ChatNavigator

Омниканальная диалоговая платформа на естественном языке

STC-S9777

Руководство администратора

НЦДА.00767-01 91 01

Версия документа: 3-170726

Версия программного обеспечения: 5.1

ЦРТ инновации

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

Иллюстрации, приведенные в документе, могут 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».

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ 7
1.1 Сведения о продукте и изготовителе 7
1.2 Служба технической поддержки 7
1.3 Назначение решения 7
2. СИСТЕМНЫЕ ТРЕБОВАНИЯ 8
2.1 Список используемых портов 9
3. СОСТАВ ПЛАТФОРМЫ 10
3.1 Подсистема взаимодействия с клиентом ChatWidget 11
3.2 Подсистема обработки запросов клиентов BotServer 12
3.3 Подсистема интеграции HttpAdapter 12
3.4 Подсистема хранения и аналитики ServiceWeb 12
3.5 Подсистема проверки целостности IntegrityVerifier 13
3.6 Резервирование компонентов платформы 13
4. УСТАНОВКА И НАСТРОЙКА 15
4.1 Установка ChatNavigator с помощью Docker 15
4.1.1 Миграция с RPM на Docker 17
4.2 Установка ChatNavigator с помощью Ansible Playbook 18
4.3 Настройка и запуск компонентов 20
4.3.1 BotServer 20
4.3.2 HttpAdapter 29
4.3.3 ServiceWeb 35
4.3.3.1 Подключение ServiceWeb к Security 47
4.3.3.2 Фильтрация по параметрам диалога 48
4.3.3.3 Настройка контекста для веб-компонентов 50
4.3.3.4 Настройка автоматической очистки базы данных 50
4.3.3.5 Маппинг ссылок для автоматического скачивания документов 50
4.3.4 IntegrityVerifier 51
4.3.5 ChatWidget 52
4.3.5.1 Кастомизация ChatWidget 54
4.3.5.2 Интеграция ChatWidget с сайтом 56
4.3.6 OmnichatAdapter 57

4.3.6.1 Настройка форматирования ответов для OmnichatAdapter 61
4.3.7 SipAdapter 63
4.3.8 ConfigService 66
4.3.9 Telegram 70
4.3.10 CrafttalkAdapter 71
4.3.11 Classifier и Classifier Trainer 74
4.3.12 NER и NER Trainer 75
4.3.13 FileServer 75
4.3.14 Kafka-adapter 77
4.3.15 Trigger-service 78
4.3.16 Max-adapter 80
4.3.17 Data-stream-service 82
4.3.18 Корректировка тайм-аутов 82
4.3.19 Изменение продолжительности сессии 83
4.3.20 Общая конфигурация MongoDB 83
4.3.21 Конфигурирование LDAP 84
4.3.21.1 Создание шифрованного AES-пароля 84
4.4 Настройка ротации лог-файлов 86
4.5 Включение статического баннера 87
5. ЗАПУСК СИСТЕМЫ 89
5.1 Авторизация 89
5.2 Интерфейс 89
6. ФОРМИРОВАНИЕ РОЛЕЙ 92
6.1 Реализованные привилегии 92
6.2 Создание новой роли 98
6.3 Изменение роли 99
6.4 Копирование роли 100
6.5 Удаление роли 100
6.6 Настройки Active Directory 100
7. УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ 102
7.1 Список пользователей 102
7.2 Создание пользователя 103

7.3 Редактирование пользователя 104
7.4 Блокировка и разблокировка пользователя 104
7.5 Удаление пользователя 105

8. РАБОТА С КАНАЛАМИ 106
8.1 Создание канала 106
8.2 Редактирование канала 107
8.3 Загрузка сценария 107
8.4 Проверка работы виджета 108
8.5 Приоритизация каналов 109
8.6 Удаление канала 110

9. СОСТОЯНИЕ КОМПОНЕНТОВ 111
9.1 Список компонентов 111
9.2 Состояние компонентов 111
9.3 Список проверок компонентов 112

10. ОПИСАНИЕ ЛОГ-ФАЙЛОВ 113
10.1 Расположение и содержимое лог-файлов 113
10.2 Описание лог-файлов 114

11. ПРОВЕРКА РАБОТОСПОСОБНОСТИ 117
11.1 Проверка загрузки сценария 117
11.2 Проверка работы ChatWidget 117
11.3 Проверка работы OmnichatAdapter 117
11.4 Проверка работы SipAdapter 118
11.5 Проверка работы IntegrityVerifier 119
11.6 Проверка работы добавленных каналов адаптера 120
11.7 Проверка ротации логов 121
11.8 Проверка работы механизма устаревания сессии 122

12. МОНИТОРИНГ КОМПОНЕНТОВ 123

13. ГЛОССАРИЙ 128

14. ИСТОРИЯ ОБНОВЛЕНИЙ 134

ВВЕДЕНИЕ

Общие положения

Настоящее руководство предназначено для персонала, производящего установку и настройку ChatNavigator.

Руководство содержит описание интерфейса интеграции решения с внешней системой.

Сведения, приведенные в руководстве, относятся к версии 5.1 ChatNavigator.

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
Курсив	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.)
Полужирный курсив	Имена файлов и путей доступа к ним
`Моноширинный`	Тексты конфигурационных файлов, исходные текстов программ, параметры API
Подчеркнутый бирюзовый	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.

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Сведения о продукте и изготовителе

Наименование: Омниканальная диалоговая платформа на естественном языке ChatNavigator

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

На базе ChatNavigator разрабатываются виртуальные консультанты и ассистенты, которые ведут с клиентом естественный диалог в чате, мессенджере или по телефону. Диалоговые ассистенты обрабатывают большие объемы обращений как от внешних клиентов компании, так и от сотрудников организации, что позволяет автоматизировать процессы обслуживания.

2 СИСТЕМНЫЕ ТРЕБОВАНИЯ

Функционирование компонентов **ChatNavigator** 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 **ChatNavigator** протестировано на следующих операционных системах:

– AlmaLinux 9.x (9.5 и выше);

– AstraLinux SE 1.8.x (1.8.3 и выше), уровень защищенности «Орел»;

– РЕД ОС 8.x (8.0 и выше);

– SberLinux 8.x (8.10 (Kazbek) и выше) — для SberLinux недоступна установка с помощью Docker.

Платформа совместима со следующим сторонним программным обеспечением:

– Postgres 18;

– MongoDB 6;

– Java 21 (OpenJDK Zulu21);

– RabbitMQ-server 3.13.

