
AgentNavigator

Платформа для работы с ИИ-агентами

STC-S9013

Руководство администратора

НЦДА.01201-01 91 01

Версия документа: 3-160725

Версия программного обеспечения: 4.15

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ	8
1.1 Сведения о продукте и изготовителе	8
1.2 Служба технической поддержки	8
1.3 Назначение решения	8
2. СИСТЕМНЫЕ ТРЕБОВАНИЯ	9
2.1 Список используемых портов	10
3. СОСТАВ ПЛАТФОРМЫ	11
3.1 Подсистема взаимодействия с клиентом ChatWidget	13
3.2 Подсистема обработки запросов клиентов BotServer	13
3.3 Подсистема интеграции HttpAdapter	14
3.4 Подсистема хранения и аналитики ServiceWeb	14
3.5 Подсистема проверки целостности IntegrityVerifier	14
3.6 Резервирование компонентов платформы	14
4. УСТАНОВКА И НАСТРОЙКА	17
4.1 Подключение локального RPM-репозитория	17
4.2 Установка пререквизитов и утилит	18
4.3 Настройка и запуск окружения	19
4.3.1 RabbitMQ	19
4.3.1.1 Общая конфигурация RabbitMQ	19
4.3.1.2 Аутентификация Rabbit с помощью TLS	20
4.3.1.3 Генерация сертификатов для RabbitMQ	21
4.3.2 MongoDB	23
4.3.2.1 Создание пользователей с аутентификацией по логину и паролю	23
4.3.2.2 Настройка авторизации	24
4.3.3 PostgreSQL	28
4.3.4 CORS	29
4.3.5 FileServer-клиент	30
4.3.6 Nginx	30
4.3.6.1 Замена сертификатов для Nginx	32
4.3.7 HASP Driver	32

4.3.8 HG server	32
4.3.9 NFS-сервер	33
4.3.10 NFS-клиент	33
4.3.11 Замена паролей сервисных учетных записей	34
4.4 Установка системы AgentNavigator	35
4.4.1 Обновление локального RPM-репозитория	35
4.4.2 Установка обязательных компонентов	35
4.4.3 Установка необязательных компонентов	36
4.4.4 Установка компонентов с изменением пути инсталляции	37
4.5 Настройка и запуск компонентов	37
4.5.1 BotServer	37
4.5.2 HttpAdapter	45
4.5.3 ServiceWeb	49
4.5.3.1 Подключение ServiceWeb к Security	58
4.5.3.2 Фильтрация по параметрам диалога	59
4.5.3.3 Настройка контекста для веб-компонентов	61
4.5.3.4 Настройка автоматической очистки базы данных	61
4.5.4 IntegrityVerifier	62
4.5.5 ChatWidget	63
4.5.5.1 Кастомизация ChatWidget	65
4.5.5.2 Интеграция ChatWidget с сайтом	67
4.5.6 OmnichatAdapter	67
4.5.6.1 Настройка форматирования ответов для OmnichatAdapter	71
4.5.7 SipAdapter	73
4.5.8 ConfigService	76
4.5.9 Telegram	78
4.5.10 CrafttalkAdapter	80
4.5.11 Classifier и Classifier Trainer	82
4.5.12 NER и NER Trainer	83
4.5.13 FileServer	84
4.5.14 python-http-adapter	85
4.5.15 database-http-adapter	86
4.5.16 Корректировка тайм-аутов	88
4.5.17 Изменение продолжительности сессии	88

4.5.18 Общая конфигурация MongoDB	89
4.5.19 Конфигурирование LDAP	90
4.5.19.1 Создание зашифрованного AES-пароля	90
4.6 Настройка ротации лог-файлов	92
4.7 Включение статического баннера	93
5. ЗАПУСК СИСТЕМЫ	95
5.1 Авторизация	95
5.2 Интерфейс	95
6. ФОРМИРОВАНИЕ РОЛЕЙ	98
6.1 Привилегии	98
6.2 Создание новой роли	105
6.3 Изменение роли	106
6.4 Копирование роли	107
6.5 Удаление роли	107
6.6 Настройки Active Directory	107
7. УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ	109
7.1 Список пользователей	109
7.2 Создание пользователя	109
7.3 Редактирование пользователя	111
7.4 Блокировка пользователя	111
7.5 Удаление пользователя	112
8. РАБОТА С КАНАЛАМИ	114
8.1 Создание канала	114
8.2 Редактирование канала	115
8.3 Загрузка сценария	115
8.4 Проверка работы виджета	116
8.5 Приоритизация каналов	117
8.6 Удаление канала	118
9. СОСТОЯНИЕ КОМПОНЕНТОВ	119
9.1 Список компонентов	119
9.2 Состояние компонентов	119
9.3 Список проверок компонентов	120

10. ОПИСАНИЕ ЛОГ-ФАЙЛОВ	121
10.1 Расположение и содержимое лог-файлов	121
10.2 Описание лог-файлов	122
11. ПРОВЕРКА РАБОТОСПОСОБНОСТИ	126
11.1 Проверка загрузки сценария	126
11.2 Проверка работы ChatWidget	126
11.3 Проверка работы OmnichatAdapter	126
11.4 Проверка работы SipAdapter	127
11.5 Проверка работы IntegrityVerifier	128
11.6 Проверка работы добавленных каналов адаптера	129
11.7 Проверка ротации логов	130
11.8 Проверка работы механизма устаревания сессии	131
12. МОНИТОРИНГ КОМПОНЕНТОВ	132
13. ГЛОССАРИЙ	138
14. ИСТОРИЯ ОБНОВЛЕНИЙ	144

ВВЕДЕНИЕ

Общие положения

Настоящее руководство предназначено для персонала, производящего установку и настройку платформы **AgentNavigator**.

Руководство содержит описание интерфейса интеграции решения с внешней системой.

Сведения, приведенные в руководстве, относятся к версии 4.15 **AgentNavigator**.

Настоящее руководство не заменяет учебную, справочную литературу и руководства от производителей операционной системы и общего программного обеспечения.

Соглашения и обозначения

В документе приняты следующие типографские соглашения:

Формат текста	Описание
Обычный	Основной текст документа
<i>Курсив</i>	Специальное понятие при первом появлении в основном тексте и списке терминов
Полужирный	Наименования компаний, программ и программных компонентов, а также наименования элементов интерфейса (заголовки, кнопки и т. п.)
<i>Полужирный курсив</i>	Имена файлов и путей доступа к ним
Моноширинный	Тексты конфигурационных файлов, исходные текстов программ, параметры API
<u>Подчеркнутый бирюзовый</u>	Внешние ссылки и ссылки на разделы настоящего руководства
Элемент > Вложенный элемент	Выбор пункта меню. Например, текст Файл > Выход должен пониматься так: выбрать меню Файл , а затем — команду Выход из меню Файл
<метка>	Фрагмент текста, который должен быть заменен реальным значением. Например, в записи D:\<DirName>\file.txt текст <DirName> используется для обозначения названия каталога в файловой системе



Указания на действия, которые необходимо выполнить в обязательном порядке.



Примечания, рекомендации и прочие сведения, требующие особого внимания.



Ссылки на другие документы в основном тексте.

Товарные знаки

Названия компаний и названия продуктов, упомянутые в документе, являются собственностью их соответствующих владельцев.

Иллюстративный материал, приведённый в документе, может содержать вымышленные имена, фамилии, номера телефонов, адреса и тому подобные сведения. Любое их совпадение с данными реальных личностей следует считать случайностью.

Интерфейс на рисунках может незначительно отличаться от интерфейса продукта в связи с его постоянной модернизацией.

Ни одна из частей этого издания не подлежит воспроизведению, передаче, хранению в поисковой системе или переводу на какой-либо язык в любой форме, любыми средствами без письменного разрешения общества с ограниченной ответственностью «ЦРТ-инновации».

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Сведения о продукте и изготовителе

Наименование: **AgentNavigator**

Платформа для работы с ИИ-агентами

Изготовитель: Общество с ограниченной ответственностью «ЦРТ-инновации»

Адрес: 194044, Санкт-Петербург, ул. Гельсингфорсская, д. 3, к. 11 лит. Д, пом. 229

Телефон: +7 (812) 325-88-48

Факс: +7 (812) 327-92-97

1.2 Служба технической поддержки

Адрес службы сервисного обслуживания и технической поддержки в интернете:

Электронная почта: support@speechpro.com

Адрес в сети Интернет: <http://www.speechpro.ru/support>

При обращении в службу технической поддержки необходимо представить четкое описание возникшей проблемы.

Предварительно подготовьте следующую информацию:

- сведения о конфигурации серверов;
- название и версии используемых операционных систем;
- версия программного обеспечения.

1.3 Назначение решения

Диалоговые и недиалоговые ИИ-агенты на базе **AgentNavigator** позволяют обрабатывать запросы пользователей и текстовые данные на естественном языке с помощью LLM и дополнительных инструментов и внешних источников данных (баз знаний, информационных систем и баз данных).

2 СИСТЕМНЫЕ ТРЕБОВАНИЯ

Функционирование компонентов **AgentNavigator** протестировано на компьютере, аппаратная конфигурация которого отвечает следующим параметрам:

- частота процессора: не менее 2,4 ГГц;
- количество физических ядер процессора: не менее 4;
- размер оперативной памяти: не менее 8 Гбайт;
- размер основного хранилища: не менее 300 Гбайт.

Рекомендуется использование SAS и RAID 10 для повышения пропускной способности хранилища.

На всех серверах необходимо наличие минимум одного USB-порта для подключения ключа лицензионной защиты, а также не менее двух портов Ethernet с пропускной способностью 100 Мбит/с.

Функционирование **AgentNavigator** протестировано на следующих операционных системах: РЕД ОС 8.0 x64, Red Hat Enterprise Linux v.9 x64, AlmaLinux 9.4 и Oracle Linux Server 9.4.

Платформа совместима со следующим сторонним программным обеспечением:

- Postgres 14;
- MongoDB 6;
- Java 21 (OpenJDK Zulu21);
- RabbitMQ-server 3.8.8.

Функционирование пользовательской части платформы успешно протестировано на компьютере, аппаратная конфигурация которого отвечает следующим параметрам:

- тип процессора: Intel Core i3 и выше (9 поколения и выше);
- частота процессора: не менее 1,8 ГГц;
- количество ядер процессора: не менее 2;
- размер доступной оперативной памяти: не менее 8 Гбайт.

Рабочий компьютер должен быть подключен к сети (локальной или Интернет). Для работы с большими файлами необходима скорость передачи данных в обоих направлениях не менее 100 Мбит/с.

Компоненты платформы поддерживают заявленные браузеры и их более новые версии:

- Chrome 133;
- Mozilla Firefox 134;
- Microsoft Edge 133;
- Яндекс.Браузер 24.10.4.

2.1 Список используемых портов

Сервис	Используемый порт
BotServer	8083
ChatWidget	9999
FileServer	9320
HttpAdapter	9030
IntegrityVerifier	10101
OmnichatAdapter	9066
ServiceWeb	8086
Telegram	8097
PrompterAdapter	9107
Classifier	9008
Classifier Trainer	9009
SipAdapter	9098
VkontakteAdapter	9100/9200
WhatsappAdapter	8090
RabbitMQ	5672 / 25672 / 15672
MongoDB	27017
PostgreSQL	5432
NFS	2049 UDP / 111 TCP
statedictionaryservice	8091
dialoghistoryservice	8092
cnreport	8093

3 СОСТАВ ПЛАТФОРМЫ

Состав и схема взаимодействия компонентов платформы **AgentNavigator** показана на рисунке ниже.

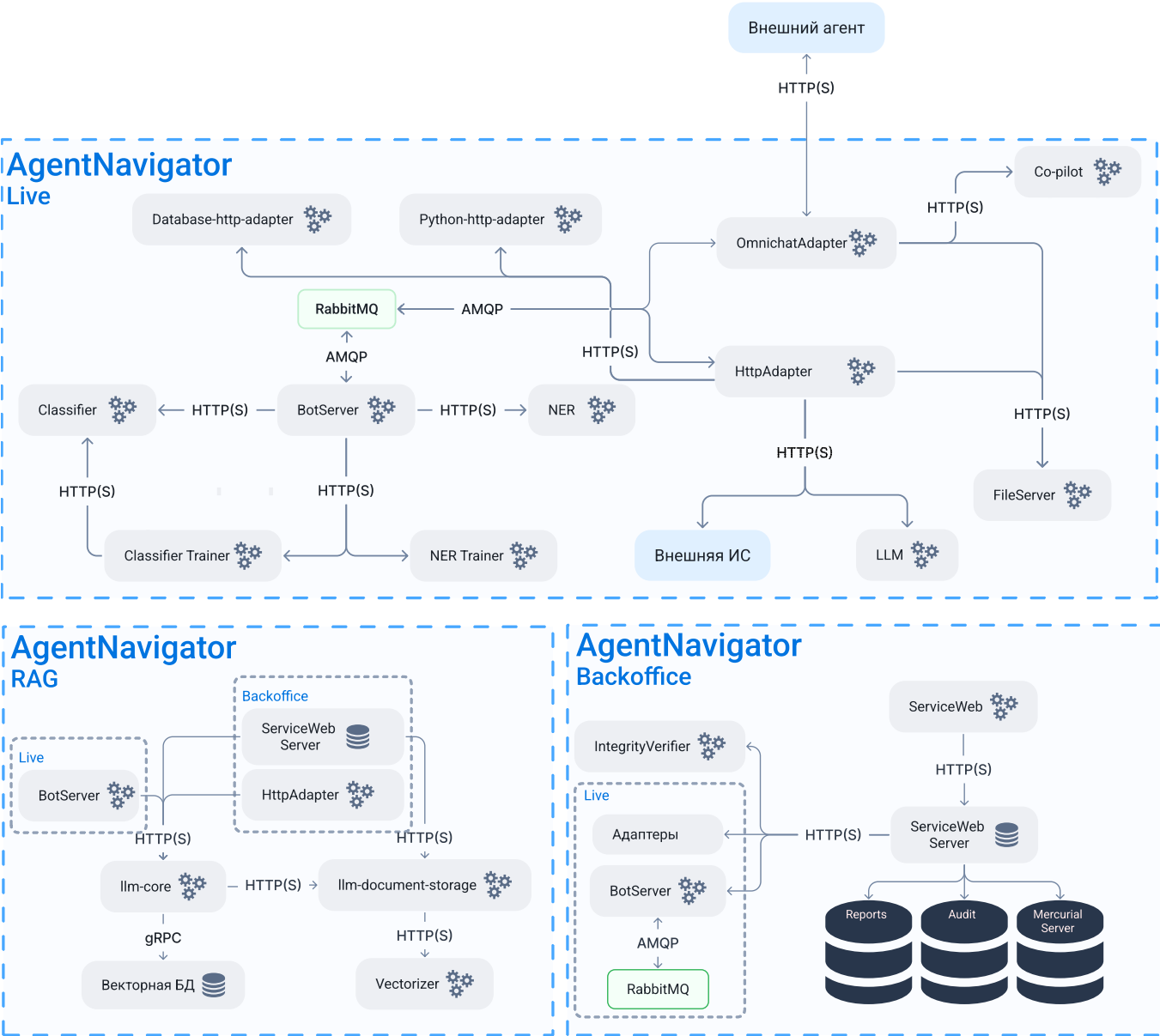


Рисунок 1 — Состав и схема взаимодействия компонентов AgentNavigator

Компонент	Назначение
Live	Часть платформы, отвечающая за обработку запросов пользователей.
BotServer	Компонент, обеспечивающий выполнение диалоговых сценариев. Управляет конфигурацией NLP-модулей (NLP — агрегирующий компонент, обеспечивающий классификацию и парсинг запросов на основе специальных паттернов).
HttpAdapter	Конфигурационный модуль, предоставляющий возможность обращаться к веб-сервисам из диалогового сценария.

Компонент	Назначение
Classifier	Компонент, обеспечивающий классификацию и парсинг запросов на основе машинного обучения.
Classifier Trainer	Компонент, обеспечивающий обучение классификатора Classifier .
NER	Компонент, обеспечивающий извлечение именованных сущностей на основе машинного обучения.
NER Trainer	Компонент, обеспечивающий обучение моделей для извлечения именованных сущностей.
Шина данных	Обеспечивает взаимодействие между внутренними модулями AgentNavigator по единому протоколу на базе RabbitMQ .
FileServer	Сервис, обеспечивающий обработку и хранение файлов из диалогов.
OmnichatAdapter	Адаптер, обеспечивающий взаимодействие с внешними чат-платформами.
SipAdapter	Адаптер, позволяющий обмениваться данными с VoiceKey.IVR SIP .
Python-http-adapter	Сервис, обеспечивающий возможность вызова и выполнения Python-скриптов из сценариев.
Database-http-adapter	Сервис, обеспечивающий возможность выполнения SQL-запросов из сценариев.
Co-pilot	Веб-приложение; пользовательский интерфейс для взаимодействия с AI-помощником в режиме диалога или в режиме навыков.
Backoffice	Часть платформы, отвечающая за хранение и обработку диалогов пользователя, мониторинг, развертывание диалогов и т. д.
IntegrityVerifier	Сервис контроля целостности компонентов платформы.
ServiceWeb	Веб-приложение, административный интерфейс для просмотра диалогов, построения отчетов, просмотра состояния компонентов, развертывания сборок сценариев и т. п.
	ServiceWeb Server HTTP/REST-сервер и веб-приложение.
	Reports База данных диалогов.
	Audit База данных аудит-логов.
	Mercurial Server Сервер, хранящий версионизируемые сценарии диалога.
RAG	Часть платформы, отвечающая за работу с базой знаний и поиск по ней.

Компонент	Назначение
Ilm-document-storage	Сервис, обеспечивающий хранение исходных документов и их метаданных в коллекциях (базе знаний).
Ilm-core	Сервис, обеспечивающий семантический поиск по векторизованным документам коллекций.
Vectorizer	Компонент, обеспечивающий взаимодействие с векторной моделью.
Векторная БД	Векторная база данных для хранения фрагментов документов и их векторов, используемых в семантическом поиске.

Платформа включает следующие подсистемы:

- [подсистема взаимодействия с клиентом](#);
- [подсистема обработки запросов клиентов](#);
- [подсистема интеграции](#);
- [подсистема хранения и аналитики](#);
- [подсистема проверки целостности](#).

3.1 Подсистема взаимодействия с клиентом ChatWidget

ChatWidget — компонент, встраиваемый на любую веб-страницу и предоставляющий функции для ведения диалога.

Может быть:

- встроенным виджетом;
- модулем API (синхронного или асинхронного), к которому подключены каналы заказчика;
- ботом для мессенджеров (Telegram).

Виджет, размещаемый на сайтах, предоставляет клиенту интерфейс для взаимодействия с базой знаний виртуального консультанта и ведения диалога.

Виджет обеспечивает:

- отображение персонажа виртуального консультанта;
- сбор сообщений, вводимых клиентом;
- отображение ответа подсистемы обработки запросов клиентов.

3.2 Подсистема обработки запросов клиентов BotServer

Подсистема обработки запросов клиентов **BotServer** предназначена для классификации и обработки запросов клиентов в соответствии с хранимыми сценариями диалогов и предоставления ответа для отображения через **ChatWidget**.

BotServer выполняет:

- обработку и классификацию запросов клиентов;

- формирование и предоставление ответа на запрос клиента;
- управление подсистемой взаимодействия с клиентом;
- управление подсистемой интеграции.

3.3 Подсистема интеграции HttpAdapter

Подсистема интеграции **HttpAdapter** предназначена для взаимодействия со смежными подсистемами, информация из которых используется во время консультации клиента.

HttpAdapter выполняет:

- получение и передачу данных в смежную систему;
- передачу ответа в подсистему обработки запросов клиентов.

3.4 Подсистема хранения и аналитики ServiceWeb

Подсистема хранения и аналитики предназначена для:

- хранения статистической информации по взаимодействию клиентов с платформой;
- хранения базы знаний и сценариев обработки диалогов;
- предоставления доступа к статистической информации;
- предоставления доступа к сохраненным диалогам с клиентами;
- предоставления доступа к разметке диалогов (сессий).

3.5 Подсистема проверки целостности IntegrityVerifier

Подсистема проверки целостности **IntegrityVerifier** анализирует компоненты на наличие повреждений.

3.6 Резервирование компонентов платформы

Схема компонентов платформы с резервированием представлена на рисунке 2.

В контуре обслуживания (верхний контур) обеспечивается горячее резервирование:

- **OmnichatAdapter**;
- **HttpAdapter**;
- **BotServer**;
- **PrompterAdapter**;
- **Шина данных**.

Расчет мощностей производился для нагрузки 15 запросов в секунду или 14000 диалогов в сутки.

Master-сервер по умолчанию установлен на площадке А, slave-сервер — на площадке В.

Slave в потоковом режиме копирует базу данных (БД) master.

В случае выхода master-сервера из строя, переключение на резервный сервер проводится в ручном режиме. Сервер площадки А отключается (если есть доступ), сервер площадки В переводится из slave в master. Сервер пользовательского интерфейса подключается к новому master автоматически.

В случае необходимости, возможно провести обратную замену master-slave, когда сервер площадки А будет восстановлен.

Емкость HDD на серверах виртуального консультанта (**Chat core**) рассчитана таким образом, что обеспечивается работа платформы без потери данных в течение трех суток. «Хвост» накопившихся сессий автоматически переносится в архивную БД, как только становится доступным сервер **PostgreSQL**.

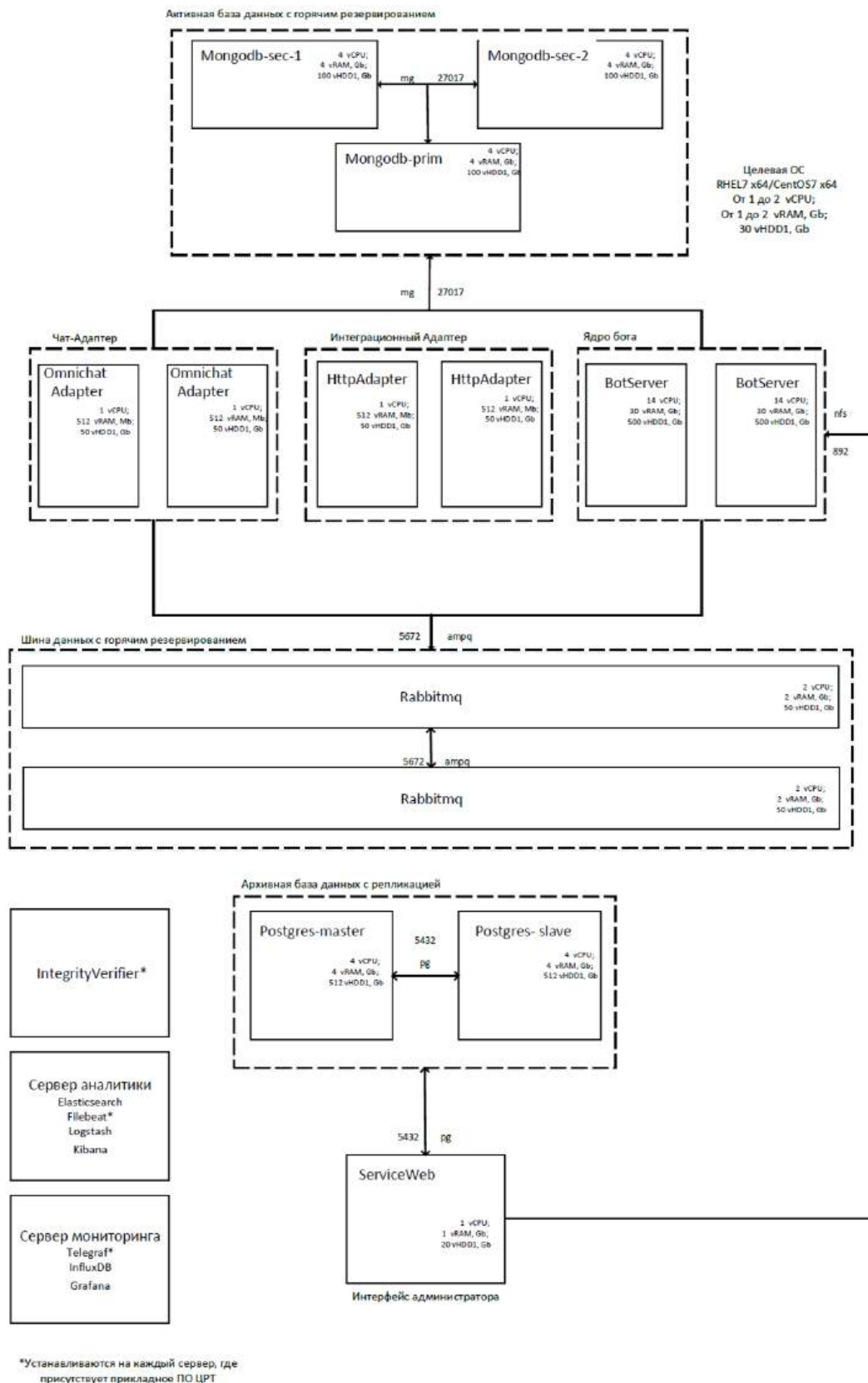


Рисунок 2 — Схема компонентов платформы AgentNavigator с резервированием

4 УСТАНОВКА И НАСТРОЙКА

4.1 Подключение локального RPM-репозитория

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер выполните только шаги 1–6.

1. Создайте каталог, в котором будет расположен репозиторий:

```
mkdir /localrepo
```

2. Зайдите в каталог:

```
cd /localrepo
```

3. В этот каталог скачайте архив с пререквизитами:

```
curl -O https://ftp.speechpro.com/puckr/vc/prerequisites/prerequisites.tgz
```

4. Чтобы распаковать архив, используйте команду:

```
tar -zxvf prerequisites.tgz
```

5. Выполните скрипт **1_create_repository.sh**.

6. Обновите кэш пакетного менеджера:

```
yum makecache --disablerepo=base,extras,updates
```



Шаги ниже необходимо дополнительно выполнить только при настройке на несколько компьютеров.

7. Установите **Nginx** и **Createrepo**:

```
yum install nginx createrepo -y
```

8. В каталог **/etc/nginx/** добавьте с заменой файл **nginx.conf** с содержимым:

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
include /usr/share/nginx/modules/*.conf;
events {
    worker_connections 1024;
}
http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';
    access_log /var/log/nginx/access.log main;
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    include /etc/nginx/conf.d/*.conf;
```

```
}
```

9. В каталог `/etc/nginx/conf.d/` добавьте файл **main.conf** с содержимым:

```
server {
    listen                80;
    server_name           $hostname;

    keepalive_timeout 30;
    keepalive_requests 1000;
    include /etc/nginx/default.d/*.conf;
}
```

10. В каталог `/etc/nginx/default.d/` добавьте файл **repo.conf** с содержимым:

```
location /vc {
    autoindex on;
    alias /localrepo;
}
```

11. Запустите **Nginx**:

```
systemctl start nginx
```

12. Добавьте **Nginx** в автозагрузку:

```
systemctl enable nginx
```

13. На каждом компьютере в составе платформы создайте в каталоге `/etc/yum.repos.d/` файл **chatbot.repo** с содержимым:

```
[chatbot]
name=Chatbot
baseurl=http://rephost/vc
enabled=1
gpgcheck=0
```

Параметр **rephost** — имя или IP-адрес компьютера, на котором развернут репозиторий по шагам, описанным выше.

14. На каждом компьютере в системе обновите кэш пакетного менеджера:

```
yum makecache --disablerepo=base,extras,updates
```

4.2 Установка пререквизитов и утилит

1. Для установки набора утилит, облегчающих управление сервером, выполните команду:

```
yum install yum-utils bash-completion vim tree zip unzip mc htop -y
```

2. Опционально, если используемый сервер является виртуальным и развернут в среде **VMware**, установите пакет с гостевыми дополнениями **open-vm-tools**:

```
yum install open-vm-tools -y
```

3. На сервере, выделенном под архивную базу данных, выполните:

```
yum install postgresql14 postgresql14-libs postgresql14-server -y
```

4. На сервере, выделенном под активную базу данных, выполните:

```
yum install mongodb-org -y
```

5. На сервере, выделенном под шину, выполните:

```
yum install rabbitmq-server -y
```

6. На каждом сервере, где будет установлен хотя бы один компонент платформы, установите **JDK**, например:

```
yum install java-21-openjdk-headless -y
```

7. На каждом сервере, где будет установлен один из следующих компонентов: **ServiceWeb**, **OmnichatAdapter**, **ChatWidget**, установите **Nginx**:

```
yum install nginx -y
```

8. На каждом сервере, где будет установлен **BotServer**, установите драйвер для HASP-ключа с помощью одной из следующих команд в зависимости от используемого ключа защиты:

Для **Sentinel HASP**:

```
yum install aksusbd -y
```

Для **Guardant**:

```
yum install grdcontrol-<version>-x86-64.rpm -y
```

9. На сервере, где будет установлен **ServiceWeb**, установите **Mercurial**:

```
yum install mercurial-4.9-1 -y
```

4.3 Настройка и запуск окружения

4.3.1 RabbitMQ

1. Добавьте файл `/etc/rabbitmq/rabbitmq.config` с содержимым:

```
[{rabbit, [
{loopback_users, []}
]}].
```

2. Добавьте программу в автозагрузку:

```
systemctl enable rabbitmq-server.service
```

3. Запустите программу:

```
systemctl start rabbitmq-server.service
```

4.3.1.1 Общая конфигурация RabbitMQ



Действия необходимы только в случае установки на несколько компьютеров.

Общая конфигурация **RabbitMQ** проводится для следующих компонентов платформы **AgentNavigator**: **BotServer**, **HttpAdapter**, **ChatWidget**, **OmnichatAdapter**, **SipAdapter**, **Telegram**, **FileServer**.

