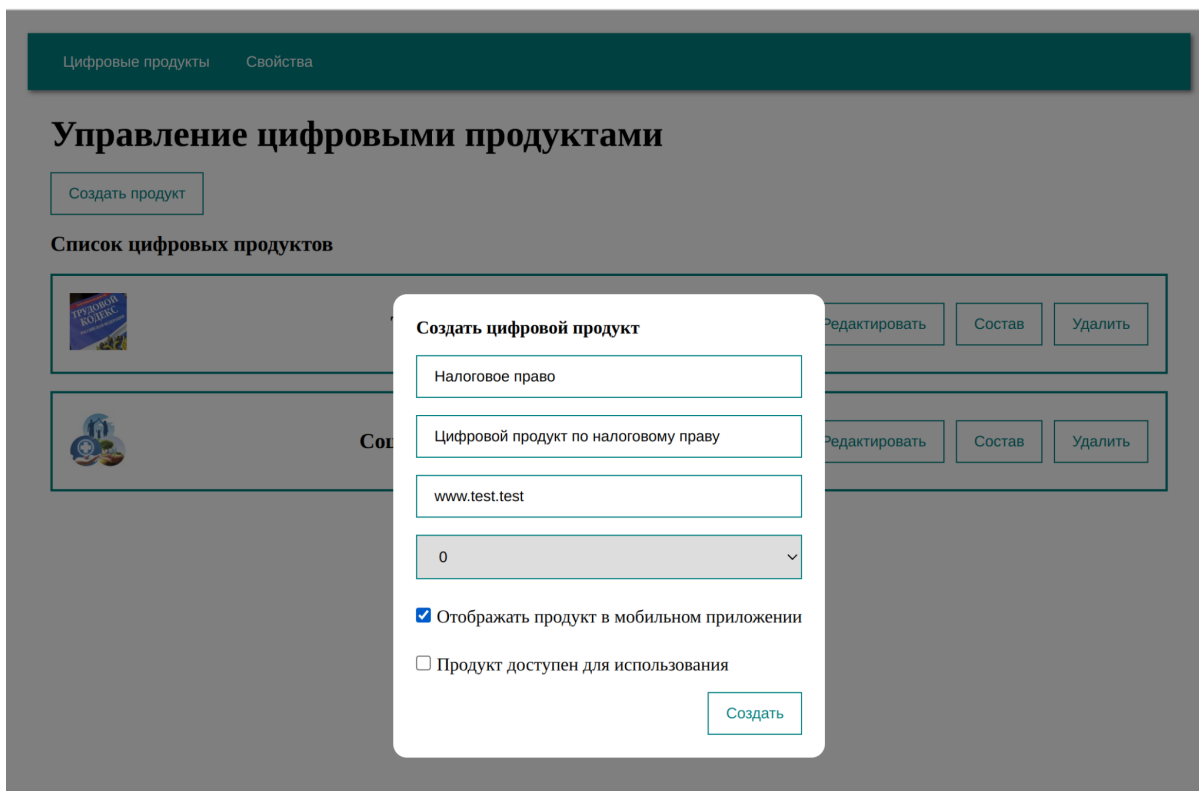
Создать продукт

Список цифровых продуктов

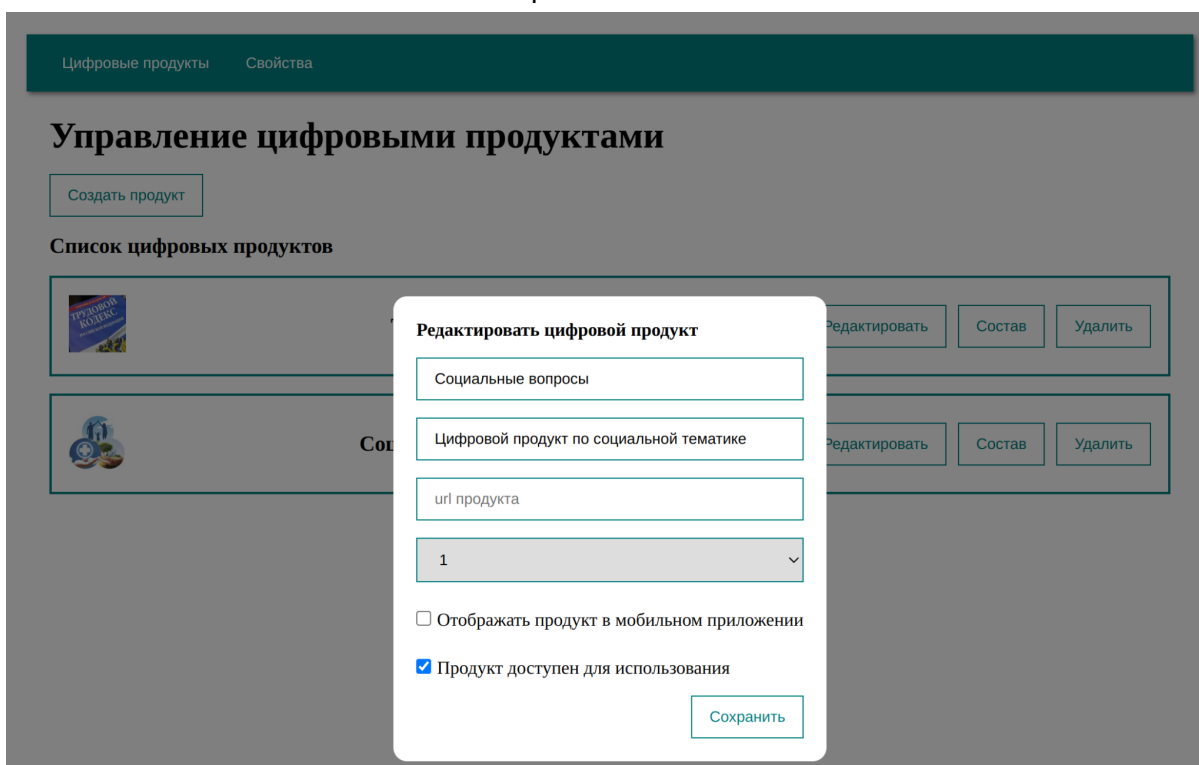
	Трудовое право	Редактировать	Состав	Удалить
Описание: Цифровой продукт по трудовому праву				
	Социальные вопросы	Редактировать	Состав	Удалить

- создать новый цифровой продукт. При нажатии кнопки “Создать продукт” открывается окно для создания нового продукта. при создании нового продукта можно указать следующие поля:
 - Название продукта
 - Описание продукта
 - url адрес продукта
 - Номер изображения (выбирается из доступных в редакторе)
 - Доступен ли продукт для использования
 - Отображается ли продукт в платформе

При нажатии кнопки “Создать” продукт сохраняется в базе данных. При нажатии вне поля окна, окно закрывается.



- Редактировать имеющийся цифровой продукт. При нажатии кнопки “Редактировать” в поле цифрового продукта открывается окно редактирования. Поля, доступные в окне, идентичны полям в окне создания цифрового продукта. При нажатии кнопки “Сохранить” все изменения сохраняются в базе. При нажатии вне поля окна окно закрывается.



- Управлять составом цифрового продукта. При нажатии кнопки “Состав” в поле цифрового продукта открывается страница управления составом ЦП. Описание интерфейса страницы приведено ниже.
- Удалить цифровой продукт. При нажатии кнопки “Удалить” продукт удаляется из базы.

При нажатии на кнопку “Свойства” в верхнем навигационном поле открывается страница редактирования базовых свойств цифровых продуктов.

Цифровые продукты

Свойства

Управление свойствами цифровых продуктов

Создать свойство

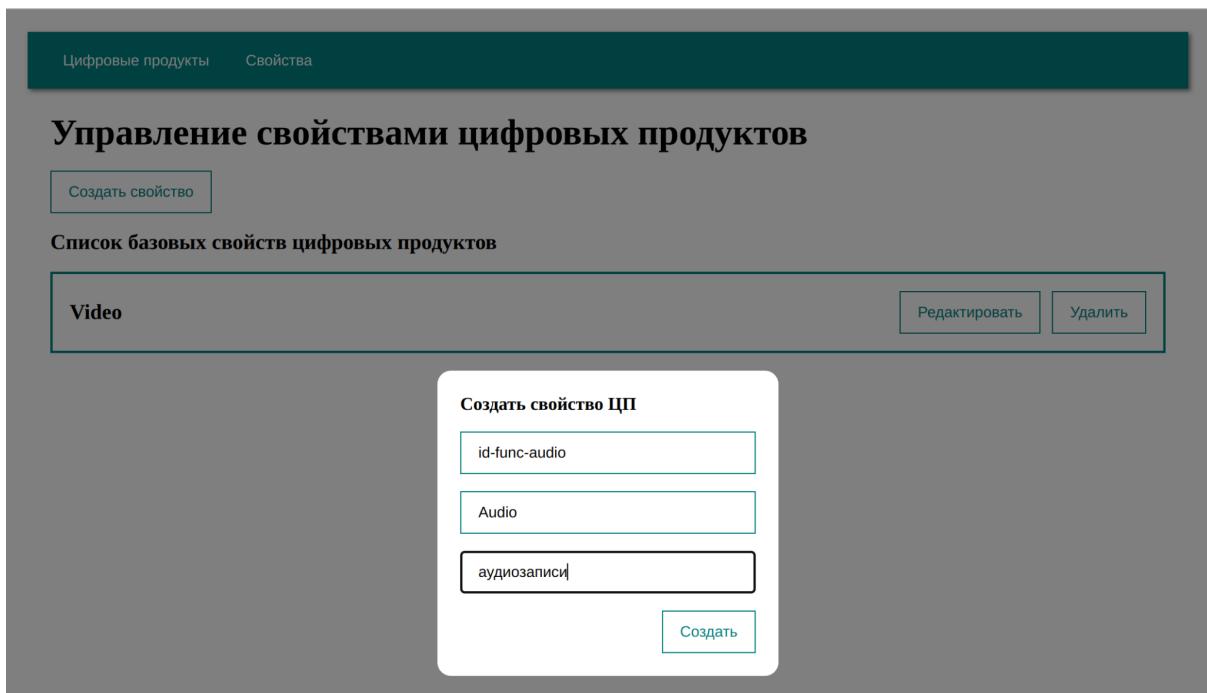
Список базовых свойств цифровых продуктов

Video	Редактировать	Удалить
-------	---------------	---------

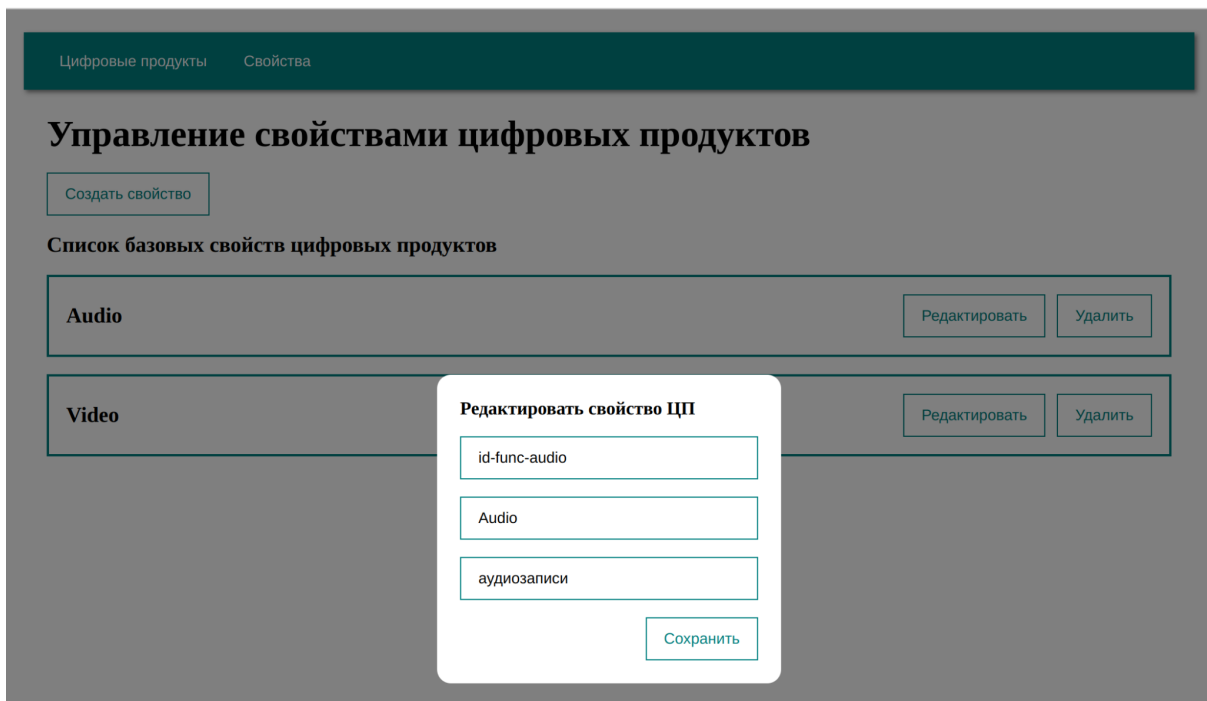
Интерфейс страницы позволяет:

- Просматривать список имеющихся базовых свойств цифровых продуктов. При нажатии на поле базового свойства появляется/скрывается его описание.
- Создать новое базовое свойство ЦП. При нажатии кнопки “Создать свойство” открывается окно создания нового базового свойства. Для базового свойства доступны поля:
 - ID свойства - строка, содержащая уникальный идентификатор базового свойства. Данное поле не может быть пустым
 - Название свойства
 - Описание свойства

При нажатии кнопки “Создать” свойство сохраняется в базе.



- Редактировать базовые свойства ЦП. При нажатии кнопки “Редактировать” в поле базового свойства открывается окно редактирования данного свойства. Поля, доступные в окне, идентичны полям в окне создания свойства. При нажатии кнопки “Сохранить” изменения сохраняются в базе. При нажатии вне окна, оно закрывается.



- Удалить базовое свойство. При нажатии кнопки “Удалить” в поле базового свойства оно удаляется из базы. При удалении базового свойства из базы также удаляются все ассоциированные с ним свойства цифровых продуктов.

Страница редактирования состава цифрового продукта предназначена для создания, редактирования и удаления свойств данного цифрового продукта. Все свойства цифрового продукта ассоциированы с тем или иным базовым свойством. Интерфейс страницы позволяет:

Цифровые продукты Свойства

Состав цифрового продукта: Трудовое право

Создать свойство

Список свойств цифрового продукта

Видео: трудовое право

Редактировать

Удалить

- Просматривать состав цифрового продукта. При нажатии на поле свойства продукта открывается описание данного свойства и id базового свойства, с которым данное свойство продукта ассоциировано
- Создать новое свойство в данном продукте. При нажатии кнопки “Создать свойство” открывается окно создания нового свойства. В окне доступны следующие поля:
 - a. id базового свойства, с которым ассоциировано данное свойство. Поле выбирается из списка имеющихся базовых свойств. Поле не может быть пустым
 - b. Название свойства в данном продукте. Данное название может отличаться от названия базового свойства
 - c. Описание свойства в данном продукте. Описание может отличаться от описания базового свойства

При нажатии кнопки “Создать” новое свойство сохраняется в базе. При нажатии вне окна, оно закрывается.

- Редактировать свойство. При нажатии кнопки “Редактировать” в поле свойства открывается окно редактирования свойства. В окне доступны поля идентичные полям в окне создания нового свойства. При нажатии кнопки “Сохранить” изменения сохраняются в базе. При нажатии вне окна, оно закрывается.
- Удалить свойство из состава цифрового продукта. При нажатии кнопки “Удалить” в поле свойства, оно удаляется из базы.

5.4. Принцип программного подключения к чату со стороны Право-бота

Комплексное программное тестирование подразумевает использование утилиты cURL, а также утилит или программ для работы с вебсокетами⁷.

Наберите в терминале:

```
curl -X GET -H "x-chat-name: lawyerBot" \
-H "x-client-token: livechat"
http://localhost/livechat/api/v3/connect
```

Должен вернуться ответ следующего вида:

```
client_id: "v3_livechat_884b7f39-71f4-4735-a30d-022b60833ecf"
host: "localhost"
password: "password"
port: 15675
```

⁷ <https://github.com/vi/websocat> или <https://github.com/websockets/wscat>

```
proto: "ws"
pub: "lbot-mob.pub.v3_livechat_884b7f39-71f4-4735-a30d-022b60833ecf"
salt: "884b7f39-71f4-4735-a30d-022b60833ecf"
show_city: false
show_fio: false
show_info_panel: false
show_phone: false
sub: "lbot-mob.sub.v3_livechat_884b7f39-71f4-4735-a30d-022b60833ecf"
token: "livechat"
user: "user"
v_host: "/"
```

Это означает, что открыто подключение через протокол `mqtt-over-websocket` с указанными параметрами безопасности:

```
host: '${proto}://${user}:${v_host}@${host}:${port}/ws'
с паролем '${password}' и id клиента '${client_id}'
```

в данном примере:

```
ws://user:@localhost:15675/ws
client_id: ${client_id}
```

Для чтения сообщений Платформа может подписаться на топик, полученный в поле `sub` (в примере: `lbot-mob.sub.v3\_livechat\_884b7f39-71f4-4735-a30d-022b60833ecf`).