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 150;

– Mozilla Firefox 152;

– Microsoft Edge 150;

– Яндекс.Браузер 26.6.1.

2.1 Список используемых портов

Сервис	Используемый порт
BotServer	8083
ChatWidget	9999
FileServer	9320
HttpAdapter	9030
IntegrityVerifier	10101
OmnichatAdapter	9066
ServiceWeb	8086
Telegram	8097
Classifier	9008
Classifier Trainer	9009
SipAdapter	9098
NER	9010
NER Trainer	9011
Data-stream-service	9104 / 9105
Kafka-adapter	9101
Max-adapter	9102
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 **ChatNavigator** показана на рисунках ниже.

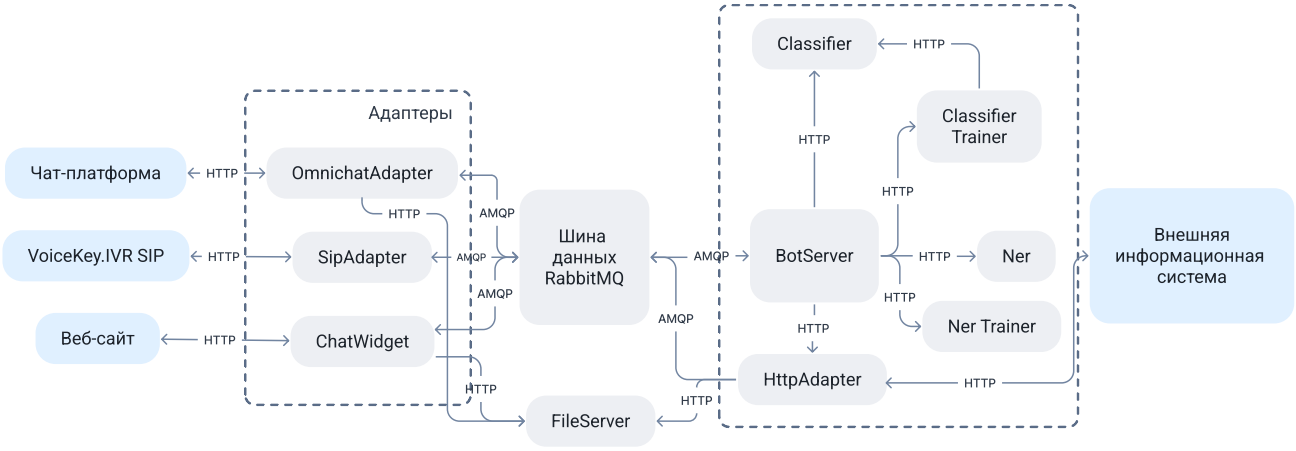


Рисунок 1 — Состав и схема взаимодействия компонентов ChatNavigator части Live

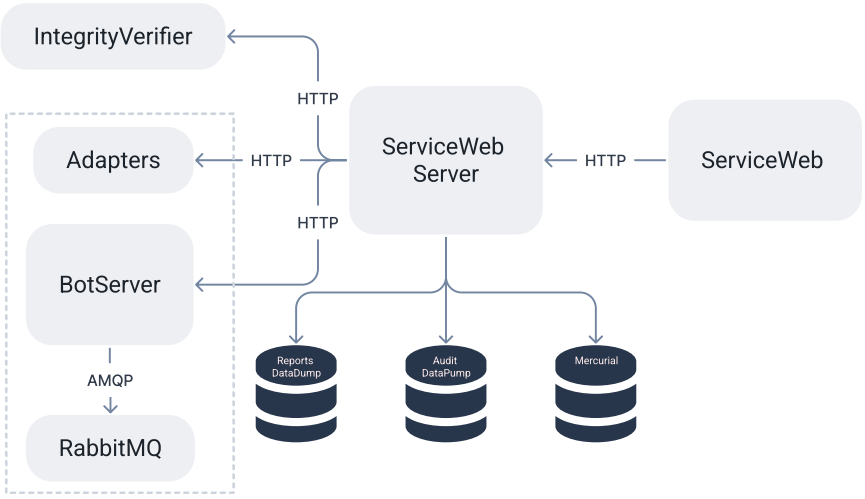


Рисунок 2 — Состав и схема взаимодействия компонентов ChatNavigator части Backoffice

Компонент	Назначение
Live Часть платформы, отвечающая за обработку запросов пользователей	
BotServer	Выполнение пайплайнов и управление конфигурацией и вызов NLU-модулей, используемых для обработки запросов клиентов
HttpAdapter	Выполнение запросов к веб-сервисам и информационным системам из пайплайна
Classifier	Классификация запросов на основе машинного обучения
Classifier Trainer	Обучение классификатора **Classifier**
NER	Извлечение именованных сущностей на основе машинного обучения
NER Trainer	Обучение моделей для извлечения именованных сущностей

Компонент	Назначение	
Шина данных	Взаимодействие между внутренними модулями ChatNavigator по единому протоколу на базе RabbitMQ	
FileServer	Обработка и хранение файлов, загруженных клиентом в рамках сессии	
OmnichatAdapter	Взаимодействие с клиентскими приложениями, мессенджерами, виджетами	
SipAdapter	Обмен данными с VoiceKey.IVR SIP	
Max-adapter	Взаимодействие с мессенджерам MAX	
Kafka-adapter	Получение и передача данных в формате JSON во внешние информационные системы	
Backoffice Часть платформы, отвечающая за хранение и обработку сессий клиента, мониторинг, разработку и настройку пайплайнов и т. д.		
IntegrityVerifier	Контроль целостности компонентов платформы	
ServiceWeb	Веб-приложение Административный интерфейс для просмотра сессий, построения отчетов, просмотра состояния платформы, разработку и настройку пайплайнов и т. п.	
	ServiceWeb Server	HTTP/REST-сервер и веб-приложение
	Reports	База данных сессии
	Audit	База данных аудит-логов
	Mercurial Server	Хранение версионируемых пайплайнов

Платформа включает следующие подсистемы:

- подсистема взаимодействия с клиентом;
- подсистема обработки запросов клиентов;
- подсистема интеграции;
- подсистема хранения и аналитики;
- подсистема проверки целостности системы.

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 асинхроного), 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 системой;

– 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 3.

В контуре обслуживания (верхний контур) обеспечивается горячее резервирование:

– OmnichatAdapter;
– HttpAdapter;
– BotServer;
– Шина данных.

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

Master-сервер по умолчанию установлен на площадке A, slave-сервер — на площадке B.