```
rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
```

```

password: guest
use-ssl: false #optional
authentication: plain #optional. external - для включения аутентификации по сертификату.
См. https://www.rabbitmq.com/access-control.html#mechanisms,
https://github.com/rabbitmq/rabbitmq-server/tree/main/deps/rabbitmq_auth_mechanism_ssl
thread-count: 1
prefetch: 4
requested-heartbeat: 60 #optional
network-recovery-interval-ms: 1000 #optional
publish-channel-count: 1 #optional. Количество каналов используемых для публикации.
https://www.rabbitmq.com/channels.html
consume-auto-ack: false #optional. true - сообщения автоматически подтверждаются в mq.
handshake-timeout: 20000 #optional. См. https://www.rabbitmq.com/networking.html
connection-timeout: 60000 #optional. TCP connection timeout.
delay-between-reconnections-ms: 5000 #optional. время задержки между попытками
переподключения
max-number-of-reconnection: 10 #optional. число попыток переподключиться
level: WARN #optional. javaх валидация сообщения. ERROR - отклонять сообщения с
ошибками.
message-ttl-ms: 1200000 #optional. TTL сообщения в во временной очереди.
https://www.rabbitmq.com/ttl.html
queue-ttl-ms: 86400000 #optional. TTL временной очереди.
https://www.rabbitmq.com/ttl.html
certificate: #optional: конфигурация для настройки tls
enable-hostname-verification: false #optional: позволяющий включить верификацию хоста
при отправки запроса, используется в mTLS
keystore: #optional: хранилище ключей
path: ./httpadapter/src/test/resources/certificate/client01.p12 #путь к хранилищу
keystore
password: '6-byrd'#пароль для хранилища и сертификатов
keys-password: '6-byrd'#optional: является паролем для сертификата. По умолчанию
равен password.
format: PKCS12 #формат, возможные значения: PKCS12, JKS
aliases: [1] #optional: предназначен для задания названия alias для сертификата
truststore: #optional: хранилище доверенных сертификатов
path: ./httpadapter/src/test/resources/certificate/truststore.jks #путь к хранилищу
truststore
password: '6-byrd' #optional: пароль для хранилища. "" по умолчанию.
format: JKS #формат, возможные значения: PKCS12, JKS
aliases: [1] #optional: предназначен для задания названия alias для сертификата

```

Для включения аутентификации с использованием сертификата требуется:

- включение на стороне клиента: **application.yml: rabbitmq.authentication: external;**
- включение на стороне сервера: **rabbitmq.conf:**
 - **auth_mechanisms.1 = EXTERNAL;**
 - определение пользователя по AgentNavigator: **ssl_cert_login_from = common_name.**
- настройка сертификатов на клиенте ([certificate](#)) и сервере.

4.3.1.2 Аутентификация Rabbit с помощью TLS



Для использования аутентификации с помощью TLS необходима версия **RabbitMQ-Server** не ниже **3.7.0.**

Выполните установку репозитория:

```
yum -y install epel-release
```

Скачайте репозиторий:

```
wget http://packages.erlang-solutions.com/erlang-solutions-1.0-1.noarch.rpm
```

Добавьте репозиторий:

```
sudo rpm -Uvh erlang-solutions-1.0-1.noarch.rpm
```

Установите **Erlang** и зависимости:

```
sudo yum -y install erlang socat logrotate
```

Скачайте **RabbitMQ**: <https://github.com/rabbitmq/rabbitmq-server/releases/download/v3.8.8/rabbitmq-server-3.8.8-1.el6.noarch.rpm>.

Добавьте ключ:

```
sudo rpm --import https://www.rabbitmq.com/rabbitmq-signing-key-public.asc
```

Добавьте ключ с помощью прокси-сервера:

```
sudo rpm --import --httpproxy https://your-proxy.example:2022  
https://www.rabbitmq.com/rabbitmq-signing-key-public.asc
```

Установите **RabbitMQ-server**:

```
sudo rpm -Uvh rabbitmq-server-3.8.8-1.el6.noarch.rpm
```

Выполните перезапуск **Rabbit**, создайте пользователя и пароль, активируйте дополнительные плагины для администрирования с помощью UI-интерфейса.

Включите консоль управления **RabbitMQ-web**:

```
sudo rabbitmq-plugins enable rabbitmq_management
```

Если **RabbitMQ** запущен на отдельном компьютере, подключаться к нему нужно извне.



Пользователь по умолчанию **guest/guest** доступен только с локального компьютера.

Создайте пользователя:

```
sudo rabbitmqctl add_user test test  
sudo rabbitmqctl set_user_tags test administrator
```

Установите разрешения для пользователя:

```
sudo rabbitmqctl set_permissions -p / test ".*" ".*" ".*"
```

4.3.1.3 Генерация сертификатов для RabbitMQ

Установите необходимые зависимости.

Установите **Erlang** и зависимости:

```
yum install python36 python36-devel python36-pip
```



Сертификат генерируется без пароля.

Сгенерируйте серверный и клиентский сертификаты:

```
git clone https://github.com/michaelklishin/tls-gen tls-gen  
cd tls-gen/basic  
make  
make verify
```

```
make info
ls -l ./result
```

Команда выше сгенерирует серверный (**server_**) и клиентский (**client_**) сертификаты.

 Разрешения для файлов-сертификатов (пользователь **RabbitMQ** должен иметь возможность их чтения) можно, например, переместить в каталог **/etc/rabbitmq/**.

Проверьте работоспособность сертификатов можно двумя командами.

Из папки со сгенерированными сертификатами запустите сервер на свободном порту (в примере ниже используется порт **8443**).

openssl server:

```
openssl s_server -accept 8443 -cert server_certificate.pem -key server_key.pem -CAfile
ca_certificate.pem
```

Подключитесь к серверу в другом терминальном окне. Если подключение произошло, в лог-файле отобразится информация о сертификате.

openssl server:

```
openssl s_client -connect localhost:8443
```

Создайте конфигурационный файл в **RabbitMQ**:

```
vi /etc/rabbitmq/rabbitmq.config
```

Конфигурация для двух листенеров (не SSL и SSL), см ниже.

rabbitmq.config

```
[
  {rabbit, [
    {tcp_listeners, []}                                # Если нужно отключить
    стандартный (незащищенный) порт
    {ssl_listeners, [5671]},
    {ssl_options, [{cacertfile, "/root/tls-gen/basic/result/ca_certificate.pem"},
                  {certfile, "/root/tls-
gen/basic/result/server_certificate.pem"},
                  {keyfile, "/root/tls-gen/basic/result/server_key.pem"},
                  {fail_if_no_peer_cert, false},
                  {verify, verify_none}                # Опция отключает обязательное
    предоставление клиентом клиентского сертификата
                  ]}
    ]}
]
```

При использовании межсетевого экрана добавьте порты в правила:

```
rabbitmq-diagnostics listeners
Interface: [::], port: 15672, protocol: http, purpose: HTTP API
Interface: [::], port: 25672, protocol: clustering, purpose: inter-node and CLI tool
communication
Interface: [::], port: 5672, protocol: amqp, purpose: AMQP 0-9-1 and AMQP 1.0
# Обычное соединение
Interface: [::], port: 5671, protocol: amqp/ssl, purpose: AMQP 0-9-1 and AMQP 1.0 over
TLS # SSL-соединение
```

Перезапустите **RabbitMQ** и проверьте доступность SSL с помощью команды, после выполнения которой произойдет подключение и отобразится SSL-информация (конфигурация, сертификат и т. п.):

если не планируется использовать клиентский сертификат:

```
openssl s_client -connect <host>:5671
```

если у клиента планируется использовать сертификат:

```
openssl s_client -connect <host>:5671 -cert client_certificate.pem -key client_key.pem -CAfile ca_certificate.pem
```



С помощью **openssl s_client** проверьте доступ как на локальном компьютере, так и на удаленном (если установка **RabbitMQ** планируется на отдельном компьютере).

Подключение сервисов к **RabbitMQ** по SSL:

```
rabbitmq:
  host: localhost          # Хост
  port: 5672               # Выставить порт для SSL-коннектора
  user: guest              # Опционально
  password: guest          # Опционально
  useSsl: true             # Выставить опцию в true
```

При старте сервиса должна отобразиться информация об успешном подключении к **RabbitMQ** по SSL-порту.

4.3.2 MongoDB

1. Отредактируйте конфигурационный файл */etc/mongod.conf*:

```
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log

storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true

processManagement:
  fork: true
  pidFilePath: /var/run/mongodb/mongod.pid

net:
  port: 27017
  bindIp: externalip
```

Параметр **externalip** — IP-адрес интерфейса в сети системы.

2. Запустите программу:

```
systemctl start mongod.service
```

По умолчанию **MongoDB** находится в автозагрузке и запуск вручную не требуется

4.3.2.1 Создание пользователей с аутентификацией по логину и паролю

Запустите консоль **mongo**, создайте суперпользователя и пользователя **mongoadmin** (сервисная учетная запись) в БД **admin**:

```
use admin
db.createUser(
  {
    user: "mongoadminadmin",
    pwd: "mongoadminpass",
    roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]
  }
)
db.createUser(
  {
    user: "mongo-root",
    pwd: "passwd",
    roles: [ { role: "root", db: "admin" } ]
  }
)
```



MongoDB используется в следующих сервисах: **BotServer**, **HttpAdapter**, **ChatWidget**, **OmnichatAdapter**, **SipAdapter**, **CrafttalkAdapter**, **Telegram**, **FileServer**, **Classifier** и **Classifier Trainer**.

Создайте пользователей для всех необходимых БД в **mongo** по аналогии с БД **BotServer**:

```
use botserver
db.createUser(
  {
    user: "user1",
    pwd: "pass1",
    roles: [ { role: "readWrite", db: "botserver" } ]
  }
)
```

Перезапустите **mongo**:

```
systemctl restart mongod
```

В конфигурационных файлах компонентов добавьте строки с пользователем и паролем, переопределив конфигурацию в каталоге **/etc/speechpro/botserver**:

```
mongodb:
  name: botserver
  username: user1
  password: pass1
  hosts:
    - localhost
```

Перезапустите **BotServer**.

4.3.2.2 Настройка авторизации

Создание привилегированных пользователей

Подключитесь к консоли **mongo** (без авторизации):

```
mongo
```

Включение авторизации в конфигурации MongoDB

Добавьте в **/etc/mongod.conf** следующий блок:

```
security:
  authorization: enabled
```

Перезапустите сервис **Mongo**:

```
systemctl restart mongod.service
```

Включение авторизации в конфигурации классификаторов

В */opt/speechpro/classifier/conf/application.yaml* измените (Сервис Исполнения):

```
mongodb:
  auth:
    enabled: true
```

В */opt/speechpro/classifier_trainer/conf/application.yaml* измените (Сервис Обучения):

```
mongodb:
  auth:
    enabled: true
```

Проверка соответствия данных пользователя в MongoDB с конфигурацией сервисов классификатора

❗ Убедитесь, что **username**, **password**, **db** пользователя, созданного на предыдущем шаге для доступа к базе данных **classifier**, совпадают с параметрами **username**, **password**, **authSource** из блока **mongodb** в конфигурации сервисов классификатора.

Проверьте корректность авторизационных данных пользователя **mongodb** в */opt/speechpro/classifier/conf/application.yaml* (Сервис Исполнения):

```
mongodb:
  auth:
    enabled: true
    username: "user1"
    password: "pass1"
    authSource: "classifier"
    authMechanism: "SCRAM-SHA-1"
```

Проверьте корректность авторизационных данных пользователя **mongodb** в */opt/speechpro/classifier_trainer/conf/application.yaml* (Сервис Обучения):

```
mongodb:
  auth:
    enabled: true
    username: "user1"
    password: "pass1"
    authSource: "classifier"
    authMechanism: "SCRAM-SHA-1"
```

Описание конфигурации Сервиса Исполнения

Пример конфигурационного файла Сервиса Исполнения */opt/speechpro/classifier/conf/application.yaml*:

```
unicorn: # Настройки запуска веб-сервера unicorn
  host: 0.0.0.0 # Хост, на котором запущен сервер
  port: 9008 # Порт, на котором запущен сервер
  workers: 1 # Количество "рабочих" процессов (всегда выставить 1)
  threads: 8 # Количество потоков, в которых работает сервер (обычно, 2*<количество ядер процессора сервера>)
  keepAlive: 30 # Время ожидания запросом ответа от сервера в секундах
```

```
timeout: 60 # Время в секундах, через которое перезагрузится "рабочий" процесс в случае долгого выполнения
```

```
mongodb: # Настройки для подключения к базе данных mongodb
  host: "127.0.0.1:27017" # Хост и порт для подключения к mongodb
  socketTimeoutMS: null
  connectTimeoutMS: 20000
  waitQueueTimeoutMS: null
  waitQueueMultiple: null
  connect: True
  serverSelectionTimeoutMS: 12000
  minPoolSize: 0
  maxPoolSize: 100
  maxIdleTimeMS: null
  auth: # Параметры для авторизации в mongodb
    enabled: false # Вкл/выкл авторизация в mongodb
    username: "user1" # Реквизиты пользователя для авторизации в mongodb
    password: "pass1" # Реквизиты пользователя для авторизации в mongodb
    authSource: "classifier" # Имя базы данных, к которой будет выполнено защищенное подключение
    authMechanism: "SCRAM-SHA-1" # Механизм шифрования паролей в mongodb

app: # Настройки для работы сервиса Исполнения
  data: # Настройки хранения и экспирации данных обученных моделей/комплектов моделей
    path: /opt/speechpro/classifier_models/ # Абсолютный путь до папки с обученными моделями/комплектами моделей
    revision:
      enabled: True # Загрузка комплектов моделей включена/выключена (True/False)
  ram:
    limit:
      entityCount: 10 # Максимальное количество моделей/комплектов моделей, одновременно загруженных в RAM и используемых для классификации
    expiration:
      checkIntervalSec: 60 # Время жизни модели в RAM
      removeIfUnusedAfterSec: 7200 # 2 hours, период запуска службы по выгрузке моделей из RAM
  algorithm: # Конфигурируемые параметры алгоритмов классификации, использующихся в сервисе
    param:
      confidence: 0.0 # Значение по умолчанию для confidence, по которому фильтруются тематики
    result:
      nBestVariants: 0 # Возвращаемое количество тематик
```

Изменяемая конфигурация */etc/speechpro/classifier/application.yaml*:

```
app:
  data:
    path: /opt/speechpro/classifier_models/
    revision:
      enabled: True
  ram:
    limit:
      entityCount: 10 # maximum 10 revisions | models loaded in ram
    expiration:
      checkIntervalSec: 60
      removeIfUnusedAfterSec: 7200 # 2 hours
  algorithm:
    param:
      confidence: 0.0
    result:
      nBestVariants: 0
```

Все поля в `/etc/speechpro/classifier/application.yml` необязательные, но сам файл по структуре должен соответствовать файлу стандартной конфигурации.

Сервер при запуске заменяет стандартные значения на значения из `/etc/speechpro/classifier/application.yml`.

Описание конфигурации Сервиса Обучения

Пример конфигурационного файла Сервиса Обучения `/opt/speechpro/classifier_trainer/conf/application.yml`:

```
unicorn: # Настройки запуска веб-сервера
  host: 0.0.0.0 # Хост, на котором запущен сервер
  port: 9009 # Порт, на котором запущен сервер
  workers: 1 # Количество "рабочих" процессов (всегда выставлять 1)
  threads: 8 # Количество потоков, в которых работает сервер (обычно, 2*<количество ядер процессора сервера>)
  keepAlive: 30 # Время ожидания запросом ответа от сервера, в секундах
  timeout: 60 # Время, через которое перезагрузится "рабочий" процесс в случае долгого выполнения, в секундах

mongodb: # Настройки для подключения к базе данных mongodb
  host: "127.0.0.1:27017" # Хост и порт для подключения к mongodb
  socketTimeoutMS: null
  connectTimeoutMS: 20000
  waitQueueTimeoutMS: null
  waitQueueMultiple: null
  connect: True
  serverSelectionTimeoutMS: 12000
  minPoolSize: 0
  maxPoolSize: 100
  maxIdleTimeMS: null
  auth: # Параметры для авторизации в mongodb
    enabled: false # Вкл/выкл авторизация в mongodb
    username: "user1" # Реквизиты пользователя для авторизации в mongodb
    password: "pass1" # Реквизиты пользователя для авторизации в mongodb
    authSource: "classifier" # Имя базы данных, к которой будет выполнено защищенное подключение
    authMechanism: "SCRAM-SHA-1" # Механизм шифрования паролей в mongodb

app: # Настройки для работы сервиса Обучения
  data: # Настройки хранения и экспирации данных обученных моделей/комплектов моделей
    path: /opt/speechpro/classifier_models/ # Абсолютный путь до папки с обученными моделями/комплектами моделей
    revision:
      enabled: True # Загрузка комплектов моделей вкл/выкл (True/False)
  hdd:
    maxUsedSpacePercent: 80 # Лимит на загрузку HDD, при достижении которого с HDD удаляется одна из устаревших моделей/комплектов при обучении новой модели/комплекта
    expiration:
      checkIntervalSec: 86400 # 1 day, время жизни модели на HDD
      removeIfUnusedAfterSec: 1209600 # 14 days, период запуска службы по выгрузке моделей с HDD
  algorithm: # Конфигурируемые параметры алгоритмов обучения, использующихся в сервисе
    cnn:
      enabled: True # Вкл/выкл алгоритм (True/False)
      avgExamplesPerClass: 50 # Метрика, при достижении которой происходит обучение модели алгоритмом CNN (среднее количество примеров на класс)
      cnnSubword:
        enabled: True # Вкл/выкл алгоритм (True/False)
        avgExamplesPerClass: 50 # Метрика, при достижении которой происходит обучение модели алгоритмом CNNSubword (среднее количество примеров на класс)
    knn:
```

```

    enabled: False # Вкл/выкл алгоритм (True/False)
  svm:
    enabled: True # Вкл/выкл алгоритм (True/False)
  component:
    classifier:
      healthcheck: # Настройки запроса поднятия модели в RAM
        enabled: True # Вкл/выкл функционал
        timeout: 10
        nodes: # Список хостов запущенных сервисов Исполнения
          - http://127.0.0.1:9008/restapi/classifier/check
    singleThreadOfTrain:
      enabled: True # Выбор режима последовательной загрузки процессора функциями сервиса
train

```

Изменяемая конфигурация */etc/speechpro/classifier_trainer/application.yaml*:

```

app:
  data:
    path: /opt/speechpro/classifier_models/
    revision:
      enabled: True
  hdd:
    maxUsedSpacePercent: 80
    expiration:
      checkIntervalSec: 86400 # 1 day
      removeIfUnusedAfterSec: 1209600 # 14 days
  algorithm:
    cnn:
      enabled: True
      avgExamplesPerClass: 50
    cnnSubword:
      enabled: True
      avgExamplesPerClass: 50
    knn:
      enabled: False
    svm:
      enabled: True
  component:
    classifier:
      healthcheck:
        enabled: True
        timeout: 10
        nodes:
          - http://127.0.0.1:9008/restapi/classifier/check
    singleThreadOfTrain:
      enabled: True

```

Все поля в */etc/speechpro/classifier_trainer/application.yaml* необязательные, но сам файл по структуре должен соответствовать файлу стандартной конфигурации.

Сервер при запуске заменяет стандартные значения на значения из */etc/speechpro/classifier_trainer/application.yaml*.

4.3.3 PostgreSQL

1. Смените текущего пользователя на пользователя **postgres**:

```
su - postgres
```

2. Инициализируйте базу:

```
/usr/pgsql-14/bin/initdb
```

3. Перечислите в файле `/var/lib/pgsql/14/data/pg_hba.conf` IP-адреса компьютеров, где установлены **ServiceWeb** или **IntegrityVerifier**, чтобы разрешить им подключение к базе данных:

```
local    all             all                                     trust
host     all             all             127.0.0.1/32          trust
host     all             all             10.0.0.5/32           md5
host     all             all             10.0.0.6/32           md5
```

10.0.0.5 и **10.0.0.6** — предполагаемые адреса искомых хостов (приведены для примера).

4. В файле `/var/lib/pgsql/14/data/postgresql.conf` раскомментируйте и приведите к виду строку:

```
listen_addresses = '*'
```

5. Смените пользователя на прежнего, нажав **Ctrl+D**, и запустите **PostgreSQL**:

```
systemctl start postgresql-14.service
```

6. Добавьте **PostgreSQL** в автозагрузку:

```
systemctl enable postgresql-14.service
```

7. Запустите консоль **PostgreSQL** от имени пользователя **postgres**:

```
sudo -u postgres psql
```

8. Создайте пользователя **reports** и одноименную базу, последовательно выполнив команды:

```
CREATE ROLE reports WITH PASSWORD 'reports' LOGIN;
CREATE DATABASE reports WITH OWNER = reports ENCODING = 'UTF-8';
\c reports
CREATE SCHEMA functions;
GRANT ALL PRIVILEGES ON SCHEMA functions TO reports;
```

9. Создайте пользователя **audit** и одноименную базу, последовательно выполнив команды:

```
CREATE ROLE audit WITH PASSWORD 'audit' LOGIN;
CREATE DATABASE audit WITH OWNER = audit ENCODING = 'UTF-8';
\c audit
CREATE SCHEMA functions;
GRANT ALL PRIVILEGES ON SCHEMA functions TO audit;
```

10. Создайте пользователя **integrityverifier** и одноименную базу, последовательно выполнив команды:

```
CREATE ROLE integrityverifier WITH PASSWORD 'integrityverifier' LOGIN;
CREATE DATABASE integrityverifier WITH OWNER = integrityverifier ENCODING = 'UTF-8';
```

11. Выйдите из **PostgreSQL**, нажав комбинацию клавиш **CTRL+D**, чтобы вид консоли сменился с «postgres=#» на «[<username>@<hostname><directory>]#», например «[root@cn-test19 lib]#».

4.3.4 CORS

```
cors:
  enabled: true # Включение поддержки CORS-запросов
  origins: '*' # Звездочку необходимо заключить в кавычки, т. к. компонент иначе не может
               # понять, что это строка. Также можно указать список origin'ов через запятую
               # (http://origin1, http://origin2, etc.)
  methods: '*' # Можно указать звездочку или конкретный список методов, например, GET,
               # POST, PUT
  headers: '*' # Опциональный
```

4.3.5 FileServer-клиент

```
fileServer:
  host: localhost # Хост, на котором развернут компонент FileServer
  port: 9320 # Порт, на котором развернут компонент FileServer
  timeout: 60000 # тайм-аут, в миллисекундах
```

4.3.6 Nginx

1. В каталог `/etc/nginx/` добавьте с заменой файл `nginx.conf` со следующим содержимым:

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
include /usr/share/nginx/modules/*.conf;
events {
    worker_connections 1024;
}
http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                   '$status $body_bytes_sent "$http_referer" '
                   '"$http_user_agent" "$http_x_forwarded_for"';
    access_log /var/log/nginx/access.log main;
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    include /etc/nginx/mime.types;
    default_type application/octet-stream;
    include /etc/nginx/conf.d/*.conf;
}
```

2. В каталог `/etc/nginx/conf.d/` добавьте файл `main.conf` со следующим содержимым:

```
server {
    listen 80;
    server_name $hostname;

    keepalive_timeout 30;
    keepalive_requests 1000;
    include /etc/nginx/default.d/*.conf;
}
```

3. В каталог `/etc/nginx/conf.d/` добавьте файл `upstreams.conf` со следующим содержимым:

```
upstream chatwidget_upstream {
    keepalive 1000;
    server localhost:9999;
}

upstream serviceweb_upstream {
    server localhost:8086;
}

upstream omnichatadapter_upstream {
    server localhost:9066;
}
```

4. На компьютере, где установлен **ServiceWeb**, в каталог `/etc/nginx/default.d/` добавьте файл `serviceweb.conf` со следующим содержимым:

```
location / {
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_pass           http://serviceweb_upstream;
    proxy_http_version  1.1;
    proxy_set_header    Connection "";
    proxy_connect_timeout      300;
    proxy_send_timeout         300;
    proxy_read_timeout         300;
    send_timeout               300;
}
```

5. На компьютере, где установлен **ChatWidget**, в каталог `/etc/nginx/default.d/` добавьте файл **chatwidget.conf** со следующим содержимым:

```
location /web-chat {
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_pass           http://chatwidget_upstream;
    proxy_http_version  1.1;
    proxy_set_header    Connection "";
}

location ~ /restapi/siteconfigs(.*) {
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_pass           http://chatwidget_upstream;
    proxy_http_version  1.1;
    proxy_set_header    Connection "";
}
```

6. На компьютере, где установлен **OmnichatAdapter**, в каталог `/etc/nginx/default.d/` добавьте файл **omnichatadapter.conf** со следующим содержимым:

```
location ~ /restapi/channels/omnichat-adapter(.*) {
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_pass           http://omnichatadapter_upstream;
    proxy_http_version  1.1;
    proxy_set_header    Connection "";
}

location ~ /v2/restapi/channels/omnichat-adapter(.*) {
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_pass           http://omnichatadapter_upstream;
    proxy_http_version  1.1;
    proxy_set_header    Connection "";
}
```

7. В каталог `/etc/nginx/default.d/` добавьте файл **sentinel.conf** со следующим содержимым:

```
location /_int_ {
    proxy_set_header    Host $host;
    proxy_set_header    X-Real-IP $remote_addr;
```

```
proxy_pass          http://localhost:1947;
proxy_http_version  1.1;
}
```

8. Добавьте программу в автозагрузку:

```
systemctl enable nginx.service
```

9. Запустите программу:

```
systemctl start nginx.service
```

4.3.6.1 Замена сертификатов для Nginx

После выпуска сертификатов для сервера произведите замену сертификатов в конфигурации сервиса **Nginx** и каталоге хранения сертификатов и приватного ключа:

```
server {
    listen      443 ssl;
    server_name domain;

    keepalive_timeout 30;
    keepalive_requests 1000;

    ssl_certificate "/etc/pki/nginx/cert.crt";
    ssl_certificate_key "/etc/pki/nginx/key.key";
    ssl_session_cache shared:SSL:1m;
    ssl_session_timeout 10m;
    ssl_ciphers HIGH:!aNULL:!MD5;
    ssl_prefer_server_ciphers on;
}
```

4.3.7 HASP Driver

Чтобы разрешить доступ к веб-интерфейсу драйвера с других хостов необходимо создать файл **/etc/hasplm/hasplm.ini** со следующим содержанием:

```
[SERVER]
ACCremote = 1
```

Веб-интерфейс драйвера будет доступен по URI **http://hostname/_int_**, где **hostname** — имя или IP-адрес сервера, где был установлен драйвер.

4.3.8 HG server

Установка проводится на компьютере, где был установлен **ServiceWeb**.

1. Создайте каталоги для работы редактора:

```
sudo -u serviceweb mkdir /etc/speechpro/serviceweb/{hg-remote,scenario}
```

2. Проинициализируйте mercurial-репозиторий:

```
(cd /etc/speechpro/serviceweb/hg-remote/ && hg init)
```

3. В каталог **/etc/speechpro/serviceweb/hg-remote/hg/** добавьте файл **hgrc** со следующим содержанием:

```
[web]
allow_push = *
push_ssl = false
```

4. В каталог **/etc/systemd/system/** добавьте файл **hg-server.service** со следующим содержанием:

```
[Unit]
Description=Remote hg server
[Service]
Type=simple
PIDFile=/var/run/hg-server.pid
WorkingDirectory=/etc/speechpro/serviceweb/hg-remote/
ExecStart=/usr/bin/hg serve
Restart=always
[Install]
WantedBy=multi-user.target
```

5. Выполните перезагрузку unit-файлов **systemd**, выполнив команду:

```
systemctl daemon-reload
```

6. Добавьте программу в автозагрузку:

```
systemctl enable hg-server.service
```

7. Запустите программу:

```
systemctl start hg-server.service
```

4.3.9 NFS-сервер

NFS-сервер настраивается на каждом сервере, где будет установлен **BotServer**, кроме того сервера, где установлен и **ServiceWeb**. В этом случае все данные передаются в пределах локальной файловой системы.

NFS-сервер публикует лог-файлы **BotServer** в **ServiceWeb** для последующего анализа и импорта в базу данных.

1. Установите **nfs-utils**:

```
yum install nfs-utils -y
```

2. В файл **/etc/exports** добавьте строку вида:

```
/var/log/speechpro/botserver          host3(ro,sync,no_subtree_check)
```

Параметр **host3** — имя или IP-адрес сервера с установленным **ServiceWeb**.

3. Запустите программу:

```
systemctl start nfs-server.service
```

4.3.10 NFS-клиент

NFS-клиент настраивается на сервере, где будет установлен **ServiceWeb**, кроме случая, когда на том же сервере установлен и **BotServer**. В этом случае все данные передаются в пределах локальной файловой системы. NFS-клиент получает лог-файлы **BotServer** и передает в **ServiceWeb** для последующего анализа и импорта в базу данных.

1. Установите **nfs-utils**:

```
yum install nfs-utils -y
```

2. Создайте каталоги для монтирования NFS-каталогов по количеству серверов с установленным **BotServer**:

```
mkdir -p /var/log/speechpro/botserver-{host1,host2}/
```

Параметры **host1** и **host2** — имена серверов с установленным **BotServer**.

3. В файл **/etc/fstab** добавьте строки вида:

```
host1:/var/log/speechpro/botserver/ /var/log/speechpro/botserver-host1/ nfs
ro, sync, soft, retrans=10 0 0
host2:/var/log/speechpro/botserver/ /var/log/speechpro/botserver-host2/ nfs
ro, sync, soft, retrans=10 0 0
```

Параметры **host1** и **host2** — имена серверов с установленным **BotServer**.

4. Смонтируйте явно (целевые NFS-сервера уже должны быть запущены ранее):

```
mount -a
```

4.3.11 Замена паролей сервисных учетных записей

Измените пароли учетных записей в **PostgreSQL**, **MongoDB**, **RabbitMQ**, согласно инструкции ниже.

1. Для смены паролей в базе данных **PostgreSQL** выполните команды в консоли на сервере с БД.

Войдите под пользователем **postgres**:

```
psql -U postgres
```

Команда для смены пароля пользователя в базе данных:

```
ALTER USER username WITH PASSWORD 'newpassword';
```

Смена пароля для пользователя **reports**:

```
ALTER USER reports WITH PASSWORD 'reportssecure';
```

Подключение можно проверить с любого клиента **psql** с помощью команды и ввести пароль при запросе на подключение:

```
psql -U reports -h hostname.domain.*
```

Аналогично смена паролей для других пользователей в базе данных **PostgreSQL**.

2. Для смены паролей в базе данных **MongoDB** не старше версии **4.0** выполните команды на сервере с БД.



Если для базы данных **MongoDB** не созданы пользователи, необходимо предварительно создать пользователей и задать пароли. Создание пользователей в базе данных **MongoDB** описано в подразделе [Создание пользователей с аутентификацией по логину и паролю](#).

Выполните команду на сервере с БД **mongo**:

```
mongo
```

Перейдите в нужную базу:

```
use namedatabase
```

Смените пароль пользователя:

```
db.updateUser("user", {pwd: "newpassword" })
```

3. Для смены паролей в **RabbitMQ** выполните следующие команды.

Доступные пользователи в **RabbitMQ**:

```
rabbitmqctl list_users
```

Смена пароля в консоли на сервере с установленным **RabbitMQ**:

```
rabbitmqctl change_password user password
```

4.4 Установка системы AgentNavigator

Установка **AgentNavigator** идентична для всех операционных систем на основе **Red Hat**.

4.4.1 Обновление локального RPM-репозитория

1. Создайте каталог **/localrepo/vc**:

```
mkdir /localrepo/vc
```

2. Перенесите в созданный каталог пакеты компонентов из дистрибутива.

3. Обновите базу данных репозитория:

```
createrepo --update /localrepo/
```

4. Обновите кэш пакетного менеджера:

```
yum makecache
```

4.4.2 Установка обязательных компонентов

Установите минимально необходимый для корректной работы **AgentNavigator** набор компонентов:

```
yum install botserver chatwidget httpadapter serviceweb integrityverifier omnichatadapter  
-y
```

Для установки в другие директории необходимо использовать ключ **--relocate**. Например, для переопределения **opt** на **customOpt** укажите ключ **--relocate /opt=/customOpt**. Ниже приведена команда для установки компонентов в переопределенные директории.

```
rpm -iv /path/service.rpm --relocate /opt=/app --relocate /etc=/app/etc --  
relocate /var/log=/app/log
```

Компонент **BotServer** обрабатывает реплики клиента в соответствии с загруженным в него сценарием.

Компонент **ChatWidget** встраивает виджет на сайт, чтобы пользователь мог взаимодействовать с **AgentNavigator** через графический интерфейс.

Компонент **HttpAdapter** обеспечивает интеграцию **AgentNavigator** с информационными системами.

Компонент **ServiceWeb** отвечает за интерфейс панели администрирования.

Компонент **IntegrityVerifier** отвечает за целостность (неизменяемость) установленных компонентов. Оповещает об изменениях в конфигурационных файлах, если таковые произошли

Компонент **OmnichatAdapter** позволяет задавать пользовательские вопросы в **AgentNavigator** через протокол HTTP.

 Компонент **IntegrityVerifier** ставится на каждый компьютер, где установлен хотя бы один из следующих компонентов: **BotServer**, **ChatWidget**, **HttpAdapter**, **ServiceWeb**, **Telegram**, **FileServer**, **OmnichatAdapter**, **SipAdapter**.

4.4.3 Установка необязательных компонентов

1. В случае интеграции с мессенджером **Telegram** установите компонент дополнительно:

```
yum install telegram -y
```

2. В случае использования компонента **FileServer** установите его:

```
yum install fileserver -y
```

3. В случае использования компонента **SipAdapter** установите его:

```
yum install sipadapter
```

4. В случае использования компонентов **Classifier** и **Classifier Trainer** создайте каталоги для моделей:

```
mkdir /opt/speechpro/classifier_models
```

Установите компоненты **Classifier** и **Classifier Trainer**:

```
yum install classifier classifiertrainer
```

```
systemctl daemon-reload
```

Запустите компоненты:

```
systemctl start classifier classifier_trainer
```

Включите классификатор в конфигурации компонента **BotServer** (файл `/etc/speechpro/botserver/application.yml`):

```
nlp:
  classifier:
    enabled: true
```

5. В случае использования компонента **CrafttalkAdapter** установите его:

```
yum install crafttalkadapter -y
```

6. В случае использования компонентов **NER** и **NER Trainer** установите их:

```
yum install ner
yum install ner_trainer
```

Перезагрузите службы:

```
systemctl daemon-reload
systemctl enable ner ner_trainer
```

7. В случае использования компонента **Co-pilot** установите его:

```
yum install co-pilot -y
```

8. В случае использования компонента **ConfigService** установите его:

```
yum install config-service -y
```

9. В случае использования компонента **python-http-adapter** установите его:

```
yum install python-http-adapter -y
```

4.4.4 Установка компонентов с изменением пути инсталляции

Для изменения пути установки компонента необходимо использовать пакетный менеджер **rpm** и опциональный параметр `--relocate`. Всего можно изменить три пути, которые используются для установки компонента:

- **/opt** — директория установки компонента;
- **/etc** — директория файлов конфигурации компонента;
- **/var/logs** — директория логов компонента;
- **/tmp** — директория временных файлов.

Пример:

```
rpm -iv /path/service.rpm --relocate /opt=/app --relocate /etc=/app/etc --  
relocate /var/log=/app/logs --relocate /tmp=/app/tmp
```

Ограничения:

- пакетный менеджер **yum** не поддерживает параметр `--relocate`;
- нельзя использовать **/etc/*** для релокации **/opt**;
- нельзя использовать **/var/log/*** для релокации **/etc**;
- все сервисы необходимо устанавливать с одинаковыми параметрами изменения пути инсталляции. Нельзя установить сервис А с параметром **/opt=/app1**, а сервис В с параметром **/opt=/app2**.

4.5 Настройка и запуск компонентов



В директории **/opt/speechpro/<component>/conf** располагаются конфигурационные файлы по умолчанию, которые помещаются туда сразу после установки компонента. Не рекомендуется их редактировать, так как после переустановки все изменения сбросятся на состояние по умолчанию.

В директории **/etc/speechpro/<component>/** располагаются конфигурационные файлы, которые переопределяют параметры конфигурационных файлов из директории **/opt**. Именно в них рекомендуется делать необходимые изменения конфигурации, так как конфигурационные файлы из директории **/etc** НЕ перезаписываются после переустановки компонента.



PDF является форматом для просмотра. Копируя содержимое из PDF-документа, проверяйте и выставляйте форматирование самостоятельно.