Для отправки сообщений Платформа должно отправлять их в топик, указанный в поле `pub` (в примере: `lbot-mob.pub.v3\_livechat\_884b7f39-71f4-4735-a30d-022b60833ecf`).

Убедиться в этом можно через веб-интерфейс брокера очередей RabbitMQ. Зайдите по ссылке <http://localhost:15672>. Для авторизации введите: `user, password`.

На вкладке Queues появятся указанные выше топики (Рисунок 43).

RabbitMQ

3.7.28

Erlang 22.3.4.7

Overview

Connections

Channels

Exchanges

Queues

Admin

Queues

▼ All queues (17)

Pagination

Page

1

 of 1 - Filter:

☐ Regex

?

Overview		Messages				Message rates			+/-
Name	Features	State	Ready	Unacked	Total	Incoming	deliver / get	ack	
bot.queue.client	D	idle	0	0	0				
bot.queue.server	D	idle	0	0	0				
lbot.queue.man	D	idle	0	0	0				
lbot.queue.mobile	D	idle	0	0	0				
lbot.queue.web	D	idle	0	0	0				
lman.queue.man	D	idle	0	0	0				
lman.queue.mobile	D	idle	0	0	0				
lman.queue.web	D	idle	0	0	0				
mqtt-subscription-v3_92623_7129f6a6-7242-469b-bac8-fa6a0e0ad90eqos1	D AD	idle	0	0	0				
mqtt-subscription-v3_92623_791937c8-556f-4d3c-9995-1805813abfdbqos1	D AD	idle	0	0	0				
mqtt-subscription-v3_92623_93f811f6-2d48-4e56-912b-cf8ae977408eqos1	D AD	idle	0	0	0				
mqtt-subscription-v3_92623_a99a8aff-4357-45f6-96ce-14d74d25cab0qos1	D AD	idle	0	0	0				
mqtt-subscription-v3_92623_b2f21830-0d13-4d27-be31-b8db39e35301qos1	D AD	idle	0	0	0				
mqtt-subscription-v3_92623_fd8f5aea-b369-4779-9de6-6e5a4e217137qos1	D AD	idle	0	0	0				
sup.queue.man	D	idle	0	0	0				
sup.queue.mobile	D	idle	0	0	0				
sup.queue.web	D	idle	0	0	0				

► Add a new queue

HTTP API

Server Docs

Tutorials

Community Support

Community Slack

Commercial Support

Plugins

GitHub

Changelog

Рисунок 43

Для чтения истории сообщений, наберите в терминале: `curl -X GET -H "x-chat-name: lawyerBot" -H "token: livechat" http://localhost/livechat/api/getAllMessage`

5.5. Переписка с ботом

Для осуществления тестирование переписки с ботом:

1) Перейти по ссылке `http://localhost/livechat/?chat=lawyer-bot-web&token=92623`, справа внизу – кнопка чата. Чтобы бот получил сообщение, необходимо нажать эту кнопку, в открывшемся окне написать сообщение и отправить его.

2) Открыть платформа в эмуляторе и написать сообщения из чата платформы.

5.6. Подключение от имени бота

Для того чтобы осуществить тестирование переписки от имени бота :

1) Перейти по ссылке
[`http://localhost/livechat/?chat=lawyer-bot-man&token=92623`](http://localhost/livechat/?chat=lawyer-bot-man&token=92623), справа внизу страницы –
кнопка чата от имени бота. Чтобы отправить сообщение от имени бота, следует нажать
эту кнопку, в открывшемся окне написать сообщение и отправить его.

6. Эксплуатация экземпляра ПО, предоставленного для проведения экспертной проверки

Порядок использования экземпляра программного обеспечения, предоставленного для проведения экспертной проверки, аналогичен заявленному в документе “Руководство пользователя Платформы "LEGALHELP"”⁸.

Платформа должно быть запущено в эмуляторе Android Studio (см. “6.1 Запуск платформы в эмуляторе Android Studio на Ubuntu для запуска стенда и экспертной проверки”), а сервисы Платформы должны быть запущены в docker-compose (см. “6.3 Запуск стенда для экспертной проверки”).

6.1 Запуск платформы в эмуляторе Android Studio на Ubuntu для запуска стенда и экспертной проверки

- 1) Запустить android-studio (Рисунок 44).

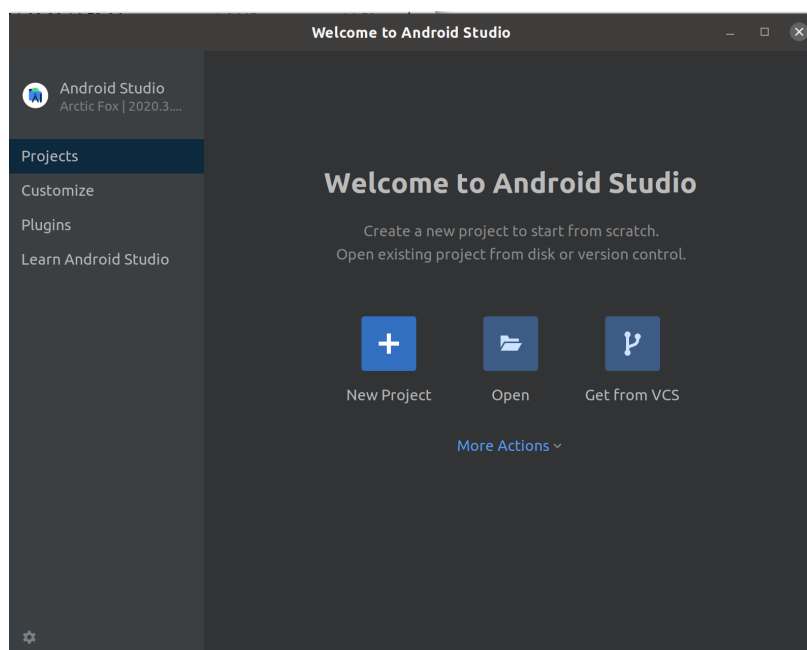


Рисунок 44

⁸ https://www.applegalhelp.ru/docs_download/3

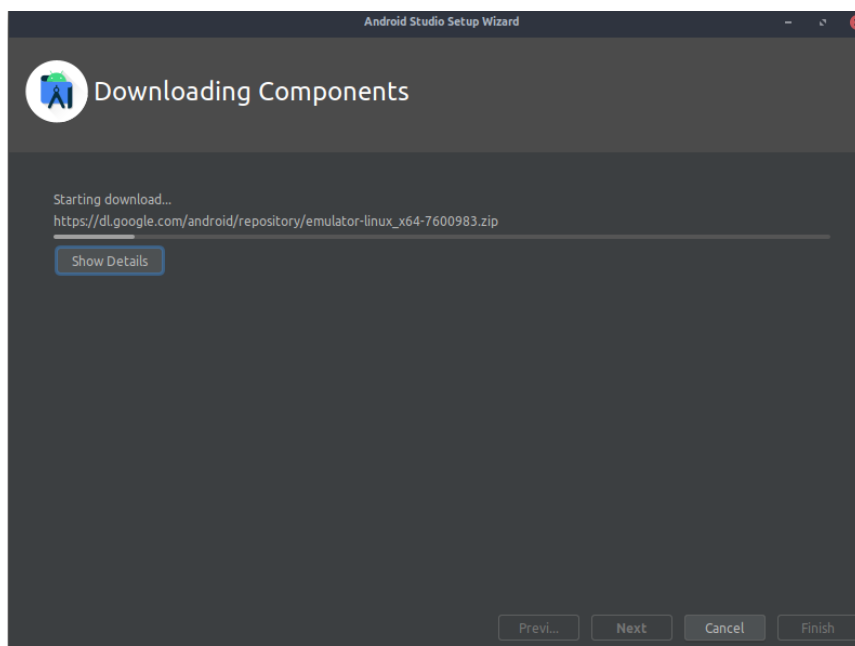


Рисунок 45

- 2) Далее открыть **AVD Manager** -> Запустить эмулятор (клик по зеленому треугольнику "play").

6.2 Сборка новых версий ПО Платформы

Чтобы собрать новую версию платформы в виде APK-файла нужно:

- 1) Запустить Android Studio и открыть вкладку **build variants**;
- 2) Если в списке модулей пусто, следует ожидать 2-3 минуты, по прошествии которых перейти в меню **Build > Rebuild project** и подождать окончания сборки модулей;
- 3) После того как в **build variants** появляется список модулей, следует открыть терминал и перейти в папку с исходным кодом Платформы:

```
cd ~/soft/src/legalhelp-mobile/android
```


выполнить `./gradlew assembledevRelease`
- 4) В результате успешного выполнения соберется требуемый APK файл.
- 5) Для установки на эмулятор перейти в папку `~/soft/src/legalhelp-mobile/android/app/build/outputs/apk/dev/release`
- 6) Необходимо перенести **app-dev-release.apk** файл в открывшийся эмулятор. Для этого его следует просто перетащить и отпустить в окне (drag and drop функционал)

6.3 Запуск стенда для экспертной проверки

- 1) `cd ~/soft/backend` – перейти в папку с бэкенд составляющей Платформы.

2) Выполнить `./run-local.sh` (предпочтительно), либо воспользоваться командой `docker-compose up --build` —, чтобы запустить докер (необходимо учитывать продолжительное время установка всех образов). Во время запуска могут быть различные ошибки, если на машине закрыты нужные порты. Чтобы не разбираться, какие порты требуется открыть - добавлен файл для запуска `run-local.sh`, который выключит конкурирующие сервисы и откроет все необходимые порты.

3) В случае, если сборка сигнализирует ошибками `https://<some_url>... does not contain Release ...` - IP заблокирован, то необходимо локально подключиться к VPN.

4) Если локально установлен `nginx/mysql/redis/rabbit`, то необходимо отключить все сервисы, выполнив команду `sudo service <name> stop`, так как стандартные порты будут заняты (см. 1.5 Требования к окружению для запуска стенда и экспертной проверки - в секции “Диагностика закрытых портов на ОС Ubuntu (Linux)”). Для решения этой задачи написан файл запуска `run-local.sh`, но решение всех возможных проблем запуска им не гарантируется.

5) После успешного запуска докера в консоли будут разноцветные надписи логов сервисов (`redis/livechat/backend` etc) и самая важная надпись `mobile-api_1 | {"level":"info","message":"Thu Sep 09 2021 13:58:05 GMT+0000 (Coordinated Universal Time) [index]: Listening on port 3050"}`

6) В случае экспертной проверки следует ввести в эмуляторе следующие данные для авторизации:

логин: 79999999999

пароль: 1234

6.4 Инструменты, моделирующие поведение внешних систем

Утилита для тестирования чат-бота со стороны Пользователя

Для старта беседы с ботом следует перейти по ссылке: <http://localhost/livechat/?chat=lawyer-bot-web&token=92623>

Утилита для тестирования работы чатов с Пользователя со стороны Юриста

Для подключения к той же беседе от имени специалиста: <http://localhost/livechat/?chat=lawyer-bot-man&token=92623>

Утилита для тестирования работы чата юриста со стороны Пользователя

Для подключения к чату с юристом со стороны Пользователя перейдите по ссылке: <http://localhost/livechat/?chat=lawyer-man-web&token=92623>

Утилита для тестирования работы чата юриста со стороны Юриста

Для подключения к чату с юристом со стороны Юриста перейдите по ссылке: <http://localhost/livechat/?chat=lawyer-man-man&token=92623>

Утилита для тестирования работы чата с техподдержкой со стороны Пользователя

Для подключения к чату с техподдержкой от имени Пользователя перейдите по ссылке: <http://localhost/livechat/?chat=support-web&token=92623>

Утилита для тестирования работы чата с техподдержкой со стороны сотрудника Технической поддержки

Для подключения к чату от имени специалиста технической поддержки перейдите по ссылке:

<http://localhost/livechat/?chat=support-man&token=92623>

Все чаты используются со стороны Платформы. Со стороны Юриста и Специалиста Технической поддержки - используются специальные утилиты в составе внешних информационных систем.

6.5 Инструменты, необходимые для администрирование подсистем

Администрирование ПО осуществляется через внешние информационные системы. Для обеспечения интеграции с ВИС разработаны REST-API у соответствующих подсистем.

Для удобства эксплуатации - REST-API серверы оснащены Swagger-интерактивной документацией, которая позволяет выполнять запросы к REST-API через веб-интерфейс. При этом возможно выполнять запросы через утилиту cURL.

6.5.1. Администрирование серверной части Модуля ЦП

Digitalproduct

GET	/digitalproduct	Get All	✓
POST	/digitalproduct	Create One	✓
GET	/digitalproduct/{item_id}	Get One	✓
PUT	/digitalproduct/{item_id}	Update One	✓
DELETE	/digitalproduct/{item_id}	Delete One	✓

Featurebase

GET	/featurebase	Get All	✓
POST	/featurebase	Create One	✓
GET	/featurebase/{item_id}	Get One	✓
PUT	/featurebase/{item_id}	Update One	✓
DELETE	/featurebase/{item_id}	Delete One	✓

Featuredigitalproduct

GET	/featuredigitalproduct/{digital_product_id}	Read Digital Product Features	✓
POST	/featuredigitalproduct/{digital_product_id}	Create Digital Product Feature	✓
GET	/featuredigitalproduct/{digital_product_id}/{feature_digital_product_id}	Read Digital Product Feature	✓
PUT	/featuredigitalproduct/{digital_product_id}/{feature_digital_product_id}	Update Digital Product Feature	✓
DELETE	/featuredigitalproduct/{digital_product_id}/{feature_digital_product_id}	Delete Digital Product Feature	✓

Appsettings

GET	/appsettings	Read Current Settings	✓
PUT	/appsettings	Update App Settings	✓

Appupdateparams

GET	/api/v1/notifications/updateParams	Read App Updates	✓
-----	------------------------------------	------------------	---

6.5.2. Администрирование состава цифровых юридических продуктов и их свойств

Фронтенд доступен для использования по адресу <http://localhost:8080>

Главная страница предназначена для управления непосредственно набором цифровых продуктов

Управление цифровыми продуктами

[Создать продукт](#)

Список цифровых продуктов

	Трудовое право	Редактировать	Состав	Удалить
	Социальные вопросы	Редактировать	Состав	Удалить

Цифровые продукты и их свойства хранятся в базе данных редактора и передаются в платформу по запросу через REST API, которые представлены как API ЦП для платформы и API для подсистемы Администрирования:

1. Получить список всех цифровых продуктов, имеющихся в базе данных;
2. Создать новый цифровой продукт в базе данных Редактора;
3. Получить конкретный цифровой продукт по его идентификатору;
4. Изменить конкретный цифровой продукт по его идентификатору;
5. Удалить конкретный цифровой продукт по его идентификатору;
6. Получить список всех базовых свойств цифровых продуктов;
7. Создать новое базовое свойство цифрового продукта;
8. Получить конкретное базовое свойство ЦП по его идентификатору;
9. Изменить конкретное базовое свойство ЦП по его идентификатору;
10. Удалить конкретное базовое свойство ЦП по его идентификатору;
11. Получить список всех свойств данного цифрового продукта;
12. Создать новое свойство у цифрового продукта;
13. Получить конкретное свойство ЦП по идентификатору продукта и идентификатору свойства;
14. Изменить конкретное свойство цифрового продукта по идентификатору продукта и идентификатору свойства;
15. Удалить конкретное базовое свойство ЦП по его идентификатору;
16. Получить список настроек для платформы (для редактора цифровых продуктов);
17. Изменить настройки для платформы;
18. Получить список настроек для платформы (для платформы).

6.5.3. Администрирование Мобильного бекенда

Ключевой точкой интеграции Платформы с серверной частью является Мобильный бекенд, который выступает фасадом, обеспечивающим управление подключением мобильных клиентов снаружи. Внутри платформы Мобильный бекенд обеспечивает соединение с внутренними сервисами и подсистемами.