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 A отключается (если есть доступ), сервер площадки B 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 A 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.

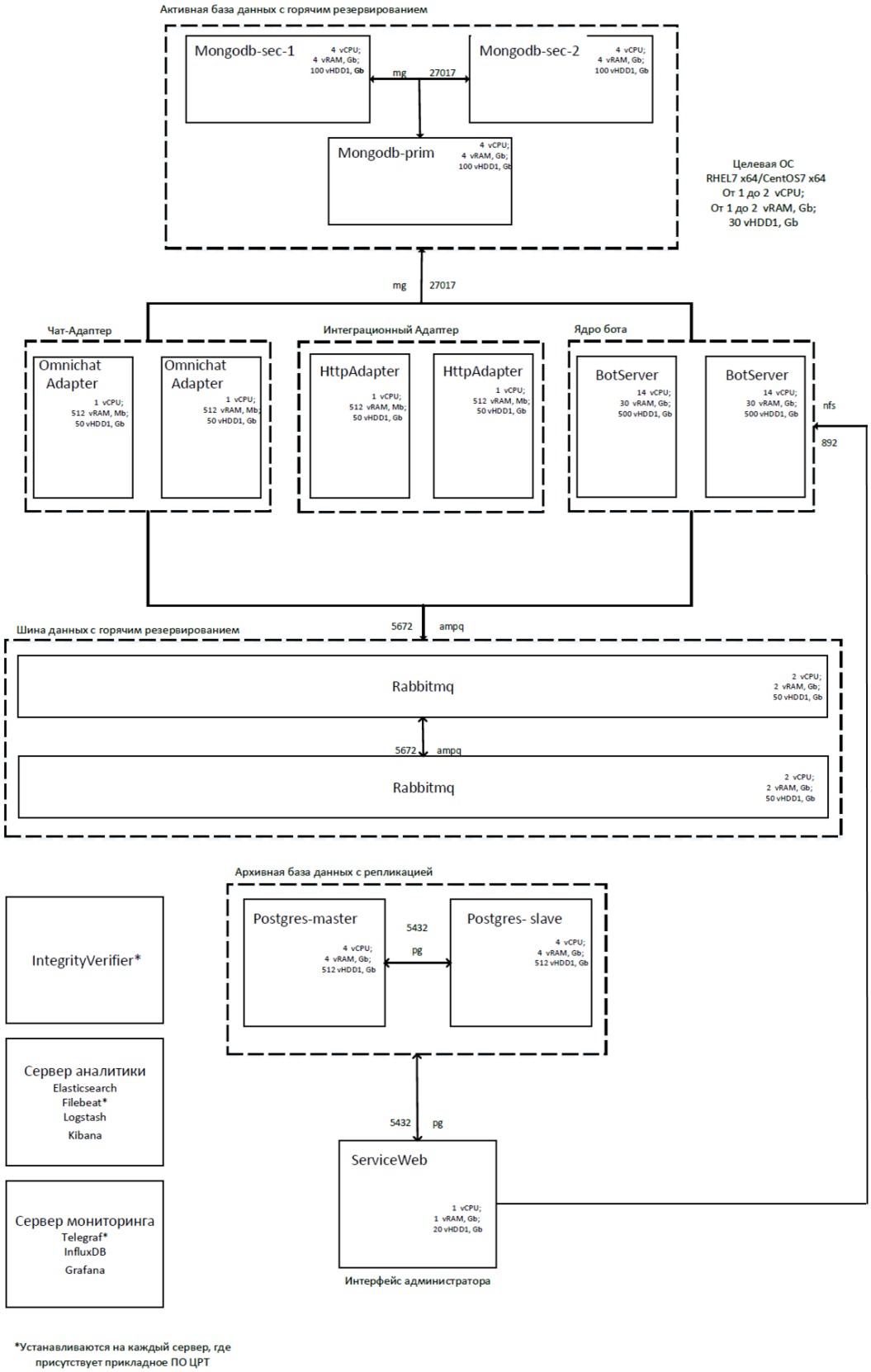


Рисунок 3 — Схема компонентов платформы ChatNavigator с резервированием

4 УСТАНОВКА И НАСТРОЙКА

4.1 Установка ChatNavigator с помощью Docker

Загрузите дистрибутивы поставки

Пример ссылок для скачивания:

Дистрибутивы обязательных компонентов ChatNavigator:

```
https://ftp.speechpro.com/puckr/vc/release/1.0-temp/docker/cn-docker-<version>.tgz.cpt
```

Дистрибутивы инфраструктурных сервисов:

```
https://ftp.speechpro.com/puckr/vc/release/1.0-temp/docker-env/cn-docker-env-
<version>.tgz.cpt
```

Дистрибутивы необязательных компонентов ChatNavigator:

```
https://ftp.speechpro.com/puckr/vc/release/1.0-temp/docker-ad/cn-docker-ad-
<version>.tgz.cpt
```

Модели NER:

```
https://ftp.speechpro.com/puckr/vc/prerequisites/ner_encoder.tar.gz
https://ftp.speechpro.com/puckr/vc/prerequisites/nltk_data.tar.gz
```

Сервисы RAG:

```
https://ftp.speechpro.com/puckr/vc/prerequisites/open/an-prereqs-docker.tgz
https://ftp.speechpro.com/puckr/vc/prerequisites/closed/rag/an-rag-docker.tgz.cpt
https://ftp.speechpro.com/puckr/vc/prerequisites/closed/semanticsearch/an-semanticsearch-
<version>-docker.tgz.cpt
```

Файлы эмбеддингов:

```
https://ftp.speechpro.com/puckr/vc/embeddings/embeddings.tgz
```

Распакуйте модели

1. Создайте директорию для моделей

```
mkdir -p ./models/llm_models/semanticsearch_models
mkdir ./models/nlu-ner
```

Для моделей NER

Распакуйте модели в *models/nlu-ner* (*./models/nlu-ner/ner_encoder* и *./models/nlu-ner/nltk_data*), чтобы получилась структура определенного типа. При изменении путей обязательно скорректируйте соответствующие монтирования в конфигурационных файлах.

```
tar -xzf ner_encoder.tar.gz -C ./models/nlu-ner
tar -xzf nltk_data.tar.gz -C ./models/nlu-ner
```

Для моделей RAG

Распакуйте модели в *models/llm_models*, пример

```
unzip docling-models_1.2.0.zip -d ./models/llm_models/
unzip nlu-common-multilang-768-2.0.0_1.1-778350.zip -d
./models/llm_models/semanticsearch_models/
```