4.5.1 BotServer

Конфигурационный файл

Пример конфигурационного файла **/etc/speechpro/botserver/application.yml**:

```
server:  
  port: 8083  
  servlet:  
    context-path: /
```

```

    encoding:
      charset: UTF-8
      enabled: true
      force: true
  jetty:
    threads:
      max: 200
      min: 6
  rabbitmq:
    host: localhost
    port: 5672
    vhost: /
    user: guest
    password: guest
    thread-count: 1
    prefetch: 16
  stat-log-rabbitmq: #логирование stat-log событий(вопросы/ответы пользователя) в очередь,
для дальнейшего использования DialogHistoryService
    enabled: false #по умолчанию не логируются, необходимо включить только в случае
интеграции с DialogHistoryService
    host: localhost
    port: 5672
    vhost: /
    user: guest
    password: guest
    thread-count: 1
    prefetch: 16
  mongodb:
    name: test
    hosts:
      - localhost
    write-concern: ACKNOWLEDGED
    min-connections-per-host: 1
    max-connections-per-host: 100
    max-wait-time-ms: 10000
    max-connection-idle-time-ms: 60000
    max-connection-life-time-ms: 7200000
    socket-timeout-ms: 30000
    add-extra-class-field: true
    insert-scenario-retry:
      max-attempts: 3
      backoff-period-ms: 1000

  botserver:
    thread-count: 4
    external-call-thread-count: 2
    script-pool-size: 4 #количество javascript-движков для распараллеливания выполнения js-
кода. Для каждого сценария в scenario-pool-size создается script-pool-size-движков. Т.е.,
для scenario-pool-size = 4 и script-pool-size = 4 будет создано 4*4 = 16 движков.
    scenario-pool-size: 4 #значение размера кеша сценария - количество сценариев
находящееся в кеше
    deployed-scenario-validation-required: false
    processing-time-warn-threshold-ms: 2000
    nl-processing-max-count: 3
    max-state-change-count: 100
    active-channel-interval-sec: 14400
    scenario-load-at-startup-enabled: true
    default-answer: Извините, мне не удалось получить информацию.
    script:
      script-timeout-ms: 60000 #максимальное время выполнения скриптов
      init-timeout-ms: 60000
      wrap-in-curly-braces: true # включение оборачивания скриптов в сценарии (script: и
т.д.) в фигурные скобки для ограничения скоупа.
    engine:
      ecma-script-version: 2021 #год версии ECMAScript
    production-mode:

```

```

    enabled: true
  session:
    timeout-enabled: true
    timeout-sec: 7200
    timeout-check-interval-sec: 600
    timeout-thread-count: 1
    timeout-state: /onSessionTimeout
    remove-after-expiry-enabled: true
    remove-after-expiry-sec: 3600
    remove-after-expiry-check-interval-sec: 300
  lock:
    enabled: true
    overwrite: true
    back-off-policy:
      type: exponential
      initial-interval: 100
      max-interval: 1000
      multiplier: 2
    retry-policy:
      type: simple
      max-attempts: 6
  orderable-request:
    enabled: true
    #На текущий момент orderable-request используется только для кейса одновременной
отправки Recognition+InboundCallEnd (Text+Text).
    #Пользователь не ожидает ответ на Recognition(Text), т.к. положил трубку
(InboundCallEnd).
    #Исходя из этого, выбран timeout 60sec (с точки зрения обработки, практически
бесконечный),
    #т.к. не ожидается быстрый ответ бота.
    max-retry-count: 600
    #Задержка перед повторной отправкой в очередь
    delay-unordered-message-ms: 100
  too-long-question:
    enabled: true
    length: 2048
    state: /questionTooLongState
    score: 0
    confidence: 0
    fusion: 0
    weight-rb: 0
    weight-ml: 0
    pattern: n/a
  reject-message:
    after-ms-enabled: true
    after-ms: 10000
    on-uncached-scenario-enabled: true
  test:
    thread-count: 1
  test-nlp:
    strategy: auto
    threads-per-core: 1
    thread-count: 0
  day-session-count-scheduler:
    delay-after-midnight-sec: 1
  scenario:
    remove-if-unused-after-days: 2
    check-interval-sec: 7200
  before-step-processing:
    state: /beforeStepProcessing
  after-step-processing:
    state: /afterStepProcessing
  bottobot:
    management:
      client:
        timeout-ms: 1000

```

```
answer-while-busy:
  enabled: true
  answer: 'Пожалуйста, подождите. Ваш запрос обрабатывается.'
httpadapter:
  host: localhost
  port: 9030
  config-url: /config
  timeout-ms: 60000

nlp:
  pattern-timeout-ms: 0 #максимальное время обработки одного паттерна
  processing-timeout-ms: 3600000 #максимальное время обработки всех паттернов по всем
уровням (т.е. данный параметр задает максимальное время обработки одного вопроса)
  processing-timeout-no-match-pattern: '*' #паттерн, который мы считаем noMatch
  pattern-time-warn-threshold-ms: 100
  pattern-conflict-score-diff-threshold: 0.2
  nl-processing-time-warn-threshold-ms: 3600000
  preprocess-new-spec: false
  num-of-max-top-level-results: 30 #количество прочих наиболее подходящих результатов при
матчинге стейта, которые будут сохранены в бд, для дальнейшего использования
  fusion:
    weight:
      min-samples-count: 198
      min-rules-count: 3
      max-samples-count: 100
      max-rules-count: 39
  classifier:
    enabled: true
    train:
      scheme: http
      host: localhost
      port: 9009
      base-path: /restapi/classifier/
      http-client:
        connect-timeout-ms: 120000 #максимальное время на установку соединения
        socket-timeout-ms: 3600000 #максимальное время на получение ответа
        connections:
          max-total: 80
          max-per-route: 20
      retry:
        max-attempts: 1
        backoff-period-ms: 1000
  classify:
    scheme: http
    host: localhost
    port: 9008
    base-path: /restapi/v2/classifier/
    http-client:
      connect-timeout-ms: 30000
      socket-timeout-ms: 2000
      connections:
        max-total: 80
        max-per-route: 20
      retry:
        max-attempts: 2
        backoff-period-ms: 100
    healthcheck:
      connect-timeout-ms: 15000
      socket-timeout-ms: 15000
      base-path: /restapi/
  ner:
    enabled: true
    mode: group #режим обучения, возможные значения: flat, group
    confidence: 0.0
    on-restart-detector:
      enabled: true
```

```
check-restart-interval-sec: 10
trainer:
  scheme: http
  host: localhost
  port: 9011
  train:
    base-path: /restapi/models/
    http-client:
      connect-timeout-ms: 120000
      socket-timeout-ms: 3600000
      connections:
        max-total: 80
        max-per-route: 20
      retry:
        max-attempts: 1
        backoff-period-ms: 1000
  engine: #получение текущих алгоритмов обучения
    base-path: /restapi/engine/
    http-client:
      connect-timeout-ms: 30000
      socket-timeout-ms: 2000
      connections:
        max-total: 80
        max-per-route: 20
      retry:
        max-attempts: 2
        backoff-period-ms: 100
executor:
  scheme: http
  host: localhost
  port: 9010
  model-groups-load-unload-delete: #загрузка/выгрузка/удаление группы моделей
    async-load-thread-count: 4 # Если по каким-то причинам, после загрузки модели,
    модель оказалась не загруженной, то будет выполнена повторная загрузка модели. Данный
    параметр определяет сколько потоков может быть выделено под эту работу.
    base-path: /restapi/modelgroups
    http-client:
      connect-timeout-ms: 120000
      socket-timeout-ms: 3600000
      connections:
        max-total: 80
        max-per-route: 20
      retry:
        max-attempts: 1
        backoff-period-ms: 1000
  find-entities:
    base-path: /restapi/modelgroups
    http-client:
      connect-timeout-ms: 500
      socket-timeout-ms: 1000
      connections:
        max-total: 80
        max-per-route: 20
      retry:
        max-attempts: 1
        backoff-period-ms: 100
  lemmatize:
    base-path: /restapi/lemmatize
    http-client:
      connect-timeout-ms: 500
      socket-timeout-ms: 1000
      connections:
        max-total: 80
        max-per-route: 20
      retry:
        max-attempts: 1
```



```

openfeign:
  httpclient:
    max-connections: 80
    max-connections-per-route: 50
  client:
    config:
      llm-core:
        url: localhost
        connect-timeout: 1_000
        read-timeout: 30_000
  config:
    import-check:
      enabled: false

multipart:
  max-file-size: -1
  max-request-size: -1
  file-size-threshold: 0

scenario:
  validation:
    script-runtime-check: #валидации скриптов
    type-error: warn #проверка переменных внутри скрипта (например, при обращении к
полю undefined (необъявленной) переменной)
#Например:
#   var a = 1;
#   var tmp = sessionData.var.field; #<- обращаемся к полю field необъявленного поля
var (undefined)

    reference-error: enabled #проверка на недопустимые ссылки (например отсутствующие
функции в скрипте) и синтаксические ошибки
#Например:
    tmpData.name = getName(); #<- функция нигде не объявлена

    external-call-inner-reactions-check: #валидации externalCall конструкции (внутренние
реакции)
    change-state: disabled
    include: enabled
    choice: warn
    extension: disabled
    timeout: disabled
    step-back: disabled
    external-call: enabled
    external-call-method-existence-check: enabled #проверка наличия externalCall метода в
haConfig.yaml
    dialog-engine: #валидации технических стейтов
    tech-states:
      before-nlp-overrides-all-check: warn #если включена опция "when: beforeNlp",
запрещается включать другие опции (т.к. они будут проигнорированы)
      path-check: enabled #существование технических стейтов в сценарии
      on-nl-processing-limit: enabled #Проверяет соответствующий технический стейт на
значение (если оно представлено) nlProcessingMaxCount (>=1) и отсутствие переходов из
технического стейта (отсутствие конструкций go, go!)
      pattern: #валидация текста паттерна
      forbidden: #проверяет паттерны (именованные и нет) (их текстовое/еще не
распарсенное представление) на "запрещенные сочетания символов" (список регулярных
выражений представлен)
      status: enabled
      regular-expressions: [ "\\| [ *]*\\|", "\\| [ *]*\\)", "\\| [ *]*\\{", "\\| [ *]*\\
\\", "\\( [ *]*\\|", "\\[ [ *]*\\|", "\\{ [ *]*\\|", "\\{ [ *]*\\)", "\\( [ *]*\\)", "\\[ [ *]
*\\)" ]
      stop-words:
        status:
          regex: enabled
          stop-words-only-status: enabled
          double-exclusion-status: enabled

```

```

wildcard-in-excluded-group-status: enabled
# опциональных элементов в стоп-группах быть не должно
optional-element-regular-expressions: [ "!\\[" ]
#исключение отделяемой * и $oneWord
any-stop-word-regular-expression: [
  #[]()| lead to parsing !* as a distinct pattern, like space.
  #e.g., {!*} -> perm(star(*)) - invalid case.
  # {!*a} - valid case as *a is a stem
  #([*]|[$]oneWord) - аналогично отрицанию *, запрещается отрицание $oneWord
  #at the beginning: !* a, !*(a), !*{a}
  "^\\s*!([*]|[$]oneWord|[$]nonEmptyGarbage) [\\s(){}]+",
  "^\\s*!([*]|[$]oneWord|[$]nonEmptyGarbage)\\s*$",
  #at the middle: a !* b
  "[\\s(){}|]+!([*]|[$]oneWord|[$]nonEmptyGarbage) [\\s(){}|]+",
  #at the end: a !*
  "[\\s(){}|]+!([*]|[$]oneWord|[$]nonEmptyGarbage)\\s*$"
]
#стоп-слова в опциях запрещены
stop-word-in-options: [
  #at the beginning
  "\\(\\s*![ËÄA-яa-zA-Z0-9*]+\\s*\\|\\.+\\)",
  #at the middle
  "\\(\\.+\\|\\s*![ËÄA-яa-zA-Z0-9*]+\\s*\\|\\.+\\)",
  #at the end: (a|a)
  "\\(\\.+\\|\\s*![ËÄA-яa-zA-Z0-9*]+\\s*\\)" ]
reaction-existence-check: #валидация существования реакций
reactions: [ "o:" ] # only 'o:' is supported
validation: at_least_once # only supported option. Должна присутствовать как
минимум одна 'o:' в сценарии
check: warn #enabled|disabled
compound-answer-check: #валидация составных ответов
enabled: true
forbidden-file-name-symbols-regexps: [ "[\\#\\!\\|\\@\\*\\&\\%\\:\\:\\/\\|\\~\\|\\|\\?\\|\\^\\|\\N\\|\\;\\|\\-\\|=]", "[\\|<\\|>\\|\\(\\|\\)\\|\\[\\|\\]\\|\\{\\|\\}\\|]", "[\\|\\/\\|\\\\\\|]", "\\|\\.\\$" ] #перечень регулярных
выражений соответствующих недопустимым именам файлов
channel-type-check: # валидация по префиксу
enabled: true
disabled-by-channel-prefixes: [ 'postal' ] # список префиксов, по которым будет
получен положительный результат при валидации
reactions:
  tr:
    allow-in-root: true

logging:
  access-log:
    annotation-enabled: true

bot-to-bot:
  requeue-max-count: 3 #обязательный параметр
  on-max-requeue: асерт #обязательный параметр, возможные значения: АСЕРТ и REJECT
  #формула:задержка перед возвратом сообщения в очередь = initialDelay *
multiplayer^N(где N - счетчик возвратов сообщения)
  initial-delay-ms: 10 #обязательный параметр, множитель в формуле для вычисления
задержки перед возвратом сообщения в очередь.
  multiplayer: 2 #обязательный параметр, число, которое возводится в степень счетчика
возвратов сообщения в формуле для вычисления задержки перед возвратом сообщения в
очередь.
  in-step-transfer-max-count: 5 #необязательный параметр, по умолчанию равен 5,
максимальное количество переходов с мастера на скилл в рамках одного шага(stepNumber)
  session-data:
    master-to-skill:
      enabled: true #необязательный параметр, по умолчанию true, включает/выключает
передачу sessionData при переходе с мастера на скилл
    skill-to-master:
      enabled: true #необязательный параметр, по умолчанию true, включает/выключает
передачу sessionData при переходе со скилла на мастер

```

```
healthcheck:
  dependent-component-timeout-ms: 30000
  mongo-timeout-ms: 30000
```

Пример конфигурационного файла */etc/speechpro/botserver/application-stands.yml*:

```
stands:
- stand-id: skillChannelId #обязательный параметр
  host: localhost #обязательный параметр
  port: 5672 #обязательный параметр
  vhost: / #обязательный параметр
  user: guest #обязательный параметр
  password: guest #обязательный параметр
  thread-count: 1 #опциональный параметр, стандартное значение: 1
  prefetch: 8 #опциональный параметр
  requested-heartbeat: 10 #опциональный параметр, стандартное значение: 60
  network-recovery-interval-ms: 1000 #опциональный параметр, стандартное значение: 1000
  publish-channel-count: 1 #опциональный параметр, стандартное значение: 1
  consume-auto-ack: false #опциональный параметр, стандартное значение: false
  delay-between-reconnections-ms: 5000 #стандартное значение: 0
  max-number-of-reconnection: 10 #стандартное значение: = 1
  message-ttl-ms: 99999 #опциональный параметр, стандартное значение: 1200000
  queue-ttl-ms: 99999 #опциональный параметр, стандартное значение: 86400000
  message-validator: ERROR #опциональный параметр, стандартное значение: WARN
```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер выполните только шаг [2](#).

1. В файле */etc/speechpro/botserver/application.yml* укажите:

```
rabbitmq:
  host: mqhost
mongodb:
  name: botserver
  hosts:
    - mghost
httpadapter:
  host: hahost
```

mqhost, **mghost** и **hahost** — имена или IP-адреса хостов, где установлены RabbitMQ, MongoDB и HttpAdapter соответственно.

2. Запустите программу:

```
systemctl start botserver.service
```

4.5.2 HttpAdapter

Конфигурационный файл

```
spring:
  main:
    banner-mode: 'off'
  profiles:
    include:
      - llm-models
      - llm-sdk-core-client
      - llm-services # используется в httpadapter и serviceweb
  config:
```

```

import: optional:configserver:http://localhost:9078
cloud:
  config:
    import-check:
      enabled: false
  openfeign:
    httpclient:
      max-connections: 200
      max-connections-per-route: 50
  client:
    config:
      giga-chat-auth-beta: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 10_000
      giga-chat-completions-beta: # используется в httpadapter и serviceweb. Может
быть настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 30_000
      giga-chat-auth-prod: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 10_000
      giga-chat-completions-prod: # используется в httpadapter и serviceweb. Может
быть настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 30_000
      yandex-gpt-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 10_000
      llm-core:
        url: localhost
        connect-timeout: 1_000
        read-timeout: 10_000

multipart:
  max-file-size: -1
  max-request-size: -1
  file-size-threshold: 0

# блок конфигурации подключения к rabbitmq
rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 8

#блок конфигурации подключения к MongoDB
mongodb:
  name: httpadapter
  hosts:
    - localhost
  write-concern: ACKNOWLEDGED

httpadapter:
  mq-processing-thread-count: 20
  script-engine-pool-count: 20

```

```

script-cache-expire-after-access-min: 60
modify-request-max-tries: 10
reject-message:
  after-ms-enabled: true
  after-ms: 30000
script:
  script-timeout-ms: 60000 #максимальное время выполнения скриптов
  engine:
    есma-script-version: 2021 #год версии ECMAScript
http:
  timeout-ms: 10000
  connections:
    max-total: 80
    max-per-route: 20
method-cache:
  after-write:
    expiration-thread-count: 1
proxy:
  enabled: false
async:
  callback:
    back-off-policy:
      type: exponential
      initial-interval: 100
      max-interval: 1000
      multiplier: 2
    retry-policy:
      type: simple
      max-attempts: 6
  context:
    enabled: true
    timeout-sec: 1
    timeout-check-interval-ms: 200
request:
  escape:
    json:
      enabled : false

#блок конфигурации настройки режимов валидации на конфигов
scenario:
  validation:
    param-count: warn # возможные режимы warn/enabled/disabled
    param-usage: warn # возможные режимы warn/enabled/disabled

server:
  port: 9030

healthcheck:
  mongo-timeout-ms: 30000

llm: # используется в httpadapter и serviceweb. Может быть настроено в config-service
  default-vendor: GIGA_CHAT # Возможные значения GIGA_CHAT и YANDEX_GPT
  giga-chat:
    enabled: false
    auth-required: true
    api-type: PROD # Возможные значения BETA и PROD
    auth:
      username: someUser
      password: somePassword
      prod:
        scope: GIGACHAT_API_CORP
      retry:
        max-attempts: 1
        back-off-period-ms: 100
      uri: /api/v2/oauth
  completions:

```

```

model-id: GigaChat-Pro # определяется для каждого внедрения отдельно
model-name: GigaChat-Pro # определяется для каждого внедрения отдельно
profanity-check: true # определяется для каждого внедрения отдельно
uri: /api/v1/chat/completions
retry:
  max-attempts: 1
  back-off-period-ms: 100
channel-specific:
- channel-id: gigaChatSpecificChannelId1
  auth:
    username: someUser1
    password: somePassword1
  completions:
    profanity-check: true
- channel-id: gigaChatSpecificChannelId2
  auth:
    username: someUser2
    password: somePassword2
  completions:
    profanity-check: false
certificate: #optional: конфигурация для настройки tls
protocol: TLS #optional: выбор протокола/алгоритма, по умолчанию SSL
keystore: #optional: хранилище ключей
  path: ./httpadapter/src/test/resources/certificate/client01.p12 #путь к хранилищу
keystore
  password: '6-bypd'#пароль для хранилища и сертификатов
  keys-password: '6-bypd'#optional: является паролем для сертификата. По умолчанию
равен password.
  format: PKCS12 #формат, возможные значения: PKCS12, JKS
  aliases: [1] #optional: предназначен для задания названия alias для сертификата
truststore: #optional: хранилище доверенных сертификатов
  path: ./httpadapter/src/test/resources/certificate/truststore.jks #путь к
хранилищу truststore
  password: '6-bypd' #optional: пароль для хранилища. "" по умолчанию.
  format: JKS #формат, возможные значения: PKCS12, JKS
  aliases: [1] #optional: предназначен для задания названия alias для сертификата
yandex-gpt:
  enabled: false
  api-key: someApiKey
  completions:
    dir-id: someDirId # определяется для каждого внедрения отдельно
    model-id: yandexgpt/latest # определяется для каждого внедрения отдельно
    model-name: YandexGPT Pro # определяется для каждого внедрения отдельно
    uri: /foundationModels/v1/completion
    retry:
      max-attempts: 1
      back-off-period-ms: 100

```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер выполните только шаг [2](#).

1. В файле `/etc/speechpro/httpadapter/application.yml` укажите:

```

rabbitmq:
  host: mqhost
mongodb:
  hosts:
    - mgghost

```

mqhost и **mgghost** — имена или IP-адреса хостов, где установлены **RabbitMQ** и **MongoDB** соответственно.

2. Запустите программу:

```
systemctl start httpadapter.service
```

4.5.3 ServiceWeb

Конфигурационный файл

```
server:
  port: 8086
  servlet:
    context-path: /
    session:
      timeout: 3600

web-socket:
  ping-delay-seconds: 10
  message-timeout-seconds: 10
  allowed-origins: '*'

serviceweb:
  security:
    headers:
      frame-options: deny #deny|sameOrigin|disable

prompter:
  enabled: ifLogsDirExist
  postgres:
    url: jdbc:postgresql://localhost:5432/reports?currentSchema=prompter ?
currentSchema=prompter — опционально
    username: reports
    password: reports
    query-execute-time-limit-in-ms: 15000
    min-pool-size: 0
    max-pool-size: 10
    flyway:
      enabled: true
      locations: classpath:migrationsprompter
      baseline-on-migrate: false # включение опции baselineOnMigrate в flyway для схемы
      #For backward compatibility
      table: schema_version
  pump:
    enabled: true
    log-file-name: prompteradapter.prompter.json.log
    logs-paths: ${prompter.pump.statLogPaths}
    initial-delay: 0
    delay: 100
    buffer: 1000
    thread-count: 1

postgres:
  reports:
    url: jdbc:postgresql://localhost:5432/reports?currentSchema=prompter ?
currentSchema=prompter — опционально
    username: reports
    password: reports
    query-execute-time-limit-in-ms: 15000
    min-pool-size: 10 #минимальное количество соединений к бд, в рамках подключения.
    max-pool-size: 10 #максимальное количество соединений к бд, в рамках подключения.
    skill: #секция подключений skill-db для получения sessionData в ответах skill
      connections: #необязательный массив подключений к базам данных skill, элемент
массива полностью соответствует параметрам подключений master
```

```

# - {dbname: reports, user: reports, password: reports, host: localhost, port:
5432, query-execute-time-limit-in-ms: 15000, min-pool-size: 10, max-pool-size: 10}
update-from-skill: # Новая секция для получения данных со skill
  enabled: true # вкл/выкл
  delay-ms: 100 # период запуска задачи для получения данных со скилла по одному
answer (session_data, internalSession_data) через новую очередь в бд (Таблица:
update_from_skill_queue)
  thread-count: 1 # количество потоков в новой задаче
  available-after-ms: 60000 # тайм-аут для повторной обработки ответа, если не
найден запись на skill или произошла ошибка
  log-requeue-after-attempts: 10 # количество попыток после которого будет
залогирован warn по ответу
  housekeeper: # очистка очереди от старых не обработанных записей
    cron-expression: 0 0 1 * * ? # cron для очистки
    interval: 30 days # возраст записи для очистки
super-admin-user:
  create-on-empty-users: true #Создавать пользователя и роль супер админ, если
отсутствуют пользователи в SW.
audit:
  url: jdbc:postgresql://localhost:5432/reports?currentSchema=prompter ?
currentSchema=prompter – опционально
  username: reports
  query-execute-time-limit-in-ms: 15000
  min-pool-size: 1
  max-pool-size: 5
flyway:
  enabled: true
  locations: classpath:migrationsaudit
  baseline-on-migrate: false # включение опции baselineOnMigrate в flyway для схемы
#For backward compatibility
  table: schema_version

scenario-loader:
  snakeyaml:
    code-point-limit: 1048576000 # 1000 * 1024 * 1024. Means, that max file size to be
parsed is 1000 MB

export:
  postgres:
    min-pool-size: 0
    max-pool-size: 5
  fetch-size: 10000
  max-rows: 1000000
  csv:
    encoding: UTF-8
    delimiter: ','
  excel:
    font-size-in-points: 11
    first-column-width: 3500 # Ширина первого столбца
  word:
    font-size-in-points: 10
  pdf:
    font-size-in-points: 8
  page-config:
    top-margin-inches: 0.5 # Отступы страницы(используется в word) от края в дюймах
    bottom-margin-inches: 0.5
    left-margin-inches: 0.5
    right-margin-inches: 0.5
    columns-ratio-width: 8, 12, 10, 15, 12, 12, 10, 10, 35 # Список из абстрактных
величин, определяющих соотношение ширины колонок таблицы друг относительно друга. Имеет
константный размер – 9, при запуске сервиса проводится валидация на соответствие этому
размеру. Текущее, константное заполнение отражает ширину столбцов максимально
приближенную к reference файлу. Пример: количество колонок – 4, columnsRatioWidth – 10,
10, 10, 10, комментарий – для всех 4 колонок, будет задана одинаковая ширина, значений
которой составляет 25% от общей ширины таблицы.

```

```
statstates:
  redirecttooperator:
    operator-state-diff-business-state-steps: 1

operatorchatdata:
  enabled: ifLogsDirExist #признак возможности для фильтрации, также есть варианты
  (enabled, disabled)
  filter:
    request-data-root-field: agent #значение поля (request_data), по которому
    определяется, что вопрос относится к операторскому чату

pump:
  enabled: true
  stat-log-file-name: botserver.stat.log
  json-stat-log-file-name: botserver.stat.json.log
  stat-logs-paths: ${pump.statLogPaths}
  audit-log-file-name: audit.log
  audit-logs-path: /var/log/speechpro
  audit:
    enabled: true
    delay: 1000
    buffer: 1000
  day-session-counts:
    enabled: true
    day-session-log-file-name: botserver.daysessioncount.json.log
    day-session-logs-paths: ${pump.daySessionCounts.daySessionLogsPaths}
    delay: 3_600_000
    buffer: 1000
  bot-to-bot-link:
    enabled: true
    bot-to-bot-link-log-file-name: botserver.bottobotlink.json.log
    bot-to-bot-link-logs-paths: ${pump.botToBotLink.botToBotLinkLogsPaths}
    delay: 1000
    buffer: 1000
  form:
    enabled: true
    form-log-file-name: botserver.form.json.log
    form-logs-paths: {node1: /var/log/speechpro/botserver}
    delay: 1000
    buffer: 1000
  session-top-hint-states:
    enabled: true
    log-file-name: botserver.sessiontophintstates.json.log
    logs-paths: ${pump.sessionTopHintStates.logsPaths}
    delay: 1000
    buffer: 1000
  delay: 100
  buffer: 1000
  clean:
    delay-ms: 3600000
    inactive-timeout: 1 days
  type: json

uri-rewrite:
  enabled: true # поддержка trailing-slash

multipart:
  max-file-size: -1
  file-size-threshold: 0
  max-request-size: -1

http-client:
  connect-timeout-ms: 180000
  socket-timeout-ms: 180000
```

```
connections:
  max-total: 80
  max-per-route: 20
retry:
  max-attempts: 1
  backoff-period-ms: 1000

hg:
  enabled: ${hg.enabled}

dictionary:
  repo:
    remote: https://bitbucket.org/speechpro/vc-content
    local: /etc/speechpro/serviceweb/vc-content

scenario-editor:
  log-limit: 1000 #макс. количество коммитов возвращаемое hg log
  repo:
    id: vc-beeline
    protocol: http
    remote: localhost:8000
    local: /etc/speechpro/serviceweb/scenario
    init-thread-count: 5
    is-lf: true

spring:
  http:
    encoding:
      enabled: true
      charset: UTF-8
  main:
    banner-mode: 'off'
  profiles:
    include: llm-models
  config:
    import: optional:configserver:http://localhost:9078
  jmx:
    enabled: true
  security:
    maximum-sessions: 100
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher
  flyway: #Миграция БД reports (схема public) при запуске приложения
    enabled: true
    locations: classpath:migrations
    baseline-on-migrate: false # включение опции baselineOnMigrate в flyway для схемы
    #For backward compatibility
    table: schema_version
    out-of-order: true
    validate-migration-naming: true
  cloud:
    openfeign:
      httpclient:
        max-connections: 80
        max-connections-per-route: 50
      client:
        config:
          ivr-session: # Настройки подключения к сервису IVR SessionService
            url: http://localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          ivr-security: # Настройки подключения к сервису IVR Security
            url: http://localhost
            connect-timeout: 1_000
            read-timeout: 30_000
```

```

    llm-core:
      url: http://localhost:9999
      connect-timeout: 1_000
      read-timeout: 30_000
    giga-chat-auth-beta: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 10_000
    giga-chat-completions-beta: # используется в httpadapter и serviceweb. Может
быть настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 30_000
    giga-chat-auth-prod: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 10_000
    giga-chat-completions-prod: # используется в httpadapter и serviceweb. Может
быть настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 30_000
    yandex-gpt-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
      url: localhost
      connect-timeout: 1_000
      read-timeout: 10_000
    llm-document-storage:
      url: http://localhost:9998
      connect-timeout: 1_000
      read-timeout: 30_000

llm: # используется в httpadapter и serviceweb. Может быть настроено в config-service
  default-vendor: GIGA_CHAT
  giga-chat:
    enabled: false
    api-type: PROD
    auth-required: true
    auth:
      username: someUser
      password: somePassword
      prod:
        scope: GIGACHAT_API_CORP
      retry:
        max-attempts: 1
        back-off-period-ms: 100
    uri: /api/v2/oauth
  completions:
    model-id: GigaChat-Pro
    model-name: GigaChat-Pro
    profanity-check: true
    top-p:
      repetition-penalty:
        temperature:
          uri: /api/v1/chat/completions
      retry:
        max-attempts: 1
        back-off-period-ms: 100
  yandex-gpt:
    enabled: false
    api-key: someApiKey
    completions:
      dir-id: someDirId
      model-id: yandexgpt/latest

```

```

    model-name: YandexGPT Pro
    uri: /foundationModels/v1/completion
    retry:
      max-attempts: 2
      back-off-period-ms: 100

prompts:
  intent-generation:
    system:
      user: 'Перед тобой документ. Сгенерируй от {min} до {max} тематик, о которых идет
речь в документе.\nРезультат верни в формате JSON списка без каких либо пояснений,
например, ["тематика1", "тематика2", "тематика3", "тематика4"].\nНе повторяй тематики.
\n\nТекст документа: {text}'
    question-generation:
      system:
        user: 'Сгенерируй от {min} до {max} фраз для интента "{subject}". Результат верни в
формате JSON списка без каких либо пояснений, например, ["фраза1", "фраза2", "фраза3",
"фраза4"]. Не дублируй фразы.'
      question-generation-with-examples:
        system:
          user: 'Сгенерируй от {min} до {max} фраз для интента "{subject}". Примеры фраз:
{examples}. Результат верни в формате JSON списка без каких либо пояснений, например,
["фраза1", "фраза2", "фраза3", "фраза4"]. Не повторяй фразы из примера и не дублируй
фразы.'

```

```
user-queries:
  - base: ou=Branches
    field-filter: sAMAccountName
user-blocking-code:
  - 514
  - 66050

kerberos:
  enabled: false
  keytab: /opt/tomcat.keytab
  principal: HTTP/applicationhost@YOURDOMAIN.COM

deploy:
  enabled: true
  path: /tmp/
  thread:
    pool: 1
  retry-on-startup:
    enabled: true
    botserver-check-internal-sec: 3
    is-continue-deploy: false
  retry-fail-channel-status-update:
    back-off-interval-ms: 10000
    max-attempts: 30

quality:
  sub-selection-interval-min: 60
  unknown-ratio: 0.3

ad-groups-roles:
  mapping:
    is-many-to-many-relations: true

synchronize-ad-users:
  # delay: 14400000 = 4hr, 10800000 = 3hr, 7200000 = 2hr, 3600000 = 1hr
  delay: 10800000
  enabled: true

dialogs:
  #postgres interval
  max-session-duration: 12 hours

housekeeper:
  audit:
    enabled: false
    cron-expression: 0 0 1 * * ?
    interval: 3 years
  reports:
    enabled: false
    cron-expression: 0 0 1 * * ?
    interval: 3 years

session-top-hint-states:
  scenario-cache-size: 100
  housekeeper:
    enabled: true
    cron-expression: 0 0 1 * * ?
    interval: 30 days

visualeditor:
  stateendpoint:
    concurrentupdate:
      enabled: false
  cache:
    enabled: true
```

```

    expire-after-access-ms: 600000
    max-size: 100
    file-name-validation:
      enabled: true
      forbidden-file-name-symbols-regexps: ["[a-яА-Я]", "[\\#\\!\\@\\*\\&\\%\\:\\/\\|\\~\\`\\'\\?\\^\\$\\%\\&\\;\\|=]", "[\\<\\>\\(\\)\\[\\]\\{\\}\\}]", "[\\/\\\\\\\\]", "\\.$"]
    knowledge-base:
      max-file-char-count: 15000
    stop-words-validation:
      enabled: true
      #see description in botserver application.yml
      optional-element-regular-expressions: ["!\\[\""]
      any-stop-word-regular-expression: [
        "\\s*!([*]|[$]oneWord)\\s(){}|+",
        "\\s*!([*]|[$]oneWord)\\s*$",
        "\\s(){}|!+!([*]|[$]oneWord)\\s(){}|+",
        "\\s(){}|!+!([*]|[$]oneWord)\\s*$"
      ]
      stop-word-in-options: [
        "\\(\\s*![a-zA-Z0-9*]+\\s*\\.+\\)",
        "\\(\\.+\\|\\s*![a-zA-Z0-9*]+\\s*\\.+\\)",
        "\\(\\.+\\|\\s*![a-zA-Z0-9*]+\\s*\\)"
      ]
    prompts:
      max-system-prompt-char-count: 2000
      max-user-prompt-char-count: 1000
    script:
      script-timeout-ms: 5000

channel:
  exclude:
    enabled: true
    by-prefixes: ['postal']

retry:
  pump:
    retry-on-incomplete-line:
      max-attempts: 5
      wait-before-next-attempt-ms: 1000

locale: ru

reports:
  questions-per-state:
    aggregate-by: TITLE

logging:
  access-log:
    annotation-enabled: true

ivr:
  security:
    enabled: false # Настройка включения интеграции ServiceWeb в систему безопасности IVR
  session:
    credentials: # Данные пользователя для интеграции ServiceWeb в систему безопасности IVR
  username: username
  password: 12345678
  domain-id: 123
  media-storage: # Настройка подключения к сервису IVR MediaStorage
    connection:
      host:
      port:
      timeout-ms: 60000

classifier:

```

```
installed: true # Установлен ли на стенде классификатор, значению по умолчанию true
```

Настройка и запуск

Важно проверить, что `/opt/speechpro/serviceweb/conf/application.yml` в конфигурациях полей `prompter→pump→logsPaths` и `pump→logsPaths`, не имеет дублирующих полей/путей с теми же параметрами, что и в `/etc/speechpro/serviceweb/application.yml`.