Веб-интерфейс интерактивной swagger-документации REST-API (Рисунок 46) расположен по адресу: <http://localhost/docs/>

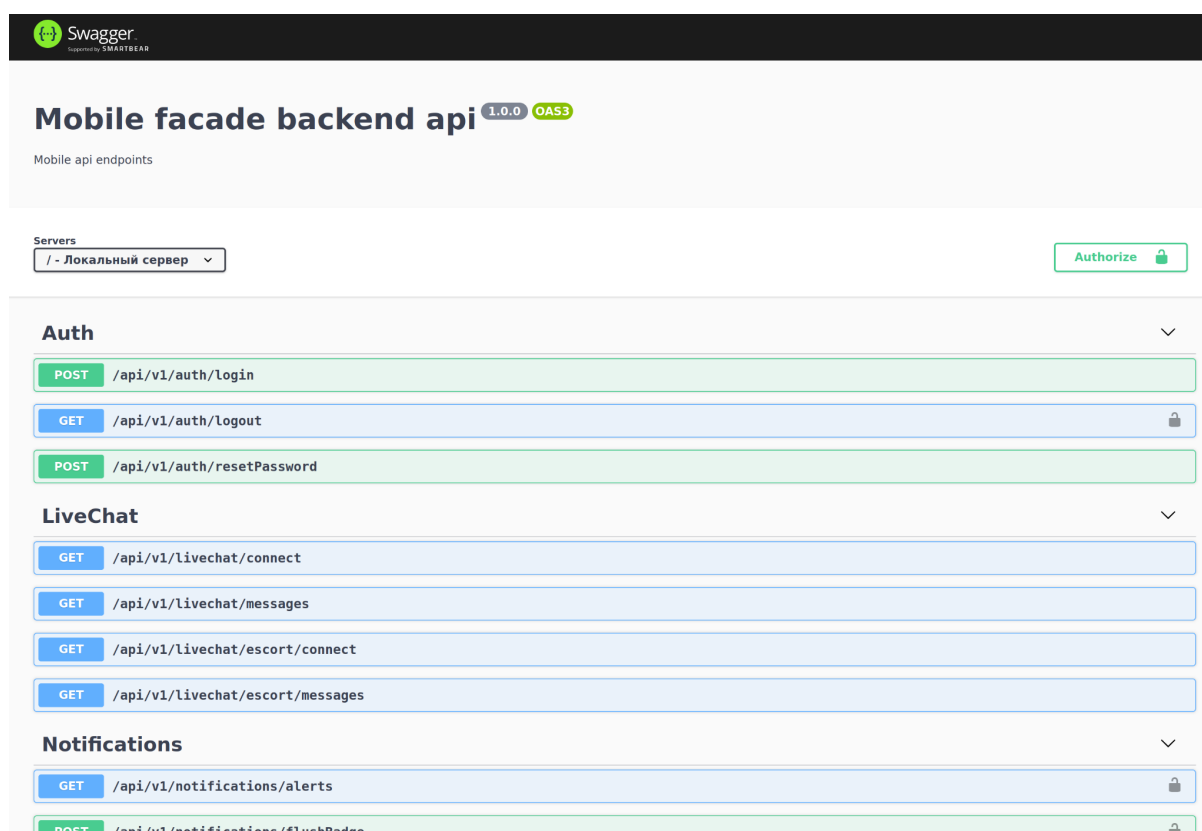


Рисунок 46. Веб-интерфейс REST-API Мобильного бекенда

6.5.2 Администрирование подсистемы авторизации

Подсистема авторизации выполняет операции CRUD над пользователями, их ролями и привилегиями. Пользователи могут быть добавлены в группы, для которых устанавливаются групповые ролевые политики и привилегии.

Веб-интерфейс подсистемы авторизации расположен по адресу (Рисунок 47): <http://localhost/passport/swagger/>

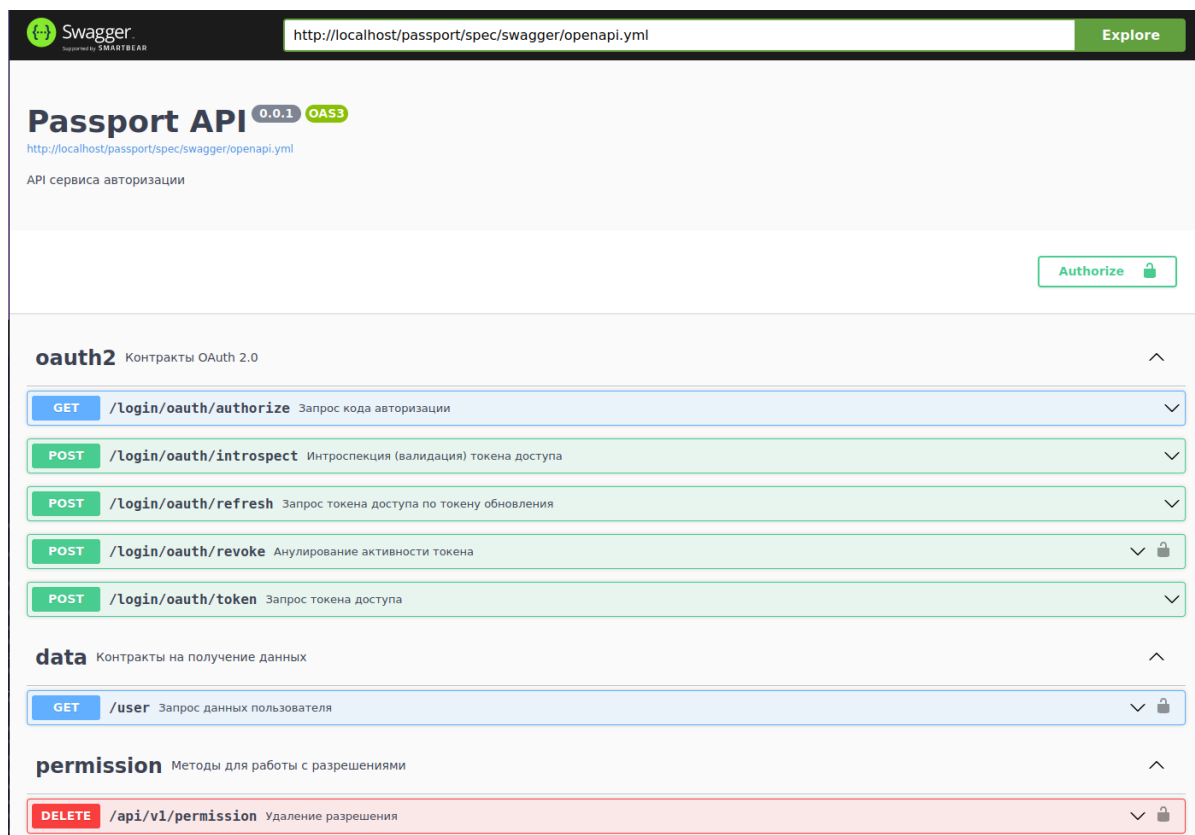


Рисунок 47. Веб-интерфейс вызовов REST-API подсистемы авторизации

Подробное описание параметров каждого запроса указано в веб-интерфейсе в swagger-документации. Для ознакомления с подробным описанием, необходимо развернуть строчку с методом REST-API (Рисунок 48).

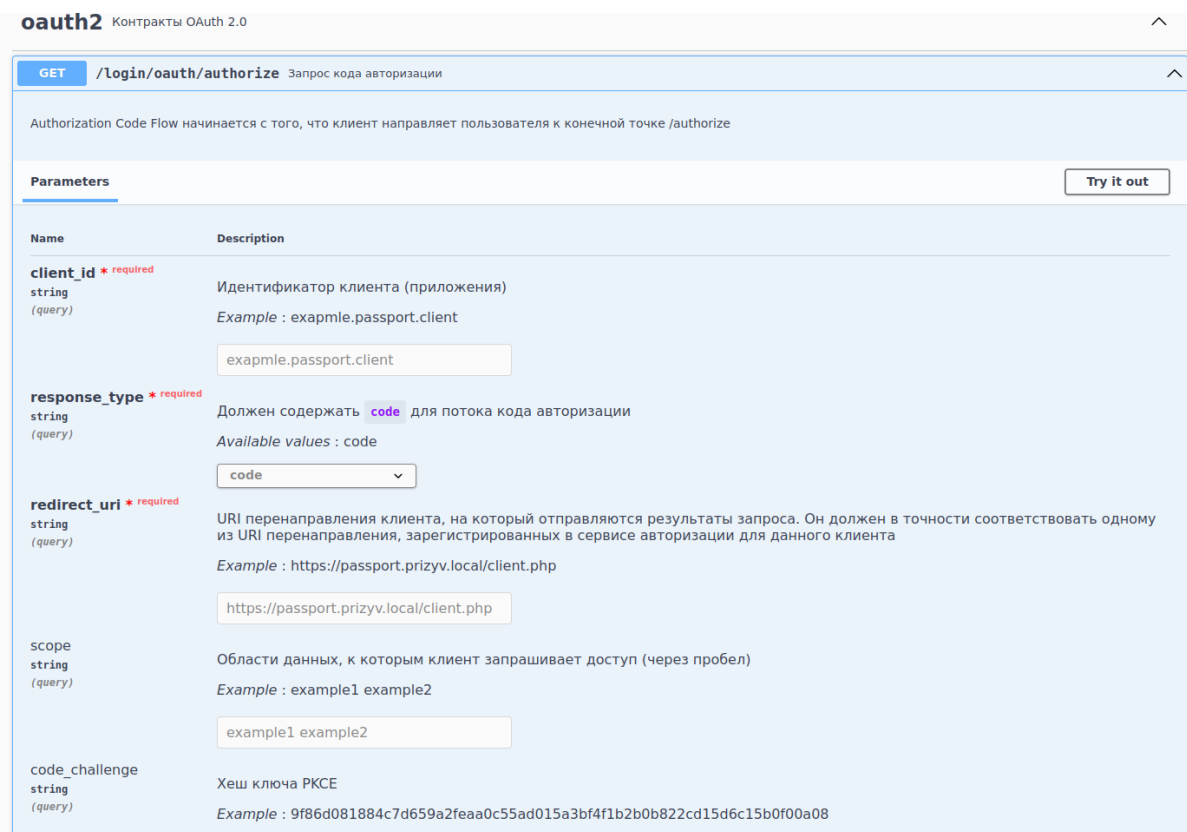


Рисунок 48. Подробное описание метода авторизации

6.5.3 Администрирование базы знаний

Веб-интерфейс для администрирования базы знаний представлен по ссылке (Рисунок 49): <http://localhost:8002/docs>

Основными операциями с базой знаний являются: проверка состояния сервиса, создание, обновление, удаление и чтение документа, отвечающего на один конкретный вопрос. Предусмотрен метод релевантного поиска по коллекции документов.

FAQ Service API 0.1.0 OAS3

/openapi.json

FAQ Service API We've had to endure much, you and I, but soon there will be order again — a new age. Aquinas spoke of the mythical City on the Hill. Soon that city will be a reality, and we will be crowned its kings, or better than kings: Gods!

default

GET	/health_check	Health Check	⌵
PUT	/faq/add_update	Faq Add	⌵
DELETE	/faq/delete	Faq Delete	⌵
POST	/faq/get_item	Faq Get Item	⌵
POST	/faq/search	Faq Search	⌵

Schemas

HTTPValidationError >

Item >

Item_idx >

RequestSearchQuery >

Рисунок 49

6.5.4 Администрирование брокера очередей RabbitMQ подсистемы чатов

Веб-интерфейс брокера очередей RabbitMQ доступен по ссылке (Рисунок 50):
<http://localhost:15672>

Для авторизации необходимо ввести данные: **user**, **password**

Администрирование подразумевает наблюдение за нагрузкой, создание, чтение, удаление очередей, каналов, соединений, управление доступом.

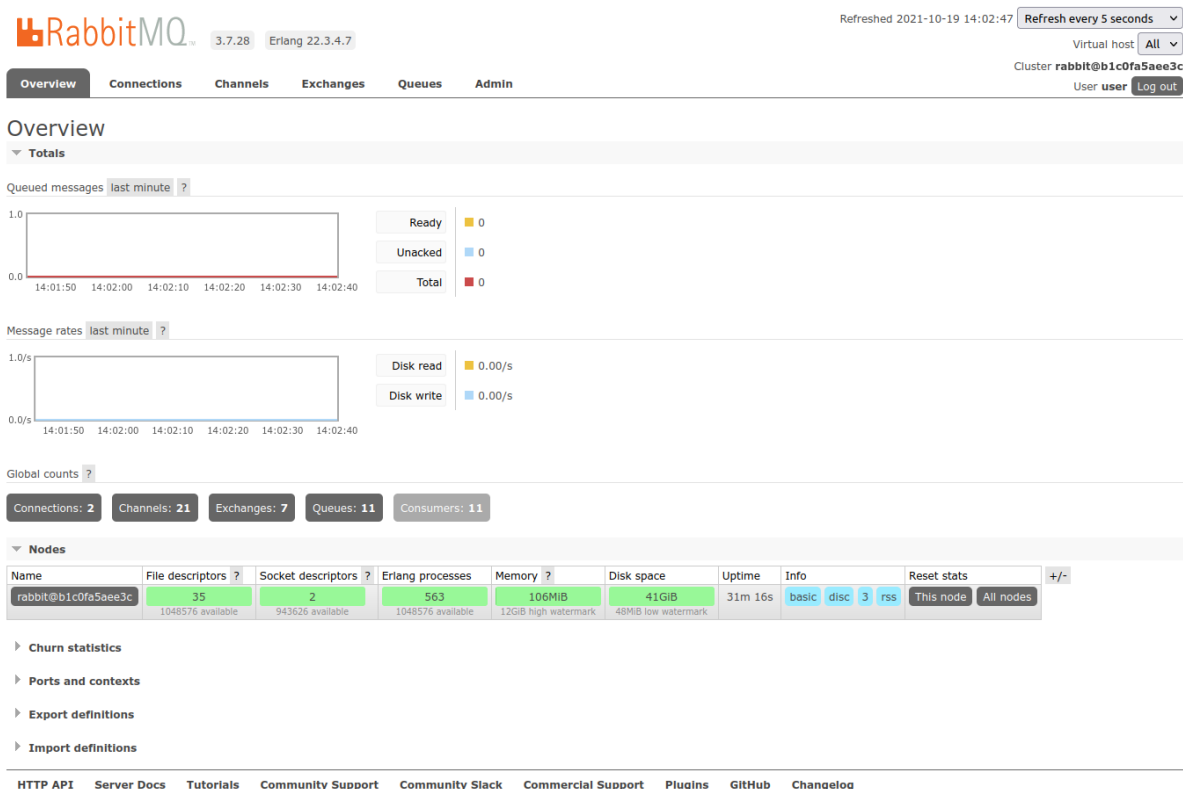


Рисунок 50

6.5.5 Администрирование с помощью средств мониторинга нагрузки

Средства мониторинга нагрузки обеспечивают мониторинг за состоянием Платформы. В отчетах отражены потребление ЦПУ, оперативной памяти, количество запросов в секунду, потребление диска.

Логи сбора метрик доступны по адресу: <http://localhost:9100/metrics>

Инструмент мониторинга Cadvisor доступен по ссылке (Рисунок 51): <http://localhost:8090/containers/>

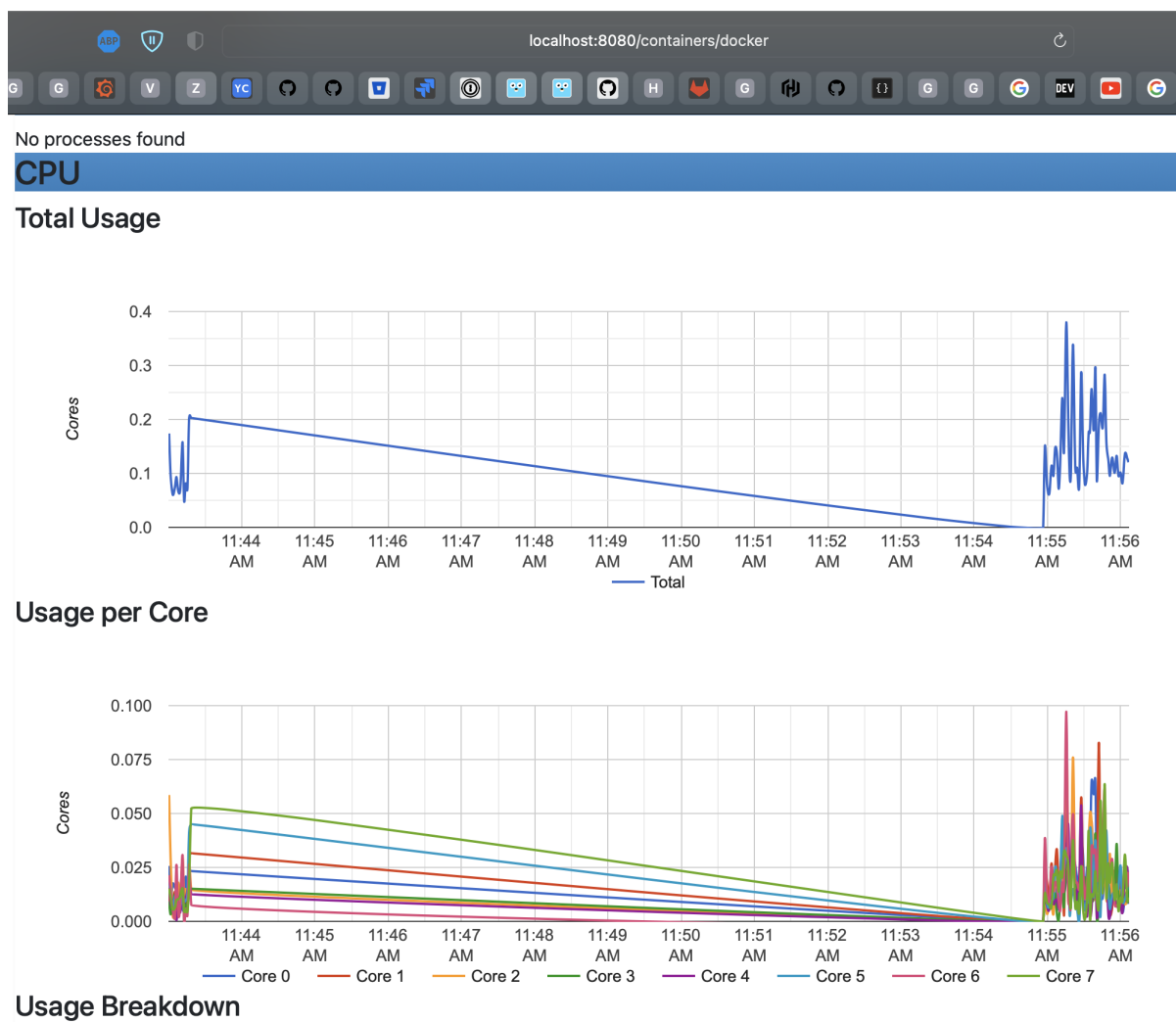


Рисунок 51

Средство мониторинга Grafana доступно по адресу (Рисунок 52):
<http://localhost:3000>

Для авторизации необходимо ввести данные: **admin, admin**

Необходимо выбрать отчет на странице Dashboards
<http://localhost:3000/dashboards>



Рисунок 52

7. Математическое обеспечение Модуля ЦП в рамках платформы

Условно чат-бот «Право-бот» можно разделить на компоненты обеспечивающие общение в чате на стороне Платформы и платформенные компоненты, обеспечивающие интеллектуальную основу – ядро классификатора интенгов Пользователей, полученных в результате анализа текстов сообщений в чате (реплик Пользователя в чате) с целью удовлетворения выявленной потребности, интенга, и автоматического формирования ответов (реплик Право-бота в чате).