Для эмбеддингов

Распакуйте файлы в *models/classifier_embeddings*, пример

```
tar -zxvf embeddings.tgz -C ./models/classifier_embeddings/cnnsubword
mv ./models/classifier_embeddings/cnnsubword/gensim* ./models/classifier_embeddings/cnn/
```

Расшифруйте архивы

Пример команды:

```
ccrypt -d <зашифрованный_архив>
```

Распакуйте архивы

Пример команды:

```
tar -zxvf <архив>
```

Загрузите Docker-образы

Загрузите образы одной командой, если они находятся в одной директории

```
for f in *__*.tar; do echo "Loading $f" && docker load -i "$f"; done
```

Исключите сервисы

Если это необходимо, измените файл *compose.override.yml* — добавьте компоненты, которые не требуют запуска в контуре. Пример:

```
  head-hunter-token-service:
    profiles: ["exclude"]
```

Также можно исключить или добавить целые блоки удалением или добавлением комментария в секции include.

Закомментируйте override-строки

В файле *docker-compose.yml* закомментируйте строки, содержащие - compose/*-override.yml.

Запустите сервисы

Первичный запуск

Выполните скрипт:

```
run.sh
```

Запуск контейнеров

```
docker compose up -d
```

Остановка контейнеров

```
docker compose down
```

Структура проекта

compose/	Директория с Docker compose-файлами сервисов
data/	Постоянный хранилища: PostgreSQL, MinIO, очереди и др.
kafka/	Конфигурация kafka
logging/	Конфигурация логирования
models/	Архивы и распакованные модели для RAG, NER, классификаторов

monitoring/	Конфигурации мониторинга
scripts/	Вспомогательные скрипты
service-configs/	Директория с конфигурациями компонентов ChatNavigator
service-logs/	Директория с лог-файлами компонентов ChatNavigator
.env	Общий файл файл переменных для всех *docker-compose*
docker-compose.yml	Точка входа для compose-файлов
compose.override.yml	Файл с исключенными компонентами
run.sh	Скрипт для первичного запуска

Для работы RAG необходимо в файле *.env* указать адрес сервера stc-net-license.

4.1.1 Миграция с RPM на Docker

Вот отредактированный вариант. Исправлены опечатки (`сценириями` → `сценариями`), убраны разговорные и тавтологичные конструкции (`сделать бэкап` → `создать резервную копию`, `выключить службы` → `отключить службы`, `дать права` → `настроить права доступа`), повторяющаяся фраза `опираясь(используя, либо модифицируя под нужды)` приведена к единому стандарту. Логика, порядок шагов и команды сохранены без изменений.

1. Создайте резервную копию базы данных PostgreSQL, используя (либо адаптировав под нужды) скрипт *scripts/postgres_backup_db.sh*.

2. Создайте резервную копию базы данных MongoDB, используя (либо адаптировав под нужды) скрипт *scripts/mongo_dump.sh*.

3. Создайте директории для баз данных MongoDB и PostgreSQL, а также Docker-сеть для конфигурации:

```
- mkdir -p ./data/{postgres_data,mongodb_data,serviceweb_data}
- docker network create cn-network || true`
```

4. Скопируйте используемые сценарии из */etc/speechpro/serviceweb/scenario* в *data/serviceweb_data/scenario*, либо измените точку монтирования в *compose/01-infrastructure.yml* и *compose/02-base.yml* для сервисов ServiceWeb и Mercurial на - ../data/serviceweb_data/scenario:/etc/speechpro/serviceweb/scenario.

5. Настройте права доступа (root) для репозитория сценариев после копирования:

```
- chown -R root data/serviceweb_data/scenario
```

6. Измените файлы конфигурации сервисов в **service-configs**, используя в качестве ориентира оригинальные файлы из */etc/speechpro/**.

7. Остановите и отключите работающие локальные службы, используя (либо адаптировав под нужды) скрипт *scripts/stop_local_services.sh*.

8. Выполните миграцию базы данных **PostgreSQL**, используя (либо адаптировав под нужды) скрипт *scripts/postgres_migrate_db_docker.sh*.

9. Выполните миграцию базы данных **MongoDB**, используя (либо адаптировав под нужды) скрипт *scripts/mongo_migrate_db_docker.sh*.

10. Запустите скрипт *run.sh* без дополнительных параметров.

4.2 Установка ChatNavigator с помощью Ansible Playbook

Создание локального репозитория rpm/deb

Выберите один из вариантов в зависимости от используемого дистрибутива.

Дистрибутивы на базе RPM

```
sudo dnf install createrepo -y
sudo mkdir /opt/localrepo/
sudo chown admin /opt/localrepo/
cd /opt/localrepo/
createrepo . --update

cat <<EOF >/etc/yum.repos.d/local.repo
[localrepo]
name=localrepo
baseurl=file:///opt/localrepo/
enabled=1
gpgcheck=0
EOF
```

Дистрибутивы на базе DEB

```
sudo apt-get install dpkg-dev -y
sudo mkdir /opt/localrepo
sudo chown admin /opt/localrepo
cd /opt/localrepo
cat <<EOF | sudo tee /etc/apt/sources.list.d/localrepo.list
deb [trusted=yes] file:/opt/localrepo ./
EOF
sudo -s
dpkg-scanpackages . /dev/null | gzip -9c >Packages.gz
```

Работа с зашифрованными архивами

1. Дешифруйте архив с помощью утилиты **ccrypt**.

```
ccrypt -d <зашифрованный_архив>
```

2. Извлеките содержимое архивов с программным обеспечением deb\rpm в директорию */opt/localrepo/*.

```
tar -zxvf <архив> -C /opt/localrepo/
```

Установка ansible

Установка через pip

1. Перейдите в директорию проекта:

```
cd ./ansible-chatbot
```

2. Создайте виртуальное окружение:

```
python3 -m venv ansible-chatbot
```

3. Активируйте его:

```
source ansible-chatbot/bin/activate
```

4. Установите Ansible:

```
pip install ansible -r requirements.txt
```

5. Установите galaxy requirements:

```
ansible-galaxy install -r requirements.yml
```

6. Деактивируйте виртуальное окружение:

```
deactivate
```

Проверка установки

```
ansible --version
```

Подготовка файла inventory перед запуском

1. Измените версии ПО на необходимые для развертывания.

2. Выберите необходимые компоненты. Компоненты, которые не устанавливаются на хост, необходимо закомментировать с помощью #.