1. Если конфигурация сервисов **AgentNavigator** соответствует стандартной и включает в себя только **BotServer**, **HttpAdapter** и **ChatWidget**, то в файле `/etc/speechpro/serviceweb/application-integration.yml` укажите:

```
integration:
  vcservices:
    botserver: # Сервис
      service-name: botserver # Название, отображаемое в ServiceWeb, вкладка "Состояние
компонентов"
      nodes:
        - {ip: 127.0.0.1, port: 8083} # Первый элемент в списке — это primary-узел
    httpadapter:
      service-name: httpadapter
      nodes:
        - {ip: 127.0.0.1, port: 9030}
    chatwidget:
      service-name: chatwidget
      nodes:
        - {ip: 127.0.0.1, port: 9999}
    integrityverifier:
      service-name: IntegrityVerifier
      nodes:
        - {ip: 127.0.0.1, port: 10101}
      max-http-timeout-ms: 3000
    deprecated-channels-archive: # не используемые каналы в системе
      abstract-service-name: # название сервиса, например sipadapter или botserver или
chatwidget
        - abstract-channel-id-1 # список идентификаторов каналов сервиса (соответствует
тем, что находились в секции integration.vcservices.{service-name}.dedicated-channel
когда канал был актуален)
        - abstract-channel-id-2
```

Если конфигурация отличается от стандартной, то в файле `/etc/speechpro/serviceweb/application-integration.yml` укажите:

```
integration:
  vcservices:
    omnichatadapter: # Сервис
      service-name: omnichatadapter # Название, отображаемое в ServiceWeb, вкладка
"Состояние компонентов"
      dedicated-channels: # Перечислите каналы, которые должны отображаться в списке
доступных каналов на вкладке "Каналы"
        - { id: omnichat-adapter-master, name: omnichatMaster, botToBotChannelType:
MASTER }
        - { id: omnichat-adapter-skill-1, name: omnichatSkill1, botToBotChannelType:
SKILL }
        - { id: omnichat-adapter-skill-2, name: omnichatSkill2 } # Конфигурация для
проверки явного отсутствия атрибута botToBotChannelType
      nodes: # Узлы сервиса
        - {ip: 127.0.0.1, port: 9066}
    sipadapter:
      service-name: sipadapter
      dedicated-channels:
        - {id: sip-adapter-incoming, name: sip-adapter-incoming}
```

```

- {id: sip-adapter-incoming-master, name: sip-adapter-incoming-master,
  botToBotChannelType: MASTER }
  nodes:
    - {ip: 127.0.0.1, port: 9098}
  on-startup:
    create-channels-enabled: true # Вкл/выкл создание и обновление каналов адаптеров в
    BotServer при старте ServiceWeb
    max-attempts: 60 # Количество попыток
    backoff-period-ms: 1000 # Интервал между попытками
    deprecated-channels-archive: # не используемые каналы в системе
    abstract-service-name: # название сервиса, например sipadapter или botserver или
    chatwidget
    - abstract-channel-id-1 # список идентификаторов каналов сервиса (соответствует
    тем, что находились в секции integration.vcservices.{service-name}.dedicated-channel
    когда канал был актуален)
    - abstract-channel-id-2

```



Компоненты **Classifier**, **Classifier Trainer**, **NER**, и **NER Trainer** нельзя указывать в данной конфигурации.

Чтобы каналы адаптеров отображались в интерфейсе, добавьте их в блок `dedicated-channels`.

Важно проверить, что файл `/opt/speechpro/serviceweb/conf/application.yml` в конфигурациях полей `prompter→pump→logsPaths` и `pump→logsPaths` не имеет дублирующих полей/путей с теми же параметрами файла `/etc/speechpro/serviceweb/application.yml`.

2. Если для мастер-бота и скилл-бота используются разные БД, укажите `connections` в секции `skill`:

```

postgres:
  reports:
    dbname: reports
    user: reports
    password: reports
    host: localhost
    port: 5432
    query-execute-time-limit-in-ms: 15000
    min-pool-size: 10 #минимальное количество соединений к БД, в рамках подключения.
    max-pool-size: 10 #максимальное количество соединений к БД, в рамках подключения.
    skill: #секция подключений skill-db для получения sessionData в ответах skill
    #connections: #необязательный массив подключений к базам данных skill, элемент
    массива полностью соответствует параметрам подключений master
    # - {dbname: reports, user: reports, password: reports, host: localhost, port:
    5432, query-execute-time-limit-in-ms: 15000, min-pool-size: 10, max-pool-size: 10}

```

3. Запустите программу:

```
systemctl start serviceweb.service
```

4.5.3.1 Подключение ServiceWeb к Security

Для подключения **ServiceWeb** к сервису **Security** системы **VoiceKey.IVR SIP** в конфигурационном файле `application.yml` укажите настройки подключения к сервису в блоке [spring.cloud.openfeign.client.config](#), установите [ivr.security.enabled=true](#), и укажите данные администратора в параметре [ivr.session.credentials](#).

Пример конфигурации:

```

spring:
  cloud:
    openfeign:
      httpclient:
        max-connections: 80

```

```

    max-connections-per-route: 50
  client:
    config:
      ivr-session: # Настройки подключения к сервису IVR SessionService
        url: http://localhost
        connect-timeout: 1_000
        read-timeout: 30_000
      ivr-security: # Настройки подключения к сервису IVR Security
        url: http://localhost
        connect-timeout: 1_000
        read-timeout: 30_000
  ivr:
    security:
      enabled: false # Настройка включения интеграции ServiceWeb в систему безопасности IVR
    session:
      credentials: # Данные администратора для интеграции ServiceWeb в систему безопасности
        IVR, могут меняться
        username: username
        password: 12345678
        domain-id: 123

```

4.5.3.2 Фильтрация по параметрам диалога

Конфигурация фильтров хранится в конфигурационном файле `/opt/speechpro/serviceweb/conf/application-dialog-filters.yml` компонента **ServiceWeb** и представляет собой следующую структуру:

```

dialogs:
  filters:
    - id: adapter-async
      channelIds: [adapter-async]
      fields:
        - id: ctn
          db: {table: questions, column: requestData, field: ctn}
          front: {target: SESSION, title: ctn, type: MULTI, values-as-string: ["value 1",
"value 2", "value 3"]}

```

Перед началом работы с файлом `/opt/speechpro/serviceweb/conf/application-dialog-filters.yml` его следует скопировать в директорию `/etc/speechpro/serviceweb` и уже в копии файла, то есть в `/etc/speechpro/serviceweb/application-dialog-filters.yml`, настраивать фильтры.

По аналогии с любой другой конфигурацией, для применения сконфигурированных фильтров перезапустите компонент командой:

```
systemctl restart serviceweb
```

Пример поиска **JSON_CONTAINS**:

```

dialogs:
  filters:
    - id: adapter-async
      channelIds: [adapter-async]
      fields:
        - id: ctn
          db: {table: questions, column: requestData, field: ctn}
          front: {target: SESSION, title: ctn, type: TEXT, validate-regex: '\w+'}

```

Пример поиска **JSON_TEXT_EXTRACT**:

```

dialogs:
  operations:
    allowed: [lower]
  filters:

```

```

- id: adapter-async
  channelIds: [adapter-async]
  fields:
    - id: operatorPenName
      db: {table: questions, column: requestData, field: agent.penName}
      front: {target: SESSION, title: "Псевдоним оператора", type: TEXT,
validateRegex: '\w+'}
      queryType: JSON_TEXT_EXTRACT
      operations:
        - name: lower

```

Значения параметров:

- **id** — идентификатор фильтра;
- **channelIds** — список идентификаторов каналов, к которым можно применять фильтрацию;
- **fields** — массив объектов полей для фильтрации;
- **id** — идентификатор поля;
- **db** — информация о поле в БД;
- **table** — название таблицы, в котором находится поле (возможные значения: **QUESTIONS**, **ANSWERS**);
- **column** — название колонки (возможные значения: для table **QUESTIONS**: **REQUEST_DATA**, для table **ANSWERS**: **RESPONSE_DATA**, **INTERNAL_SESSION_DATA**, **SESSION_DATA**, **TEMP_DATA**);
- **field** — json-поле (например, `someField` в `request_data: { 'someField': 1 }`);
- **queryType** — тип **query**;
- **JSON_CONTAINS** — по умолчанию. Поиск по '@>' — contains (использует GIN-индекс поддерживаемый по умолчанию). Например, `requestData @> {'a': 'aBc'}`;
- **JSON_TEXT_EXTRACT** — поиск по '->>' — json extract (требуется создания отдельного индекса). Например, `requestData ->>'a' = 'aBc'`;
- **operations** — список дополнительных операций над колонкой и значением. Поддерживается только для **query-type=JSON_TEXT_EXTRACT**. Список разрешенных имен конфигурируется в *dialogs.operations.allowed*;
- **name** — имя операции. Например, `lower: lower (requestData ->>'a') = lower('aBc')`;
- **front** — информация о поле на FE;
- **target** — тип фильтрации, на FE фильтры группируются в UI по этим значениям, параметр влияет только на отображение в UI (возможные значения: **SESSION** — поле размещается в разделе **Параметры диалогов** или **PHRASE** — поле размещается в разделе **Параметры реплик**);
- **title** — название компонента, которое отображается в UI;
- **type** — тип компонента (возможные значения: **TEXT** — текст-бокс, **BOOLEAN** — чекбокс, **MULTIVAL** — поле ввода множественных значений, **MULTI** — селект-бокс для нескольких элементов, **SELECT** — селект-бокс для одного элемента);

- **validateRegex** — регулярное выражение для валидации текстового поля (**TEXT**). Если введенное значение в текстовом поле не совпадает с регулярным выражением, то поле выделяется красной рамкой;
- **valuesAsString** — массив строк, куда записываются значения для фильтров типа **MULTI** и **SELECT**. Для типа **SELECT** значение по умолчанию должно идти первым в массиве.

4.5.3.3 Настройка контекста для веб-компонентов

Для изменения контекста достаточно установить значение в соответствующем конфигурационном файле `/etc/speechpro/serviceweb/application.yml`:

```
server:
  context-path: / # Контекст здесь
```

Если для веб-приложений установлен контекст отличный от серверных приложений, то в соответствующей папке со сборкой файлов веб-приложения в файле **contextPath.js** заменить (или закомментировать) код:

```
window.contextPath = (() => {
  const path = window.location.pathname.replace(/(sceditor|vseditor)\/?/, "")
  return path.endsWith("/") ? path.slice(0, -1) : path
})();
```

на:

```
window.contextPath = "/myNewContext"
```

В общем случае замена контекста на отличный от контекста серверной части не рекомендуется.

В настоящее время поддержка контекста доступна для компонентов: **ServiceWeb**, **Визуальный Редактор**, **Текстовый Редактор**.

4.5.3.4 Настройка автоматической очистки базы данных

По умолчанию очистка архивной БД выключена.

```
housekeeper:
  enabled: false
  cronExpression: 0 0 1 * * ?
  #postgres interval
  interval: 3 years
```

enabled: установите **true**, чтобы включить автоматическую очистку базы данных;

cronExpression: период запуска задачи на очистку в формате Cron Expression http://docwiki.embarcadero.com/Connect/en/Writing_a_CRON_Expression;

interval: интервал времени, позже которого данные будут удалены, в формате postgres interval <https://postgrespro.ru/docs/postgresql/14/functions-datetime>.

Удаляемые данные: диалоги, разметка диалогов, информацию по диалогам (содержимое объектов `sessionData/tempData` и пр.), логи журнала аудита.

4.5.4 IntegrityVerifier

Конфигурационный файл

```
server:
  port: 10101
spring:
  datasource:
    url: jdbc:postgresql://host1:5432/integrityverifier
    username: integrityverifier
    password: integrityverifier
    hikari:
      minimum-idle: 1
      maximum-pool-size: 2
  flyway: #Миграция БД reports (схема public) при запуске приложения
    enabled: true

integrityverifier:
  #Invalid value. Property "nodeId" must be not null and not empty and be overwritten
  #in /etc/speechpro/integrityverifier/application.yml file
  node-id: null
  rescan-period-seconds: 43200
  auto-rescan-enabled: true
  master-snapshot-paths:
    - "/opt/speechpro/"
```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага [2](#).

1. В файле `/etc/speechpro/integrityverifier/application.yml` укажите:

```
postgres:
  host: pghost
```

Где **pghost** — сервер PostgreSQL.

2. В файле `/etc/speechpro/integrityverifier/application.yml` укажите:

```
integrityverifier:
  node-id: "node1"
```



Значение **node-id** у каждого экземпляра должно быть уникальным.

3. Создайте схему базы данных **integrityverifier** в PostgreSQL:

```
(cd /opt/speechpro/integrityverifier/bin/ && bash migrate.sh migrate)
```

4. Создайте снимок текущего состояния компонентов комплекса:

```
(cd /opt/speechpro/integrityverifier/bin/ && bash snapshots.sh)
```

5. Запустите программу:

```
systemctl start integrityverifier.service
```

4.5.5 ChatWidget

Конфигурационный файл

```
server:
  port: 9999
  servlet:
    context-path: /
chatwidget:
  session:
    timeout-sec: 3600
    timeout-check-interval-sec: 1800
    timeout-check-delay-sec: 1800
  polling:
    timeout-ms: 30000
    timeout-core-pool-size: 3
  history:
    enable: true
    remove-delay-sec: 3600
  processing-error-message: Сервис временно недоступен.
  reject-on-uncached-scenario-message: Сервис будет доступен в ближайшее время.
  pass-request-data-in-init: # Конфигурация передачи requestData при инициализации виджета
    enable: true
    field-name: data #префикс параметра requestData в url запросе. Пример
data.name=Oleg&data.surname=Sidorov
  build-url:
    server: localhost # protocol+hostname+port (пример: http://hostname:9999) или
    protocol+ip+port (пример: http://127.0.0.1:1), используется для формирования url до
    канала. Является опциональным параметром, если не указан, то будет использоваться текущий
    IP-адрес ChatWidget.
  build-url:
    server: #настройка для формирования uri при построении ссылки на каналы в serviceweb.
    По умолчанию значение не заполнено, будет выбрано автоматически по настройкам системы.
    Пример заполненного значения: http://127.0.0.1:9999

file-server:
  host: localhost
  port: 9320
  timeout: 60000
file:
  max-size-kb: 50000
  unexpected-ext: [exe, dll, php, jar, js, html, css]
  special-chars: "#!@()*&\\%/:<>'|\"~`?^"
  invalid-file-message: К сожалению файл не отвечает нашим требованиям

mongodb:
  hosts:
    - localhost
  name: wa
  add-extra-class-field: true
rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 5
  prefetch: 10

spring:
  main:
    show_banner: false
```

```
jmx:
  enabled: true

redirect-to-operator:
  enabled: false
  always: false
  key-phrases-rule: #ключевые фразы перевод на оператора
    enabled: false
    key-phrases:
      - Перевод на оператора.
  flag-rule: # redirect to operator by reaction 'o:'
    enabled: true
  type-rule: # redirect to operator by response message (rejectMessage or
ProcessingError)
    enabled: true

xmpp:
  host:
  port:
  start-encrypted:
  domain:
  subdomain:
  secret-key:
  multiple-allowed: true
  resource: chatwidget
  operator-jid:
  status:
    online: В сети
  history:
    botname: Чат-Бот
    username: Пользователь
#   https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html
    timestamp-format: YYYY-MM-dd HH:mm
    line-delimiter: "\n"
    template:
      header: |
        История диалога (${sessionId}):
      user: |
        (${timestamp}) ${redirect-to-operator.xmpp.history.username}: ${text}
      bot: |
        (${timestamp}) ${redirect-to-operator.xmpp.history.botname}: ${text}
  component:
    name: ChatNavigator
    description: Интеграция с ChatNavigator
    max-threadpool-size: 17
    max-queue-size: 1000
    enforce-iq-result: true

logging:
  access-log:
  annotation-enabled: true

healthcheck:
  dependent-component-timeout-ms: 30000
  mongo-timeout-ms: 1000
```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер выполните только шаг 2.

1. В файле `/etc/speechpro/chatwidget/application.yml` укажите:

```
rabbitmq:
  host: mqhost
```

```
mongodb:
  name: chatwidget
  hosts:
    - mghost
```

mghost и **mghost** — имена или IP-адреса хостов, где установлены RabbitMQ и MongoDB соответственно.

2. Запустите программу:

```
systemctl start chatwidget.service
```

4.5.5.1 Кастомизация ChatWidget

1. Создайте **siteconfig.yml**:

```
vi /etc/speechpro/chatwidget/siteconfig.yml
```

2. Скопируйте нужные параметры из примера:

```
chatwidget:
  siteConfigs:
    - appKey: <channel_name_here> #для каждого канала секция настраивается отдельно,
      если не указать — конфигурация будет применена для всех каналов
      widget:
        name: 'Чат-Бот' # заголовок виджета
        botName: 'Бот' # имя бота в блоке сообщения
        userName: 'Я' # имя пользователя в блоке сообщения
        greeting: 'Добрый день!' # приветствие у кнопки открытия виджета;
        отображается, если задан avatarImg
        inputPlaceholder: 'Ваш вопрос' # плейсхолдер поля ввода сообщения
        loadHistoryMessage: 'Загрузить историю сообщений' # текст кнопки загрузки
        истории сообщений
        feedbackMsg: 'Помог ли вам ответ?' # текст сообщения фидбека
        feedbackMsgYes: 'Да' # текст положительного фидбека
        feedbackMsgNo: 'Нет' # текст отрицательного фидбека
        sendFileMsg: 'Отправка файла...' # текст при отправке файла
        bigFileErrorMsg: 'Ошибка отправки. Файл слишком большой.' # ошибка отправки
        файла
        sendButtonText: 'OK' # текст кнопки отправки сообщения
        avatarDiameter: '70px' # диаметр аватарки; отображается, если задан
        avatarImg
        avatarImg: 'userpic.png' # изображения для аватара
        bodyBgImg: 'bg.png' # изображение фона
        position: ['top', 'left'] # расположение виджета (если не задано, то виджет
        внизу справа)
        titleImg: 'request-init-icon.png' # изображение в шапке виджета (перед
        названием)
        titleSecondaryImg: 'dog-icon.png' # изображение в шапке виджета (после
        названия)
        titleReplaceImg: 'gif/loading-icon.gif' # замена второго изображения в шапке
        виджета
        titleReplaceImgTimeoutMs: 20000 # тайм-аут замены второго изображения
        chatFontSize: '16px' # размер текста сообщений
        inputFontSize: '14px' # размер текста поля ввода сообщения
        bodyBgColor: '#e6fff9' # цвет фона
        chatBgColor: '#afc8f3' # цвет фона сообщений виджета (и вопросов, и ответов)
        chatFontColor: '#42742c' # цвет текста сообщений виджета
        chatQuestionBgColor: '#fff' # цвет фона сообщений-вопросов
        chatAnswerBgColor: '#fff' # цвет фона сообщений-ответов
        headBgColor: '#6bc1e4' # цвет фона хедера виджета
        headFontColor: '#fff' # цвет текста хедера виджета
        mainBgColor: '#eee' # цвет фона виджета
        inputBgColor: '#0c4d8a' # цвет фона поля ввода сообщения
        sendButtonColor: '#f50a0a' # цвет кнопки для отправки сообщения
```

```

sendButtonTextcolor: '#FFFFFF' # цвет текста кнопки для отправки сообщения
scrollColor: '#24b3ed' # цвет полосы прокрутки в окне виджета
font: 'Open Sans' # шрифт всех текстовых элементов виджета
greetingFont: 'Open Sans' # шрифт приветственного сообщения; если задан и
greetingFont, и font, то применится greetingFont
inputFont: 'Open Sans' # шрифт текст поля ввода сообщения; если задан и
inputFont, и font, то применится inputFont
chatFont: 'Open Sans' # шрифт текста сообщений; если задан и chatFont, и
font, то применится chatFont
headerFont: 'Open Sans' # шрифт текста в хедере виджета; если задан и
headerFont, и font, то применится headerFont
fileUploading: false # показывать кнопку загрузки файлов
showAvatar: true # закрепить аватар
shadowCloseButton: true # дополнительная кнопка закрытия виджета (при
отсутствии хедера)
invertGreeting: true # поменять местами приветствие и аватарку
showBgTitle: false # показывать лого компании
horizontalStretch: true # растянуть виджет по горизонтали
verticalStretch: true # растянуть виджет по вертикали
showMessageTime: false # показывать время отправки и получения сообщений
messageBoxArrow: false # добавить стрелку-капельку для блока сообщения
# цвет стрелки задается параметрами
chatQuestionBgColor и chatAnswerBgColor
linkAsButton: false # оформить список в виде кнопок
messageHeader: true # добавить хедеры для каждого сообщения (отображается
имя и аватар отправителя, время отправки)
# аватары отправителей должны храниться в папке с изображениями
# как аватар робота должно использоваться изображение avatar-wa-a-lk.svg
# как аватар клиента должно использоваться изображение avatar-wa-qf.png
hideHeader: true # скрыть хедер виджета
disableTransition: true # убрать визуальные эффекты открытия и закрытия
alwaysOpen: true # всегда отображать виджет
responsive: true # включить адаптивный дизайн для мобильных устройств
maxInputLength: 140 # максимальное количество символов в поле ввода вопроса
draggable: true # перетаскивать окно виджета
closeOnHeader: true # отображать иконку закрытия в шапке виджета
techErrorMsg: 'Сервис временно недоступен' # сообщение баннера в случае
недоступности виджета
hideInput: false # скрыть поле ввода сообщения

```



Путь к папке с изображениями: `/opt/speechpro/chatwidget/webapp/img`.

Если задаются параметры настройки шрифтов, то для каждого нового используемого шрифта необходимо добавить в файл **fonts/fonts.css** описание набора шрифтов следующего вида:

```

@font-face {
  font-family: "Open Sans"; // название обязательно соответствует значению, указанному в
siteconfig
  src: url("/fonts/OpenSans-Regular-webfont.woff2") format("woff2"), // месторасположение
шрифта на локальной машине или в Интернете
      url("/fonts/OpenSans-Regular-webfont.woff") format("woff"); // через запятую
указываются различные форматы подключаемых шрифтов
}

```

Шрифты должны быть добавлены в `/opt/speechpro/chatwidget/webapp/fonts`.

3. Запустите утилиту обновления конфигурации **ChatWidget**, считывающую информацию из файла конфигураций и сохраняющую в **MongoDB**.

Путь к файлу конфигураций задается в качестве аргумента при вызове скрипта (**siteconfig.sh**, пример запуска смотрите далее) либо, в случае отсутствия аргумента, берется путь по умолчанию (`/etc/speechpro/chatwidget/siteconfig.yml`). Утилита запускается файлом **siteconfig.sh** либо через класс

UpdateSiteConfigUtil.java. При запуске утилиты происходит удаление сохраненных ранее конфигураций и сохранение новых из файла.

Путь к утилите: `/opt/speechpro/chatwidget/bin/siteconfig.sh`.

Пример запуска утилиты без аргументов, путем до файла `siteconfig.yml` будет `/etc/speechpro/chatwidget/siteconfig.yml`:

```
cd /opt/speechpro/chatwidget/bin/ && ./siteconfig.sh
```

Пример запуска утилиты с аргументом:

`"./siteconfig.sh /opt/speechpro/siteconfig/siteconfig.yml"` → путем до файла `siteconfig.yml` будет `"/opt/speechpro/siteconfig/siteconfig.yml"`

4. Рекомендуется сохранить содержание базы во внешний файл:

```
mongodump --username=mongoadminadmin --password=mongoadminpass --
authenticationDatabase=admin --db=chatwidget --out=/root/back_mongo/chatwidget_$(date +%Y-%m-%d)
```

5. Если нужно изменить параметры в `siteconfig.yml`, достаточно отредактировать его параметры и заново запустить скрипт из шага 3, чтобы изменения сохранились. Чтобы не затрагивать все каналы, можно выполнить скрипт с аргументом **appKey** только для того канала из `siteconfig.yml`, для которого делались изменения:

```
./siteconfig.sh --appKey test
```

4.5.5.2 Интеграция ChatWidget с сайтом

При необходимости интеграции виджета с сайтом подключите **jQuery** и в head страницы веб-ресурса скопируйте содержимое ниже:

```
<script> $(document).ready(function ()
{ var e = document.createElement('script'); e.async = true; e.setAttribute('src',
'https://<url>'); document.body.appendChild(e); }
); </script>
```

Где `https://<url>` — ссылка на js-файл (`https://адрес_стенда/js/wa-min.js`).

Добавьте настройки **CORS** для **ChatWidget** в файл `/etc/speechpro/chatwidget/application.yml`:

```
cors:
  enabled: true
  origins: '*'
  methods: '*'
  headers: '*'
```

4.5.6 OmnichatAdapter

Конфигурационный файл

```
omnichatadapter:
  processing-time-warn-threshold-ms: 2000
  response:
    question:
      sync:
        timeout-enabled: true
        timeout-sec: 300 #время ожидания обработки вопроса
        timeout-check-interval-ms: 10000 #интервал проверки вопросов
        max-thread-wait-timeout-factor: 1.5
      async:
        timeout-enabled: true
```

```

        timeout-sec: 300
        timeout-check-interval-ms: 10000
    http-timeout: 1000
    connections:
        max-total: 80
        max-per-route: 20
    proxy:
        enabled: false
    retry:
        max-attempts: 3
        backoff-period-ms: 1000
        thread-count: 20
    reply:
        default-formatting: empty
        formattings:
            - name: omnichat-html
              default-configuration:
                replacement-patterns:
                    - pattern: "!?\[\([^\]]*>\]\)\[([^\ ]*>\]) ( \"[^\\"(\)]\\\" ]+\") ?\]"
                      replace-to: "<a href=\"\$2\">\$1</a>"
                    - pattern: "\\n"
                      replace-to: "<br>"
                    - pattern: "\\*#"
                      replace-to: "*"
            - name: empty
              defaultConfiguration:
                replacementPatterns:
                    - pattern:
                      replace-to:
        thread-count: 1 #число потоков (пул), обрабатывающих сообщения, полученные из
rabbitmq
    session:
        timeout-enabled: true
        timeout-sec: 7200
        timeout-check-interval-sec: 60
    file-response-type: multipart
server:
    port: 9066
    servlet:
        encoding:
            enabled: true
            charset: UTF-8
            force: true
management:
    endpoints:
        web:
            exposure:
                include: [health, prometheus] # перечисление эндпоинтов, которые будут доступны
по запросу GET /actuator/<endpoint_name>
    metrics:
        export:
            prometheus:
                enabled: false # эндпоинт prometheus по умолчанию не доступен, для получения
метрик нужно проставить значение true

spring:
    main:
        banner-mode: 'off'
    http:
        encoding:
            enabled: true
            charset: UTF-8
            force: true
        converters:
            preferred-json-mapper: jackson
    servlet:

```

```

    multipart:
      max-file-size: -1
      max-request-size: -1
      file-size-threshold: 0
rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 8
mongodb:
  name: omnichatadapter
  hosts:
    - localhost
  write-concern: ACKNOWLEDGED
  min-connections-per-host: 20
  max-connections-per-host: 300
  threads-allowed-to-block-for-connection-multiplier: 5
  max-wait-time-ms: 5000
  max-connection-idle-time-ms: 1800000
  max-connection-life-time-ms: 7200000
  socket-timeout: 5000
  add-extra-class-field: true

redirect-to-operator:
  key-phrases:
    - Перевод на оператора.
redirect-to-csi:
  key-phrase: запросить csi.
error-code:
  key-phrases:
    - техническая ошибка
    - технических проблем

channels:
  channel-prefix: omnichat-adapter
  default-channels:
    enabled: true
    action: add_if_not_exist
    channels:
      - channel-id: omnichat-adapter
        response-endpoint:
          host: localhost
          port: 9077
          path: /
          scheme: http

prompt:
  enabled: true #включен запрос подсказок или нет (если параметр выключен, запросы на
  ботсервер/промптерадаптер все равно могут уходить (для записи истории))
  prompter: true #получим ли мы подсказки от промптера (суфлера)
  bot: true #получим ли мы подсказки от бота (правил)
  prompt-on-redirect-to-operator-enabled: false # если вопрос пользователя привел к
  переводу на оператора, то будет ли по этому вопросу запрос подсказок (работает только в
  режиме always)
  prompt-mode: afterQuestion # режим запроса подсказок. Возможные значения:
  onQuestion/afterQuestion/always
                                # onQuestion - получим подсказки только по запросу подсказок
от оператора (специальный тип запроса)
                                # afterQuestion - получим подсказки
                                # * на первый запрос подсказки от оператора
                                # * на все реплики пользователя после первого запроса
подсказки от оператора

```

```

# * если запрос подсказки в сессии был, то все
последующие запросы подсказки игнорируются (подсказки не возвращаются и ничего не пишется
в историю диалога)
# always - поведение как раньше (запрос подсказки на вопросы
пользователя и игнорирование запросов подсказок от оператора)

healthcheck:
  dependent-component-timeout-ms: 30000
  mongo-timeout-ms: 1000
  filesaver:
    enabled: true
  prompteradapter:
    enabled: true

event-group:
  # Параметры по обработке групп сообщений от botServer. Необходимо
  для синхронизации порядка отправляемых пользователю сообщений
  retry-count: 3      # Количество попыток на обработку текущей группы. В случае
  превышения лимита, система проигнорирует тот факт, что прошлая группа еще не
  обработалась, и отправит ответ по текущей группе
  retry-timeout: 250  # Пауза в мс между попытками.

logging:
  access-log:
    annotation-enabled: true

```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага [2](#).

1. В файле `/etc/speechpro/omnichatadapter/application.yml` укажите:

```

rabbitmq:
  host: mqghost
mongodb:
  hosts:
    - mgghost

```

mqghost и **mgghost** — имена или IP-адреса хостов, где установлены RabbitMQ и MongoDB соответственно.

2. Для создания каналов укажите в файле `/etc/speechpro/omnichatadapter/application.yml`:

```

channels:
  channel-prefix: omnichat-adapter
  default-channels:
    enabled: true
    action: add_if_not_exist
    channels:
      - channel-id: omnichat-adapter
        response-endpoint:
          host: tpsystem
          port: 9077
          path: /
          scheme: http

```

Чтобы каналы адаптеров отображались в интерфейсе, добавьте их в блок `dedicatedChannels` файла `/etc/speechpro/serviceweb/application-integration.yml`.

Параметр **tpsystem** — IP-адрес или имя сторонней системы, куда будут поступать ответы. В совокупности с параметрами **port**, **path** и **scheme** из конфигурационного файла выше **endpoint-URI** будет выглядеть как **http://tpsystem:9077/**.

3. Запустите программу:

```
systemctl start omnichatadapter.service
```

4.5.6.1 Настройка форматирования ответов для OmnichatAdapter

Настройки форматирования ответов задаются в конфигурационном файле **/etc/speechpro/omnichatadapter/application.yml**.

В конфигурации компонента-адаптера **OmnichatAdapter** можно задать правила форматирования ответов для каждого канала адаптера и настроить форматирование в зависимости от значений параметров для каждого канала.

Также есть возможность задать форматирование по умолчанию для компонента, которое будет применяться для всех каналов.