ПО Платформы основано на взаимодействия компонентов, которые осуществляют управление диалогом (Dialogue Manager, DM) на основе гибридного подхода, который заключается в использовании:

- бизнес-правил диалога, которые определяют правила (политики) настройки чата (например, время ожидания отклика системы);

- «учителя» – обучающей выборки бизнес-сценариев диалога (текстовое сообщение-вопрос – интенг-потребность) на основе экспертных знаний бизнес-скриптов по оказанию юридических консультаций (БЗ);

- алгоритмов машинного дообучения с целью классификации на основе анализа и соотнесения текстовых сообщений-вопросов, полученных из чата (реплик Пользователя в чате) к определенным интенгам, а также формирование новых знаний бизнес-сценариев диалога (текстовое сообщение-вопрос – интенг-потребность) и дополнение БЗ, соответственно.

Реализуется задача классификации при обучении по прецедентам.

Ядро интеллектуального чат-бота в точности реализует нейронную сеть прямого распространения.

Количество внутренних слоёв – два по шестьдесят четыре нейрона.

В результате обучения максимизируется метрика качества ROC-AUC⁹, функцией потерь выступает категориальная кросс-энтропия¹⁰.

В качестве baseline классификатора были использованы логистическая регрессия и случайный лес, а в качестве основного – SVM с различными ядерными функциями¹¹. сравнение моделей проводилось по метрикам ROC AUC, F1 weighted, balanced accuracy. В таблице 5 приведены результаты сравнения моделей на тестовом наборе данных.

Таблица 5 – Результаты сравнения моделей на тестовом наборе данных

Метрика	Naive classifier	Logistic regression	Random forest	SVC, linear	SVC, polynomial (deg = 3)	SVC, rbf	SVC, sigmoidal
ROC AUC	0.50	0.98	0.96	0.98	0.99	0.99	0.99

⁹ https://en.wikipedia.org/wiki/Receiver_operating_characteristic

¹⁰ https://en.wikipedia.org/wiki/Cross_entropy

¹¹ Cortes C., Vapnik V. Support-vector networks //Machine learning. – 1995. – Т. 20. – №. 3. – С. 273-297.

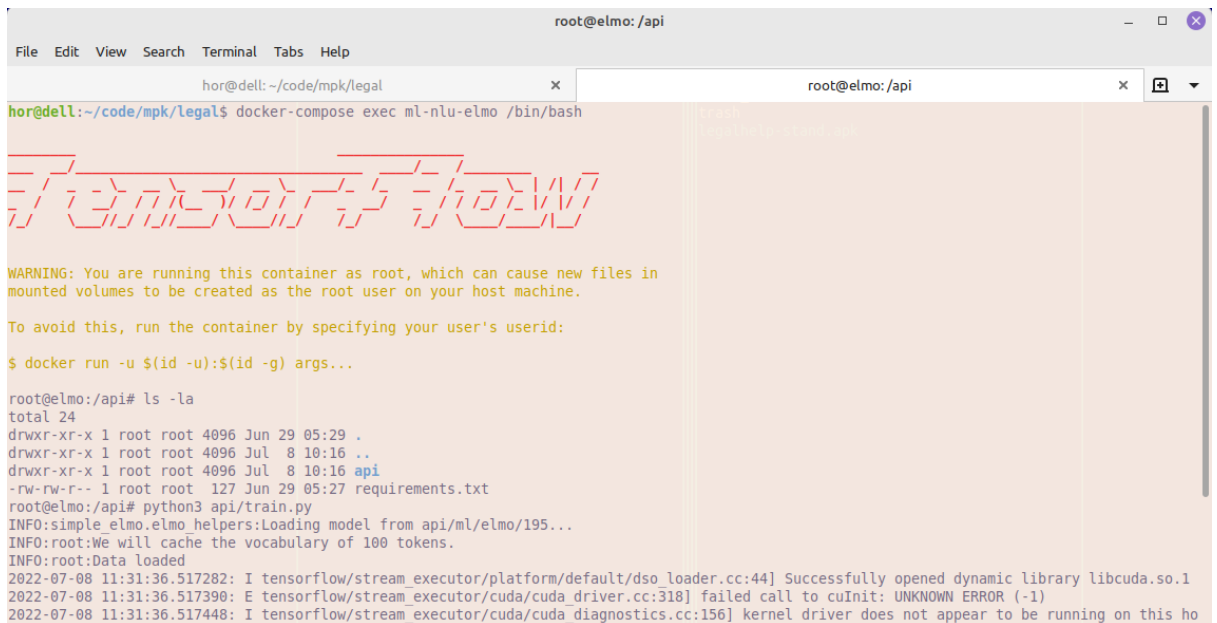
F1, weighted	0.30	0.92	0.88	0.92	0.94	0.95	0.93
balanced accuracy	0.17	0.88	0.80	0.88	0.90	0.92	0.88

Как видно из сравнительной таблицы, наилучшие показатели BERT показывает более высокие показатели по метрикам по сравнению с ELMO.

Теоретический и вычислительный анализ моделей позволяет сделать выбор моделей ELMO, BERT для использования в ансамблях моделей для классификации намерений пользователя из текстовых сообщений в чатах с Право-ботом.

7.1. Тренировка нейронных сетей интеллектуального чат-бота для разделов Модуля ЦП

```
docker-compose exec ml-nlu-elmo /bin/bash
```



```
root@elmo: /api
File Edit View Search Terminal Tabs Help
hor@dell: ~/code/mpk/legal x root@elmo: /api x
hor@dell:~/code/mpk/legal$ docker-compose exec ml-nlu-elmo /bin/bash

WARNING: You are running this container as root, which can cause new files in
mounted volumes to be created as the root user on your host machine.

To avoid this, run the container by specifying your user's userid:

$ docker run -u $(id -u):$(id -g) args...

root@elmo:/api# ls -la
total 24
drwxr-xr-x 1 root root 4096 Jun 29 05:29 .
drwxr-xr-x 1 root root 4096 Jul 8 10:16 ..
drwxr-xr-x 1 root root 4096 Jul 8 10:16 api
-rw-rw-r-- 1 root root 127 Jun 29 05:27 requirements.txt
root@elmo:/api# python3 api/train.py
INFO:simple_elmo.elmo helpers:Loading model from api/ml/elmo/195...
INFO:root:We will cache the vocabulary of 100 tokens.
INFO:root:Data loaded
2022-07-08 11:31:36.517282: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcuda.so.1
2022-07-08 11:31:36.517390: E tensorflow/stream_executor/cuda/cuda_driver.cc:318] failed call to cuInit: UNKNOWN ERROR (-1)
2022-07-08 11:31:36.517448: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this ho
```

Установка Ubuntu

Прежде чем приступить к установке новой версии операционной системы, необходимо скачать образ и записать его на USB-флеш-носитель.

Подготовка к установке

Необходимо выполнить следующие действия:

- 1) Загрузить образ Ubuntu 20.04 с официального сайта.

Дистрибутив можно скачать с официального [веб-сайта](http://releases.ubuntu.com/20.04/)¹² Ubuntu (Рисунок 4). На странице релиза опубликованы две ссылки: версия с графическим интерфейсом – Desktop и серверная – Server.

¹² <http://releases.ubuntu.com/20.04/>

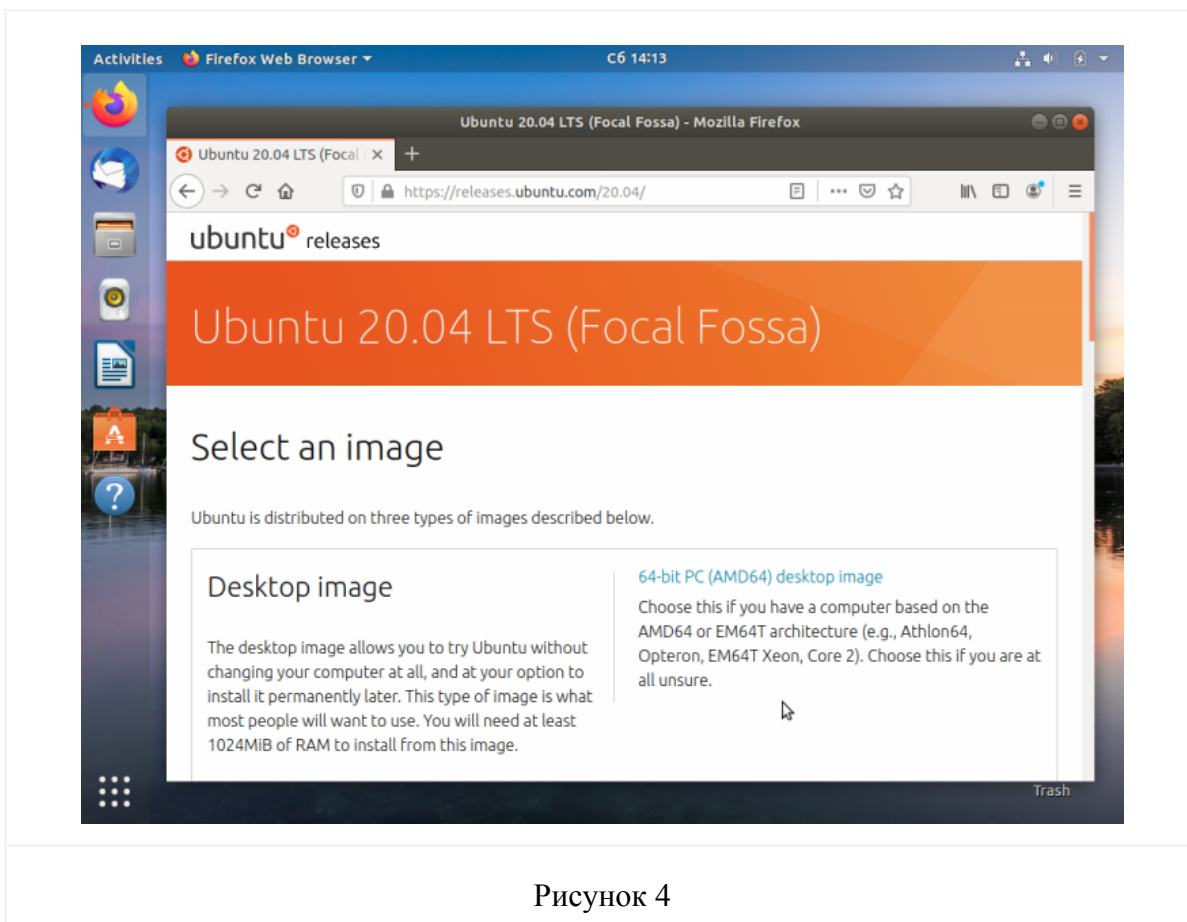


Рисунок 4

Нужна настольная версия (ссылка **Desktop image**). Файл, содержащий дистрибутив, называется **ubuntu-20.04-desktop-amd64.iso**. Скачайте и сохраните его в папке на локальном диске. Обратите внимание на то, что речь идёт о 64-разрядной версии. Прежде, чем приступить к установке, убедитесь, что ваш компьютер способен в принципе работать с 64-разрядными системами.

2) Осуществить запись образа.

Перед началом установки содержимое скачанного файла необходимо записать на USB-флешку или DVD-диск.

Для того, чтобы записать ISO-образ на USB-флеш носитель, рекомендуем использовать утилиту **Etcher**. Скачивается с сайта [balena.io](https://www.balena.io/etcher/)¹³, есть установщик для Windows, Linux и Mac, а также портативная версия, не требующая установки.

Запись в программе **Etcher** выполняется таким образом: выберите нужный файл, убедитесь, что установлена подходящая по размеру флешка, нажмите кнопку **Flash** (Рисунок 5).

¹³ <https://www.balena.io/etcher/>

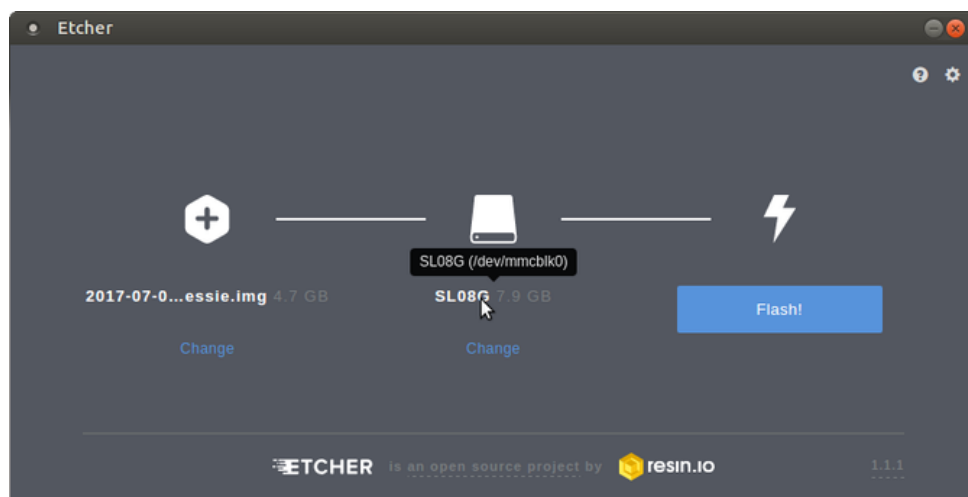


Рисунок 5

Кроме этой программы рекомендуем также утилиту для Windows – Rufus, загрузить её можно с [сайта Rufus](https://rufus.ie/)¹⁴ (Рисунок 6).

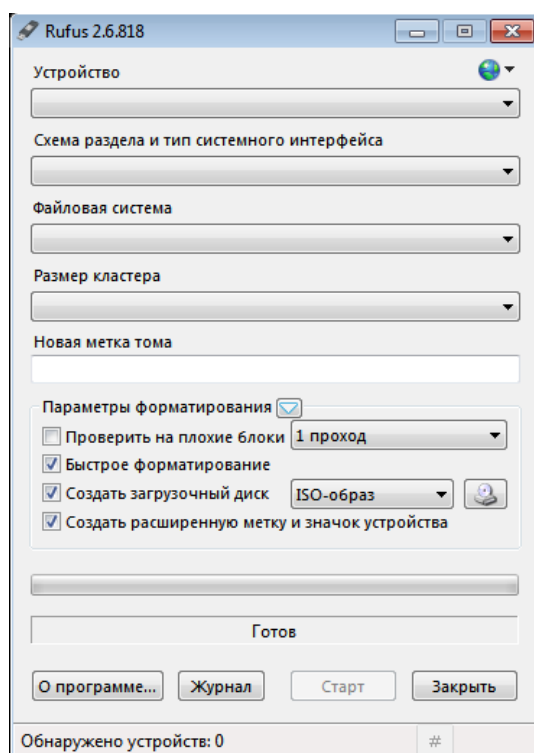


Рисунок 6

После того, как образ с дистрибутивом Ubuntu записан на носитель, можно приступать к установке.

¹⁴ <https://rufus.ie/>

Установка Ubuntu 20.04

Необходимо вставить флешку в USB-порт или поместить DVD-диск в привод. После чего следует перезагрузить компьютер. В момент запуска следует выбрать загрузку с USB или DVD-носителя (зависит от конкретного компьютера). Далее начнется **установка Ubuntu 20.04**.

Следует выполнить следующие действия:

- 1) Подождать запуска установщика Ubuntu с USB или DVD-носителя.

После начала загрузки с носителя появится окно (Рисунок 7).

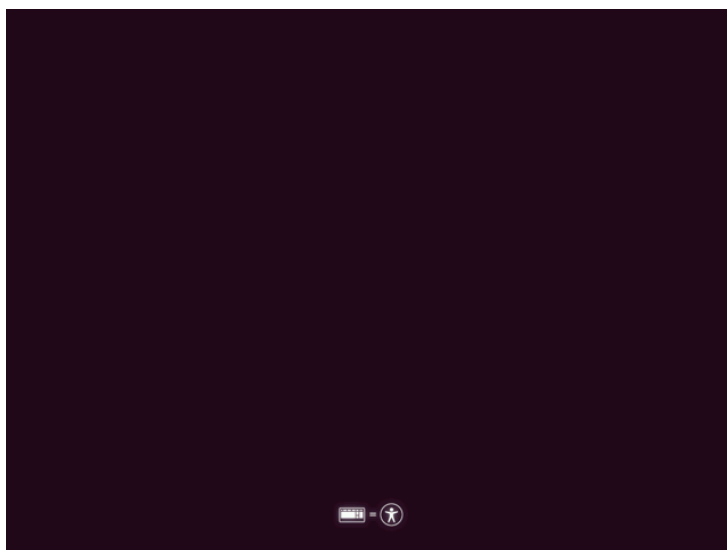


Рисунок 7

Это означает, что загрузка установщика началась. Далее установщик проверяет файлы дистрибутива, в окне появляются их названия (Рисунок 8).