3. Измените **instance:** на DNS-имя или IP-адрес сервера, на который производится установка.

4. После создания локального репозитория с компонентами ChatNavigator в файле *inventory* добавьте:

```
repositories:
  - { name: "localrepo", baseurl: "///opt/localrepo", description: "Local repository" }
```

Запуск Ansible Playbook и установка компонентов ChatNavigator

Выполните одну из команд в зависимости от используемого пользователя.

Пример команды для пользователя root

```
ansible-playbook install.yml -i inventory -u root -k
```

Пример команды для пользователя ansible с правами sudo

```
ansible-playbook install.yml -i inventory -vv -u ansible -b -k -K
```

4.3 Настройка и запуск компонентов

Для установки вручную и/или с помощью Ansible Playbook

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.

Для установки с помощью Docker

Конфигурационные файлы располагаются по пути *<директория_развертывания_Docker>/service-configs*. Содержимое этой директории аналогично */etc/speechpro/<component>/* и выполняет те же функции.
Некоторые примеры далее в руководстве приведены для директории */etc*, но их так же можно выполнить для */service-configs*, изменив директорию в команде.

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

4.3.1 BotServer

Конфигурационный файл

Пример конфигурационного файла */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
mq-events:
  audit: # логирование audit событий через очередь, а не файл. Если выключено, что будет логирование в файл.
    enabled: true # if disabled, audit-logging in files will be used
    queue: audit
  stat-log:
    serviceweb: # логирование событий через очередь, а не файл. Если выключено, что будет логирование в файл.
      enabled: true # if disabled, stat-logging in files will be used
```

```
      queue: stat-log-serviceweb
    dialoghistoryservice: # 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

s3:
  endpoint: http://localhost:3900
  access-key:
  secret-key:
  connect-timeout-seconds: 10
  socket-timeout-seconds: 60

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
    enable:
      jvm: true # значение true включает метрики для мониторинга JVM
  endpoint:
    prometheus:
      enabled: true

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: 300000 #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
    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
  ignore-message:
    ignore-outdated-step-number: false # во включенном состоянии botserver начинает
игнорировать события ExternalCallEvent (от httpadapter), которые задержались, а диалог
ушел вперед из-за новых вопросов пользователя
  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.'
  file-processing:
    state: /fileProcessing
    extensions:
      - pdf
      - txt
      - md
      - html
      - docx
      - doc
      - rtf
      - odt
      - pptx
      - xlsx
      - png
      - jpeg
      - csv
    invalidFileAnswer: 'Файлы имеют невалидный формат.'
    processingErrorAnswer: 'Не удалось обработать файлы.'
  export-processing:
    state: /exportProcessing
    processingErrorAnswer: 'Не удалось обработать экспорт.'
  json-processing:
    state: /jsonProcessing
  loop:
    count-limit: 100

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
  classification-max-iteration-count: 10 # ограничение количества итераций LLM-
классификации
  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
            backoff-period-ms: 100
      status:
        base-path: /restapi/status
        http-client:
          connect-timeout-ms: 500
          socket-timeout-ms: 1000
          connections:
            max-total: 80
            max-per-route: 20
          retry:
            max-attempts: 3
            backoff-period-ms: 100
  tokenizer: #необязательный параметр добавляет токенизацию по regex перед отправкой на
edu.stanford.nlp
    #(?<= ) - look behind, checking what ' ' stays before token \d+
    #(?= ) - look ahead, checking what ' ' stays after token \d+
    # ' ' (space) itself not included in token, when matcher is applied
    #(?<= )\d+(?= )|^\d+(?= )|(?<= )\d+$ - split numbers always
    regex: (?<= )\d+(?= )|^\d+(?= )|(?<= )\d+$
    #stanford 4.5.0 tokenizes: "word" -> [``, word, '']; 'word' -> [`, word, ']. So this
is the fix.
    replaces: { "`": "'", "´": "'", "`": "'", "``": "''", "´´": "''", "''": "''", "«":
"''", "»": "''" }
  #nl-options:
  #  threshold-level: 0.5
  #  spellcheck:
  #    enabled: true
  #    timeout: 100
  #    sort: NOORDER
  #    limit: 3
  #    depth: 4
  #  matcher-to-classifier-threshold: 1.0
```

```
  #  classifier-threshold: 0.0
  #  classifier-no-match-threshold: 0.0
  #  classifier:
  #    enabled: true
  #    weight-ml: 0.5
  #    classifier-type: flat
  #  check-mandatory-words: true
  #  check-star-mandatory-words: true
  level-result: auto
  iterative-matching:         # Настройки многоуровневой классификации
    enabled: false
    max-iteration-count: 20   # Максимальное количество итераций в рамках одного шага
  keyboard-layout-corrector:
    enabled: false

scenario-loader:
  validate-llm-core-availability: true # проверка доступности сервиса llm-core при
загрузке сценария с llm (реакции semanticSearch)
  snakeyaml:
    code-point-limit: 1048576000 # 1000 * 1024 * 1024. Means, that max file size to be
parsed is 1000 MB

spring:
  main:
    banner-mode: 'off'
#       Профили настроек, следует использовать в паре с настройкой сервиса конфигурации
#  profiles:
#    include:
#      - llm-models
#      - llm-sdk-core-client

#       Настройка сервиса конфигурации.
#       Желательна для заполнения, если есть необходимость использовать одни и те же
настройки для нескольких сервисов (httpadater, botserver, serviceweb)
#  config:
#    import: optional:configserver:http://localhost:9078
  jmx:
    enabled: true
  cloud:
    openfeign:
      httpclient:
        max-connections: 80
        max-connections-per-route: 50
        hc5:
          enabled: true
      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:
      tempData.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*![ËёA-яа-zA-Z0-9*]+\\s*\\|.+\\)",
          #at the middle
          "\\(.+\\|\\s*![ËёA-яа-zA-Z0-9*]+\\s*\\|.+\\)",
          #at the end: (a|!a)
          "\\(.+\\|\\s*![ËёA-яа-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: [ "[\\#\\!\\@\\*\\&\\%\\:\\|\\~\\`\\?\\^\\№\\;
\\-\\=]", "[\\<\\>\\(\\)\\[\\]\\{\\}]", "[\\/\\\\]", "\\.$" ] #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: accept  #обязательный параметр, возможные значения: ACCEPT 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

postal:
  llm-response-wait-timeout-sec: 60

debug:
  enabled: false
  default-remove-after-expiry-days: 30
  channels:
    - channel-id: channel1
      remove-after-expiry-days: 45
    - channel-id: channel2
```

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 */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
```