Настройка форматирования по умолчанию для всех каналов:

```
omnichatadapter:
  response:
    reply:
      default-formatting: empty
      formattings:
        - name: omnichat-html
          default-configuration:
            replacement-patterns: #замены выполняются по порядку
            - pattern: "!?\[\([^\]]*>\]\)\[\([^\]]*>\]\)\(["'`^\[\]\(\)\|\\"]+\")?\\\" "
              replace-to: "<a href=\"\${2}\">\${1}</a>"
            - pattern: "\\n"
              replace-to: "<br>"
            - pattern: "\\*#"
              replace-to: "*"
# Условное форматирование. Это возможность добавить доп. форматирование в
# зависимости от параметров requestData.
# conditions:
# - conditions:
#   fields:
#     - field: sender_id
#       value: specified
#   replacement-patterns:
#     - pattern: "sender_id"
#       replace-to: "SenderIdReplayced"
- name: empty #форматирование ответов отсутствует.
  default-configuration:
    replacement-patterns:
      - pattern:
        replace-to:
```

Настройка форматирования для конкретных каналов:

```
channels:
  default-channels:
    channels:
      - channel-id: omnichat-html
        response-formatting:
          use: omnichat-html # Указание, какую конфигурацию требуется использовать для
канала; валидируется наличие форматирования в списке formattings
```

```

- channel-id: omnichat-html-formatting-inheritance
  response-formatting:
    use: omnichat-html
    # Добавляет к указанной в use конфигурации замены для использования на этом
канале
    # Порядок применения форматирования: сначала срабатывает форматирование для
канала, затем форматирование по умолчанию
    default-configuration:
      replacement-patterns:
        - pattern: "text"
          replace-to: "change"

- channel-id: omnichat-custom
  response-formatting:
    # Если use не описывается, то создается новая конфигурация для канала.
Переиспользовать такую конфигурацию нельзя
    default-configuration: # Если конфигурация описывается новая, блок default -
обязателен
      replacement-patterns:
        - pattern: "text"
          replace-to: "change"

```

Настройка форматирования по входным параметрам запросов:

```

omnichatadapter:
  response:
    reply:
      default-formatting: empty # Указание форматирования, применяемого к ответу, если
для канала не указано использовать другое
      formattings:
        - name: conditional
          conditions: # Опциональный блок, если срабатывает один из condition, то
применяется форматирование из condition.replacement-patterns (default блок игнорируется)
            - condition: # Условия для входящего запроса. Если запрос проходит по
параметрам - срабатывают замены для этого condition. Если срабатывает condition, дефолт и
другие условия применены не будут
              fields: # Условие по полям запроса генерируется по И, условие по
нескольким разным condition генерируется по ИЛИ
                - field: ChannelType
                  value: 2
              replacement-patterns: # Конфигурация паттернов на замену; будут применены
к ответу, если запрос пройдет по параметрам
                - pattern: "ChannelType"
                  replace-to: "ChannelTypeReplayed"
            default-configuration:
              replacement-patterns:
                - pattern: "text"
                  replace-to: "replayed text"

channels:
  default-channels:
    channels:
      - channel-id: omnichat-html-formatting-inheritance
        response-formatting:
          use: omnichat-html
          conditions: # Добавляет к указанной в use конфигурации дополнительные условия
для использования на этом канале
            - condition: # Сначала применяются condition из конфигурации для канала,
затем из конфигурации по умолчанию. При срабатывании первого condition, остальные
игнорируются
              fields:
                - field: MetaData.field.value
                  value: fieldValue
                - field: MetaData.value
                  value: value
                - field: channel_name

```

```

        value: test-channel
        replacement-patterns: # Конфигурация паттернов на замену; будут применены
к ответу, если запрос пройдет по параметрам
        - pattern: "MetaData"
          replace-to: "MetaData replayced"
        default-configuration: # Добавляет к указанной в use конфигурации замены для
использования на этом канале
        replacement-patterns:
        - pattern: "text"
          replace-to: "change"

    - channel-id: omnichat-custom # Пример описания новой конфигурации для конкретного
канала с использованием conditions
      response-formatting:
        conditions:
        - condition:
            fields:
            - field: MetaData.field.value
              value: fieldValue
            - field: MetaData.value
              value: value
            replacement-patterns:
            - pattern: "MetaData"
              replace-to: "MetaData replayced"
        default-configuration: # Если конфигурация описывается новая, блок default -
обязателен
        replacement-patterns:
        - pattern: "text"
          replace-to: "change"

```

Для **REST API** недоступны для использования в **condition** следующие поля:

- API 1 версии: **ApiVersion**, **sessionId**, **message**, **timestamp**, **questionNumber**, **replyTo**;
- API 2 версии: **ApiVersion**, **requestId**, **sessionId**, **requestTime**, **questionNumber**, **replyTo**, **data**, **agent**, **errors**, **crossLinks**.

4.5.7 SipAdapter

Конфигурационный файл

```

sipadapter:
  processing-time-warn-threshold-ms: 2000
  prompt-on-redirect-to-operator-enabled: false
  response:
    question:
      sync:
        timeout-enabled: true
        timeout-sec: 300
        timeout-check-interval-ms: 10000
        max-thread-wait-timeout-factor: 1.5
      async:
        timeout-enabled: true
        timeout-sec: 300
        timeout-check-interval-ms: 10000
  http-timeout: 1000
  connections:
    max-total: 80
    max-per-route: 20
  proxy:
    enabled: false
  retry:
    max-attempts: 3

```

```

    backoff-period-ms: 1000
  reply:
    default-formatting: empty
    formattings:
      - name: sip-html
        default-configuration:
          replacement-patterns:
            - pattern: "!?\\[[([\\^\\](<>]+)\\]\\\\\\\\([\\^ \\](<>]+) ( \\\"[\\^\\\\(\\\\)\\\\\\\\\"]+\\\")?\\\\) \"
              replace-to: "<a href=\\\"$2\\\">$1</a>"
            - pattern: "\\n"
              replace-to: "<br>"
            - pattern: "\\*#"
              replace-to: "*"
      - name: empty
        default-configuration:
          replacement-patterns:
            - pattern:
              replace-to:
    thread-count: 1 #число потоков (пул), обрабатывающих сообщения, полученные из
rabbitmq
  session:
    timeout-enabled: true
    timeout-sec: 7200
    timeout-check-interval-sec: 60
  flow-rule:
    type: manual
    flow: BOT

server:
  port: 9098
spring:
  main:
    banner-mode: 'off'
  http:
    encoding:
      enabled: true
      charset: UTF-8
      force: true
    converters:
      preferred-json-mapper: jackson
  servlet:
    multipart:
      max-file-size: -1
      max-request-size: -1
      file-size-threshold: 0
rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 8
mongodb:
  name: sipadapter
  hosts:
    - localhost
  write-concern: ACKNOWLEDGED
  min-connections-per-host: 20
  max-connections-per-host: 300
  threads-allowed-to-block-for-connection-multiplier: 5
  max-wait-time-ms: 5000
  max-connection-idle-time-ms: 1800000
  max-connection-life-time-ms: 7200000
  socket-timeout-ms: 5000
  add-extra-class-field: true

```

```
redirect-to-operator:
  key-phrases:
    - Перевод на оператора.
redirect-to-csi:
  key-phrase: запросить csi.
error-code:
  key-phrases:
    - техническая ошибка
    - технических проблем

validate-request-message:
  enabled: false
  patterns:
    - ^[?]{5,}
  replace-message: 63f31db1327f4d5a9b3961d1e03b9837
  test-messages:
    - ??????
    - Тестовое сообщение

channels:
  channel-prefix: sip-adapter
  default-channels:
    enabled: true
    action: add_if_not_exist
    channels:
      - channel-id: sip-adapter
        response-endpoint:
          host: localhost
          port: 9099
          path: /
          scheme: http

healthcheck:
  dependent-component-timeout-ms: 30000
  mongo-timeout-ms: 1000

event-group:                # Параметры по обработке групп сообщений от botServer. Необходимо
для синхронизации порядка отправляемых пользователю сообщений
  retry-count: 3             # Количество попыток на обработку текущей группы. В случае
превышения лимита, система проигнорирует тот факт, что прошлая группа еще не
обработалась, и отправит ответ по текущей группе
  retry-timeout: 250         # Пауза в мс между попытками.

logging:
  access-log:
    annotation-enabled: true
```



О порядке установки и настройки голосовой платформы **VoiceKey.PLATFORM** для работы совместно с **AgentNavigator** по протоколу SIP см. документ «VoiceKey.IVR Подсистема телефонного взаимодействия по протоколу SIP. Руководство по установке и настройке НЦДА.00744-01 93».

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага [2](#).

1. В файле `/etc/speechpro/sipadapter/application.yml` укажите:

```
rabbitmq:
  host: mqhost
mongodb:
  hosts:
```

```
- mghost
```

mghost, **mghost** — имена или IP-адреса хостов, где установлены RabbitMQ и MongoDB соответственно.

2. Для создания каналов укажите в файле `/etc/speechpro/sipadapter/application.yml`:

```
channels:
  channel-prefix: sip-adapter
  default-channels:
    enabled: true
    action: add_if_not_exist
    channels:
      - channel-id: sip-adapter
        response-endpoint:
          host: localhost
          port: 9099
          path: /
          scheme: http
```

Чтобы каналы адаптеров отображались в интерфейсе, добавьте их в блок [dedicatedChannels](#) файла `/etc/speechpro/serviceweb/application-integration.yml`.

3. Запустите программу:

```
systemctl start sipadapter.service
```

4.5.8 ConfigService

Конфигурационный файл

```
server:
  port: 9078
spring:
  application:
    name: config-service
  profiles:
    active:
      - native
  cloud:
    config:
      server:
        jdbc:
          enabled: false
        native:
          search-locations:
            - classpath:/config-repo # путь до директории с настройками сервисов
```

Интеграции сервиса с ConfigService

Для интеграции сервиса с **ConfigService** добавьте в локальные настройки:

```
spring:
  application:
    name: change-this-service-name # имя сервиса
  config:
    import: optional:configserver:http://localhost:9078 # путь до config-service
```

Если получение настроек из **ConfigService** является обязательным, то уберите **optional:** из `spring.config.import`.

Настройте тайм-ауты и Retry (подробнее в [официальной документации Spring Cloud Config](#)).

Приоритет конфигурационных файлов:

1. *{application-name}-{profile-name}.yml* на config-service
2. *application-{profile-name}.yml* на config-service
3. *{application-name}.yml* на config-service
4. *application.yml* на config-service
5. Локальные настройки сервиса

Доступные настройки

1. *application-llm-models.yml* (их используют `HttpAdapter`, `BotServer`, `ServiceWeb`):

```
llm:
  models:
    - id: GigaChat-Pro
      name: "GigaChat Pro"
      vendor: GIGA_CHAT
    - id: GigaChat
      name: "GigaChat Lite"
      vendor: GIGA_CHAT
    - id: GigaChatPlus
      name: "GigaChat Lite+"
      vendor: GIGA_CHAT
```

2. *application-llm-sdk-core-client.yml* (их используют `HttpAdapter`, `BotServer`):

```
spring:
  cloud:
    openfeign:
      client:
        config:
          llm-core:
            url: http://localhost:9999
```

2. *application-llm-services.yml* (их используют `HttpAdapter`, `ServiceWeb`):

```
spring:
  cloud:
    openfeign:
      client:
        config:
          giga-chat-auth-beta:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 10_000
          giga-chat-completions-beta:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          giga-chat-auth-prod:
            url: http://localhost:9443
            connect-timeout: 1_000
            read-timeout: 10_000
          giga-chat-completions-prod:
            url: http://localhost:9443
            connect-timeout: 1_000
            read-timeout: 30_000
          yandex-gpt-completions:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 10_000
```

```
llm:
```

```

default-vendor: GIGA_CHAT
giga-chat:
  api-type: PROD
  auth:
    username: someUser
    password: somePassword
  prod:
    scope: GIGACHAT_API_CORP
  retry:
    max-attempts: 1
    back-off-period-ms: 100
  uri: /api/v2/oauth
completions:
  model-id: GigaChat-Pro
  model-name: GigaChat-Pro
  profanity-check: true
  top-p:
  repetition-penalty:
  temperature:
  uri: /api/v1/chat/completions
  retry:
    max-attempts: 1
    back-off-period-ms: 100
yandex-gpt:
  api-key: someApiKey
  completions:
    dir-id: someDirId
    model-id: yandexgpt/latest
    model-name: YandexGPT Pro
    uri: /foundationModels/v1/completion
  retry:
    max-attempts: 2
    back-off-period-ms: 100

```

В этой, объединенной конфигурации отсутствуют настройки `llm.giga-chat.enabled`, `llm.yandex-gpt.enabled` для того, чтобы уже непосредственно решить вопрос о включение этих интеграций. Но в общем случае, они также могут быть указаны в общем файле конфигурации(в `config-service`)

4.5.9 Telegram

Конфигурационный файл

```

server:
  port: 8087

telegram:
  url: https://api.telegram.org
# url: https://149.154.167.200
  http-timeout: 60000
  session-timeout: 1800000
  session-auto-cleanup-interval: 2400000
  botan-url: https://api.botan.io/track
  proxy:
    enabled: false
    host:
    port:
    excluded-hosts: [localhost, 127.0.0.1]
  get-me:
    retry-interval-ms: 1000
    max-retry-count: 3600
  web-hook:
    enabled: false

rabbitmq:

```

```

host: localhost
port: 5672
vhost: /
user: guest
password: guest
thread-count: 1
prefetch: 8
mongodb:
  name: telegram
  hosts:
    - localhost
  write-concern: ACKNOWLEDGED

telegram: #Настройка интеграции
  channel-to-token:
    <channelId>: <token> (например, telegram:
251332029:AAHcPPTmaybIgwLYQwn5uSgdF2oeRMpDXJM)
  web-hook:
    enabled: false #optional: false
  proxy: #optional
    enabled: true #optional: false
    host: proxy.stc
    port: 3128

healthcheck:
  dependent-component-timeout-ms: 30000
  mongo-timeout-ms: 1000

```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага [2](#).

2. В файле `/etc/speechpro/telegram/application.yml` укажите:

```

rabbitmq:
  host: mqhost
mongodb:
  hosts:
    - mgghost

```

mqhost и **mgghost** — имена или IP-адреса хостов, где установлены **RabbitMQ** и **MongoDB** соответственно.

2. В файле `/etc/speechpro/telegram/application.yml` укажите:

```

telegram:
  channel-to-token:
    <channelId>: <token> (например, telegram:
251332029:AAHcPPTmaybIgwLYQwn5uSgdF2oeRMpDXJM)
  web-hook:
    enabled: false #optional: false

```

Параметр **telegram** — имя канала, который будет создан автоматически при запуске компонента, а через двоеточие указан токен API-мессенджера.

3. Запустите программу:

```
systemctl start telegram.service
```

4.5.10 CrafttalkAdapter

Конфигурационный файл

```
server:
  port: 9076
  servlet:
    encoding:
      enabled: true
      charset: UTF-8
      force: true
spring:
  main:
    banner-mode: 'off'
  mvc:
    converters:
      preferred-json-mapper: jackson
crafttalkadapter:
  commands:
    redirect-to-operator:
      template: "/switchredirect aiassist2_crt intent_id=\"%s\"" # команда для перевода
на оператора в чат-платформу craftTalk (идентификатор агента, включен в описание команды)
      default-intent-id: "route_default" #значение intent_id по умолчанию
    finish-dialog:
      template: "/switchredirect aiassist2_crt intent_id=\"%s\"" # команда для передачи
признак окончания диалога в чат-платформу CraftTalk (идентификатор агента, включен в
описание команды)
      default-intent-id: "error" #значение intent_id по умолчанию
  processing-time-warn-threshold-ms: 2000
  response:
    question:
      async:
        timeout-enabled: true
        timeout-sec: 300
        timeout-check-interval-ms: 10000
  http-timeout: 1000
  connections:
    max-total: 80
    max-per-route: 20
  proxy:
    enabled: false
  retry:
    max-attempts: 3
    backoff-period-ms: 1000
    thread-count: 20
  reply:
    default-formatting: empty
    formattings:
      - name: craft-talk-adapter-html
        default-configuration:
          replacement-patterns:
            - pattern: "!\?\[\([^\]\<>]+\)\]\[\([^\ \]\<>+\) ( \"[^\(\\)\]\<>]+\")?\]"
              replace-to: "<a href=\"%2\">$1</a>"
            - pattern: "\\n"
              replace-to: "<br>"
            - pattern: "\\*#"
              replace-to: "*"
      - name: empty
        default-configuration:
          replacement-patterns:
            - pattern:
              replace-to:
  thread-count: 1 #число потоков (пул), обрабатывающих сообщения, полученные из
rabbitmq
```

```

session:
  timeout-enabled: true
  timeout-sec: 6000
  timeout-check-interval-sec: 60
authentication:
  key-cloak:
    - auth-service-id: key-cloak
      host: localhost
      port: 80
      path: /auth/realms/LDAP
      scheme: https
      login:
      password:
      client-id:
      client-secret:
    craft-talk-api:
      - auth-service-id: craft-talk-api
        host: localhost
        port: 443
        path: /api/external/auth/authorize
        scheme: https
        login:
        password:
        # Список точек авторизации типа keyCloak
        # Уникальный идентификатор точки.

Используется при настройке каналов

rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 8

mongodb:
  name: crafttalkadapter
  hosts:
    - localhost
  write-concern: ACKNOWLEDGED
  min-connections-per-host: 20
  max-connections-per-host: 300
  threads-allowed-to-block-for-connection-multiplier: 5
  max-wait-time-ms: 5000
  max-connection-idle-time-ms: 1800000
  max-connection-life-time-ms: 7200000
  socket-timeout-ms: 5000
  add-extra-class-field: true

channels:
  channel-prefix: craft-talk-adapter
  default-channels:
    enabled: true
    action: add_if_not_exist
    channels:
      - channel-id: craft-talk-adapter
        auth-service-id: craft-talk-api # Идентификатор точки авторизации
        response-endpoint: #описание точки для обратных запросов. Данные представлены в
        # качестве примера
        host: localhost
        port: 9087
        path: /
        scheme: http
  response-message:
    operator:
      operator-id: __SYSTEM__
      operator-fname: System

```

```
operator-lname: System
dialog-id:
```

```
logging:
  access-log:
    annotation-enabled: true
```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага [2](#).

1. В файле `/etc/speechpro/crafttalkadapter/application.yml` укажите:

```
rabbitmq:
  host: mqhost
mongodb:
  hosts:
    - mqhost
```

Параметр **tpssystem** — IP-адрес или имя сторонней системы, куда будут поступать ответы. В совокупности с параметрами **port**, **path** и **scheme** из конфигурационного файла выше **endpoint-URI** будет выглядеть как **`http://tpssystem:9077/`**.

mqhost и **mqghost** — имена или IP-адреса хостов, где установлены RabbitMQ и MongoDB соответственно.

2. Для создания каналов укажите в файле `/etc/speechpro/crafttalkadapter/application.yml`:

```
channels:
  - channel-id: craft-talk-adapter
    auth-service-id: craft-talk-api # Идентификатор точки авторизации
    response-endpoint: #описание точки для обратных запросов. Данные представлены в
качестве примера
    host: localhost
    port: 9087
    path: /
    scheme: http
```

Чтобы каналы адаптеров отображались в интерфейсе, добавьте их в блок [dedicated-channels](#) файла `/etc/speechpro/serviceweb/application-integration.yml`.

```
integration:
  vcservices:
    crafttalkadapter:
      service-name: crafttalkadapter
      dedicated-channels:
        - {id: craft-talk-adapter, name: craft-talk-adapter}
      nodes:
        - {ip: 127.0.0.1, port: 9076}
```

3. Запустите программу:

```
systemctl start crafttalkadapter.service
```

4.5.11 Classifier и Classifier Trainer



При обновлении на новую версию **AgentNavigator** следует очищать каталог с комплектами моделей, а также удалять содержимое коллекции `classifier.revisions` базы данных **MongoDB**.

Скачайте архив с эмбедингами по ссылке:
<https://ftp.speechpro.com/puckr/vc/embeddings/embeddings.tgz>, распакуйте файлы.

Создайте папку `/opt/speechpro/resources/classifier_embeddings/cnn`.

Добавьте в папку `/opt/speechpro/resources/classifier_embeddings/cnn` файлы эмбедингов, необходимые для обучения на CNN-модели:

- `gensim_cbow_300d_20mc_30e_12t_0.025a_0.015ma_tf1_lk4_bln.npy`;
- `gensim_cbow_300d_20mc_30e_12t_0.025a_0.015ma_tf1_lk4_bln.dic`.

Создайте папку `/opt/speechpro/resources/classifier_embeddings/cnnsubword`.

Добавьте в папку `/opt/speechpro/resources/classifier_embeddings/cnnsubword` файлы эмбедингов, необходимые для обучения на CNNSubword-модели:

- `sw2v_300d_1e_8t_0.025a_4w_15n.npy`, `sw2v_300d_1e_8t_0.025a_4w_15n.dic`;
- `subword.model`.

Файлы эмбедингов поставляются вместе с дистрибутивом.

4.5.12 NER и NER Trainer

Скачайте архив с Encoder.

```
curl -O https://ftp.speechpro.com/puckr/vc/prerequisites/ner_encoder.tar.gz
```

Добавьте в папку `/opt/speechpro/ner_encoder/1.0/ner_encoder/` согласно настройкам конфигурации по умолчанию секции `app.encoder.path`.

Скачайте архив с NLTK.

```
curl -O https://ftp.speechpro.com/puckr/vc/prerequisites/nltk_data.tar.gz
```

Для NER положите по пути `/opt/speechpro/ner/lib/nltk_data/1.0/nltk_data` согласно настройкам конфигурации по умолчанию секции `app.nltk.punkt.path`. Для сервиса NER Trainer положите по пути `/opt/speechpro/ner_trainer/lib/nltk_data/1.0/nltk_data` согласно настройкам конфигурации по умолчанию секции `app.nltk.punkt.path`.

Конфигурационный файл

```
unicorn:
  host: 0.0.0.0
  port: 9010
  workers: 1
  threads: 8
  keepAlive: 30
  timeout: 60

app:
  data:
    path: /opt/speechpro/ner_models
  encoder:
    path: /opt/speechpro/ner_encoder/1.0/ner_encoder
  nltk:
    punkt:
      path: /opt/speechpro/ner/lib/nltk_data/1.0/nltk_data
```

4.5.13 FileServer

Конфигурационный файл

Файл *application.yml*:

```
server:
  port: 9320
spring:
  main:
    show_banner: false
  servler:
    multipart:
      max-file-size: -1    #максимальный размер файла, (-1 неограниченный)
      max-request-size: -1  #максимальный размер запроса, (-1 неограниченный)
      file-size-threshold: 1KB  #порог, после которого файл будет записываться на диск
(порция)
fileserver:
  directory: /etc/speechpro/fileserver/storage  #базовая директория в которую будут
сохраняться файлы
  base-url: http://localhost:9320  #базовый урл, используется при формировании урла (для
скачивания файла) после загрузки файла
  expiration:  #конфиг очистки файлов с mongo и файловой системы
  timeout-sec: 7300  #время при превышении которого, во время очистки, инфо о файле в
БД и сам файл будут удалены
  check-interval-sec: 3600  #интервал между запусками очистки
  pool-size: 1  #количество потоков, выделяемых на выполнение очистки
  smb-uploading-buffer-size-mbytes: 2
  proxy:
    enabled: true  # Если не используется прокси выставить параметр в false
    host: proxy.stc  # конфигурация прокси сервера
    port: 3128
    excluded-hosts: [localhost, 127.0.0.1]  # исключения прокси сервера
  http:
    timeout-ms: 60000  # тайм-аут на получения данных (в случае если файл скачивается
файлсервером по переданному url)
mongodb:
  name: fileserver
  hosts:
    - localhost
  write-concern: ACKNOWLEDGED
logging:
  access-log:
    annotation-enabled: true

healthcheck:
  mongo-timeout-ms: 1000
```

Настройте сетевой доступ к общедоступному каталогу в файле *haConfig.yaml*:

```
haConfigs:
  - methodName: sendFileBySMB
    httpMethod: POST
    urlTemplate: http://localhost:9320/restapi/integration-protocols/smb/upload
    parameters: [id, uploadFolder]
    headers:
      Content-Type: application/json
    bodyTemplate: |
      {
        "fileId": ${id},
        "uploadFolder": ${uploadFolder},
        "connection": {
          "host": "192.168.4.191",
          "sharedFolder": "share",
```

```

        "credentials": {
            "user": "testuser",
            "password": "Zaq12wsx"
        }
    }
}

```

4.5.14 python-http-adapter

Конфигурационный файл

Файл *application.yml*:

```

server:
  port: 8095

script:
  timeout-script-in-second : 30
  scripts-dir-path : file:/home/scripts/
  python:
    executable: "python3"

spring:
  jackson:
    date-format: "yyyy-MM-dd HH:mm:ss.SSS"
    serialization:
      write-dates-as-timestamps: false

```

Сервис поддерживает значения по умолчанию, которые хранятся в конфигурационном файле *application.yml*. Основные настройки включают:

- Путь к директории со скриптами (**script.scripts-dir-path**) — указывает путь к папке, в которой хранятся скрипты, и поддерживает 3 формата пути до файла:
 - путь относительно директории запуска сервиса (префикс **file:**);
 - путь абсолютный (префикс **file:**);
 - путь относительно classpath JVM (префикс **classpath:**).
- Ограничение на время исполнения скриптов (**script.timeout-script-in-second**) — определяет максимальное время выполнения скрипта в секундах. При превышении этого времени будет выброшено исключение, и сервис вернет ответ с HTTP-статусом 422.
- Формат даты для сериализации (**spring.jackson.date-format**) — даты должны передаваться в строковом формате (предпочтительный формат), который поддерживает корректное восприятие скриптом. Если дата передана не в строковом формате, то она будет сериализована по формату [ISO 8061](#) (**yyyy-MM-dd HH:mm:ss**) без часового пояса.

Требования к скриптам

1. Все библиотеки, используемые для работы скрипта, должны быть установлены в систему (глобально). Скрипты не должны зависеть от виртуальных окружений для выполнения.

2. Каждый скрипт должен содержать исполняемую функцию **main**, которая будет точкой входа.

3. При вызове скрипта функция **main** запускается автоматически (например, с использованием условия):

```
if __name__ == "__main__":
```

```
main()
```

4. Внутри функции **main** должен быть реализован парсер командной строки, например с использованием библиотеки **argparse**.

5. Все параметры должны задаваться с префиксом **--**.

```
parser = argparse.ArgumentParser()
parser.add_argument('--par1', type=str, help="First argument")
parser.add_argument('--par2', type=str, help="Second argument")
parser.add_argument('--par3', type=str, help="Third argument")
```

```
args = parser.parse_args()
```

6. Скрипт должен выводить только конечный результат своей работы, без промежуточных или отладочных сообщений в стандартный вывод командной строки.

4.5.15 database-http-adapter

Выполнение процедур в СУБД

Поддерживаемые базы данных

Сервис был протестирован с использованием следующих версий баз данных:

- PostgreSQL 14;
- MS SQL 2017;
- Oracle 23 (free версия), Oracle XE 18 и Oracle XE 21.

Конфигурационный файл

Файл **application.yml**:

```
server:
  port: 8094

logging:
  access-log:
    annotation-enabled: true

spring:
  datasource:
    url: jdbc:postgresql://localhost:5432/database?escapeSyntaxCallMode=call
    username: username
    password: password
  jackson:
    date-format: "yyyy-MM-dd HH:mm:ss.SSS" # https://www.iso.org/iso-8601-date-and-time-
format.html
    serialization:
      write-dates-as-timestamps: false

procedure:
  call-timeout-in-seconds: 60
  defaults-path: classpath:procedure-defaults.json
```

Файл значение по умолчанию

```
{
  "example_proc": {
    "example_arg0": null,
    "example_arg1": 0
  }
}
```

Значения параметров по умолчанию (*procedure.defaults-path*)

Сервис поддерживает значения параметров по умолчанию для заданных процедур. Эти значения и имена процедур конфигурируются в специальном файле значений по умолчанию, указанном в конфигурации (**procedure.defaults-path**). **procedure.defaults-path** поддерживает 3 вида пути до файла:

- путь относительно директории запуска сервиса (префикс file:);
- путь абсолютный (префикс file:);
- путь относительно classpath JVM (префикс classpath:).

Если у параметров процедуры в самой СУБД есть значения по умолчанию, то эти значения необходимо продублировать в файле значений по умолчанию. В противном случае и для них придется выставлять значения в запросе.

Формат дат (*spring.jackson.date-format*)

Даты должны передаваться в запросе в том формате, в котором их понимают JDBC-драйвер СУБД и сама СУБД. По умолчанию используется формат ISO 8061 (2024-05-29 10:27:17) без timezone. Формат даты описывается в конфигурации сервиса (**spring.jackson.date-format**). При выборе формата дат нужно учитывать, что в некоторых СУБД (например в MS SQL) есть сразу несколько типов данных, отвечающих за хранение дат. Изменив формат дат, необходимо изменить файл значений по умолчанию и все заготовленные запросы в соответствии с новым форматом дат.

Тайм-аут вызова процедуры (*procedure.call-timeout-in-seconds*)

Можно ограничить время вызова процедуры, задав количество секунд в конфигурации (**procedure.call-timeout-in-seconds**). Если тайм-аут сработает, то ответ от сервиса будет код 504.

Выполнение SQL-запросов в СУБД

Поддерживаемые базы данных

Сервис был протестирован с использованием следующих версий баз данных:

- PostgreSQL 14;
- MS SQL 2017;
- MySQL 8;
- SQLite 3.47.1.

Конфигурационный файл

Файл **application.yml**:

```
server:
  port: 8094

sql-query:
  limit: 1000
  timeout: 30
  pool-config:
    max-pool-size: 5
    min-idle: 2
  database-configs:
```

```

- database-name: database-postgresql
  url: jdbc:postgresql://localhost:5432/databasePostgresql
  username: username
  password: password
  transaction-isolation-level: SERIALIZABLE

- database-name: database-mysql
  url: jdbc:postgresql://localhost:3306/databaseMysql
  username: username
  password: password
  transaction-isolation-level: SERIALIZABLE

- database-name: database-mssql
  url: jdbc:postgresql://localhost:1433/databaseMssql
  username: username
  password: password
  transaction-isolation-level: SERIALIZABLE

- databaseName: database-sqlite
  url: jdbc:sqlite:deploy/datasources/init-scripts/sqlite/database-sqlite.db
  username: ""
  password: ""
  transaction-isolation-level: serializable

```

Сервис подключается ко всем базам данных, указанным в конфигурационном файле, во время запуска.

Значения параметров по умолчанию

Сервис поддерживает параметры по умолчанию, которые настраиваются в конфигурационном файле:

- **limit** — ограничение на количество строк, возвращаемых запросом SELECT (по умолчанию 1000);
- **timeout** — максимальное время выполнения SQL-запроса (по умолчанию 30 секунд);
- **pool-config** — раздел настроек пула соединения к каждой базе данных:
 - **max-pool-size** — максимальное количество параллельных соединений с базой данных (по умолчанию 5);
 - **min-idle** — минимальное количество соединений, которые должны оставаться свободными (по умолчанию 2).

4.5.16 Корректировка тайм-аутов

Для правильной работы **AgentNavigator** тайм-ауты сессий на адаптерах должны быть меньше, чем у **BotServer**, то есть 7200 секунд или 2 часа.

Для адаптера **OmnichatAdapter** укажите:

```

omnichatadapter:
  session:
    timeoutSec: 3600

```

Запретите создание сессий с **userid=null**. Действия выполняются в **mongo shell**:

```

use test;
db.runCommand({collMod: 'sessions', validator: {userId: {$ne: null}}});

```

4.5.17 Изменение продолжительности сессии

1. В файле **/etc/speechpro/botserver/application.yml** добавьте конфигурацию:

```

botserver:
  session:

```

```

    timeout-enabled: true
    timeout-sec: <требуемое время тайм-аута в секундах + 5%. например: 6000+0.05*6000 =
7560>
    timeout-check-interval-sec: <0.01 * timeout-sec. Например: 60>
    remove-after-expiry-enabled: true
    remove-after-expiry-sec: <тайм-аут на удаление истекшей сессии = timeout-sec*0.5.
Например, 3000>
    remove-after-expiry-check-interval-sec: <0.01 * remove-after-expiry-sec. Например:
30>

```

2. Перезапустите компонент **BotServer**.

3. Добавьте конфигурацию, изменяющую продолжительность сессии, в конфигурационные файлы всех адаптеров и перезапустите их. Например:

В файле */etc/speechpro/omnichatadapter/application.yml* добавьте конфигурацию:

```

omnichatadapter:
  session:
    timeout-enabled: true
    timeout-sec: <требуемое время тайм-аута в секундах. например: 6000>
    timeout-check-interval-sec: <периодичность проверки тайм-аута в секундах,
определяемая формулой timeout-check-interval-sec=timeout-sec * 0.01. Например: 60>

```

Перезапустите компонент **OmnichatAdapter**.

4.5.18 Общая конфигурация MongoDB



Действия необходимы только в случае установки на несколько компьютеров.

Общая конфигурация **MongoDB** проводится для следующих компонентов платформы **AgentNavigator**: **BotServer**, **HttpAdapter**, **ChatWidget**, **OmnichatAdapter**, **SipAdapter**, **Telegram**, **FileServer**.

```

mongodb:
  name: test #название БД.
  username: #optional
  password: #optional
  hosts: #FT/LB конфигурация.
    #Из документации:
    #Put as many servers as you can in the list and the system will figure out the
rest. This can either be a list of mongodb
    #servers in the same replica set or a list of mongos servers in the same sharded
cluster.
    - localhost
  write-concern: ACKNOWLEDGED #optional. https://www.javadoc.io/doc/org.mongodb/mongo-
java-driver/3.4.3/com/mongodb/WriteConcern.html
  cursor-finalizer-enabled: false #optional. Автоматическая финализация курсора.
  min-connections-per-host: #optional
  max-connections-per-host: #optional
  threads-allowed-to-block-for-connection-multiplier: #optional
#https://stackoverflow.com/questions/6520439/how-to-configure-mongodb-java-driver-
mongooptions-for-production-use
  max-wait-time-ms: #optional
  max-connection-idle-time-ms: #optional
  max-connection-life-time-ms: #optional
  socket-timeout-ms: 10_000 #optional
  heartbeat-socket-timeout-ms: #optional
  add-extra-class-field: false #optional. True: добавлять название класса в виде
отдельного поля в объект, который сохраняется в MongoDB для сериализации/десериализации.
  connect-timeout-ms: #optional

```

```
server-selection-timeout-ms: #optional
```

4.5.19 Конфигурирование LDAP

4.5.19.1 Создание зашифрованного AES-пароля

1. Выполните команду `/opt/speechpro/serviceweb/bin > bash encodePassword.sh passString:`

```
[root@cn-dev11] /opt/speechpro/serviceweb/bin > bash encodePassword.sh test
ERROR StatusLogger No log4j2 configuration file found. Using default configuration:
logging only errors to the console.
Encode password: test
Encoded string: 2Fq0GuWbcDXHiMWURleqdg==
```

2. В файле `etc/speechpro/serviceweb/application.yml` добавьте конфигурацию:

```
ldap:
  enabled: false
  connection:
    url:
    base:
    admin-user-dn:
    admin-password:
    security: startTls
  authentications:
    - dn: ou=Branches
      user-name-attr: sAMAccountName
  mapping:
    user: {shortName: sAMAccountName, fullName: displayName, email: mail, position:
sapTitle, city: 1, enabled: userAccountControl}
    group: {cn: cn, name: name, description: description, enabled: active, members:
member}
  group-queries:
    - base: ou=Branches
      member-field: member
  user-queries:
    - base: ou=Branches
      field-filter: sAMAccountName
  user-blocking-code:
    - 514
    - 66050

ad-groups-roles:
  mapping:
    is-many-to-many-relations: true

synchronize-ad-users:
  # delay: 14400000 = 4hr, 10800000 = 3hr, 7200000 = 2hr, 3600000 = 1hr
  delay: 10800000
  enabled: true
```

Параметры описаны в таблице ниже:

Параметр	Описание
ldap.enabled	Включение/выключение интеграции компонента с LDAP. Значение true/false .
ldap.connection.url	Адрес Active Directory .
ldap.connection.base	Корневой каталог.