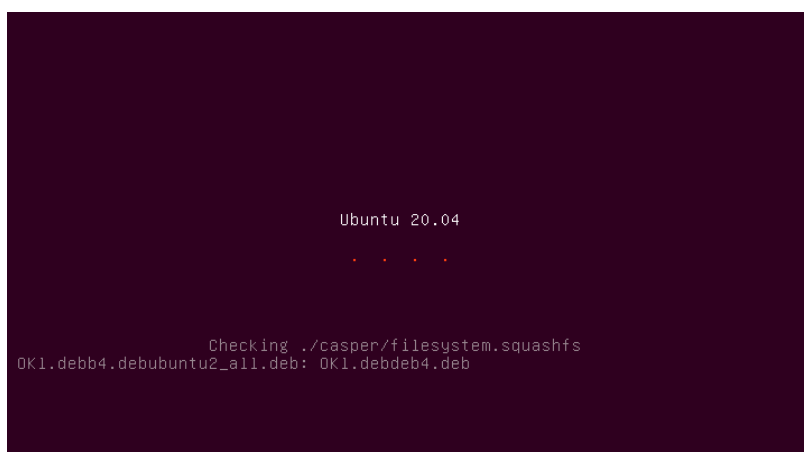


Рисунок 8

По окончании проверки запускается графическая оболочка (Рисунок 9).

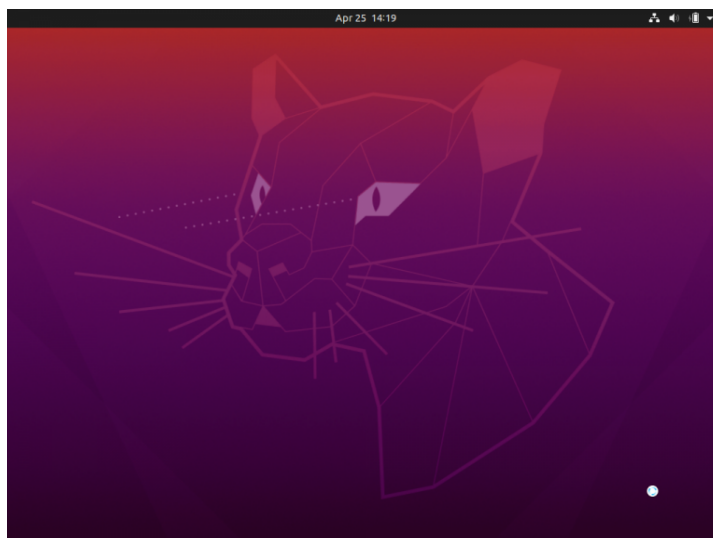


Рисунок 9

2) Загрузить LIVE-Систему.

С помощью меню в левой части окна можно выбрать язык установщика. Программа предлагает запустить систему в Live-режиме или приступить к инсталляции. Рекомендуется выбрать режим **Попробовать Ubuntu**.

В этом случае операционная система стартует полностью с носителя. Если при дальнейшей установке возникнут ошибки, их можно будет исправить.

3) Запустить на рабочем столе платформа с иконкой “Установить Ubuntu 20.04”.

Для того, чтобы запустить Установщик, после загрузки системы следует дважды кликнуть левой кнопкой мыши по значку **Установить Ubuntu 20.04 LTS** (Рисунок 10).

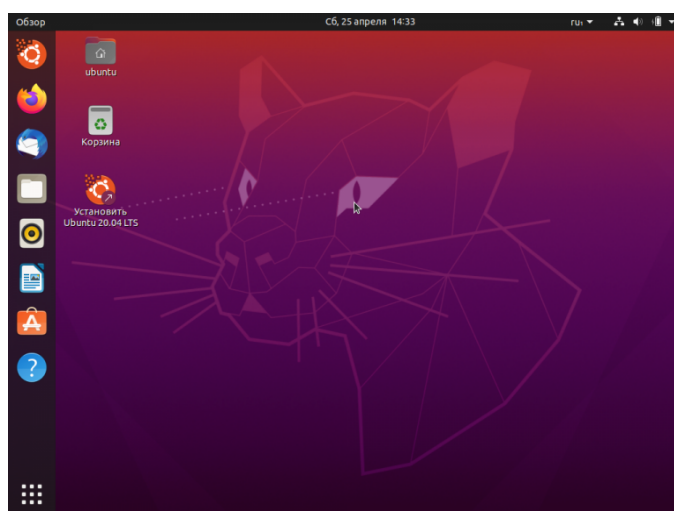


Рисунок 10

4) Выбрать язык.

Следует выбрать язык интерфейса будущей системы и нажать кнопку **Продолжить** (Рисунок 11, 12).

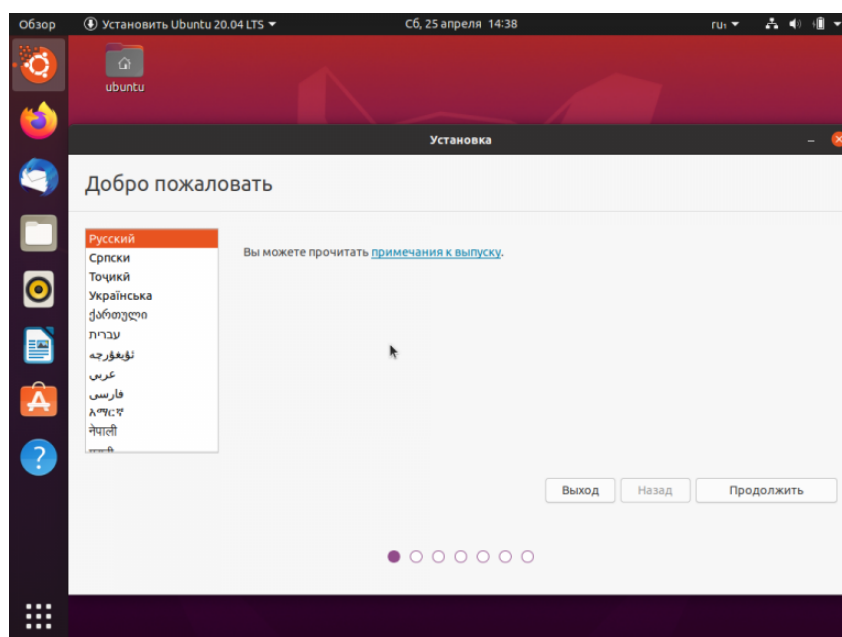


Рисунок 11

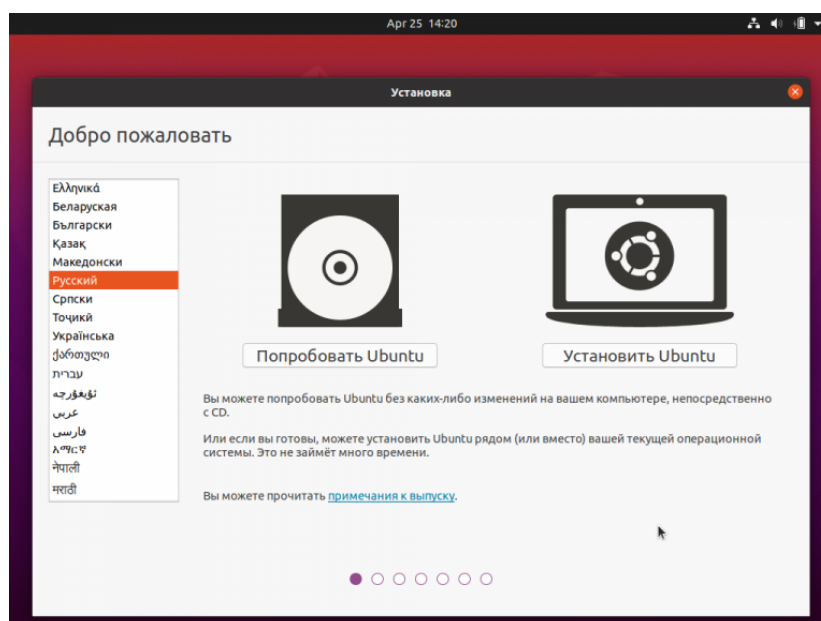


Рисунок 12

5) Выбор раскладки клавиатуры.

Чтобы выбрать раскладку клавиатуры по умолчанию, следует нажать кнопку **Продолжить** (Рисунок 13). Если в процессе установки потребуется вводить латинские буквы, можно переключить раскладку с помощью сочетания клавиш **Alt+Shift** или **Win+Пробел**.

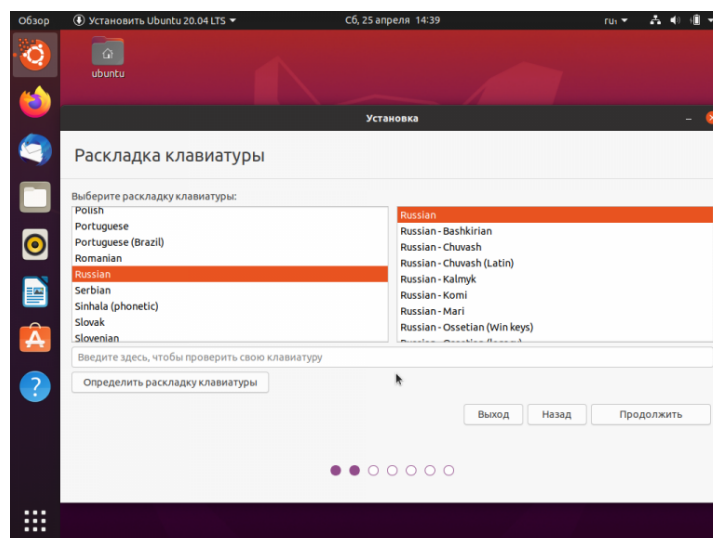


Рисунок 13

6) Выбрать программное обеспечение.

Рекомендуется выбрать пункт **Обычная установка**, в этом случае Установщик установит типовой набор приложений. Если выбрать опцию **Минимальная установка**, то дополнительные программы установлены не будут. Также рекомендуем оставить включенной опцию **Загрузить обновления во время установки Ubuntu**, так как к моменту завершения установки уже могут быть установлены важные исправления (Рисунок 14).

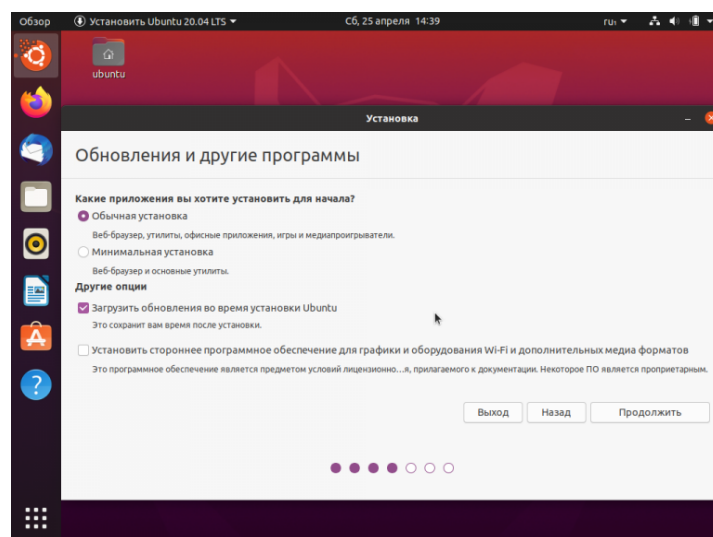


Рисунок 14

Флажок **Установить стороннее программное обеспечение для графики и оборудования Wi-Fi и дополнительных медиа форматов** устанавливается по желанию. По завершении настройки нажмите кнопку **Продолжить**.

7) Осуществить разметку диска.

Установщик определяет есть ли на диске другие операционные системы. Если их нет, можно нажать кнопку **Установить сейчас** (Рисунок 15). Диск будет автоматически размечен.

Если на диске уже установлена Windows, можно выбрать пункт **Установить рядом с Windows**. В этом случае компьютер можно будет загружать с Ubuntu или Windows по желанию.

Если выбрать опцию **Другой вариант**, Установщик даст возможность выполнить собственную разметку диска. Это удобно, например, при установке Ubuntu на SSD, для того, чтобы не создавался раздел swap.

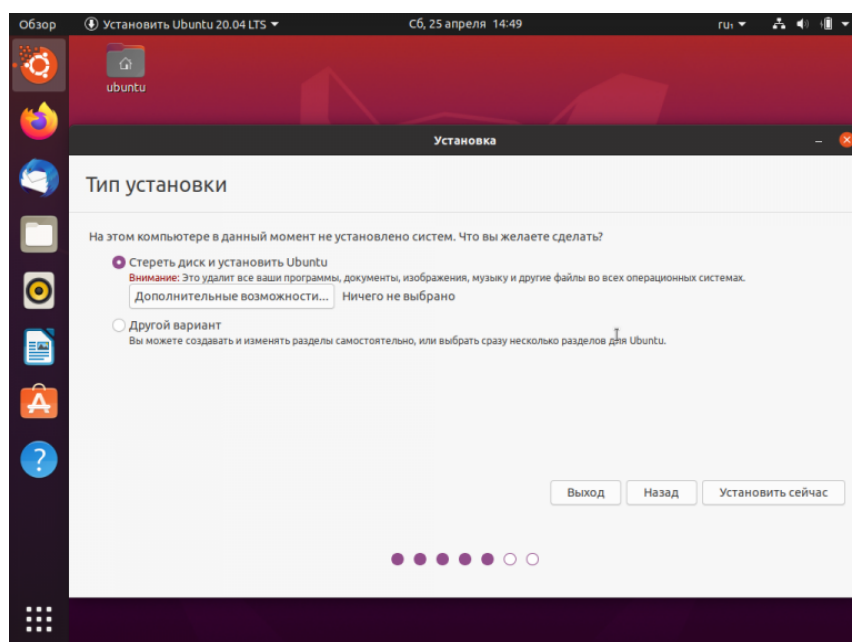


Рисунок 15

8) Подтвердить изменения разметки диска.

Установщик Ubuntu в процессе разметки диска непосредственно физические действия с диском не выполняет, а только лишь показывает проект изменений (Рисунок 16). Это удобно, если разметка спланирована неверно и её нужно поменять.

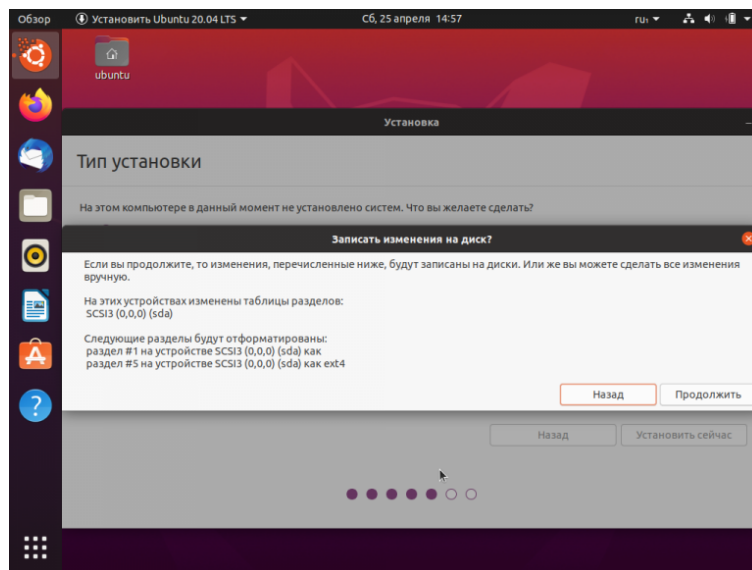


Рисунок 16

Следует просмотреть список предлагаемых изменений и нажать кнопку **Продолжить**, чтобы принять их. После этого шага установка Ubuntu 20.04 уже началась в фоне, но следует настроить дополнительно несколько параметров.

9) Установить часовой пояс.

Для того, чтобы система показывала правильное время, необходимо указать часовой пояс. Это можно сделать прямо на карте, кликнув левой кнопкой мыши в нужную область. Также можно найти и указать город в строке поиска под картой. Выбрав часовой пояс, нажмите кнопку **Продолжить** (Рисунок 17).