4.3.2 HttpAdapter

Конфигурационный файл

```
spring:
  main:
    banner-mode: 'off'
#       Профили настроек, следует использовать в паре с настройкой сервиса конфигурации
#  profiles:
#    include:
#      - llm-models
#      - llm-sdk-core-client
#      - llm-services # используется в httpadapter и serviceweb

#       Настройка сервиса конфигурации.
#       Желательна для заполнения, если есть необходимость использовать одни и те же
настройки для нескольких сервисов (httpadater, botserver, serviceweb)
#  config:
#    import: optional:configserver:http://localhost:9078
  cloud:
    config:
      import-check:
        enabled: false
    openfeign:
      httpclient:
        max-connections: 200
        max-connections-per-route: 50
        hc5:
          enabled: true
```

```
          socket-timeout: 5 # особенно актуально для медленных ресурсов
          socket-timeout-unit: seconds
          connection-request-timeout: 180000
          connection-request-timeout-unit: MILLISECONDS

      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
          cotype-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          qwen-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          deep-seek-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          chat-gpt-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          llm-core:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 10_000
          long-doc:
            url: http://localhost:8400
            connect-timeout: 1_000
            read-timeout: 600_000 # 10 minutes
          llm-document-parser:
            url: http://localhost:33120
            connect-timeout: 1_000
            read-timeout: 1_200_000 # 20 minutes
          database-http-adapter:
            url: http://localhost:8094
```

```
            connect-timeout: 1_000
            read-timeout: 30_000
          python-http-adapter:
            url: http://localhost:8095
            connect-timeout: 1_000
            read-timeout: 30_000
  grpc:
    client:
      channels:
        data-stream-service: # подключение к data-stream-service по gRPC
          target: static://localhost:9105
          bypass-certificate-validation: false

  datasource:
    url: jdbc:postgresql://localhost:5432/httpadapter
    username: httpadapter
    password: httpadapter
    type: com.zaxxer.hikari.HikariDataSource
    hikari:
      minimum-idle: 1
      maximum-pool-size: 10
  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
  flyway:
    enabled: true

s3:
  endpoint: http://localhost:3900
  access-key:
  secret-key:
  connect-timeout-seconds: 10
  socket-timeout-seconds: 60

file-server:
  host: localhost
  port: 9320
  timeout: 60_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


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:
      ecma-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:
  postgres-timeout-ms: 30000

llm: # используется в httpadapter и serviceweb. Может быть настроено в config-service
  default-vendor: GIGA_CHAT
  giga-chat:
    enabled: false
    auth-required: true
    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 # 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
      connection-properties:
        max-connections: 200
        max-connections-per-route: 50
    channel-specific:
      - channel-id: gigaChatSpecificChannelId1
        auth:
          username: someUser1
          password: somePassword1
          #prod:
            #scope: GIGACHAT_API_CORP
        completions:
          profanity-check: true
          top-p:    # (Возможные значения: >= 0 и <= 1, например 0.55)
        # certificate:
        # ... состав параметров аналогичен параметру llm.giga-chat.certifacte
      - channel-id: gigaChatSpecificChannelId2
        auth:
          username: someUser2
          password: somePassword2
        completions:
          profanity-check: false
          top-p:    # (Возможные значения: >= 0 и <= 1, например 0.55)
    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: #пароль для хранилища и сертификатов
        keys-password: #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: #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
      model-id: yandexgpt/latest
      model-name: YandexGPT Pro
      uri: /foundationModels/v1/completion
      retry:
        max-attempts: 2
        back-off-period-ms: 100
  cotype:
    enabled: false
    bearer-token: Bearer token
    completions:
      model-id: cotype_pro_2
      model-name: CotypePro
      uri: /v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  qwen:
    enabled: false
    bearer-token: Bearer token
    completions:
      model-id: /models/Qwen3-32B
      model-name: Qwen 3 32B
```

```
      uri: /v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  deep-seek:
    enabled: false
    bearer-token: Bearer token
    completions:
      model-id: deepseek-chat
      model-name: Deepseek
      uri: /api/v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  chat-gpt:
    enabled: false
    api-key: someApiKey
    completions:
      model-id: gpt-oss-20b
      model-name: GPT-OSS-20b
      uri: /api/v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1

  stream:
    data-stream-service:
      # Отдельные настройки, которые будут переданы в BotServer.
      # Отдельные, так как потребитель этих настроек может иметь другую сеть.
      # Сам HttpAdapter использует настройки spring.grpc.client.channels.data-stream-
service
      stream-info:
        host: localhost
        port: 9105
        secure: false

ai-algo:
  ignore-errors: true

mq-events:
  audit: # логирование audit событий через очередь, а не файл. Если выключено, что будет
логирование в файл.
    enabled: true # if disabled, audit-logging in files will be used
    queue: audit
```

Настройка и запуск

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

1. В файле ***/httpadapter/application.yml*** укажите:

```
rabbitmq:
  host: mqhost
```

mqhost — имя или IP-адрес хоста, где установлен **RabbitMQ**.