Параметр	Описание
ldap.connection.admin-user-dn	Путь к пользователю (администратору).
ldap.connection.admin-password	Пароль.
ldap.connection.security	Типы безопасного соединения. Значение могут быть startTls (tls) или simple (небезопасное).
ldap.authentication.dn	Каталог, в котором находится пользователь.
ldap.authentication.user-name-attr	Поле объекта, по которому происходит авторизация. Возможные значения: uid , cn .
ldap.mapping.user	Определение соответствия полей объекта пользователя из Active Directory и полей объекта компонента ServiceWeb .
ldap.mapping.group	Определение соответствия полей объекта групп из Active Directory и полей объекта компонента ServiceWeb .
ldap.group-query.base	Каталог, в котором лежит список групп.
ldap.group-query.field-filter	Обязательное поле в объекте.
ldap.user-query.base	Каталог со списком пользователей.
ldap.user-query.field-filter	Поле, по которому происходит поиск объекта, скорее всего это уникальное поле со значением login пользователя, например uid , cn .
ldap.user-blocking-code	Список с кодами блокирования учетных записей пользователей.
synchronize-ad-users.delay	Период запуска синхронизации пользователей из Active Directory в компоненте ServiceWeb в миллисекундах.
synchronize-ad-users.enabled	Включение/выключение синхронизации, значение true/false .
ldap.group-query.prefix	Необязательное поле. Используется для выгрузки список групп с определенным префиксом. Например, указан общий каталог, в котором большое количество групп, а нужны только группы с префиксом « Sys CN_content_ ». Параметр будет выглядеть так: prefix: {field: cn, value: Sys CN_content_} field: cn , где cn – это поле объекта Group , value: Sys CN_content_ – это префикс значения.

4.6 Настройка ротации лог-файлов

По умолчанию лог-файлы компонентов архивируются ежедневно или по достижении размера 500 Мбайт. Архивы удаляются через 62 дня, либо по достижении определенного размера. Размер зависит от группы архивов, т. к. каждая имеет разный размер.

Эти параметры настраиваются в конфигурационных файлах `/opt/speechpro/<component>/conf/log4j-prod.xml`:

```
<Properties>
  <Property name="component">serviceweb</Property>
  <Property name="pattern">%d %p [%t] %m %n</Property>
  <Property name="lastModified">62d</Property>
</Properties>
<Appenders>
  <RollingRandomAccessFile
    name="mainSync"
    fileName="${basePath:-/var/log/speechpro}/${component}/${component}.main.log"
    filePattern="${basePath:-/var/log/speechpro}/${component}/${component}.main.%
d.%i.log.gz">
    <PatternLayout>
      <Pattern>${pattern}</Pattern>
    </PatternLayout>
    <Policies>
      <TimeBasedTriggeringPolicy/>
      <SizeBasedTriggeringPolicy size="${mainMaxFileSize:-500 MB}"/>
    </Policies>
    <DefaultRolloverStrategy>
      <Delete basePath="${basePath}/${component}">
        <IfFileName glob="${component}.main.*.log*">
          <IfAny>
            <IfLastModified age="${lastModified}" />
            <IfAccumulatedFileSize exceeds="200MB"/>
          </IfAny>
        </IfFileName>
      </Delete>
    </DefaultRolloverStrategy>
  </RollingRandomAccessFile>
  <Async name="main">
    <AppenderRef ref="mainSync"/>
  </Async>
</Appenders>
```

TimeBasedTriggeringPolicy — параметр отвечает за ежедневную архивацию.

SizeBasedTriggeringPolicy — параметр отвечает за архивацию по достижении определенного размера.

IfFileName — параметры для удаления применяются к файлам/архивам, которые попадают под такое имя.

IfLastModified — параметр отвечает за удаление файлов старше 62 дней.

IfAccumulatedFileSize — параметр отвечает за удаление самого старого архива при достижении размера всех архивов этой группы размера в 200 Мбайт.

4.7 Включение статического баннера

При проведении технических работ можно выводить статический баннер с информацией для пользователя и временным запретом писать в чат.

Подготовка siteconfig

Создайте файл **stub.json** по адресу **/etc/speechpro/chatwidget/parsedsiteconfigs**. Для этого используйте скрипт **siteconfig.sh**:

```
./siteconfig.sh --exportType json_file --
outputFolder /etc/speechpro/chatwidget/parsedsiteconfigs --appKey stub
```

Конфигурация Nginx

Выполните действия:

1. Добавьте в файл **/etc/nginx/conf.d/main.conf** следующую секцию:

```
server {
    listen      9999;
    server_name $hostname;

    # Remove trail slash because it can produce 500 error when accessing to files
    rewrite ^/(.*)/$ /$1 permanent;

    set $SERVICE_HOME /opt/speechpro/chatwidget;
    set $SERVICE_OVERRIDING_DIR /etc/speechpro/chatwidget;

    root $SERVICE_HOME/webapp;

    include /etc/nginx/stub.d/chatwidget.conf;
}
```

2. Создайте файл **/etc/nginx/stub.d/chatwidget.conf** со следующим содержимым:

```
location ~* /web-chat/(.*)/test {
    alias $SERVICE_HOME/webapp/test.html;
    default_type text/html;
}

# First capture group is channel name. Non-greedy because of ignoring channel name and
# getting full path to webapp file
# Second capture group is path to webapp file.
# For example: /web-chat/notExistChannel/css/wa-min.css
# $1 - notExistChannel
# $2 - css/wa-min.css
location ~* /web-chat/(.*?)/(.*) {
    alias $SERVICE_HOME/webapp/$2;
}

location ~* /restapi/siteconfigs/(.*) {
    alias $SERVICE_OVERRIDING_DIR/parsedsiteconfigs/stub.json;
}
```

3. В случае необходимости отображения статического баннера включите (уберите знаки комментария) конфигурацию статического баннера, описанную выше, в **/etc/nginx/conf.d/main.conf** и примените изменение конфигурации **Nginx** с помощью одной из следующих команд:

```
nginx -s reload
```

Или

```
systemctl restart nginx
```

5 ЗАПУСК СИСТЕМЫ

5.1 Авторизация

Откройте один из рекомендованных браузеров. В адресной строке укажите IP-адрес сервера и порт **8086** в формате *https://IP-адрес сервера:8086/*.

Авторизуйтесь в интерфейсе платформы, заполнив поля **Логин** и **Пароль**. Чтобы посмотреть введенный пароль нажмите , чтобы скрыть нажмите . Нажмите **Войти**.

Смените язык интерфейса с помощью значка  в правом верхнем углу страницы. Выберите язык из выпадающего списка. Доступны три языка — русский, английский и испанский.

По умолчанию зарегистрирован пользователь с ролью **SuperAdmin** (для входа укажите **admin** в поле **Логин** и **admin123** в поле **Пароль**).

Если настроена интеграция с **Active Directory**, пользователь может авторизоваться с помощью логина и пароля от своей доменной учетной записи Windows. Интеграция с **Active Directory** подразумевает выполнение определения соответствия (маппинг) между группами **Active Directory** и ролями, созданными в **AgentNavigator**.

Если указанный пользователь отсутствует в системе или произошла ошибка при вводе данных, в форме авторизации появится сообщение.

5.2 Интерфейс

Если авторизация прошла успешно, в окне браузера откроется графический интерфейс **AgentNavigator** со следующими элементами:

- 1: главное меню;
- 2: выход из учетной записи;
- 3: рабочая вкладка.

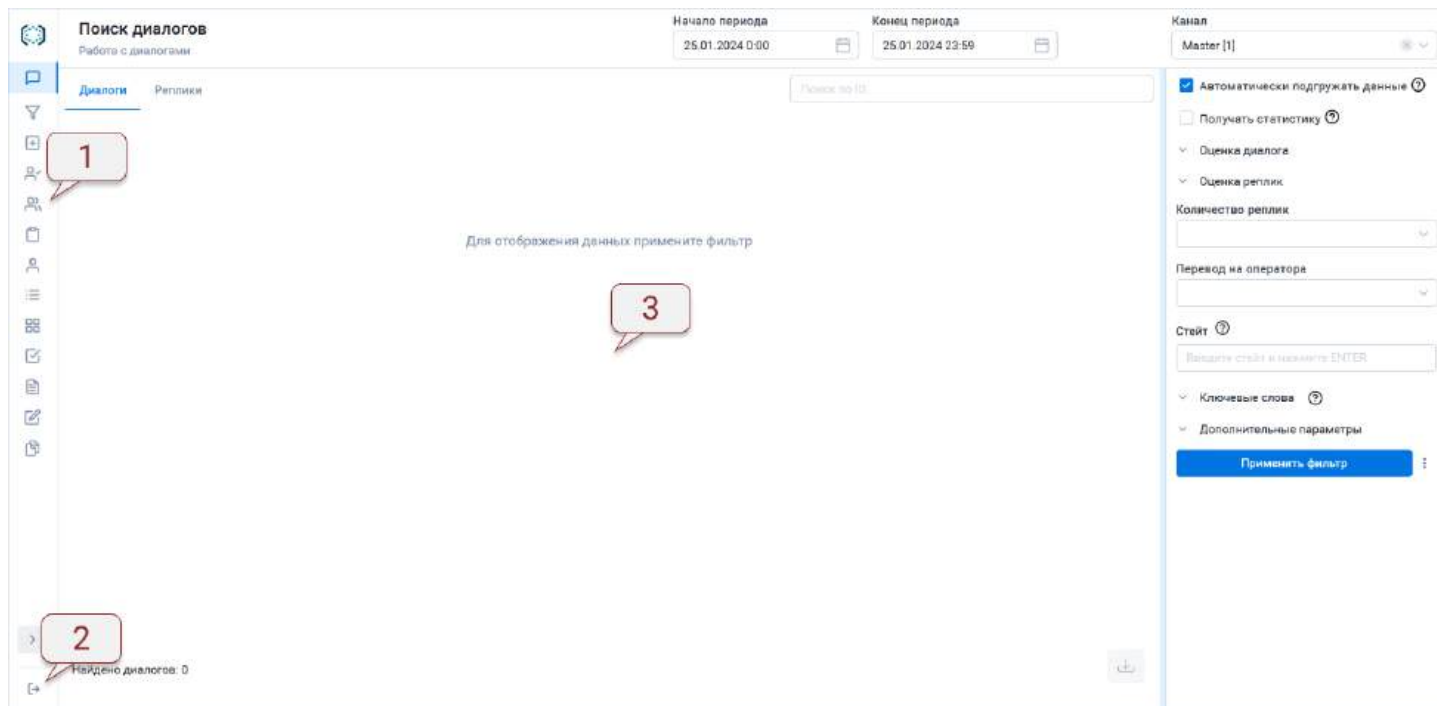


Рисунок 3 — Графический интерфейс AgentNavigator

Главное меню по умолчанию свернуто, чтобы его развернуть, нажмите .

Для завершения работы нажмите .

Для перехода к рабочей вкладке нажмите соответствующую кнопку в главном меню (рис. 4).

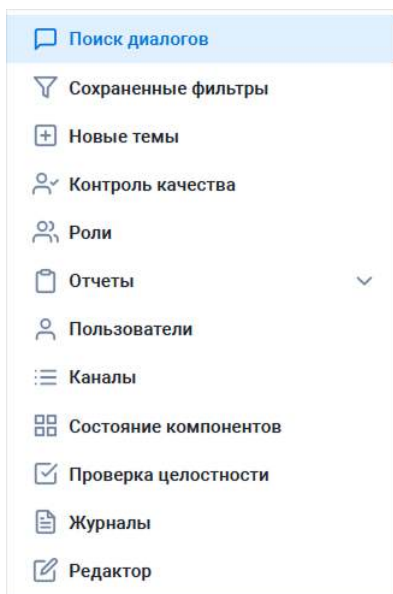


Рисунок 4 — Главное меню

Главное меню для конкретного пользователя зависит от назначенных ему привилегий.

Поиск диалогов — переход к сортировке шагов сессий и работы с ними.

Сохраненные фильтры — переход к сохраненным фильтрам диалогов.

Новые темы — переход к работе с разделами и темами, которые необходимо добавить в сценарий.

Контроль качества — переход к определению качества ответов виртуального консультанта.

Роли — переход к формированию ролей, настройке для каждой роли прав и уровня доступа.

Отчеты — дополнительное меню с возможными для построения отчетами.

Пользователи — переход к управлению пользователями.

Каналы — переход к:

- созданию, просмотру, переименованию и удалению каналов;
- получению ссылки для отправки запроса в канал;
- загрузке сценария в канал;
- просмотру информации о загрузке;
- выгрузке сценария из канала.

Состояние компонентов — переход к просмотру статусов компонентов платформы.

Проверка целостности — переход к просмотру последнего измененного состояния платформы и целостности ее компонентов и модулей.

Журналы — переход к просмотру и анализу событий и действий пользователей.

Редактор — переход к редактору сценария диалога.

Коллекции документов — переход к управлению документам для семантического поиска.



Подробно о вкладках **Поиск диалогов**, **Новые темы**, **Контроль качества**, **Отчеты**, **Редактор**, **Коллекции документов** описано в документе «AgentNavigator Платформа для работы с ИИ-агентами STC-S9013. Руководство пользователя НЦДА.01201-01 92 01».



Подробно о вкладках **Проверка целостности**, **Журналы** описано в документе «AgentNavigator Платформа для работы с ИИ-агентами STC-S9013. Руководство администратора безопасности НЦДА.01201-01 95 01».

6 ФОРМИРОВАНИЕ РОЛЕЙ

Для создания, изменения и удаления ролей воспользуйтесь вкладкой **Роли**.

Вкладка **Роли** содержит следующие элементы (рис. 5):

- 1: перечень созданных ролей;
- 2: поле поиска привилегий.
- 3: перечень назначенных привилегий для выбранной роли;
- 4: перечень всех системных привилегий.

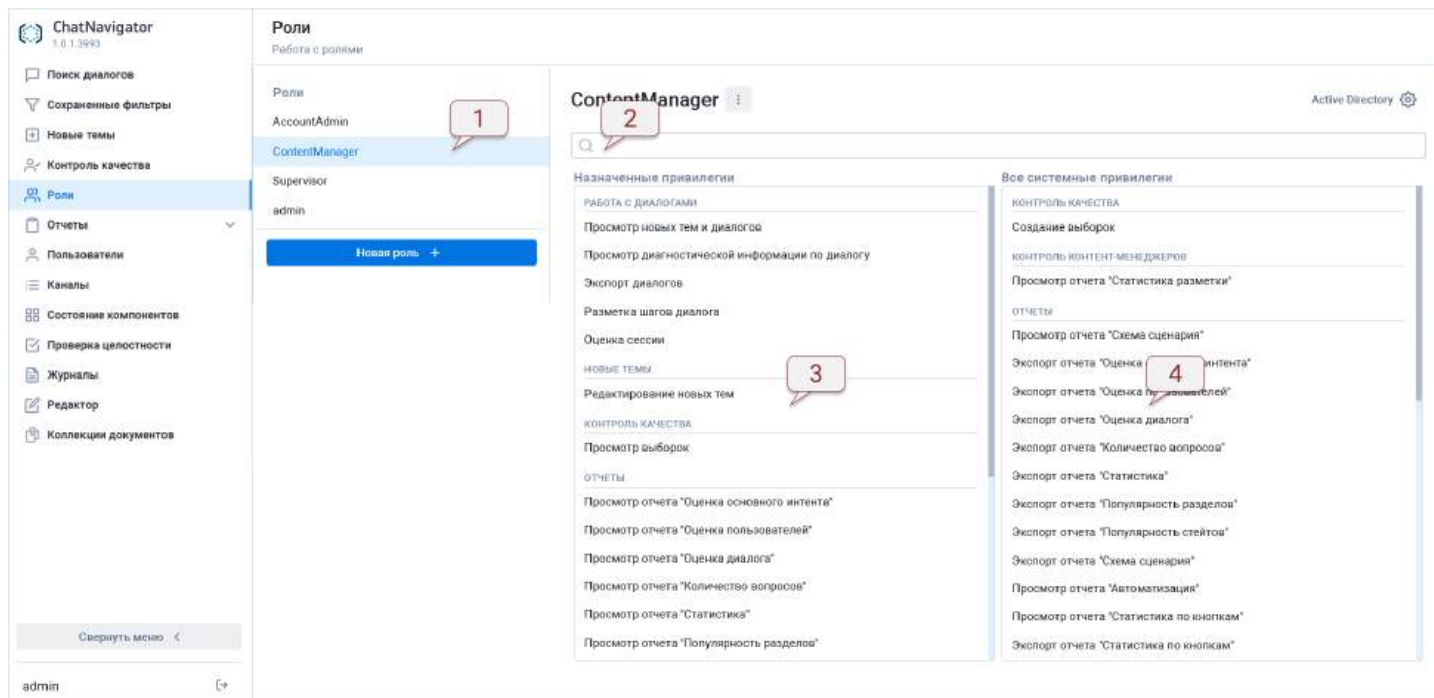


Рисунок 5 — Вкладка Роли

6.1 Привилегии

В отношении пользователей предусмотрены следующие основные права:

1. Право на управление ролями: создание, изменение (переименование роли и изменение состава привилегий) и удаление ролей.
2. Право на управление пользователями: создание, изменение данных пользователя, блокировка и удаление пользователей.
3. Право на управление ролями пользователя: назначение или снятие роли для конкретного пользователя.
4. Право на настройку интеграции с **Active Directory (AD)**: определение соответствия (маппинг) между группами **AD** и ролями, созданными в **AgentNavigator**.

Права созданной роли определяются назначенными для нее привилегиями.

Привилегии для роли назначаются из перечня системных привилегий, которые приведены ниже в таблице.

Группа	Привилегия	Комментарий
Работа с диалогами	Просмотр новых тем и диалогов	Позволяет просматривать диалоги/реплики (как общий список, так и отдельные диалоги) по доступным данному пользователю каналам, пользоваться всеми фильтрами. Пользователь увидит сам диалог, его шаги, результаты оценки сессии и разметки шагов. Если при поиске по ID диалога диалог находится в канале, недоступном пользователю, то такой диалог не отображается.
	Просмотр диагностической информации по диалогу	При просмотре диалога позволяет переключиться в режим отображения отладочной информации по диалогу: <code>sessionData</code> , <code>responseData</code> и т. п. Если пользователю предоставляется эта привилегия, то «Просмотр новых тем и диалогов», а также «Разметка шагов диалога» активируются автоматически.
	Экспорт диалогов	Привилегия на экспорт отфильтрованных диалогов. Если пользователю предоставляется эта привилегия, то «Просмотр новых тем и диалогов» активируется автоматически.
	Экспорт статистики диалогов	Привилегия на экспорт статистики длительности диалогов. Если пользователю предоставляется эта привилегия, то «Просмотр новых тем и диалогов» активируется автоматически.
	Разметка шагов диалога	Привилегия на разметку шагов диалога. Если пользователю предоставляется эта привилегия, то «Просмотр новых тем и диалогов» активируется автоматически.
	Оценка сессии	Привилегия на оценку сессии. Если пользователю предоставляется эта привилегия, то «Просмотр новых тем и диалогов» активируется автоматически.
	Сохранение фильтров	Привилегия позволяет сохранять фильтры, просматривать собственные сохраненные фильтры, изменять и удалять собственные сохраненные фильтры.
	Просмотр всех сохраненных фильтров	Привилегия позволяет сохранять фильтры, просматривать все (свои и чужие) фильтры, изменять и удалять собственные сохраненные фильтры. Если пользователю предоставляется эта привилегия, то «Сохранение фильтров» активируется автоматически.

Группа	Привилегия	Комментарий
	Управление всеми сохраненными фильтрами	Привилегия позволяет сохранять фильтры, просматривать все (свои и чужие) фильтры, изменять и удалять все (свои и чужие) сохраненные фильтры. Если пользователю предоставляется эта привилегия, то «Просмотр всех сохраненных фильтров» активируется автоматически.
Новые темы	Редактирование новых тем	Привилегия на добавление, изменение и удаление новых тем.
Контроль качества	Просмотр выборок	Позволяет просматривать календарь, статистическую информацию, искать и просматривать выборки.
	Создание выборок	Предоставляет право на создание выборок.
Контроль контент-менеджеров	Просмотр отчета	Можно просматривать статистику по «своим» контент-менеджерам.
Отчеты	Просмотр отчета X	Сколько отчетов, столько и привилегий.
	Экспорт отчетов X	Предоставляет право на экспорт соответствующего отчета.
Управление ролями и пользователями	Создание/редактирование пользователей	Предоставляет право на создание пользователя, изменение данных пользователя, привязку учетной записи Mercurial (если была дана привилегия на просмотр сценария диалога). Не предоставляет право назначения пользователю ролей.
	Удаление/блокировка пользователей	Предоставляет право на удаление/блокировку пользователей. Если пользователю предоставляется эта привилегия, то «Создание/редактирование пользователей» активируется автоматически.
	Создание/редактирование ролей	Предоставляет право на создание/редактирование ролей.
	Удаление ролей	Предоставляет право на удаление ролей. Если пользователю предоставляется эта привилегия, то «Создание/редактирование ролей» активируется автоматически.
	Управление ролями пользователя	Предоставляет право на управление ролями пользователя. Если пользователю предоставляется эта привилегия, то «Создание/редактирование ролей» активируется автоматически.

Группа	Привилегия	Комментарий
	Маппинг ролей с ролями AD	Предоставляет право на маппинг ролей с ролями AD . Если пользователю предоставляется эта привилегия, то «Создание/редактирование ролей» активируется автоматически.
Управление каналами	Просмотр каналов	Предоставляет право на просмотр каналов (чтение).
	Управление каналами	Предоставляет право на управление каналами: создание и удаление. Если пользователю предоставляется эта привилегия, то «Просмотр каналов» активируется автоматически.
	Просмотр всех каналов с возможностью загрузки	Просмотр каналов адаптеров и виджета с возможностью загружать сценарии в каналы и управление всеми каналами (чтение и запись). Родительская привилегия — root.
	Настройка приоритета каналов	Предоставляет возможность устанавливать приоритет канала. Если задать высокий приоритет каналу, тогда сценарии из этого канала будут загружаться в кэш BotServer в первую очередь при обращении пользователя. Если пользователю предоставляется эта привилегия, то «Просмотр каналов», «Управление каналами» и «Просмотр всех каналов» активируется автоматически.
	Назначение пользователей на каналы	Предоставляет возможность назначить пользователей на канал. Если пользователю предоставляется эта привилегия, то «Просмотр каналов», «Управление каналами» и «Просмотр всех каналов» активируется автоматически.
Состояние компонентов	Просмотр «Состояние компонентов»	Предоставляет право на просмотр «Состояние компонентов».
Проверка целостности	Просмотр «Проверка целостности»	Предоставляет право на просмотр «Проверка целостности».
Журналы	Просмотр журнала событий	Предоставляет право на просмотр журнала событий.
	Экспорт журнала событий	Предоставляет право на экспорт отфильтрованного журнала событий в файл заданного формата: .pdf , .xlsx (.xls), .docx (.doc), .csv .

Группа	Привилегия	Комментарий
Работа со сценарием	Просмотр сценария диалога	Активируется автоматически, если к учетной записи пользователя привязана учетная запись Mercurial . Позволяет открывать для чтения файлы в Текстовом редакторе и просматривать сценарий без его редактирования в Визуальном редакторе . Дальнейшее управление правами по работе со сценарием находится на стороне Mercurial . Права на чтение или на чтение и запись назначаются при создании канала (подробнее см. раздел Добавление канала документа «AgentNavigator Платформа для работы с ИИ-агентами STC-S9013. Руководство администратора НЦДА.01201-01 91 01»).
	Текстовый редактор	Предоставляет право на работу в Текстовом редакторе .
	Визуальный редактор	Предоставляет право на работу в Визуальном редакторе .
	Редактирование сценария диалога	Предоставляет право на редактирование сценария диалога.
	Просмотр документов Базы знаний	Предоставляет право на просмотр документов Базы знаний .
	Редактирование документов Базы знаний	Предоставляет право на редактирование документов Базы знаний . Если пользователю предоставляется эта привилегия, то «Просмотр документов Базы знаний».
Co-pilot	Доступ к интерфейсу Co-pilot	Предоставляет право на работу в Co-pilot .
Работа с коллекциями документов	Просмотр коллекций и документов	Предоставляет право просматривать собственные коллекции и документы в них, просматривать чужие коллекции и документы, к которым пользователю дан доступ на чтение.

Группа	Привилегия	Комментарий
	Управление коллекциями и документами	Предоставляет право: – создавать новые коллекции; – загружать документы в собственные коллекции (в новые и существующие); – удалять собственные коллекции; – удалять документы в собственных коллекциях; – загружать документы в чужие коллекции, к которым пользователю дан доступ на запись; – удалять чужие коллекции, к которым пользователю дан доступ на запись; – удалять документы в чужих коллекциях, к которым пользователю дан доступ на запись. Если пользователю предоставляется эта привилегия, то «Просмотр коллекций и документов» активируется автоматически.
	Просмотр всех коллекций и документов	Предоставляет право просматривать все коллекции и документы в них. Если пользователю предоставляется эта привилегия, то «Просмотр коллекций и документов» активируется автоматически.
	Управление всеми коллекциями и документами	Предоставляет право загружать документы во все коллекции, удалять документы во всех коллекциях, удалять все коллекции. Если пользователю предоставляется эта привилегия, то «Управление коллекциями и документами» и «Просмотр всех коллекций и документов» активируется автоматически.
	Назначение пользователей на коллекцию	Позволяет предоставлять доступ к коллекции на чтение/запись для других пользователей. Если пользователю предоставляется эта привилегия, то «Просмотр коллекций и документов» активируется автоматически.

Работа с назначением и снятием привилегий идет согласно отношениям parent-child. Если выдается привилегия, у которой есть родительские привилегии, то родительские выдаются вплоть до корневого уровня (на первом уровне иерархии в общем случае больше, чем одна привилегия), а дочерние привилегии не трогаются. Если снимается привилегия, у которой есть дочерние, то дочерние снимаются вплоть до последнего уровня иерархии, а родительские привилегии не трогаются.

При наведении курсора на привилегию в перечне **Все системные привилегии** или **Назначенные привилегии**, цветом выделяются связанные с ней привилегии.

После развертывания платформы в ней по умолчанию присутствует один пользователь со всеми правами и ролью **SuperAdmin**, а также еще три роли: **AccountAdmin**, **Supervisor** и **ContentManager**.

Все базовые роли доступны для редактирования.

Базовые пользователи и роли представлены ниже в таблице.

Привилегия	SuperAdmin	AccountAdmin	Supervisor	ContentManager
Просмотр новых тем и диалогов	+	+	+	+
Просмотр диагностической информации по диалогу	+	+	+	+
Экспорт диалогов	+	+	+	+
Экспорт статистики диалогов	+	+	+	+
Разметка шагов диалога	+	+	+	+
Оценка сессии	+	+	+	+
Редактирование новых тем	+	-	+	+
Просмотр выборок	+	-	+	+
Создание выборок	+	-	+	-
Просмотр отчета (по контент-менеджерам)	+	-	+	-
Просмотр отчета X	+	+	+	+
Экспорт отчетов X	+	+	+	-
Создание/редактирование пользователей	+	-	-	-
Удаление/блокировка пользователей	+	-	-	-
Создание/редактирование ролей	+	-	-	-
Удаление ролей	+	-	-	-
Управление ролями пользователя	+	-	-	-
Маппинг ролей с ролями AD	+	-	-	-
Просмотр каналов	+	-	-	+
Управление каналами	+	-	-	+
Просмотр всех каналов	+	-	-	-
Настройка приоритета каналов	+	-	-	-
Назначение пользователей на каналы	+	-	-	-

Привилегия	SuperAdmin	AccountAdmin	Supervisor	ContentManager
Просмотр «Состояние компонентов»	+	-	-	-
Просмотр «Проверка целостности»	+	-	-	-
Просмотр журнала событий	+	-	-	-
Просмотр сценария диалога	+	-	-	+
Текстовый редактор	+	-	-	+
Визуальный редактор	+	-	-	+
Редактирование сценария диалога	+	-	-	+
Просмотр документов Базы знаний	+	-	-	+
Редактирование документов Базы знаний	+	-	-	+
Доступ к интерфейсу Co-pilot	+	-	-	-
Сохранение фильтров	+	-	+	+
Просмотр всех сохраненных фильтров	+	-	+	+
Управление всеми сохраненными фильтрами	+	-	+	-
Просмотр коллекций и документов	+	+	+	+
Управление коллекциями и документами	+	-	-	+
Просмотр всех коллекций и документов	+	+	+	+
Управление всеми коллекциями и документами	+	-	-	-
Назначение пользователей на коллекцию	+	-	-	+

6.2 Создание новой роли

Чтобы добавить новую роль, нажмите **Новая роль +**, укажите название роли и нажмите **Создать**.

Чтобы назначить привилегии для новой роли в перечне **Все системные привилегии**, выберите нужную привилегию или группу привилегий. При наведении курсора на привилегию в перечне **Все системные привилегии** цветом также выделяются связанные с ней привилегии.

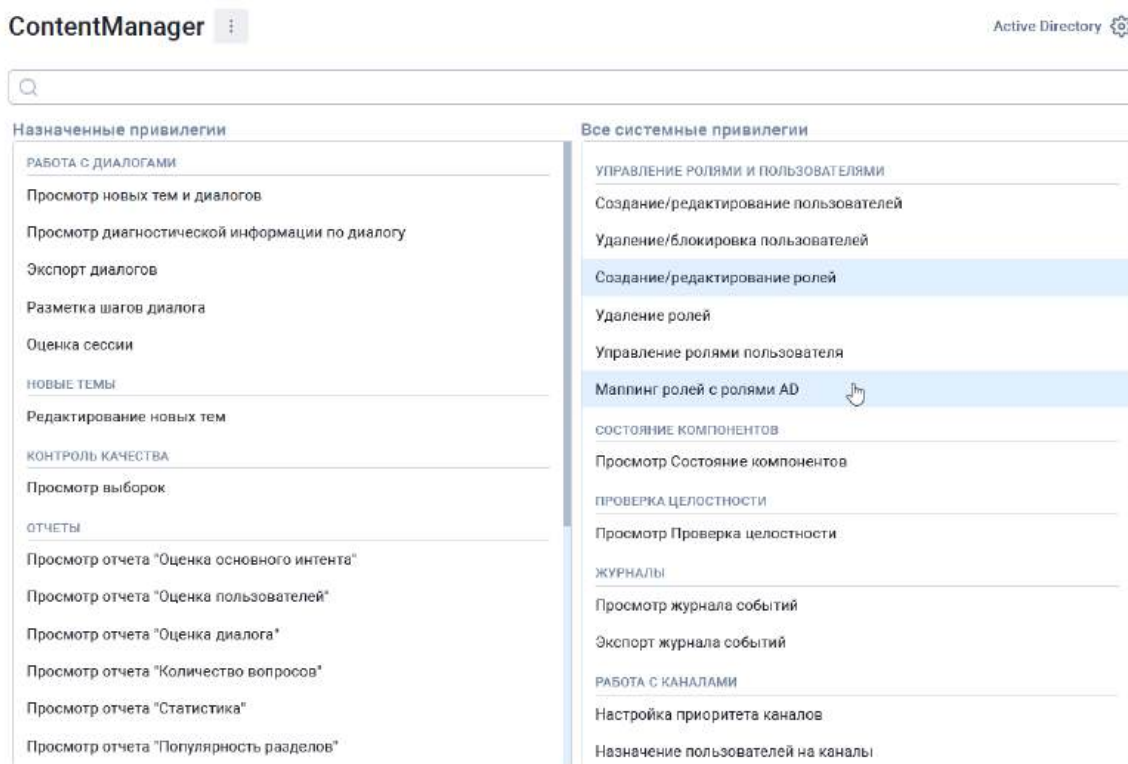


Рисунок 6 — Назначение привилегий

Выбранные привилегии в перечне **Все системные привилегии** автоматически переносятся в перечень **Назначенные привилегии**. Ошибочно назначенную привилегию можно вернуть обратно в перечень **Все системные привилегии**, если выбрать ее в перечне **Назначенные привилегии**. При наведении курсора на привилегию в перечне **Назначенные привилегии** цветом также выделяются связанные с ней привилегии.