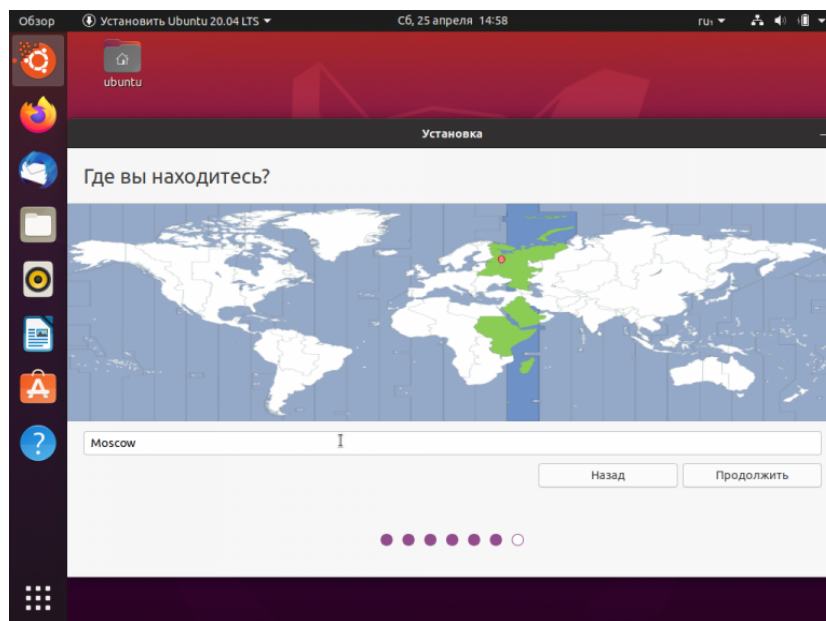


Рисунок 17

10) Создать учетную запись пользователя.

Каждый пользователь в Ubuntu работает под отдельной учётной записью, создаваемой на этом шаге. Следует ввести Ваше имя, имя Вашего компьютера, а также имя (логин) пользователя.

Затем следует задать пароль. Система подскажет правильность пароля. Опции в нижнем ряду определяют, будет система требовать ввода пароля при входе или нет. Когда все значения будут введены следует нажать кнопку **Продолжить** (Рисунок 18).

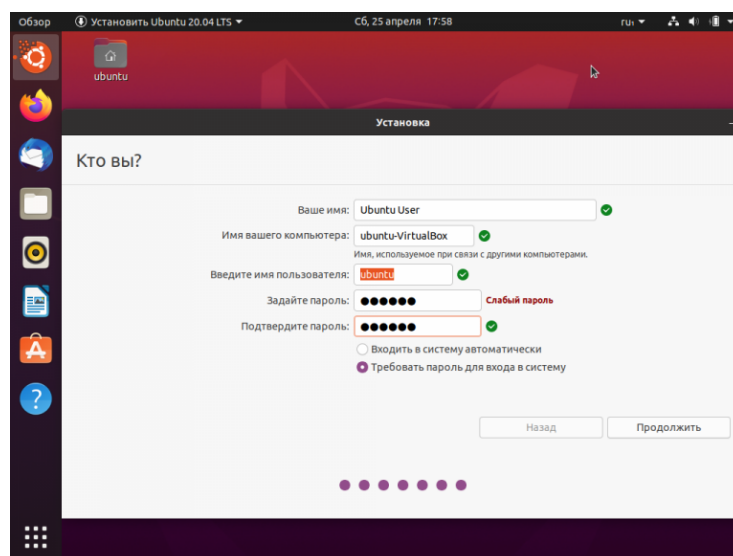


Рисунок 18

11) Установить систему.

После сбора требуемой информации необходимо дождаться пока завершится установка операционной системы Ubuntu (Рисунок 19).

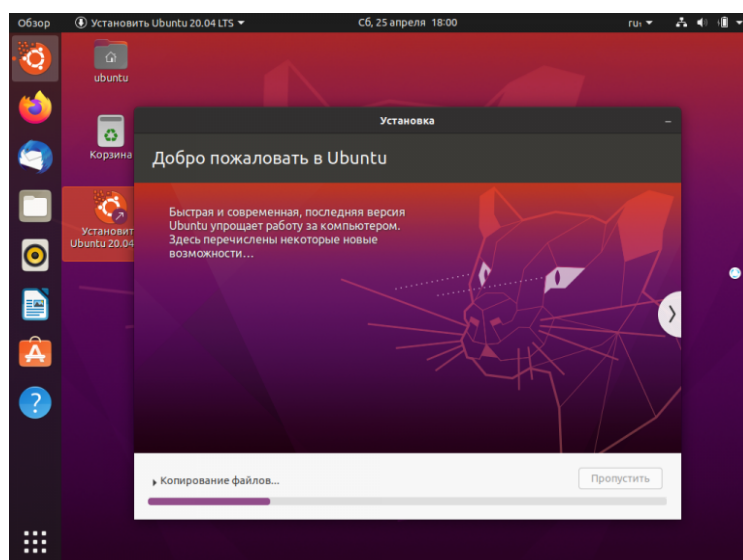


Рисунок 19

12) Осуществить перезагрузку.

По окончании копирования файлов система все еще находится в Live-режиме. Установщик предложит сделать перезагрузку для того, чтобы начать работу уже с установленной на диск операционной системой. Для того, чтобы начать работу с Ubuntu следует нажать кнопку **Перезагрузить** (Рисунок 20).

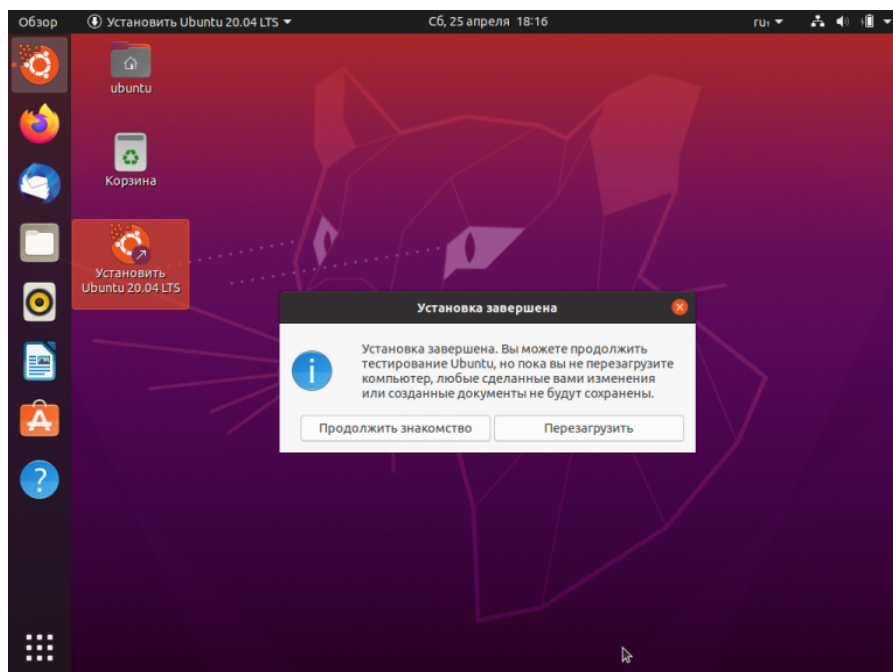


Рисунок 20

Система предложит извлечь носитель с ISO-образом перед перезагрузкой. Вытащите флешку из порта или диск из DVD-привода, после чего следует нажать клавишу **Enter** (Рисунок 21).

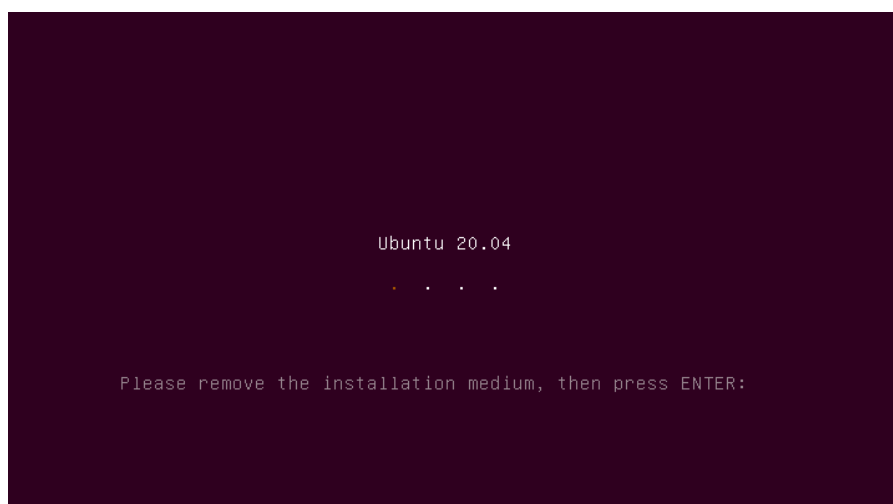


Рисунок 21

13) Осуществить первый вход в систему.

Новая система начинает загружаться, отображая экран с версией (Рисунок 22).

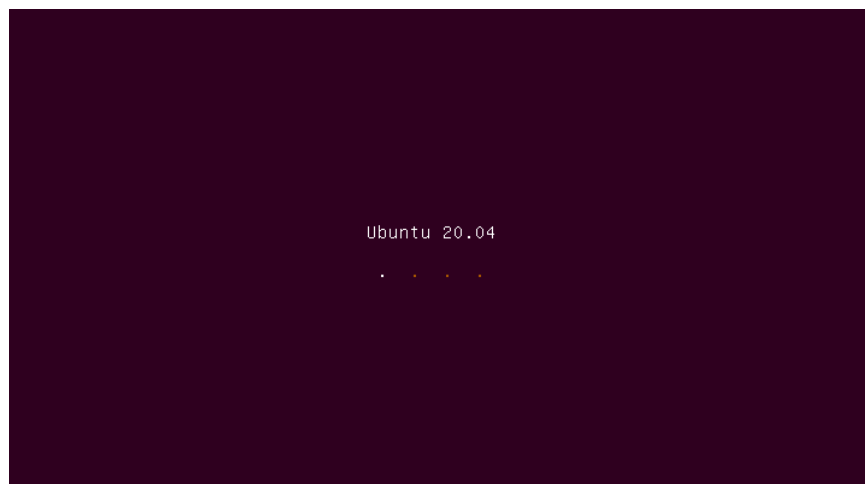


Рисунок 22

Как только графическая оболочка будет загружена, система предложит выбрать пользователя, выполняющего вход (Рисунок 23).

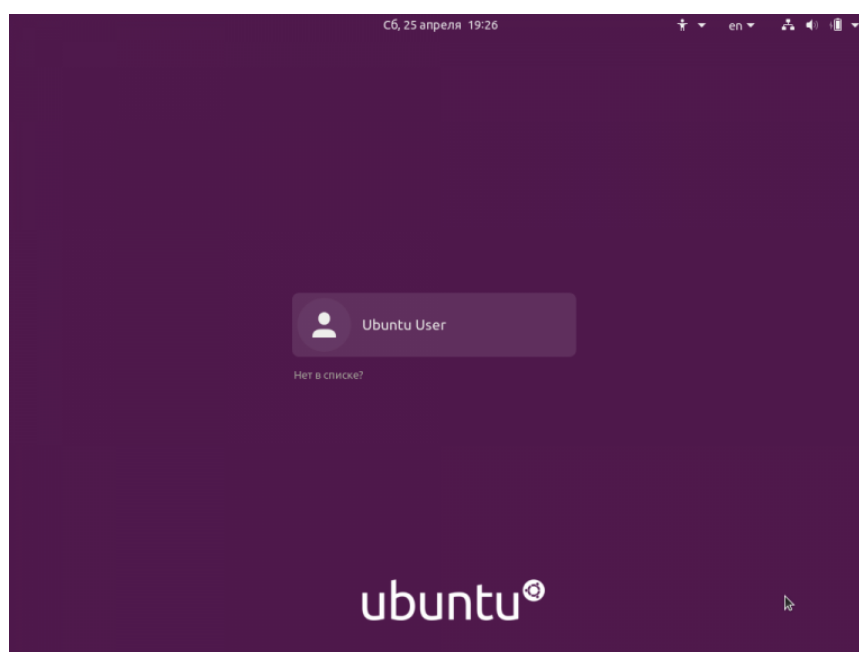


Рисунок 23

При нажатии левой кнопкой мыши на имя пользователя, появится поле ввода пароля. Следует ввести пароль учетной записи (Рисунок 24). Показ пароля при вводе можно включить, нажав левой кнопкой мыши на изображение перечеркнутого глаза в правой части поля.

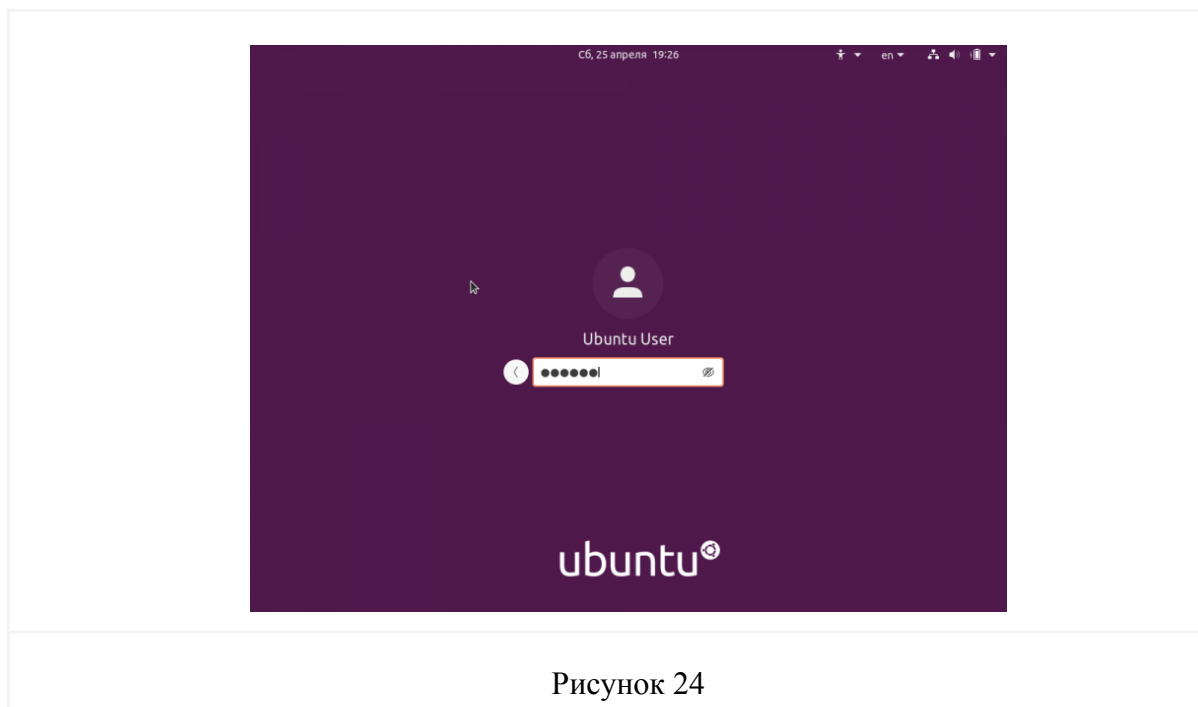


Рисунок 24

Установка Android Studio на Ubuntu

Для установки Android Studio на Ubuntu необходимо выполнить следующие действия:

- 1) Следует предварительно скачать **android-studio**. Для этого нужно перейти на сайт <https://developer.android.com/studio> и пролистать вниз сайта, где выбрать скачивание **android-studio-2020.3.1.24-linux.tar.gz** (Рисунок 25). Следует ознакомиться и согласиться с условиями.