2. Запустите программу:

```
systemctl start httpadapter.service
```

4.3.3 ServiceWeb

Конфигурационный файл

```
server:
  port: 8086
  servlet:
    context-path: /
    session:
      cookie:
        name: JSESSIONID
        http-only: true
        secure: false  # для отправки cookie только через защищенное соединение (https)
установить значение true
      timeout: 3600
  forward-headers-strategy: framework

web-socket:
  ping-delay-seconds: 10 # как часто отправляем сообщения типа ping(для поддержки
соединения) подключенным клиентам
  message-timeout-seconds: 10 # таймаут отправки сообщения
  allowed-origins: '*' # настройка разрешенного заголовка Origin при подключении, по
умолчанию * - позволяющий любой Origin


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
найдена 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

s3:
  endpoint: http://localhost:3900
  access-key:
  secret-key:
  connect-timeout-seconds: 10
  socket-timeout-seconds: 60

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


rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 16

mq-events:
  audit: # логирование audit событий через очередь, а не файл. Если выключено, что будет
логирование в файл.
    enabled: true # if disabled, audit-logging in files will be used
    queue: audit
  stat-log:
    enabled: true
    thread-count: 3
    queues: { node1: stat-log-serviceweb }
    scheduling:
      enabled: true
      delay-ms: 100

pump:
  stat-log:
    enabled: false
    stat-log-file-name: botserver.stat.json.log
    stat-logs-paths: { node1: /var/log/speechpro/botserver }
    delay-ms: 100
    buffer: 1000
  audit-log-file-name: audit.log
  audit-logs-path: /var/log/speechpro
  audit:
    enabled: false
    delay: 1000
    buffer: 1000
  day-session-counts:
    enabled: false
    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: false
    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: false
    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: false
    log-file-name: botserver.sessiontophintstates.json.log
    logs-paths: ${pump.sessionTopHintStates.logsPaths}
    delay: 1000
    buffer: 1000

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'
#       Профили настроек, следует использовать в паре с настройкой сервиса конфигурации
#  profiles:
#    include:
#      - llm-models
#      - llm-services # используется в httpadapter и serviceweb

#       Настройка сервиса конфигурации.
```

```
#       Желательна для заполнения, если есть необходимость использовать одни и те же
настройки для нескольких сервисов (httpadater, botserver, serviceweb)
#  config:
#    import: optional:configserver:http://localhost:9078

  config:
    import: optional:classpath:registry-links-mapping.yml # импорт настроек из файла с
маппингом. Если используется configserver, то импорты указываются друг за другом, через
запятую

  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
        hc5:
          enabled: true
      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
          cotype-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          qwen-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          deep-seek-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          chat-gpt-completions: # используется в httpadapter и serviceweb. Может быть
настроено в config-service
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_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
    auth-required: true
    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:
    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
  cotype:
    enabled: false
    bearer-token: Bearer token
    completions:
      model-id: cotype_pro_2
      model-name: CotypePro
      uri: /v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  qwen:
    enabled: false
    bearer-token: Bearer token
    completions:
      model-id: /models/Qwen3-32B
      model-name: Qwen 3 32B
      uri: /v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  deep-seek:
    enabled: false
    bearer-token: Bearer token
    completions:
      model-id: deepseek-chat
      model-name: Deepseek
      uri: /api/v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  chat-gpt:
    enabled: false
    api-key: someApiKey
    completions:
      model-id: gpt-oss-20b
      model-name: GPT-OSS-20b
      uri: /api/v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1

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.'



rag:
  llm:
    document-storage:
```

```
      connect-timeout-seconds: 1
      read-timeout-seconds: 36000
      long-poll-read-timeout-seconds: 60 # llm-document-storage default timeout(poll from
queue) is 30 seconds
      documents-subscription-error-timeout-seconds: 60
      documents-subscription-not-found-timeout-seconds: 1
      status-update-enabled: true
      collection-documents-page-size: 10_000
      import-file-threads-count: 10
      parser-options:
        parser-name:
          default-value: docling
          values: [PyMuPDF, docling]
        pdf-backend:
          default-value: dlparse_v4
          values: [pypdfium2, dlparse_v1, dlparse_v2, dlparse_v4]
        pipeline:
          default-value: standart
        do-ocr:
          default-value: false
          values: [true, false]
        ocr-lang:
          values: [ru, en]
        ocr-engine:
          default-value: easyocr
          values: [easyocr]
        force-ocr:
          default-value: false
          values: [true, false]
        image-export-mode:
          default-value: embedded
          values: [embedded, placeholder, referenced]
        image-scale:
          default-value: 1.0
        do-picture-description:
          default-value: false
          values: [true, false]
        do-table-structure:
          default-value: false
          values: [true, false]
        table-mode:
          default-value: FAST
          values: [FAST, ACCURATE]
        table-cell-matching:
          default-value: false
          values: [true, false]
        do-code-enrichment:
          default-value: false
          values: [true, false]
        do-formula-enrichment:
          default-value: false
          values: [true, false]
        md-page-break-placeholder:
          default-value: " "
        page-range:
          from:
          to:
        document-timeout:
          default-value: 3600


    directory-import:
      enabled: false
      watcher-thread-pool-size: 2 # количество потоков, следящих за изменениями в
каталогах
      max-file-size-kb: 4096 # максимальный размер файла для импорта в килобайтах
```

```
      max-parallel-imports: 4 # количество файлов для параллельного импорта. Хорошее
первое приближение: -Xmx / max-file-size-kb
      delete-files-enabled: true
      delete-file-hashes-enabled: true
      registry-check-enabled: false
      downloader:
        connect-timeout: 1_000 # тайм-аут подключения
        read-timeout: 30_000 # тайм-аут ожидание ответа
        follow-redirects: false # нужно ли переходить по ссылкам с перенаправлением
        certificate: #optional: конфигурация для настройки tls
          protocol: TLS #optional: выбор протокола/алгоритма, по умолчанию SSL
          enable-hostname-verification: false #optional: позволяющий включить верификацию
хоста при отправки запроса, используется в mTLS
          keystore: #optional: хранилище ключей
            path:  #путь к хранилищу keystore
            password: ''#пароль для хранилища и сертификатов
            keys-password: ''#optional: является паролем для сертификата. По умолчанию
равен password.
            format: PKCS12 #формат, возможные значения: PKCS12, JKS
            aliases: [1] #optional: предназначен для задания названия aliase для
сертификата
          truststore: #optional: хранилище доверенных сертификатов
            path: #путь к хранилищу truststore
            password: '' #optional: пароль для хранилища. "" по умолчанию.
            format: JKS #формат, возможные значения: PKCS12, JKS
            aliases: [1] #optional: предназначен для задания названия aliase для
сертификата

ldap:
  enabled: false
  enable-hostname-verification: true # возможность выключить проверку hostname по
сертификатам
  connection:
    url:
    base:
    admin-user-dn:
    admin-password:
    secret:     # используется для кодирования/декодирования пароля, например:
ge7a8qxc1o3jpm23
    security: startTls
  authentications:
    - dn: ou=Branches
      user-name-attr: sAMAccountName
  mapping:
    user: {shortName: sAMAccountName, fullName: displayName, email: mail, position:
sapTitle, city: l, 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

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-attems: 30

quality:
  sub-selection-interval-min: 60
  unknown-ration: 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: ["[а-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:
  clean:
    delay-ms: 3600000
    inactive-timeout: 1 days
  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

security-policies:
  user-inactivity-period-in-days: 90   # допустимые значения: от 90 до 180
  max-failed-attempts: 5  # допустимы значения: от 2 до 5
  lockout-duration-minutes: 30  # допустимы значения: от 30 до 180
  max-temp-locks: 2  # допустимы значения: от 1 до 10
  password:
    expiration: true
    expire-days: 90 # допустимы значения: от 30 до 90
    expiration-notice-days: 7 # допустимы значения: от 3 до 14
    same-password-validation:
      enabled: false
```

```
      history-depth: 25  # допустимы значения: от 1 до 25
```

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 */serviceweb/application.yml*.