После того, как перечень **Назначенные привилегии** будет сформирован, нажмите **Сохранить**. Сохранение привилегий подтверждается сообщением **Привилегии сохранены**.

6.3 Изменение роли

Чтобы переименовать роль, удалить или копировать, выберите ее в перечне ролей и нажмите **⋮** (рис. 7).

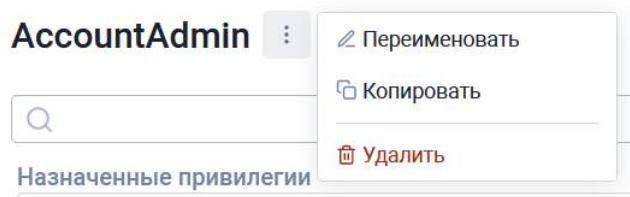


Рисунок 7 — Меню для роли

Чтобы изменить состав привилегий роли, нажмите на привилегию в перечне **Назначенные привилегии** привилегия удалится из списка назначенных и вернется в список всех системных привилегий.

Нажмите на привилегию в перечне **Все системные привилегии**, чтобы добавить ее к роли.

Воспользуйтесь строкой поиска для поиска по тексту в названии привилегий. Начните набирать название привилегии, в перечнях выше отобразятся привилегии, в названиях которых присутствует набранный текст (рис. 8). Поиск по привилегиям регистрозависимый.

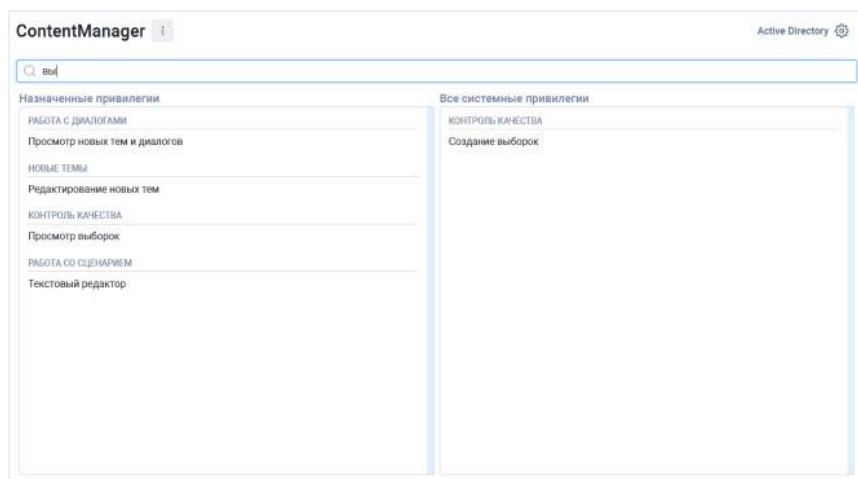


Рисунок 8 — Поиск привилегий

6.4 Копирование роли

Чтобы скопировать роль, нажмите  и выберите **Копировать**.

При создании дубликата сбрасываются все привилегии продублированной роли. Измените наименование и состав привилегий, выполнив действия из подраздела [Изменение роли](#) документа «AgentNavigator Платформа для работы с ИИ-агентами STC-S9013. Руководство администратора НЦДА.01201-01 01».

6.5 Удаление роли

Чтобы удалить роль, выберите ее в перечне ролей, нажмите  и выберите **Удалить**.

В диалоговом окне подтвердите удаление роли, нажав кнопку **Удалить**, или отмените удаление (рис. 9).

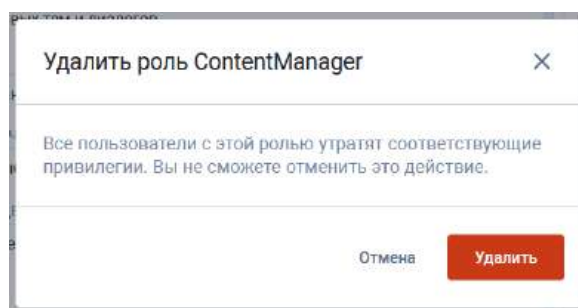


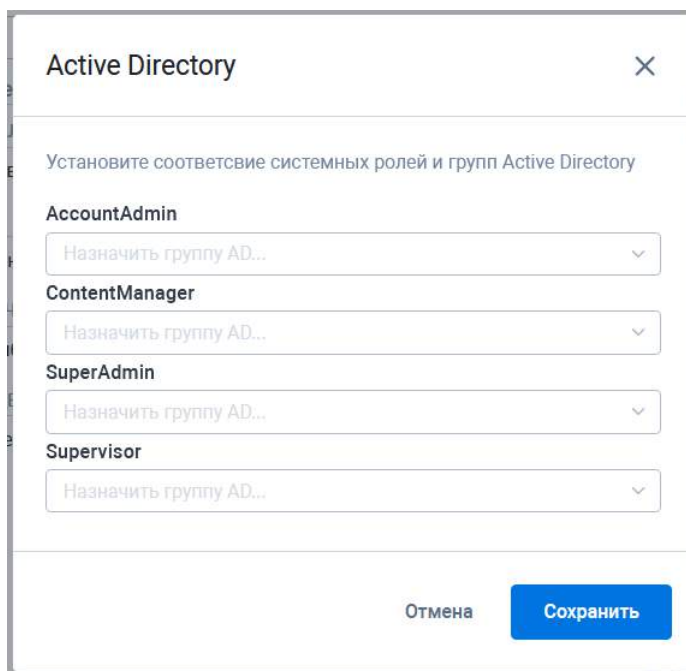
Рисунок 9 — Подтверждение удаления роли

Удаление роли подтвердится сообщением.

6.6 Настройки Active Directory

Чтобы выполнить маппинг системных ролей и групп AD, нажмите .

В открывшемся окне (рис. 10) для каждой роли из выпадающего списка выберите соответствующую группу **AD** и нажмите **ОК**.



Active Directory

Установите соответствие системных ролей и групп Active Directory

AccountAdmin
Назначить группу AD...

ContentManager
Назначить группу AD...

SuperAdmin
Назначить группу AD...

Supervisor
Назначить группу AD...

Отмена Сохранить

Рисунок 10 – Настройки **Active Directory**

7 УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ

Для добавления, редактирования и удаления учетных записей пользователей перейдите во вкладку **Пользователи**.

Вкладка **Пользователи** содержит (рис. 11):

- 1: кнопку добавления пользователя;
- 2: элементы поиска и сортировки пользователей;
- 3: список пользователей;
- 4: общие сведения о пользователях.



Рисунок 11 — Вкладка Пользователи

7.1 Список пользователей

Список зарегистрированных пользователей отображается в виде таблицы со столбцами:

- **Логин** — логин пользователя для входа в интерфейс платформы;
- **Роли** — роли пользователя;
- **Email** — электронная почта пользователя;
- **ФИО** — фамилия, имя, отчество пользователя.

Чтобы найти в списке пользователя, начните вводить **Логин** или **ФИО** в поле поиска.

Установите флажок **Только Active Directory**, чтобы отобразились только пользователи, которые для входа в интерфейс платформы используют логин и пароль своей доменной учетной записи Windows.

7.2 Создание пользователя

Для создания нового пользователя нажмите **Новый пользователь** в правом верхнем углу (рис. 10, 1). В открывшемся окне заполните обязательные поля **Логин**, **Пароль**, **Повторите пароль**, **ФИО**, **Электронная почта** (рис. 12). При необходимости укажите подразделение и должность.

Рисунок 12 — Добавление учетной записи пользователя



Длина пароля должна быть от 8 до 20 символов.

Пароль должен содержать не менее трех видов символов из следующих подмножеств: A-Z, a-z, 0-9, ~+\$&#@% и другие спецсимволы.

Выберите роли (рис. 13). Можно выбрать несколько ролей.

Рисунок 13 — Выбор ролей

Выберите канал для работы с **Co-pilot** (рис.14). Можно выбрать только один канал типа **OMNICHATADAPTER**.

Рисунок 14 — Выбор канала для работы с Co-pilot

В поле **Примечание** можно указать дополнительную информацию о пользователе.

Укажите логин и пароль учетной записи **Mercurial** и нажмите **Создать** (рис. 15).

Рисунок 15 — Учетная запись Mercurial

Если в форме ввести некорректные данные и/или не заполнить обязательные поля, то после нажатия кнопки **Создать** поля с некорректными значениями будут подсвечены (рис. 16).



Рисунок 16 — Подсветка полей с некоректными данными учетной записи пользователя

7.3 Редактирование пользователя

i Для редактирования доступны только учетные записи **Mercurial**. Редактирование учетных записей **Active Directory** с помощью интерфейса администратора невозможно. Данные учетных записей **Active Directory** синхронизируются автоматически.

Для редактирования учетной записи пользователя нажмите  в соответствующей строке списка (рис. 17).

Логин	Роли	Email	ФИО	
admin	admin	admin@servername.com	Бобров Григорий Романович	
content	ContentManager	content@servername.com	Волкова Дарья Михайловна	
supervisor	Supervisor	supervisor@servername.com	Аксёнов Вадим Лукьянович	
Всего пользователей: 3 Из них заблокировано: 0				Показывать на странице 25 1-1 < >

Рисунок 17 — Выбор пользователя для редактирования

Откроется окно, аналогичное окну для создания пользователя. Укажите новые пароли, отредактируйте другие параметры учетной записи. После внесения изменений нажмите **Сохранить** или отмените результаты редактирования кнопкой **Отмена**.

7.4 Блокировка пользователя

i Блокировать учетные записи **Active Directory** невозможно. Данные учетных записей **Active Directory** синхронизируются автоматически.

Для блокировки учетной записи пользователя нажмите  в соответствующей строке списка, нажмите **Заблокировать** и затем **Сохранить**.

По умолчанию учетные записи заблокированных пользователей в списке не отображаются. Информация о них присутствует в общих сведениях о пользователях. Установите флажок **Показывать заблокированных** (рис. 18) — учетные записи заблокированных пользователей в списке отобразятся светло-серым цветом.

Логин или ФИО	☒ Показывать заблокированных	☐ Только Active Directory	Новый пользователь +
Логин	Роли	Email	ФИО
admin	admin	admin@servername.com	Бобров Григорий Романович
content	ContentManager	content@servername.com	Волкова Дарья Михайловна
supervisor	Supervisor	supervisor@servername.com	Аксёнов Вадим Лукьянович
Всего пользователей: 3		Показывать на странице: 25	
Из них заблокировано: 1		1-1	

Рисунок 18 — Отображение заблокированных пользователей в списке

Чтобы разблокировать пользователя, нажмите  в соответствующей строке. В открывшемся окне нажмите **Разблокировать** (рис. 19) и нажмите **Сохранить**.

Редактировать пользователя

Пользователь заблокирован

Разблокировать

Логин

irene22

Латинские буквы и цифры. От 3 до 30 символов

Новый пароль

Введите пароль

Длина пароля от 8 до 20 символов. Пароль должен содержать символы не менее 3-х видов из следующих подмножеств: A-Z, a-z, 0-9, +!\$%&@% и другие спецсимволы

Повторите пароль

Введите пароль повторно

ФИО

Введите полное имя

Электронная почта

zemtsova@speechpro.com

Удалить

Разблокировать

Отмена

Сохранить

Рисунок 19 — Разблокирование пользователя

7.5 Удаление пользователя

 Удаление учетных записей **Active Directory** невозможно. Данные учетных записей **Active Directory** синхронизируются автоматически.

Для удаления учетной записи пользователя нажмите  в соответствующей строке списка или на кнопку **Удалить** в окне **Редактировать пользователя**.

В открывшемся диалоговом окне подтвердите удаление (рис. 20) кнопкой **Удалить** или отмените удаление кнопкой **Отмена**.

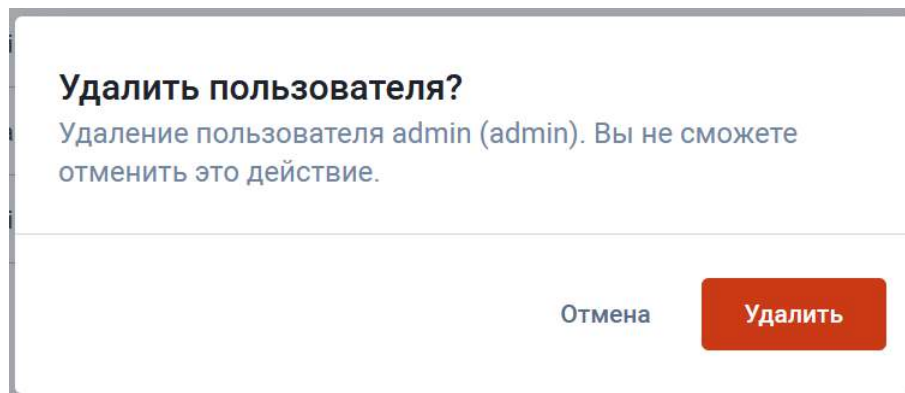


Рисунок 20 — Окно подтверждения удаления пользователя

8 РАБОТА С КАНАЛАМИ

Для добавления, редактирования, а также просмотра списка имеющихся каналов виджета воспользуйтесь вкладкой **Каналы**.

Вкладка **Каналы** содержит следующие элементы (рис. 21):

- 1 — поиск канала;
- 2 — кнопка создания канала;
- 3 — список каналов.

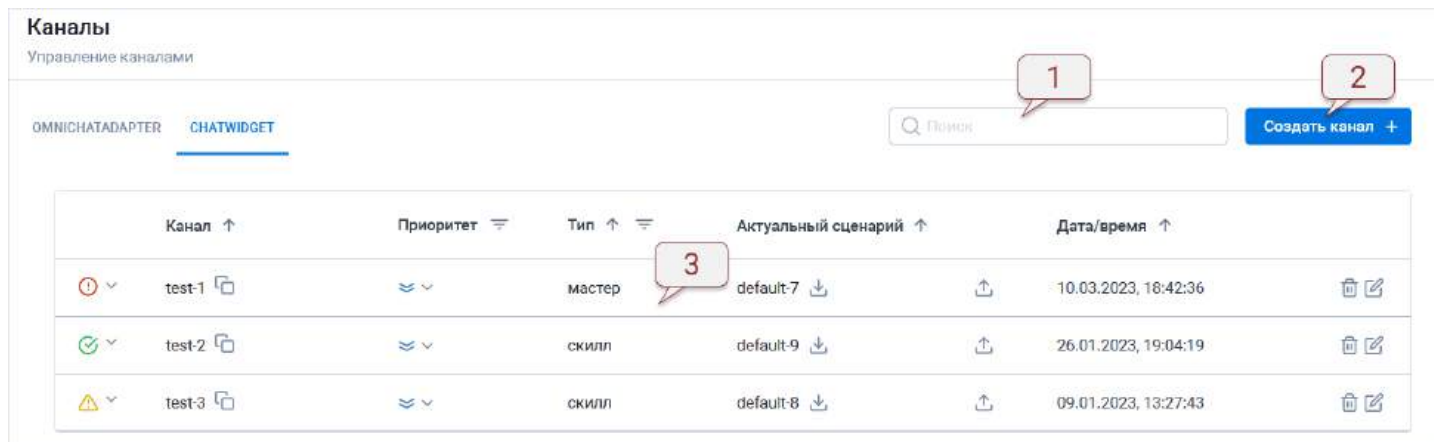


Рисунок 21 — Вкладка Каналы

Каналы разных типов отображаются на разных вкладках, например, **CHATWIDGET**, **OMNICHATADAPTER**, **SIPADAPTER**.

Для фильтрации каналов по какому-то признаку нажмите  в шапке таблицы и выберите из выпадающего списка значение. Кнопка станет активной — . Возможна одновременная фильтрация по приоритету и по типу канала.

Для сортировки каналов по какому-либо полю по возрастанию нажмите  в шапке таблицы. Кнопка станет активной — . Для сортировки по убыванию нажмите на кнопку еще раз, кнопка примет вид . Для сброса сортировки нажмите на кнопку еще раз, чтобы она приняла вид . Сортировка возможна только по одному полю.

8.1 Создание канала

Для добавления канала нажмите **Создать канал** в верхней части области управления каналами. С помощью интерфейса **AgentNavigator** можно создать только каналы типа **CHATWIDGET**.

В открывшемся окне введите имя канала, назначьте на канал пользователей с правом на запись и с правом на чтение сценария и нажмите **Создать** (рис. 22).

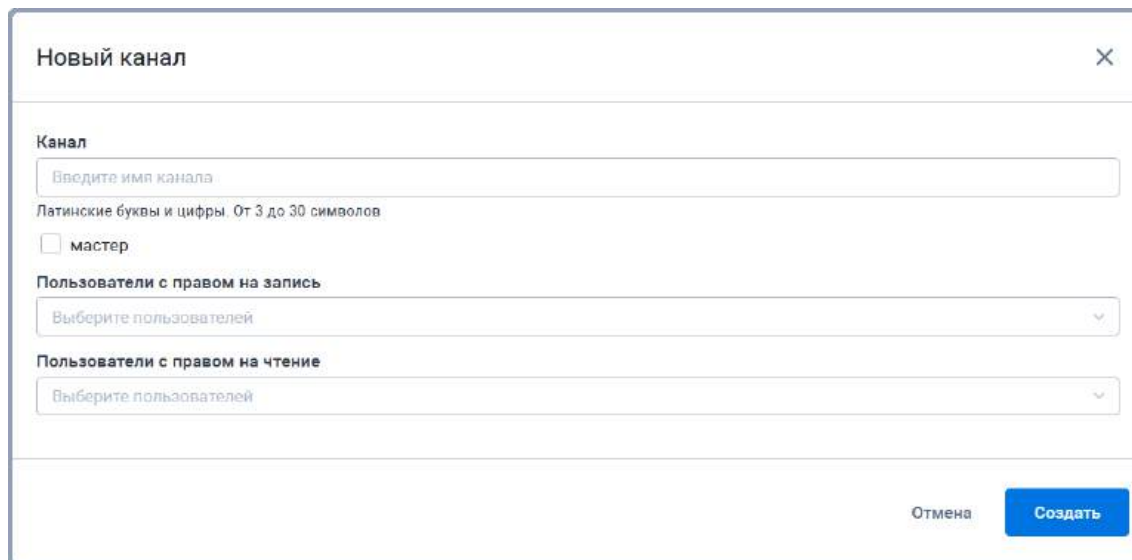


Рисунок 22 — Добавление канала

Имя канала должно быть от 3 до 30 символов и состоять из символов латинского алфавита и цифр.

8.2 Редактирование канала

 Для редактирования канала пользователю должна быть назначена роль, у которой есть соответствующая [привилегия на управление каналами](#).

Для изменения списка пользователей, обладающих правом работать со сценарием, нажмите  в строке канала. Внесите изменения и нажмите **Сохранить** (рис. 23).

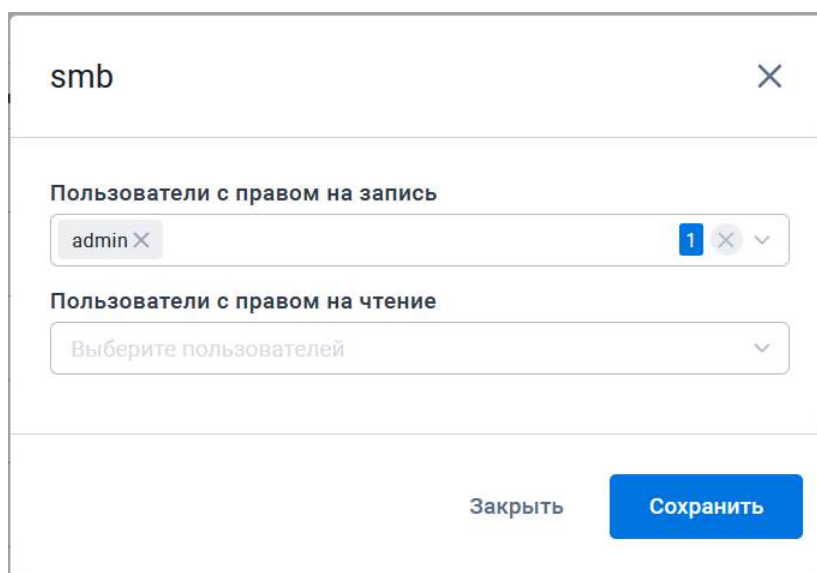


Рисунок 23 — Редактирование списка пользователей

8.3 Загрузка сценария

Нажмите  рядом с названием нужного канала в списке (рис. 24, 1).

Каналы
Управление каналами

OMNICHATADAPTER CHATWIDGET

Поиск

Создать канал +

Канал ↑	Приоритет	Тип ↑	Актуальный сценарий ↑	1	Дата/время ↑	
test-1		мастер	default-7		10.03.2023, 18:42:36	
test-2		скилл	default-9	2	26.01.2023, 19:04:19	
test-3		скилл	default-8		09.01.2023, 13:27:43	

Рисунок 24 — Выбор сценария

С помощью окна выбора файла найдите и откройте необходимый архив со сценарием. Во время загрузки сценария отобразится информационное сообщение. После завершения загрузки рядом с названием канала появятся название загруженного сценария, время загрузки и статус загрузки. Возможные статусы:

- ✓ — сценарий загружен успешно;
- ⚠ — сценарий загружен успешно, но есть предупреждения;
- ❗ — при загрузке сценария произошла ошибка, сценарий не был загружен.

При нажатии на ✓ (рис. 24, 2) отобразится подробная информация о загрузке:

test-3	скилл	default-3	05.05.2022, 16:51:35
Последняя попытка: default-3 Сценарий загружен успешно с предупреждениями! Длительность: 00:00:00.197 patterns.sc [0.0] Нет [o] в сценарии scenario.sc [1.0] У стейта должно быть название. Стейт без названия: / sendFile.sc [16.12] Не удалось выполнить скрипт с пустым контекстом. Ошибка компиляции скрипта, выдается исключение: TypeError: Cannot get property "file" of null [17.12] Не удалось выполнить скрипт с пустым контекстом. Ошибка компиляции скрипта, выдается исключение: TypeError: Cannot get property "file" of null			

Рисунок 25 — Пример подробной информации о загрузке

8.4 Проверка работы виджета

Откройте виджет канала, на который загружается сценарий. Для этого на вкладке **Каналы** нажмите на значок  рядом с названием канала. Перейдите по скопированной ссылке.

В окне виджета отображается бот с загруженным на канал сценарием (рис. 26), при помощи которого можно проверить, как работают добавленные шаблоны.



Рисунок 26 — Окно виджета

8.5 Приоритизация каналов

Если задать высокий приоритет каналу, тогда сценарии из этого канала будут загружаться в кэш **BotServer** в первую очередь при обращении пользователя. Также можно задать очередность выгрузки сценариев из кэша **BotServer**.

Приоритет канала задается на вкладке **Каналы** и может быть в последствии изменен. Чтобы задать приоритет нажмите  и выберите высокий, средний или низкий приоритет для канала.



Управлять приоритетами каналов, загружаемых в кэш **BotServer**, может только пользователь, которому назначена привилегия **Настройка приоритета каналам**.

Очередность загрузки сценариев в кэш

После загрузки/перезагрузки **BotServer** в кэш автоматически загружаются сценарии в следующей последовательности:

- 1) сценарии каналов, для которых установлен **Высокий** приоритет;
- 2) сценарии каналов, для которых установлен **Средний** приоритет;
- 3) сценарии каналов, для которых установлен **Низкий** приоритет.

Сценарии из каналов с одинаковым приоритетом загружаются в кэш в зависимости от последней даты обращения к ним: чем позже сценарий был использован в последний раз (были обращения к нему), тем раньше он будет загружен в кэш.



Устаревшие и неактивные сценарии в кэш **BotServer** не загружаются. Каналы **Postal** всегда имеют **Низкий** приоритет.

Бот отвечает на обращение пользователя только после полной загрузки сценариев в кэш.

Для активации автоматической загрузки сценариев в файле **application.yaml** установите значение **true** для параметра **botserver.scenarioLoadAtStartupEnabled**.

Очередность выгрузки сценариев из кэша

Для загрузки новых сценариев в заполненный кэш **BotServer** предварительно выгрузите неиспользуемые сценарии.

При необходимости загрузки нового сценария из кэша **BotServer** автоматически выгружаются устаревшие сценарии, без информирования пользователя о выгрузке.

Если в кэше нет устаревших сценариев, то из кэша автоматически выгружаются сценарии каналов с наименьшим приоритетом в следующей последовательности:

- 1) сценарии из каналов **Postal**;
- 2) сценарии из каналов **Widget**;
- 3) сценарии из текстовых каналов **Omnichat** и адаптеров внешнего трафика (**VK, Telegram** и т. п.);
- 4) сценарии из голосового канала **SIP**.

Если среди каналов одного типа существует несколько каналов с одинаковым приоритетом, то из кэша первым выгружается тот сценарий, дата последнего обращения к которому наиболее удаленная.

После загрузки сценария в кэш отобразится информация о загрузке нового сценария и информация о выгруженном сценарии.

8.6 Удаление канала

Для удаления канала нажмите  в строке канала.

9 СОСТОЯНИЕ КОМПОНЕНТОВ

На вкладке **Состояние компонентов** отображается информация об активности компонентов, времени проверки и версии компонентов.

Время проверки: 12.05.2022, 10:48

✓	botserver @127.0.0.1 Версия: 1.0.1.3281
✓	IntegrityVerifier @localhost Версия: 1.0.1.3281
✓	httpadapter @127.0.0.1 Версия: 1.0.1.3281
✓	chatwidget @127.0.0.1 Версия: 1.0.1.3281

Рисунок 27 — Вкладка **Состояние компонентов**

9.1 Список компонентов

На странице состояния компонентов показан список компонентов, которые были установлены в рамках данной конфигурации:

- **ChatWidget** — подсистема взаимодействия с клиентом;
- **BotServer** — подсистема обработки запросов клиентов;
- **HttpAdapter** — подсистема интеграции;
- **IntegrityVerifier** — подсистема проверки целостности.

Описание работы компонентов см. в разделе [СОСТАВ ПЛАТФОРМЫ](#).

Время проверки — общий элемент для все компонентов, который отображает дату и время, в которые системой в последний раз было зафиксировано активное состояние компонентов.

Для каждого компонента будут отображены следующие элементы:

- **Имя компонента**;
- **IP-адрес** компьютера, на котором установлен компонент;
- **Имя хоста** — имя компьютера, на котором установлен компонент.
- **Версия** — версия компонента.

9.2 Состояние компонентов

При недоступности компонентов **AgentNavigator** происходит переключение на резервный контур.

В конфигурационных файлах адаптеров можно настроить тайм-ауты по ответу **healthcheck** для данного компонента и для зависимых компонентов. А также настроить, нужно ли опрашивать зависимые компоненты при проверке жизнеспособности адаптера.

9.3 Список проверок компонентов

Список проверок приведен в таблице ниже:

Проверка	Команда	Результат
Проверка установленных компонентов AgentNavigator	<pre> yum list installed e -i 'botserver chatwidget classifier classifiertrainer fileserver httpadapter integrityverifier omnichatadapter serviceweb sipadapter' rpm -q --qf '%{NAME}\n' botserver chatwidget classifier classifiertrainer fileserver httpadapter integrityverifier omnichatadapter serviceweb sipadapter 2>/dev/null </pre>	Отображает установленные пакеты
Проверка статуса служб компонентов	<pre> systemctl status botserver chatwidget classifier classifiertrainer fileserver httpadapter integrityverifier omnichatadapter serviceweb sipadapter </pre>	Отображает текущий статус служб компонентов AgentNavigator
Проверка статуса автозагрузки компонентов при старте сервера	<pre> systemctl list-unit-files egrep -i 'botserver chatwidget classifier classifiertrainer fileserver httpadapter integrityverifier omnichatadapter serviceweb sipadapter' </pre>	Отображает статус автоматической загрузки: enabled / disabled
Проверка работоспособности установленных компонентов AgentNavigator	<pre> systemctl list-units egrep -i 'botserver chatwidget classifier classifiertrainer fileserver httpadapter integrityverifier omnichatadapter serviceweb sipadapter' </pre>	Вывод статуса службы из лог-файла

10 ОПИСАНИЕ ЛОГ-ФАЙЛОВ

10.1 Расположение и содержимое лог-файлов

Расположение и содержимое лог-файлов компонентов представлено в таблице ниже:

Компонент	Расположение	Лог-файлы
BotServer	<i>/var/log/speechpro/botserver/</i>	<i>audit.log botserver.access.log boserver.main.log botserver.stat.json.log</i>
ChatWidget	<i>/var/log/speechpro/chatwidget/</i>	<i>audit.log chatwidget.access.log chatwidget.main.log</i>
HttpAdapter	<i>/var/log/speechpro/httpadapter/</i>	<i>audit.log httpadapter.access.log httpadapter.main.log</i>
OmnichatAdapter	<i>/var/log/speechpro/omnichatadapter/</i>	<i>audit.log omnichatadapter.access.log omnichatadapter.main.log</i>
SipAdapter	<i>/var/log/speechpro/sipadapter/</i>	<i>audit.log sipadapter.access.log sipadapter.main.log</i>
IntegrityVerifier	<i>/var/log/speechpro/integrityverifier/</i>	<i>audit.log integrityverifier.main.log</i>
ServiceWeb	<i>/var/log/speechpro/serviceweb/</i>	<i>audit.log serviceweb.access.log serviceweb.main.log serviceweb.pump.log</i>
FileServer	<i>/var/log/speechpro/fileserver/</i>	<i>audit.log fileservеr.access.log fileservеr.main.log</i>
NER	<i>/var/log/speechpro/ner/</i>	<i>ner.access.log ner.main.log</i>
NER Trainer	<i>/var/log/speechpro/ner_trainer/</i>	<i>ner_trainer.access.log ner_trainer.main.log</i>
Classifier	<i>/var/log/speechpro/classifier/</i>	<i>classifier.access.log classifier.main.log</i>
Classifier Trainer	<i>/var/log/speechpro/classifier_trainer/</i>	<i>classifier_trainer.access.log classifier_trainer.main.log</i>

Компонент	Расположение	Лог-файлы
Co-pilot	<i>/var/log/speechpro/co-pilot/</i>	<i>co-pilot-{YYYY}-{MM}-{DD}.log</i>
ConfigService	<i>/var/log/speechpro/config-service/</i>	<i>audit.log</i> <i>config-service.access.log</i> <i>config-service.main.log</i>
database-http-adapter	<i>/var/log/speechpro/database-http-adapter/</i>	<i>audit.log</i> <i>database-http-adapter.access.log</i> <i>database-http-adapter.main.log</i>
python-http-adapter	<i>/var/log/speechpro/python-http-adapter/</i>	<i>audit.log</i> <i>python-http-adapter.access.log</i> <i>python-http-adapter.main.log</i>

10.2 Описание лог-файлов

В файл *access.log* сохраняются все внешние коммуникации с указанием времени события.

Файл *audit.log* для адаптеров и компонента **BotServer** сохраняет следующие события: включение, выключение, перезапуск компонента.

Файл *main.log* сохраняет все события компонента.

Файл *stat.log* для компонента **BotServer** сохраняет входящие вопросы пользователя (из адаптеров) и ответы бота (исходящие ответы в адаптеры).

Файл *pump.log* сохраняет данные о процессе пампинга *stat.log* в **ServiceWeb**.

Каждый компонент ведет журнал *audit.log* в стандарте **UTF-8**.

В этот журнал попадают события 3-х типов:

- системные (работоспособность сервиса): запуск/остановка сервиса; серьезные ошибки, требующие поддержки; потери и восстановление связи между компонентами и смежными системами;
- пользовательские (действия пользователя с платформой): изменение конфигурации, другие действия над платформой (разметка диалога и т. д.);
- безопасности (события в подсистеме АА): вход/выход из учетной записи, изменение пользователя (смена пароля, других атрибутов), попытки авторизоваться в интерфейсе.

События журналирования записываются через стандартную систему логирования.