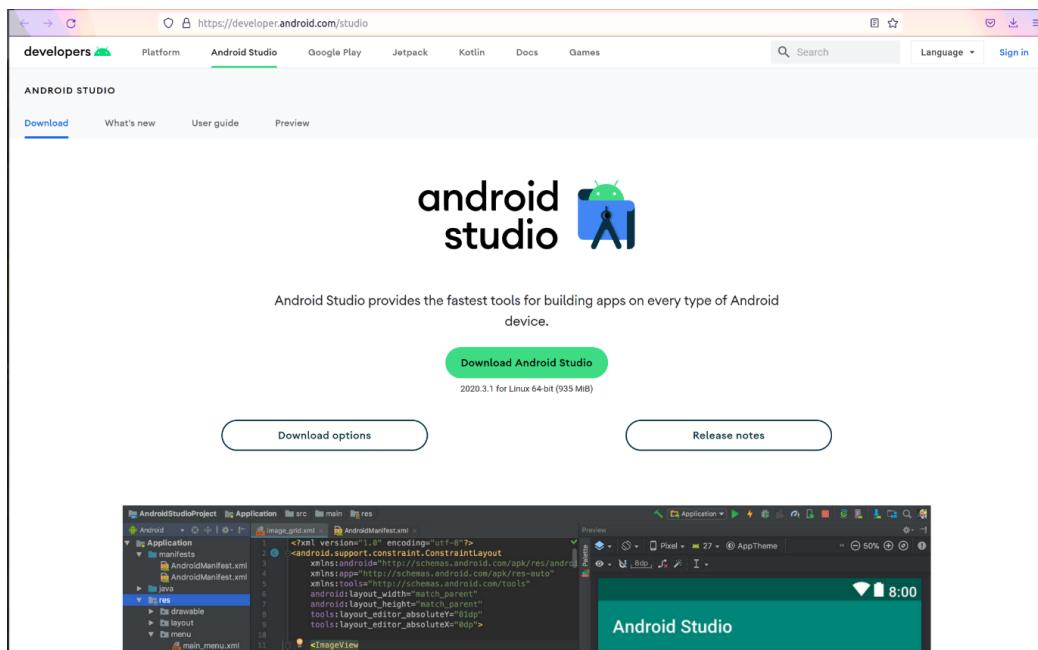


Рисунок 25

2) Установить необходимые библиотеки используя команду

```
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
lib32z1 libbz2-1.0:i386
```

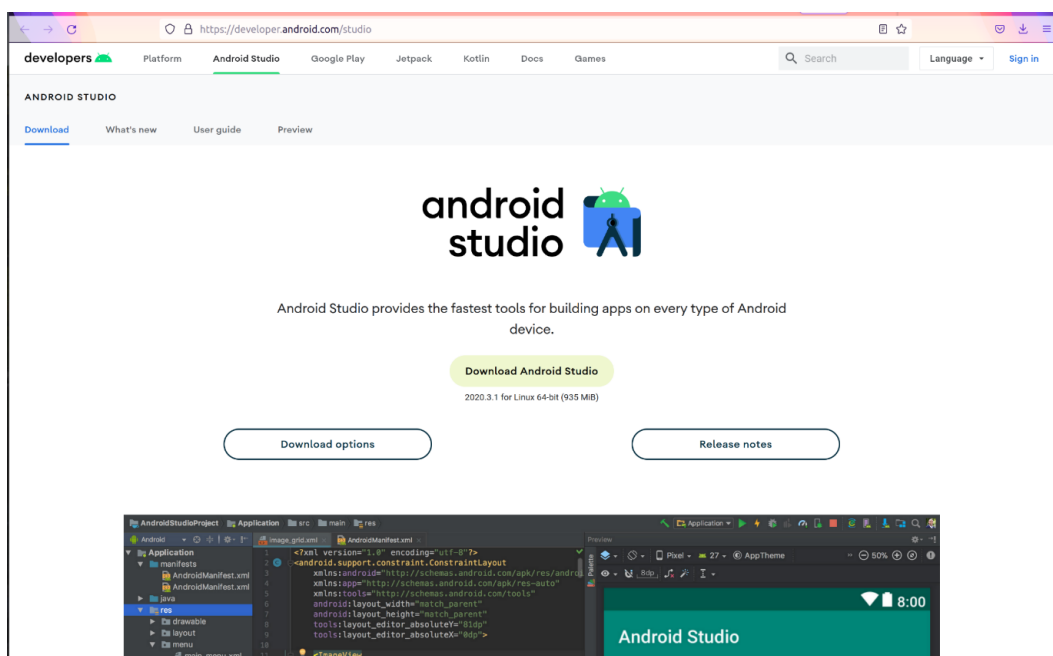


Рисунок 26

3) После скачивания Android Studio необходимо зайти в папку со скачанным архивом и распаковать его

`tar -xzf android-studio-2020.3.1.24-linux.tar.gz` (Рисунок 27)

- 4) необходимо разархивировать папку в `/home/$USER`, где `$USER` - имя вашего пользователя Ubuntu;
- 5) Открыть терминал, перейти в папку `android-studio` командой `cd /home/$USER/android-studio/` и выполнить `./bin/studio.sh` для запуска инсталлятора Android Studio.
- 6) В открывшемся окне установки IDE нажать `Next -> Standard + Next -> Выбрать тему + Next -> Согласиться с лицензией -> Finish`
- 7) Дождаться окончания установки.

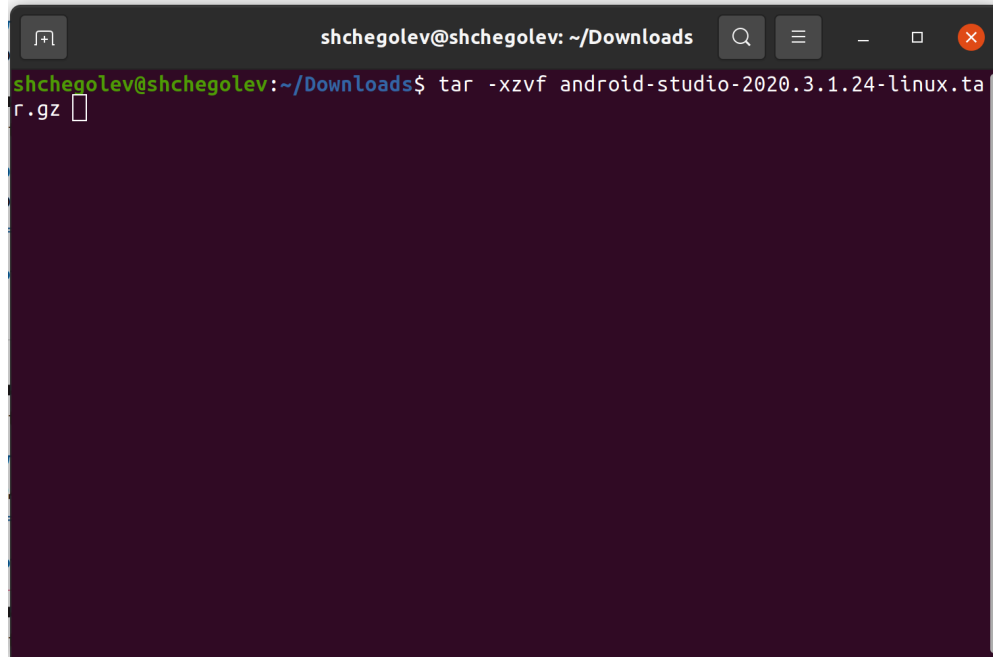
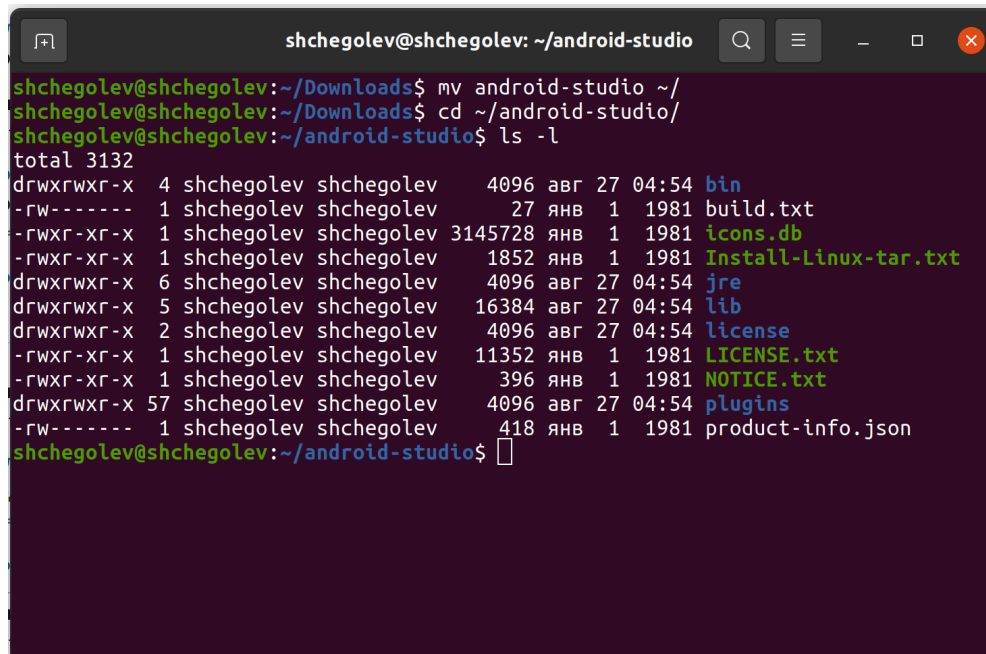


Рисунок 27

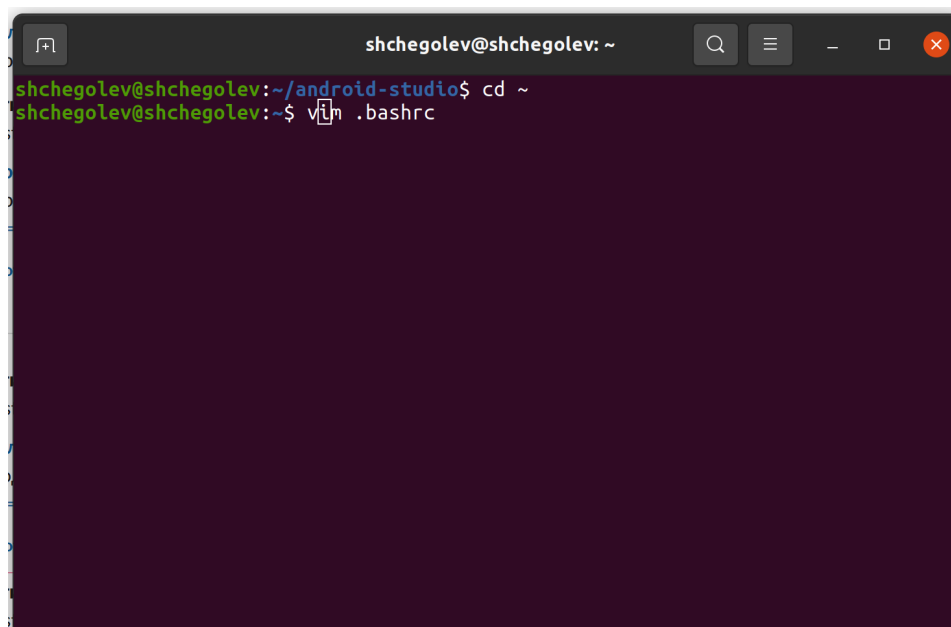
Далее следует перенести распакованную папку в желаемое место (Рисунок 28).



```
shchegolev@shchegolev: ~/android-studio
shchegolev@shchegolev:~/Downloads$ mv android-studio ~/
shchegolev@shchegolev:~/Downloads$ cd ~/android-studio/
shchegolev@shchegolev:~/android-studio$ ls -l
total 3132
drwxrwxr-x 4 shchegolev shchegolev 4096 авг 27 04:54 bin
-rw----- 1 shchegolev shchegolev 27 янв 1 1981 build.txt
-rwxr-xr-x 1 shchegolev shchegolev 3145728 янв 1 1981 icons.db
-rwxr-xr-x 1 shchegolev shchegolev 1852 янв 1 1981 Install-Linux-tar.txt
drwxrwxr-x 6 shchegolev shchegolev 4096 авг 27 04:54 jre
drwxrwxr-x 5 shchegolev shchegolev 16384 авг 27 04:54 lib
drwxrwxr-x 2 shchegolev shchegolev 4096 авг 27 04:54 license
-rwxr-xr-x 1 shchegolev shchegolev 11352 янв 1 1981 LICENSE.txt
-rwxr-xr-x 1 shchegolev shchegolev 396 янв 1 1981 NOTICE.txt
drwxrwxr-x 57 shchegolev shchegolev 4096 авг 27 04:54 plugins
-rw----- 1 shchegolev shchegolev 418 янв 1 1981 product-info.json
shchegolev@shchegolev:~/android-studio$
```

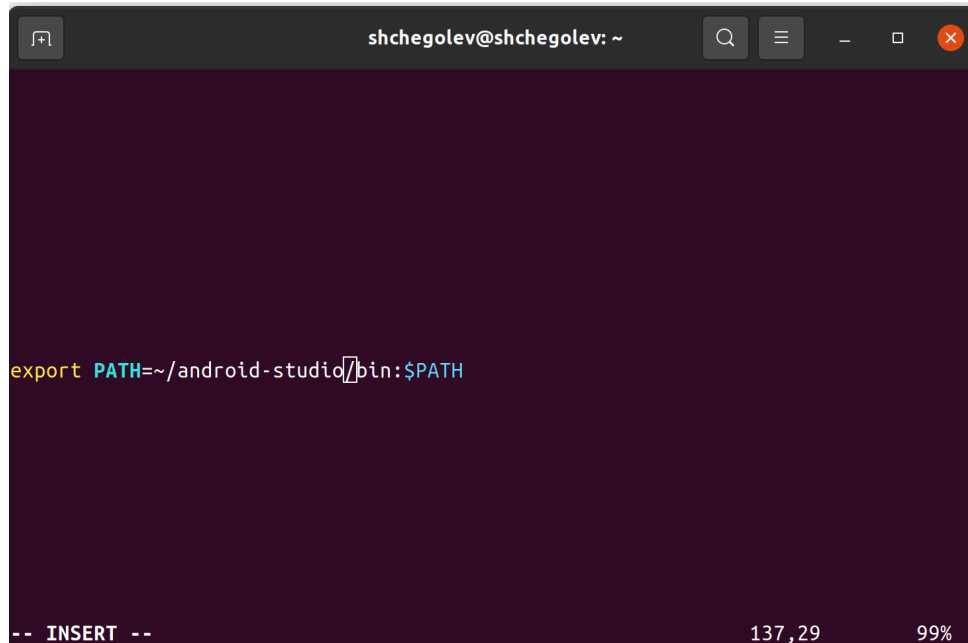
Рисунок 28

8) Добавить android-studio/bin в PATH. Для этого следует добавить строку `export PATH=~/android-studio/bin:$PATH` в файл `~/.bashrc` (Рисунок 29,30).



```
shchegolev@shchegolev: ~/
shchegolev@shchegolev:~/android-studio$ cd ~
shchegolev@shchegolev:~$ vim .bashrc
```

Рисунок 29



```
shchegolev@shchegolev: ~  
  
export PATH=~/android-studio/bin:$PATH  
  
-- INSERT -- 137,29 99%
```

Рисунок 30

- 9) Запустить установщик android-studio командой `studio.sh` (Рисунок 31,32).

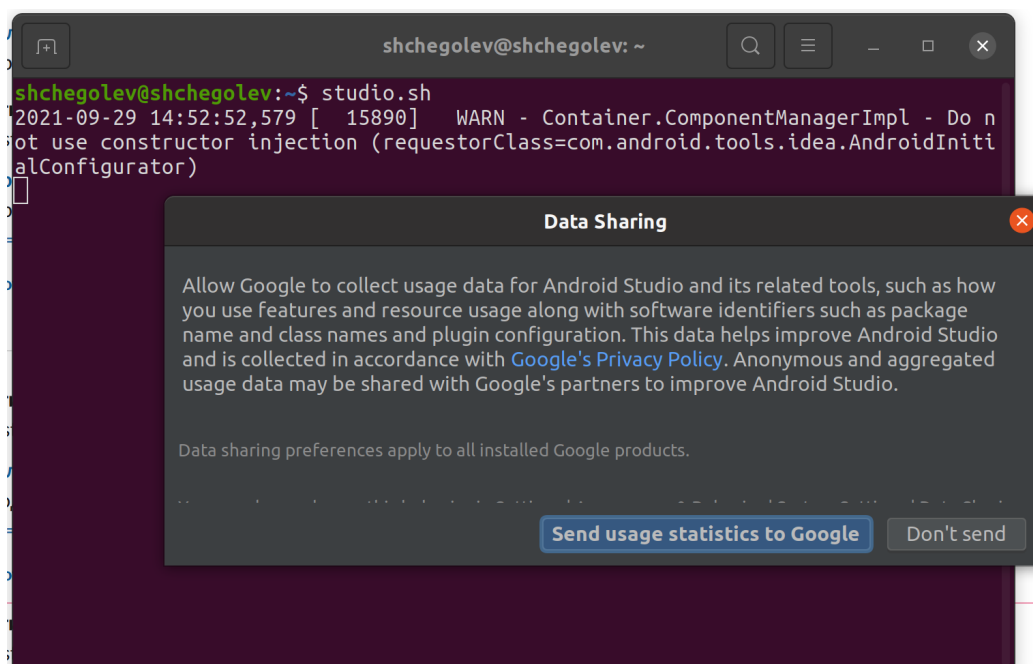


Рисунок 31

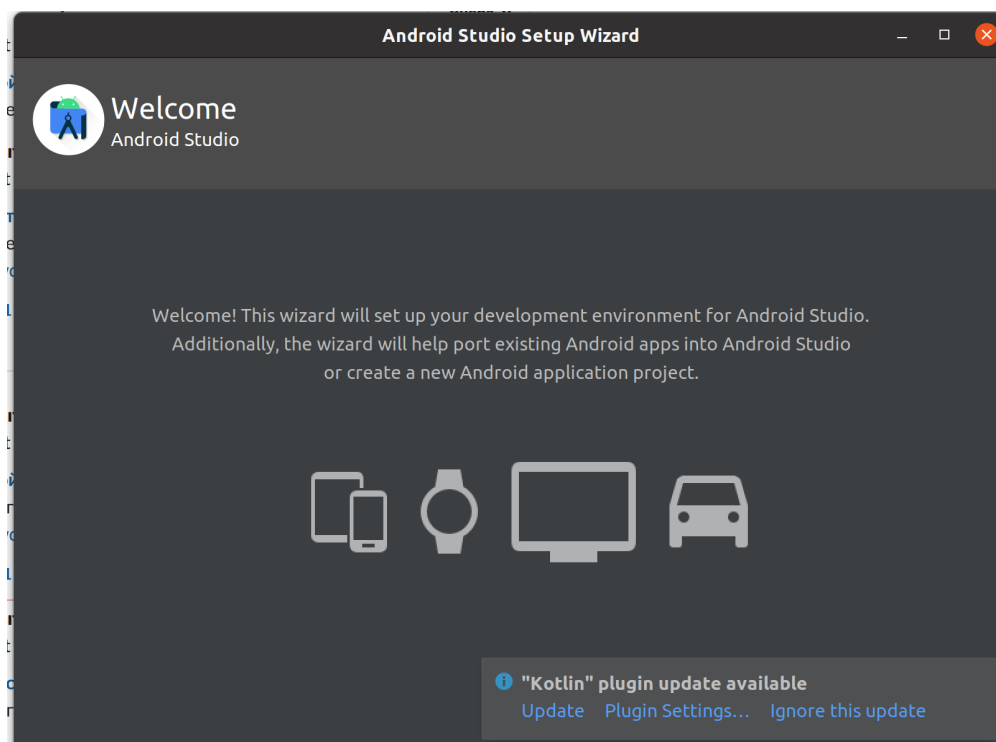


Рисунок 32

10) Следует выбрать стандартную установку (Рисунок 33).