Пример файла **.audit.log*:

```
{
  "timestamp": "2007-12-03T10:15:30+01:00",
  "level": "INFO",
  "type": "system",
  "component": "botserver",
```

```

"code": "STOP",
"message": "BotServer stopped"
}
{
  "timestamp": "2007-12-03T10:15:30+01:00",
  "level": "INFO",
  "type": "user",
  "component": "serviceweb",
  "code": "STEP_COMMENTED",
  "message": "User commented step <step_number> in <session_id> as <comment>"
  "username": "guest@servername.com",
  "role": "guest",
  "data": {"session_id": "<someSessionId>"}
}
{
  "timestamp": "2007-12-03T10:15:30+01:00",
  "level": "ERROR",
  "type": "system",
  "component": "httpadapter",
  "code": "SYS-00001",
  "message": "Service <someService> does not respond"
}

```

Расшифровка событий в лог-файлах:

Код	Описание
login	Пользователь {userName} вошел в интерфейс платформы
login_error	Неудачная попытка входа пользователя {userName} с устройства {ip}
logout	Пользователь {userName} вышел из системы с устройства {ip}
upload_file	Пользователь {userName} загрузил файл {fileName} на канал {channelName}
create_section	Пользователь {userName} создал раздел {sectionName}
remove_section	Пользователь {userName} удалил раздел {sectionName}
build_report	Пользователь {userName} построил отчет {reportName} с {dateFrom} по {dateTo}
download_report	Пользователь {userName} выгрузил отчет {reportName} с {dateFrom} по {dateTo}
add_user	Пользователь {userName} создал пользователя {itemUserName} с ролью {itemRole}
change_user_profile	Пользователь {userName} изменил у пользователя {itemUserName} {userAttributes}
deactivate_user	Пользователь {userName} заблокировал пользователя {itemUserName}
activate_user	Пользователь {userName} разблокировал пользователя {itemUserName}
remove_user	Пользователь {userName} удалил пользователя {itemUserName}
add_channel_to_user	Пользователь {userName} добавил пользователю {itemUserName} канал {channelName}

Код	Описание
remove_channel_from_user	Пользователь {userName} удалил у пользователя {itemUserName} канал {channelName}
change_channel_to_user	Пользователь {userName} изменил у пользователя {itemUserName} каналы: {channelName}
create_channel	Пользователь {userName} создал канал {channelName}
remove_channel	Пользователь {userName} удалил канал {channelName}
change_channel	Пользователь {userName} изменил у канала {channelName} Атрибуты: {channelAttributes}
service_start	{service} был запущен
service_stop	{service} был выключен
find_user	Пользователь {userName} получил данные о пользователе {userName}
find_users	Пользователь {userName} получил данные о всех пользователях системы
update_mercurial_credentials	Пользователь {userName} изменил у пользователя {userName} данные учетной записи Mercurial {HgLoginPassword}
delete_mercurial_credentials	Пользователь {userName} удалил у пользователя {userName} данные учетной записи Mercurial
set_mercurial_access	Пользователь {userName} установил у пользователя {userName} флаг hgReadOnly {hgReadOnly}
check_hg_credentials	Пользователь {userName} получил у пользователя {service} флаг hgReadOnly
find_user_roles	Пользователь {userName} получил список ролей пользователя {userName}
assign_role_to_user	Пользователь {userName} добавил пользователю {userName} роль {roleName}
unassign_role_from_user	Пользователь {userName} удалил у пользователя {userName} роль {roleName}
get_roles	Пользователь {userName} получил данные о роли {roleName}
get_role	Пользователь {userName} получил данные всех ролях системы
create_role	Пользователь {userName} создал роль {roleName}
update_role	Пользователь {userName} изменил у роли {roleName} {roleDescription}
delete_role	Пользователь {userName} удалил роль {roleName}
get_role_permissions	Пользователь {userName} получил у роли {roleName} список привилегий
assign_permission	Пользователь {userName} добавил роли {roleName} привилегию {permissionName}

Код	Описание
unassign_permissions	Пользователь {userName} удалил у роли {roleName} привилегию {permissionName}
replace_permissions	Пользователь {userName} установил роли {roleName} привилегии {permissionsName}
get_pages	Пользователь {userName} получил данные о страницах системы
get_page_actions	Пользователь {userName} получил у страницы {pageName} перечень доступных действий {actions}
get_permission	Пользователь {userName} получил данные о привилегии {permissionName}
get_permissions	Пользователь {userName} получил данные о всех привилегиях системы

11 ПРОВЕРКА РАБОТОСПОСОБНОСТИ

11.1 Проверка загрузки сценария

Для проверки выполните следующие действия:

1. Авторизуйтесь в интерфейсе администратора **ServiceWeb**.
2. Перейдите во вкладку **Каналы** и создайте канал, нажав на соответствующую кнопку и заполнив требуемые поля.
3. Выберите файл сценария с помощью кнопки **Выбрать сценарий** рядом с названием канала.
4. Убедитесь, что рядом с названием канала появилось название сценария, время загрузки и значок **Завершено**.

11.2 Проверка работы ChatWidget

Для проверки выполните следующие действия:

1. Перейдите во вкладку **Каналы**, нажмите на иконку  напротив созданного канала.
2. Откройте новую страницу браузера, вставьте скопированную ссылку вида ***http://<hostname>/web-chat/<ключ интеграции>/test*** и перейдите по ней.
3. В открывшемся окне задайте вопрос «...».
4. Убедитесь, что в ответ пришла фраза «...».
5. Зайдите в интерфейс **ServiceWeb**, во вкладке **Поиск диалогов** выберите созданный канал в выпадающем списке **Канал**.
6. Выставьте временной диапазон в фильтрах справа сверху, который включает время создания данного диалога.
7. Нажмите **Применить фильтр**.
8. Убедитесь, что данный диалог отобразился в рабочей области вкладки **Поиск диалогов**.

11.3 Проверка работы OmnichatAdapter

Для проверки выполните следующие действия:

1. Авторизуйтесь в интерфейсе администратора **ServiceWeb**.
2. Перейдите во вкладку **Каналы** и загрузите сценарий в канал **omnichat-adapter**.
3. Чтобы задать вопрос в **OmnichatAdapter**, отправьте **POST**-запрос:

Endpoint	<code>http://<hostname>:9066/v2/restapi/channels/omnichat-adapter/test</code>
Заголовки запроса	Content-Type: application/json Charset: UTF-8
Тело запроса	<pre>{ "request_id": "\${__UUID}", "appeal_id": "\${__UUID}", "request_time": "2018-09-27T18:18:53.076+0300", "message_number": "1", "sender_id": "256",</pre>

	<pre> "sender_type": "client", "meta_data": { "region": "MSK", "phone": "89214789012", "page": "http://vtb.ru" }, </pre>
	<pre> "cross_links": [{ "system_code": "crm", "id": "98754" }, { "system_code": "billing", "id": "ivanovii" }], "full_name": "Иванов Иван Иванович", "channel_id": "10", "channel_name": "Мобильноеприложение", "agent": {}, "skill_id": "49", "skill_name": "МБ_Массовыйсегмент ", "data": [{ "uid": "", "type": "text", "content": "Текст вопроса" }] } </pre>

- Во вкладке **Поиск диалогов** интерфейса администратора выберите канал **omnichat-adapter** (справа сверху) и нажмите **Применить фильтр**.
- В рабочей области вкладки **Поиск диалогов** должен отобразиться диалог, созданный через OmnichatAdapter.

11.4 Проверка работы SipAdapter

Для проверки выполните следующие действия:

- Авторизуйтесь в интерфейсе администратора **ServiceWeb**.
- Перейдите во вкладку **Каналы** и загрузите сценарий в канал **sip-adapter-incoming**.
- Чтобы задать вопрос в **SipAdapter**, отправьте **POST**-запрос:

Endpoint	http://<hostname>:9098/restapi/channels/sip-adapter-incoming/test
Заголовки запроса	Content-Type: application/json Charset: UTF-8
Тело запроса	<pre> { "header": { "messageId": "1", "sessionId": "1", "messageNumber": 1, "eventNumber": 1 }, "event": { "_type": "recognition", "mainHypothesis": { "text": "привет", "confidence": 1.0 } } } </pre>

	<pre> }, "messageData": { } } </pre>
	<pre> "cross_links": [{ "system_code": "crm", "id": "98754" }, { "system_code": "billing", "id": "ivanovii" }], "full_name": "Иванов Иван Иванович", "channel_id": "10", "channel_name": "Мобильноеприложение", "agent": {}, "skill_id": "49", "skill_name": "МБ_Массовыйсегмент ", "data": [{ "uid": "", "type": "text", "content": "Текст вопроса" }] } </pre>

4. Во вкладке **Поиск диалогов** интерфейса администратора выберите канал **sip-adapter** (справа сверху) и нажмите **Применить фильтр**.
5. В рабочей области вкладки **Поиск диалогов** отобразится диалог, созданный через **SipAdapter**.

11.5 Проверка работы IntegrityVerifier

Для проверки выполните следующие действия:

1. Зайдите по протоколу **SSH** на сервер с установленными компонентами **AgentNavigator**.
2. Убедитесь, что в интерфейсе администратора во вкладке **Проверка целостности** все компоненты помечены зеленым. Это означает, что целостность компонентов не нарушена с момента последнего снимка состояния.
3. Произвольно измените один из конфигурационных файлов, например **/opt/speechpro/botserver/conf/application.yml**.
4. В интерфейсе администратора во вкладке **Проверка целостности** выполните **Ручное сканирование**.
5. Компонент **BotServer** помечен красным, так как целостность была нарушена.
6. При нажатии на компонент, на форму выводится детализация ошибки:

Путь к оригинальному файлу: `opt/speechpro/botserver/conf/application.yml`
 Путь к измененному файлу: `opt/speechpro/botserver/conf/application.yml`
 Хэш оригинала: `6b2bba5c28957afb0fb171d502501891`
 Текущий хэш: `e4c50f0af85c15457d33d4aa7d2dc64c`
 Размер оригинала в байтах: `806`
 Текущий размер байтах: `808`
 Статус: `WRONG_HASH`

11.6 Проверка работы добавленных каналов адаптера

Проверка актуальна для дополнительных компонентов, для которых предварительно создается канал адаптера с помощью следующих файлов:

OmnichatAdapter: */etc/speechpro/omnichatadapter/application.yml*

```
channels:
  channel-prefix: omnichat-adapter
  default-channels:
    enabled: true
    action: addIfNotExist
    channels:
      - channel-id: omnichat-adapter
        response-endpoint:
          host: localhost
          port: 9077
          path: /
          scheme: http
      - channel-id: omnichat-adapter-2
        response-endpoint:
          host: localhost
          port: 9077
          path: /
          scheme: http
```

SipAdapter: */etc/speechpro/sipadapter/application.yml*

```
channels:
  channel-prefix: sip-adapter
  default-channels:
    enabled: true
    action: addIfNotExist
    channels:
      - channel-id: sip-adapter-incoming
        response-endpoint:
          host: localhost
          port: 9099
          path: /
          scheme: http
      - channel-id: sip-adapter-outgoing
        response-endpoint:
          host: localhost
          port: 9099
          path: /
          scheme: http
```

Подробнее о проверке работы **SipAdapter** см. п. [Проверка работы SipAdapter](#).

ServiceWeb: */etc/speechpro/serviceweb/application-integration.yml*

```
integration:
  vcservices:
    omnichatadapter:
      service-name: OmnichatAdapter
      dedicated-channels:
        - omnichat-adapter
        - omnichat-adapter-2
      nodes:
        - {ip: localhost, port: 9066}
```

Для проверки выполните следующие действия:

1. Авторизуйтесь в интерфейсе администратора **ServiceWeb**.
2. Перейдите во вкладку **Каналы** и загрузите в созданный канал сценарий.
3. Чтобы задать вопрос в созданный канал **OmnichatAdapter**, отправьте **POST**-запрос:

Endpoint	<code>http://<hostname>:9066/v2/restapi/channels/<название созданного канала>/test</code>
Заголовки запроса	<code>Content-Type: application/json Charset: UTF-8</code>
Тело запроса	<pre>{ "request_id": "\${__UUID}", "appeal_id": "\${__UUID}", "request_time": "2018-09-27T18:18:53.076+0300", "message_number": "1", "sender_id": "256", "sender_type": "client", "meta_data": { "region": "MSK", "phone": "89214789012", "page": "http://vtb.ru" }, "cross_links": [{ "system_code": "crm", "id": "98754" }, { "system_code": "billing", "id": "ivanovii" }], "full_name": "Иванов Иван Иванович", "channel_id": "10", "channel_name": "Мобильноеприложение", "agent": {}, "skill_id": "49", "skill_name": "МБ_Массовыйсегмент ", "data": [{ "uid": "", "type": "text", "content": "Текст вопроса" }] }</pre>

11.7 Проверка ротации логов

Для проверки автоматической архивации лог-файлов компонентов при достижении определенного размера (на примере компонента **BotServer**) выполните следующие действия:

1. В файле `/opt/speechpro/botserver/conf/log4j-prod.xml` найдите блок:

```
...
<Appenders>
  <RollingRandomAccessFile
    name="mainSync"
    fileName="${basePath:-/var/log/speechpro}/${component}/${component}.main.
log"
    filePattern="${basePath:-/var/log/speechpro}/${component}/${component}.ma
in.%d.%i.log.gz">
```

```
<PatternLayout>
  <Pattern>%d [${context:-}] %p [%t] %m %n</Pattern>
</PatternLayout>
<Policies>
  <TimeBasedTriggeringPolicy/>
  <SizeBasedTriggeringPolicy size="${mainMaxFileSize:-500 MB}" />
</Policies>
```

...

- Измените значение параметра `mainMaxFileSize` с `500 MB` на `1 KB`.
- Выполните **POST**-запрос на `/restapi/test/nlp/procces`. В заголовке запроса необходимо передать `accountid: default`, в качестве `queryParam` укажите канал (`channelId:номер канала`).
- Дождитесь, пока файл `/var/log/speechpro/botserver/botserver.main.log` достигнет размера `1 Кбайт`.
- Когда файл достигает размера `1 Кбайт`, он должен быть архивирован и иметь название типа `botserver.main.<дата архивации>.1.log.gz`.

11.8 Проверка работы механизма устаревания сессии

Для проверки того, что диалоги устаревают по истечении тайм-аута (на примере компонента **BotServer**) выполните следующие действия:

- В файле `/opt/speechpro/botserver/conf/application.yml` найдите блок:

```
session:
  timeout-enabled: true
  timeout-sec: 7200
  timeout-check-interval-sec: 600
  timeout-thread-count: 1
  timeout-state: /onSessionTimeout
  remove-after-expiry-enabled: true
  remove-after-expiry-sec: 3600
  remove-after-expiry-check-interval-sec: 300
```

- Для параметров `timeout-sec` и `timeout-check-interval-sec` установите значение `1`.
- Перезапустите компонент с помощью команды:

```
systemctl restart botserver
```

- Откройте `/var/log/speechpro/botserver/botserver.main.log` на просмотр лог-файлы компонента **BotServer** (файл `/var/log/speechpro/botserver/botserver.main.log`).
- Создайте диалог любым из способов.
- Убедитесь, что по истечению тайм-аута, установленного в параметре `timeout-sec`, в лог-файлах компонента **BotServer** появились следующие строки:

```
2019-05-31 14:06:48,684 [C:test,S:9241411b-b791-4797-bf18-3281d91ae4d2,CN:,N:2,A:default,U:c5d9d9f2-03b1-f486-fa19-9f3cda64069d,D:vc-tele2,RID:]
INFO [session-expiration-0] 1 of sessions expired after 5.884 ms
2019-05-31 14:06:50,685 [C:,,S:,,CN:,N:,A:,U:,D:,RID:] INFO [session-expiration-0] Start
expiring sessions before 2019-05-31T11:06:40.685Z...
```

12 МОНИТОРИНГ КОМПОНЕНТОВ

Для отслеживания состояния компонентов у компонентов есть [endpoint](#), при обращении к которому возвращается состояние компонента и всех зависящих от него компонентов и ресурсов.

Список компонентов, для которых доступен мониторинг:

- ChatWidget;
- MrsipAdapter;
- Telegram;
- VkontakteAdapter;
- OmnichatAdapter;
- SipAdapter.

Дерево зависимостей компонентов и их ресурсов:

Основной компонент	Зависимый компонент (уровень 1)	Зависимый компонент (уровень 2)	Зависимый компонент (уровень 3)
OmnichatAdapter			
	MongoDB		
	BotServer		
		MongoDB	
		HttpAdapter	MongoDB
		Classifier	MongoDB
	FileServer		
		MongoDB	
	PrompterAdapter		
		MongoDB	
SipAdapter	MongoDB		
	BotServer		
		MongoDB	
		HttpAdapter	MongoDB
		Classifier	MongoDB

Запрос

```
GET / http://<Host>:<port>/restapi/healthcheck
```

Формат запроса

В запросе ничего не передается.

Формат ответа

Параметры ответа

name string	обязательный	название компонента	
isOk boolean	обязательный	статус компонента	
instanceData object	необязательный	адрес компонента, если используется несколько инстансов	см. Параметры объекта instanceData
resourceToStatuses array	необязательный	список статусов зависимых ресурсов	см. Параметры статусов зависимых ресурсов и компонентов
description string	необязательный	описание ошибки	
dependentComponentStatuses array	необязательный	список статусов зависимых компонентов	см. Параметры статусов зависимых ресурсов и компонентов

Параметры объекта instanceData

host string	обязательный	имя хоста
node string	обязательный	имя ноды

Параметры статусов зависимых ресурсов и компонентов

name string	обязательный	название компонента	
isOk boolean	обязательный	статус зависимого компонента или ресурса	
instanceData object	необязательный	адрес компонента, если используется несколько инстансов	см. Параметры объекта instanceData
description string	необязательный	описание ошибки	

Пример запроса

```
POST / http://<Host>:<port>/restapi/healthcheck HTTP/1.1
Content-Type: application/json
Host: servername.com
```

Пример ответа при успешной проверке

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
  "name": "omnichatAdapter",
```

```
"isOk": true,
"instanceData": null,
"description": null,
"resourceStatuses": [
  {
    "name": "mongo",
    "isOk": true,
    "description": null
  }
],
"dependentComponentStatuses": [
  {
    "name": "botServer",
    "isOk": true,
    "instanceData": null,
    "description": null,
    "resourceStatuses": [
      {
        "name": "mongo",
        "isOk": true,
        "description": null
      }
    ],
    "dependentComponentStatuses": [
      {
        "name": "httpAdapter",
        "isOk": true,
        "instanceData": null,
        "description": null,
        "resourceStatuses": [
          {
            "name": "mongo",
            "isOk": true,
            "description": null
          }
        ],
        "dependentComponentStatuses": []
      }
    ],
    "name": "classifier",
    "isOk": true,
    "instanceData": null,
    "description": null,
    "resourceStatuses": [
      {
        "name": "mongo",
        "isOk": true,
        "description": null
      }
    ],
    "dependentComponentStatuses": []
  }
],
},
{
  "name": "classifier",
  "isOk": true,
  "instanceData": null,
  "description": null,
  "resourceStatuses": [
    {
      "name": "mongo",
      "isOk": true,
      "description": null
    }
  ],
  "dependentComponentStatuses": []
}
],
},
{
```

```

    "name": "fileServer",
    "isOk": true,
    "instanceData": null,
    "description": null,
    "resourceStatuses": [
      {
        "name": "mongo",
        "isOk": true,
        "description": null
      }
    ],
    "dependentComponentStatuses": []
  },
  {
    "name": "prompterAdapter",
    "isOk": true,
    "instanceData": null,
    "description": null,
    "resourceStatuses": [
      {
        "name": "mongo",
        "isOk": true,
        "description": null
      }
    ],
    "dependentComponentStatuses": [
      {
        "name": "prompter",
        "isOk": true,
        "instanceData": {
          "port": 9007,
          "host": "localhost"
        },
        "description": null,
        "resourceStatuses": [
          {
            "name": "mongo",
            "isOk": true,
            "description": null
          }
        ]
      }
    ],
    "dependentComponentStatuses": []
  }
]
}

```

Пример ответа при недоступном BotServer

```

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
{
  "name": "omnichatAdapter",
  "isOk": false,
  "instanceData": null,

```

```
"description": "Dependent components status fails",
"resourceStatuses": [
  {
    "name": "mongo",
    "isOk": true,
    "description": null
  }
],
"dependentComponentStatuses": [
  {
    "name": "botServer",
    "isOk": false,
    "instanceData": null,
    "description": "Fail to get response: Fail to get response after 30000 ms",
    "resourceStatuses": null,
    "dependentComponentStatuses": []
  },
  {
    "name": "fileServer",
    "isOk": true,
    "instanceData": null,
    "description": null,
    "resourceStatuses": [
      {
        "name": "mongo",
        "isOk": true,
        "description": null
      }
    ],
    "dependentComponentStatuses": []
  },
  {
    "name": "prompterAdapter",
    "isOk": true,
    "instanceData": null,
    "description": null,
    "resourceStatuses": [
      {
        "name": "mongo",
        "isOk": true,
        "description": null
      }
    ],
    "dependentComponentStatuses": [
      {
        "name": "prompter",
        "isOk": true,
        "instanceData": {
          "port": 9007,
          "host": "localhost"
        },
        "description": null,
        "resourceStatuses": [
          {
            "name": "mongo",
```

```
        "isOk": true,  
        "description": null  
      },  
    ],  
    "dependentComponentStatuses": []  
  }  
]  
}  
]
```

Коды ответов

200	Ok	Успешный запрос
-----	----	-----------------

13 ГЛОССАРИЙ

Термины и определения

Адаптер	Модуль на стороне AgentNavigator, обеспечивающий взаимодействие с чат-платформой или напрямую с внешними каналами коммуникации. Адаптер специфицирует структуру и логику обработки запросов, проходящих между AgentNavigator и внешней системой (чат-платформой или внешним каналом коммуникации). На адаптере создаются каналы, в которые загружаются сценарии обслуживания. Примеры адаптеров: OmnichatAdapter, HttpAdapter и т. д.
Активный канал	Канал, в котором есть активные сессии, а также канал, к которому были обращения последние n минут/часов/дней.
Виджет	Модуль, устанавливаемый на сайт клиента для взаимодействия клиента с оператором или чат-ботом (обычно предоставляется разработчиком онлайн-консультанта).
Внешние каналы коммуникации (биз.) Фронт-часть платформы	Telegram, веб-чат на сайте, SDK (мобильное приложение) и т. п.
Выборка	Заданное количество случайных сессий, равномерно распределенных в течение календарных суток.
Голосовое взаимодействие	Обмен информацией в голосовом канале.
Голосовая платформа	Платформа, которая обеспечивает функциональность распознавания (ASR) и синтеза (TTS) речи, взаимодействие по внешним каналам коммуникации на базе IVR, SIP и т. п.
Голосовой поток	Сказанная пользователем фраза или озвученный системой ответ. Аудиоданные в формате audio/16.
Группа ЦРТ (ЦРТ)	Группа ИТ-компаний, разработчик инновационных систем в сфере мультимодальной биометрии, распознавания и синтеза речи, многоканальной записи, обработки и анализа аудио- и видеоинформации.
Диалог	Любое взаимодействие системы с клиентом: в голосовом канале, в мобильном приложении, чат-боте, SMS и т. д. Применительно к системе AgentNavigator — это процесс обмена сообщениями между клиентом и виртуальным консультантом/оператором. Состоит из отдельных шагов. Диалог происходит в рамках сессии.

Диалоговая платформа	Состоит из комплекса технических решений, позволяющих строить сценарии диалогов по правилам или автоматически обучаться на данных с помощью технологии машинного обучения, выполнять взаимодействие текстом на естественном языке, просматривать и проводить разметку диалогов, строить различного вида аналитические отчеты. При подключении к голосовой платформе (создается омниканальное решение) является сервером приложений, позволяющим автоматизировать обращения в голосовом канале. При подключении к рабочему месту оператора (в чат-канале или после преобразования голоса в текст) обеспечивает классификацию текстовых обращений с возможностью формирования ответов или подсказок.
Запрос	HTTP-запрос между чат-платформой и виртуальным консультантом. Структура запросов описывается в API.
Информационная система	Любая внешняя (с точки зрения чат-платформы) система, содержащая информацию, которая используется при автоматизации обращений.
Инцидент	Любое событие, не являющееся частью стандартных операций по предоставлению услуги, которое привело или может привести к нарушению или снижению качества этой услуги. В контексте данного документа — событие, не описанное в рамках стандартных сценариев поведения системы.
Канал	Сущность, которая объединяет канал AgentNavigator и учетную запись АТС для обеспечения процесса оповещения клиентов.
Кейс обслуживания	Набор возможных правил маршрутизации сообщений клиента, определяющий точки взаимодействия чат-платформы и бота, а также порядок, набор и структуру передаваемых по API запросов.
Контакт-центр (КЦ) или Центр обслуживания (вызовов)	Набор программно-аппаратных средств, предназначенных для обслуживания обращений в различных каналах.
Контент-менеджер	Должностное лицо организации, специалист, ответственный контент для голосовых, чат или омниканальных сервисов.
Клиент	Пользователь услуг Заказчика, в том числе по договору об оказании услуг.
Обращение клиента (биз.) Сессия (тех.)	Ограниченное во времени, нацеленное на выявление и решение причины обращения взаимодействие Клиента по внешним каналам коммуникации. Каждое обращение клиента имеет уникальный номер и обрабатывается в соответствии с потоком обслуживания и конкретным кейсом обслуживания.

Онлайн-консультант	Программное обеспечение, обеспечивающее взаимодействие пользователей или клиентов с операторами или сотрудниками в чат-канале (мессенджер, сайт, мобильное приложение и пр.) посредством текстовой переписки. Онлайн-консультант обычно является частью Контакт-центра и обеспечивает взаимодействие с каналами связи, защиту от атак, маршрутизацию трафика, распределения обращений по скилл-группам.
Оператор	Сотрудник контактного центра Заказчика, общающийся с абонентом и осуществляющий его обслуживание.
Маршрутизация	Процесс классификации вопроса клиента и его переадресация на другие каналы в соответствии с логикой, заложенной в сценарии.
Мастер-бот	Сценарий обслуживания, в котором реализована логика маршрутизации вопросов клиента на скилл-боты. Мастер-бот находится в мастер-канале.
Мастер-канал	Канал, созданный с типом «мастер». Используется для маршрутизации диалога на скилл-каналы.
Паттерн (шаблон)	Строка специального вида, которая ставит себе в соответствие некоторое множество фраз.
Паттерн матчинг	Сопоставление фразы с шаблоном. В случае, если фразе соответствует несколько шаблонов, выбирается наиболее подходящий из них.
Перемаршрутизация	Процесс повторного (второй раз и более за одну сессию) матчинга вопроса в сценарии мастер-бота. Перемаршрутизация вызывается по конструкции goToMaster в сценарии скилл-бота.
Промежуточный сертификат	Одно из звеньев в цепочке доверия, которое позволяет связать серверный сертификат (внешней системы) с корневым центром сертификации.
Пользователь	Сотрудник компании, участвующий в бизнес-процессах обслуживания клиентов.
Поток обслуживания	Определяет, с кем клиент непосредственно общается — с виртуальным консультантом или с оператором. От потока обслуживания зависит, какие сведения должна передавать чат-платформа виртуальному консультанту.
Разметка сессий	Проставление меток, тональности, оценок ошибок и комментариев для каждого шага сессии и определение общей оценки за сессию (диалог) в целом.
Распознавание речи	Преобразование речи в текст.
Реплика	Составная часть диалога. Ответ на слова собеседника.
Семантический поиск	Поиск релевантных текстовых документов на основе смыслового (семантического) значения запрашиваемой фразы документов, а не их словарного состава.
Синтез речи	Озвучивание текста.

Система	Комплекс средств автоматического обслуживания.
Системный администратор	Должностное лицо организации, специалист по обслуживанию программных и технических средств отвечающий за работу системы в штатном режиме.
Скилл (skillName)	Параметр таблицы маршрутизации, который обозначает наименование канала, задаваемое в таблице маршрутизации. Уникален в рамках таблицы маршрутизации. Необходим для создания уникальных в рамках таблицы маршрутизации наименований скилл-ботов, которые могут иметь одинаковый channelID, т. к. находятся на разных стендах.
Скилл-бот	Сценарий обслуживания клиента по ограниченному набору тематик или отдельный навык, решающий определенную задачу клиента. Маршрутизация в скилл-бот выполняется из мастер-бота. Скилл-бот находится в скилл-канале.
Скилл-канал	Канал, созданный с типом «скилл», реализующий один из сценариев маршрутизации в него диалога из мастер-канала и обратно. Может располагаться как в другом канале той же инсталляции, что и мастер-канал, так и в другой инсталляции.
Сообщение клиента	Выраженное в текстовом виде высказывание клиента, которое обрабатывается в соответствии с заданной логикой обслуживания. Каждое сообщение клиента имеет свой уникальный номер.
Составной ответ	Ответ, состоящий из последовательности звуковых файлов и текста для синтеза (фрагментов) в произвольном порядке и количестве.
Стейт (тема)	Основной структурный элемент сценария, который может содержать шаблоны для классификации вопросов и набор реакций — действий робота, включая ответ клиенту.
Сценарий обслуживания Сценарий	Логика, в соответствии с которой виртуальный консультант осуществляет диалог с клиентом, определяет тему сообщения и формирует ответ для клиента. Также архивированный набор файлов, содержащих описание такой логики и загружаемых в канал.
Точка подключения	ВАЖНО! Не путать с «внешним каналом коммуникации». Канал обработки сессий в AgentNavigator . Канал определяет, какой именно сценарий обслуживания будет применен к поступившей в него сессии. В канале может находиться только один сценарий.
Устаревший сценарий	Сценарий, у которого нет активных сессий и есть более новая ревизия.
Фонограмма диалога	Запись разговора клиента с ботом в стереоформате так, как он звучал в телефонном канале.
Фрагмент ответа	Звуковой файл или текст для синтеза.

Центр обслуживания (ЦОВ, КЦ)	Комплекс оборудования информационного обслуживания абонентов (РД 45.191-2001). Является аналогом так называемого «Call Centre» — центра обслуживания телефонных соединений. Подразумевается набор программно-аппаратных средств, предназначенных для обслуживания обращений в различных каналах.
Чат-платформа	Платформа, которая обеспечивает возможность подключения AgentNavigator к внешним каналам коммуникации и АРМ Оператора.
Шаг диалога	Сообщение клиента и полученный на него ответ бота/оператора.
Keystore	java-хранилище, которое содержит клиентские ключи и сертификаты для прохождения аутентификации на внешнем сервере.
BotServer	Компонент, обеспечивающий выполнение диалоговых сценариев и управляющий конфигурацией NLP-модулей.
channelID	Идентификатор канала, прописываемый при создании канала в конфигурации. Уникальный в рамках одного стенда.
ID сессии	Уникальный идентификатор сессии.
Truststore	java-хранилище, которое содержит серверные сертификаты или цепочки сертификатов (приводящих к сертификату, выпущенному корневым центром сертификации) для проверки доверия к серверу, которые хранятся во внешней системе и экспортируются в доверенный сертификат через openssl.

Расшифровка аббревиатур

AES	Сокращение от англ. Advanced Encryption Standard — симметричный алгоритм блочного шифрования.
API	Сокращение от англ. Application Programming Interface — «программный интерфейс приложения». Описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
HASP	Сокращение от англ. Hardware Against Software Piracy — аппаратно-программная система защиты программ и данных от нелегального использования и несанкционированного распространения.
LDAP	Сокращение от англ. Lightweight Directory Access Protocol — «облегченный протокол доступа к каталогам». Протокол, использующий TCP/IP и позволяющий производить операции аутентификации, поиска и сравнения, а также операции добавления, изменения или удаления записей.

NER	Сокращение от англ. Named Entity Recognition — «извлечение именованных сущностей». Извлечение именованных сущностей используется для уточнения тематики, чтобы выдать клиенту наиболее релевантный ответ: ср. «хочу оформить карту», «хочу оформить кредитную карту», «хочу оформить дебетовую карту» – сущность «тип карты» определяет, какой ответ выдаст робот по теме «Оформление карты». Также используется для заполнения слотов. Например в вопросе «хочу записаться к терапевту в 128-ю поликлинику» можно выделить две сущности – специальность врача и название медицинской организации, определяющие, куда робот запишет клиента. В зависимости от того, какие обязательные слоты еще не заполнены, робот задаст уточняющие вопросы.
NLP	Сокращение от англ. Natural Language Processing — «обработка естественного языка». Направление в машинном обучении, посвященное распознаванию, генерации и обработке устной и письменной человеческой речи.
SipAdapter	Адаптер для интеграции с VoiceKey.PLATFORM. Выполняет функции передачи распознанного текста и параметров по звонку от VoiceKey.PLATFORM в AgentNavigator и передачи текста ответа и параметров для синтеза, распознавания и результатов обработки перебиваний.
БД	База данных
СУБД	Система управления базами данных.

14 ИСТОРИЯ ОБНОВЛЕНИЙ

3-160725

Изменения конфигурации

В конфигурации [ServiceWeb](#) добавлены:

- секция web-socket;
- в секцию rag.llm.document-storage:
 - long-poll-read-timeout-seconds, documents-subscription-error-timeout-seconds, documents-subscription-not-found-timeout-seconds, status-update-enabled, import-file-threads-count, collection-documents-page-size.
- в секцию rag.llm.directory-import добавлены настройки delete-files-enabled, delete-file-hashes-enabled и registry-check-enabled.

2-140725

Изменения конфигурации

В конфигурации [HttpAdapter](#) в секцию certificate добавлен protocol.

1-280325

Обновлена компонентная схема.

В системных требованиях обновлена версия Java. Изменена команда установки [JDK](#).

Изменения конфигурации

В конфигурации [BotServer](#) добавлен nlp.keyboard-layout-corrector.

В конфигурации [HttpAdapter](#) и [ServiceWeb](#) в секцию llm.giga-chat добавлено поле auth-required.

В конфигурации [ServiceWeb](#) добавлены:

- spring.flyway.baseline-on-migrate;
- postgres.audit.flyway.baseline-on-migrate;
- prompter.postgres.flyway.baseline-on-migrate;
- prompts.question-generation;
- prompts.question-generation-with-examples;
- spring.profiles.includes.llm-services;
- секция llm;
- секция web-socket;
- в секцию rag.llm.document-storage добавлены:
 - long-poll-read-timeout-seconds, documents-subscription-error-timeout-seconds, documents-subscription-not-found-timeout-seconds, status-update-enabled;
- в секцию spring.cloud.openfeign.client.config добавлены:
 - giga-chat-auth-beta;

- giga-chat-completions-beta;
- giga-chat-auth-prod;
- giga-chat-completions-prod;
- yandex-gpt-completions.

В конфигурации [HttpAdapter](#) добавлены:

- spring.profiles.includes.llm-services;
- в секции llm.giga-chat добавлена секция channel-specific.

В конфигурации [ConfigService](#):

- добавлен файл application-llm-services.yml;
- изменился формат и расширение файла, содержащего список существующих моделей: теперь application-llm-models.yml.

В конфигурации [ChatWidget](#) в секции chatwidget добавлена секция build-url.