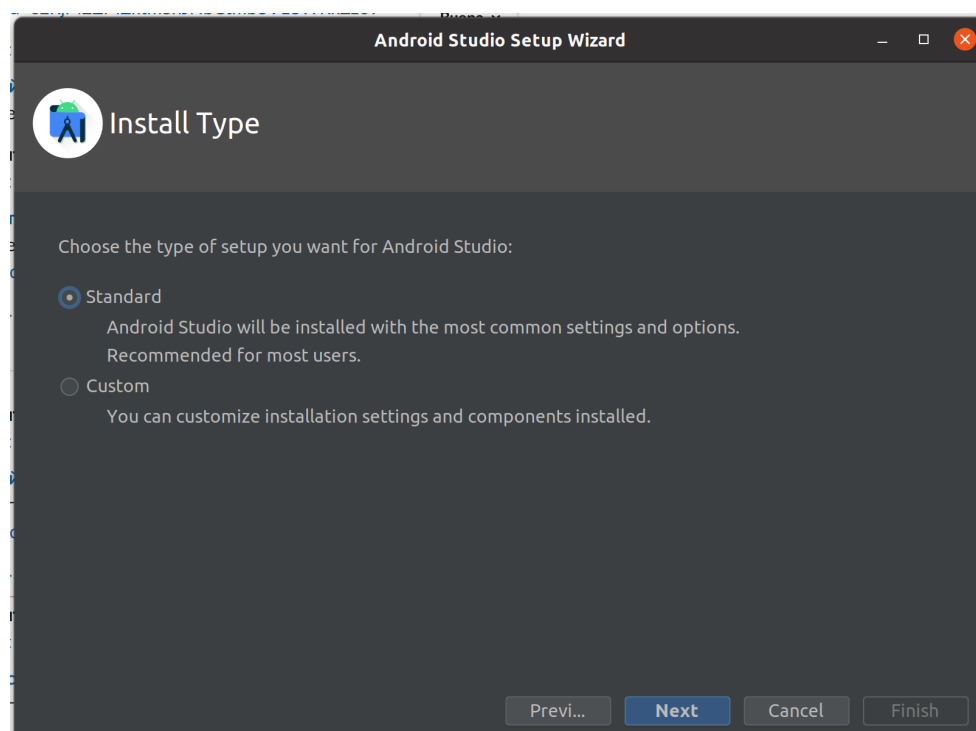


Рисунок 33

11) Выбрать тему оформления (Рисунок 34).

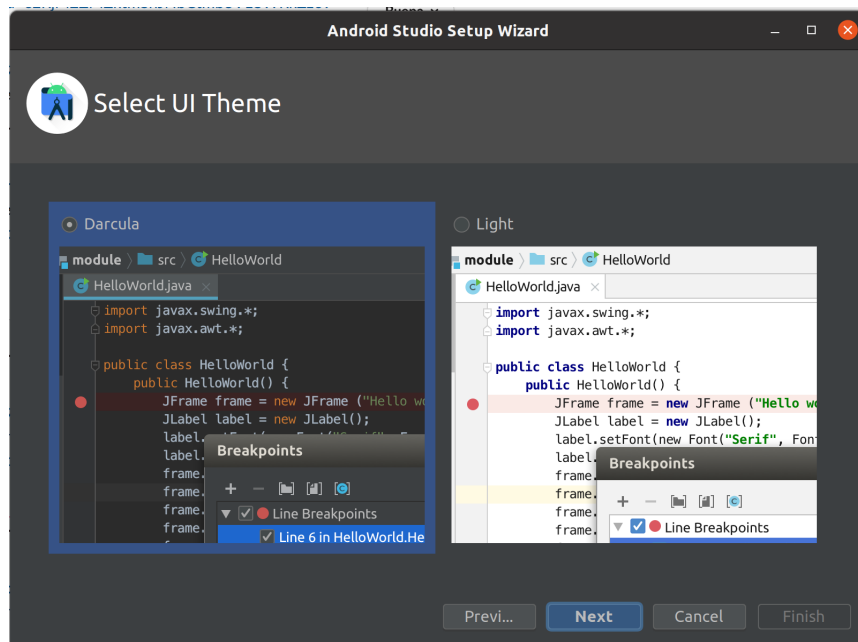


Рисунок 34

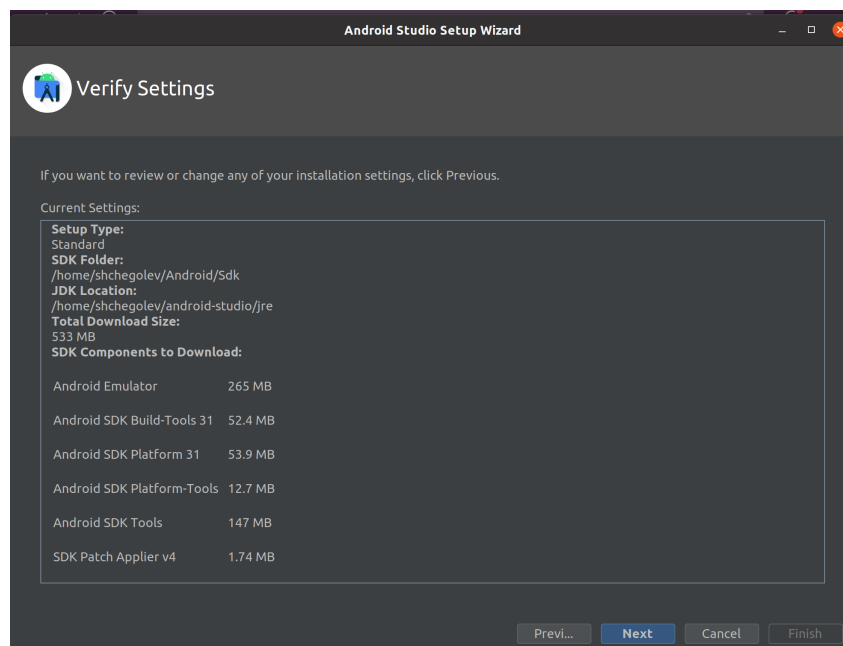


Рисунок 35

12) Запустить установку (Рисунок 36).

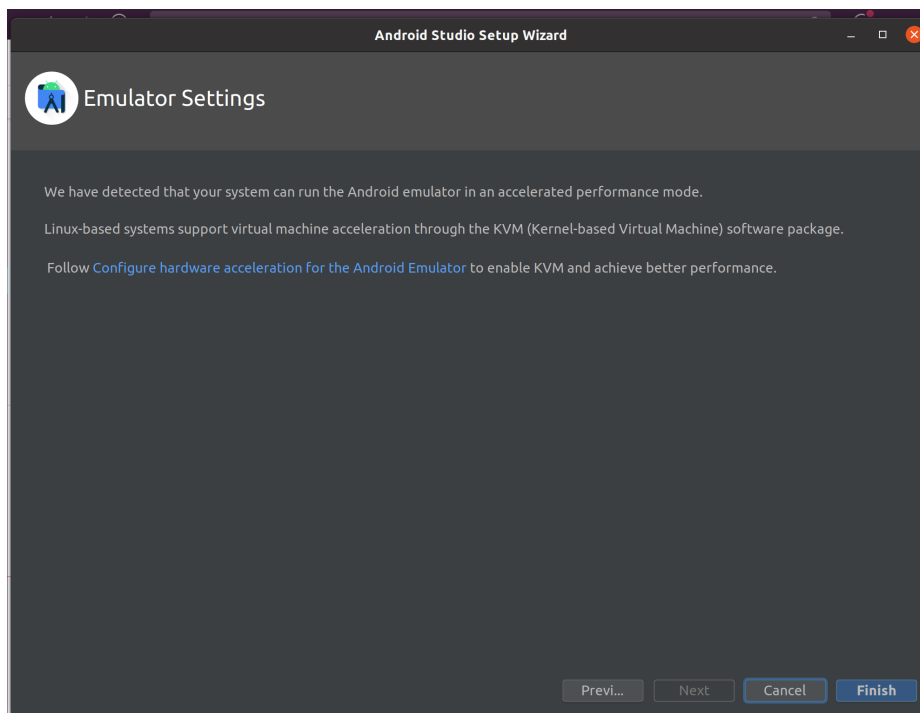


Рисунок 36

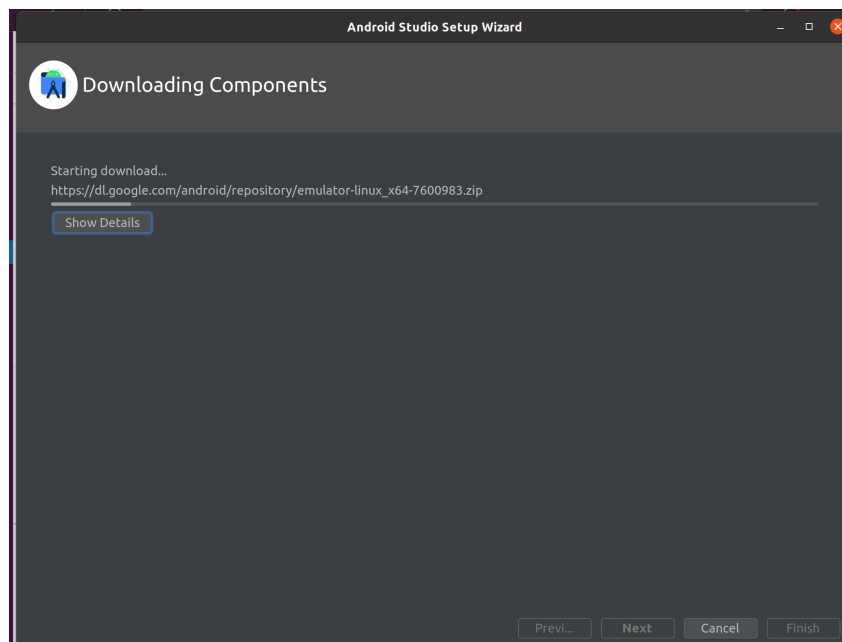


Рисунок 37

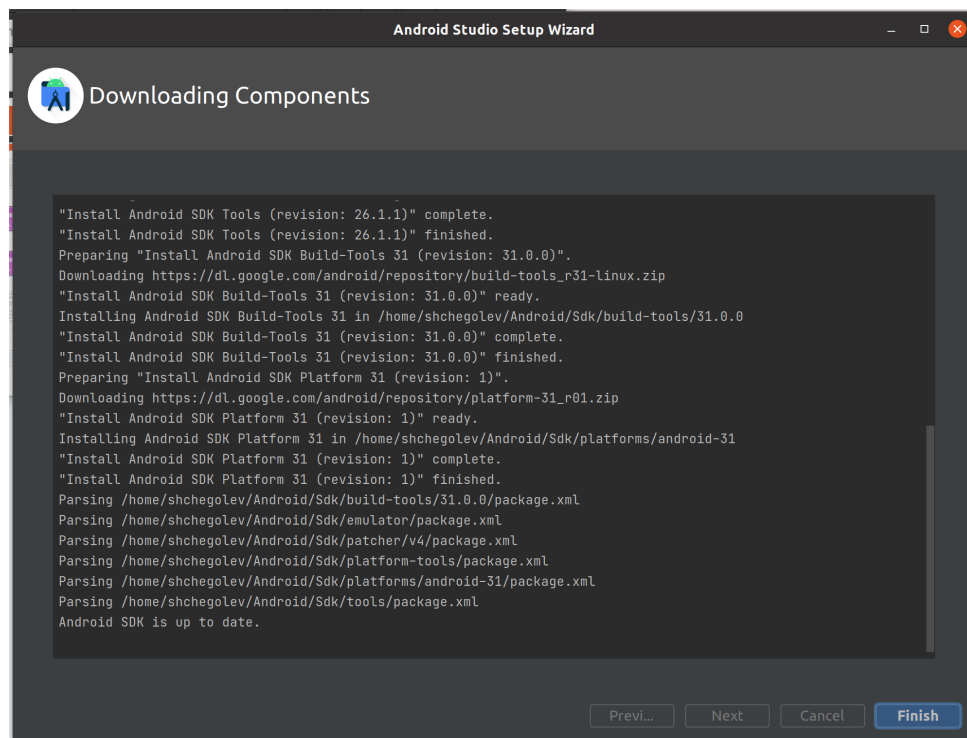


Рисунок 38

4.4.1 Завершение установки Android studio и установка дополнительных пакетов

- 1) Запустить Android Studio: открыть терминал и выполнить
`cd /home/$USER/android-studio/`
`./bin/studio.sh`

Где \$USER - домашняя папка пользователя Ubuntu

В открывшемся окне после установки IDE следует выбрать путь Open -> Project -> ~/.soft/src/legahelp-mobile -> android

- 2) Если появилась ошибка **invalid gradle jdk** и справа внизу есть опция - использовать новую версию gradle JDK, следует нажать и установить новую версию (дождаться окончания установки).

- 3) В верхнем правом углу следует найти значок зеленой коробки и телефона с названием AVD Manager -> Create new virtual device -> Выбрать Phone -> Pixel 5 (например) -> Next -> Выбрать Pie (28 API Level) и рядом активировать ссылку Download

- 4) В случае появления надписи в блоке Recommendation "Your CPU does not support required features (VT-x or SVM)" запустить эмулятор невозможно, так как необходимо устройство, поддерживающее виртуализацию.

- 5) После нажатия на ссылку Download -> Accept (подтверждение соглашения внизу) -> Ожидать окончания (скачается 1Гб) -> Finish

- 6) Происходит возврат на окно выбора версии, далее следует выбрать скачанную 28ую API версию -> Next -> Finish

- 7) Далее следует закрыть окно AVD Manager-a
- 8) В случае, если в правом нижнем углу все еще видно, что прогружаются модули, то необходимо дождаться окончания (сменяющиеся записи Gradle: ... в самом нижнем правом углу окна IDE).

Установка Docker и Docker compose на Ubuntu

Для установки актуального пакета **Docker** необходимо иметь стабильное подключение к интернету и пароль от **root** пользователя системы.

Внимание! Для удобства установки Docker и Docker-compose в комплект поставки ПО добавлен инсталлятор. Откройте терминал в директории `~/soft/backend` и запустите `./setup_environment.sh`. В случае, если все пакеты и зависимости установлены корректно (Рисунок 39), данный раздел можно пропустить. Если Docker или Docker-compose установлены некорректно, проведите установку вручную, как описано ниже.

```
Testing Basic dependencies:
git                                OK
gcc                                OK
g++                                OK
autoconf                           OK
automake                            OK
libtool                             OK
python                             OK
make                                OK
Testing Node.JS installations:
node                               OK
npm                                 OK
Testing Docker installations:
docker                             OK
```

Рисунок 39. Успешное завершение установки Docker и Docker-compose в инсталляторе

4.3.1 Подключение репозитория и обновление списка пакетов в системе

Предварительно необходимо произвести обновление существующих пакетов для установки ПО.

Для этого необходимо выполнить команду `sudo apt update`.

Затем необходимо установить рекомендованное ПО для обеспечения возможности выполнения команд группы `apt` через протокол `https`.

```
sudo apt install apt-transport-https ca-certificates curl
software-properties-common
```

4.3.2 Установка Docker и Docker-compose

Для установки Docker-а необходимо добавить GPG ключ официального репозитория Docker в список

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
```

Следует добавить Docker репозиторий в список APT источников

```
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu focal stable"
```

Далее установить Docker следующей командой: `sudo apt install docker-ce`

Для установки Docker-Compose необходимо выполнить команду

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.27.4/docker-c
ompose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

4.3.3 Настройка Docker

Необходимо включить сервис Docker-a командой `sudo service docker start`

Чтобы не вызывать все команды Docker-a от лица администратора необходимо добавить Docker в sudo группу `sudo usermod -aG docker ${USER}`

Чтобы новые права вступили в группу необходимо выполнить команду `su - ${USER}`

Далее необходимо установить корректные права доступа для исполнения команды docker-compose `sudo chmod +x /usr/local/bin/docker-compose`

Затем необходимо установить права на запущенном Docker сожете для корректной работы системы и избежания проблем с правами выполнения команд `sudo chmod 666 /var/run/docker.sock`

4.3.4 Проверка установки

Для проверки того, что Docker был корректно установлен следует выполнить команду `sudo service docker status`

Если в графе status написано Active – установка прошла успешно.

Для проверки того, что Docker-Compose был корректно установлен необходимо выполнить `docker-compose --version`