1. Если конфигурация сервисов **ChatNavigator** соответствует стандартной и включает в себя только **BotServer**, **HttpAdapter** и **ChatWidget**, то в файле */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 sipadaper или botserver или
chatwidget
      - abstract-channel-id-1  # список идентификаторов каналов сервиса(соответствует
тем, что находились в секции integration.vcservices.{service-name}.dedicated-channeld
когда канал был актуален)
      - abstract-channel-id-2
```

Если конфигурация отличается от стандартной, то в файле */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 отображаются 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 sipadaper или botserver или
chatwidget
      - abstract-channel-id-1  # список идентификаторов каналов сервиса(соответствует
тем, что находились в секции integration.vcservices.{service-name}.dedicated-channeld
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 */serviceweb/application.yml*.

2. Если для мастер-бота и скилл-бата 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
```

4.3.3.1 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
```

4.3.3.2 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 (требует 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.

4.3.3.3 Настройка контекста для веб-компонентов

Для изменения контекста достаточно установить значение в соответствующем конфигурационном файле */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**.

4.3.3.4 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.

4.3.3.5 Маппинг ссылок для автоматического скачивания документов

Конфигурационный файл ***registry-links-mapping.yml*** обеспечивает маппинг ссылок для скачивания документов для импорта в коллекцию документов в RAG.

```
links:
  mappings:
    - host: servername1.ru # хост, для которого необходимо выполнять данные
преобразования ссылок
```

```
      original-link: https://servername1.ru/document?moduleId=1&documentId=$document_id #
темплейт для ссылки на превью документа, где указывается расположение его id
      download-link: https://servername1.ru/rtf?moduleId=1&documentId=$document_id #
темплейт для ссылки на скачивание документа, где указывается, куда подставлять id
    - host: servername2.ru
      original-link: http://servername2.ru/Document/View/$document_id
      download-link: http://servername2.ru/file/pdf?eoNumber=$document_id
```

4.3.4 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 */integrityverifier/application.yml* укажите:

```
postgres:
    host: pghost
```

Где **pghost** — сервер PostgreSQL.

2. В файле */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**. Пример команды для файла конфигурации при установки или с помощью **Ansible**:

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

4. Создайте снимок текущего состояния компонентов комплекса. Пример команды для файла конфигурации при установки или с помощью **Ansible**:

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

5. Запустите программу:

```
systemctl start integrityverifier.service
```

4.3.5 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.surename=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 параметров, 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
  extensions: [doc, docx, pdf, txt, md, xls, xlsx, png, jpg, jpeg, ppt, pptx, html]
  special-chars: "#!@*&\\%/:<>'|\"~`?^"
  invalid-file-message: К сожалению, файл не отвечает нашим требованиям


rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 5
  prefetch: 10

spring:
  main:
```

```
    show_banner: false
  jmx:
    enabled: true

spring:
  main:
    banner-mode: 'off'
  jmx:
    enabled: true
  datasource:
    url: jdbc:postgresql://localhost:5432/chatwidget
    username: chatwidget
    password: chatwidget
    type: com.zaxxer.hikari.HikariDataSource
    hikari:
      minimum-idle: 1
      maximum-pool-size: 10
  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
  flyway:
    enabled: true

retry:
  max-attempts: 10
  backoff:
    delay: 100

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:
  postgres-timeout-ms: 30000

mq-events:
  audit: # логирование audit событий через очередь, а не файл. Если выключено, что будет
логирование в файл.
    enabled: true # if disabled, audit-logging in files will be used
    queue: audit
```

Настройка и запуск

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

1. В файле */chatwidget/application.yml* укажите:

```
rabbitmq:
  host: mqhost
```

mqhost — имя или IP-адрес хоста, где установлен RabbitMQ.

2. Запустите программу:

```
systemctl start chatwidget.service
```

4.3.5.1 Кастомизация ChatWidget

1. Создайте ***siteconfig.yml***. Пример команды для файла конфигурации при установки или с помощью Ansible:

```
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: '#e6ff93'  # 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 хэдере 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
```

4.3.5.2 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 */chatwidget/application.yml*:

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

4.3.6 OmnichatAdapter

Конфигурационный файл

```
omnichatadapter:
  processing-time-warn-threshold-ms: 2000
  unsupported-bot-event-message: Извините, мне не удалось получить информацию.
  response:
    question:
      sync:
        timeout-enabled: true
        timeout-sec: 3000 #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:
#     Форматирование по умолчанию.
      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:
```

```
    thread-count: 1  #число потоков (пул), обрабатывающих сообщения, полученные из
rabbitmq
  session:
    timeout-enabled: true
    timeout-sec: 6000
    timeout-check-interval-sec: 60
  file-response-type: multipart
  session-memory-size: # системные настройки по регулированию буфера памяти сессии
    question-count: 20
    request-count: 20
server:
  port: 9066
  servlet:
    encoding:
      enabled: true
      charset: UTF-8
      force: true

spring:
  main:
    banner-mode: 'off'
  servlet:
    multipart:
      max-file-size: -1
      file-size-threshold: 0
      max-request-size: -1
  mvc:
    converters:
      preferred-json-mapper: jackson
  datasource:
    url: jdbc:postgresql://localhost:5432/omnichatadapter
    username: omnichatadapter
    password: omnichatadapter
    type: com.zaxxer.hikari.HikariDataSource
    hikari:
      minimum-idle: 1
      maximum-pool-size: 10
  flyway:
    enabled: true
    baseline-on-migrate: true
  retry:
    enabled: true
retry:
  max-attempts: 10
  backoff:
    delay: 100
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
  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

file-server:
  host: localhost
  port: 9320
  timeout: 60000
file:
  max-size-kb: 15360
  extensions: [doc, docx, pdf, txt, md, xls, xlsx, png, jpg, jpeg, ppt, pptx, html, rtf,
odt, csv]
  special-chars: "#!@*&\\%/:<>'|\"~`?^"
  invalid-file-message: Файлы имеют невалидный формат или размер файлов превышает
допустимое значение.

multipart:
  max-file-size: 52428800
  max-request-size: -1
  file-size-threshold: 512000

channels:
  channel-prefix: omnichat-adapter
  default-channels:
    enabled: true
    action: add_if_not_exist
    channels:
      - channel-id: omnichat-adapter
#         response-formatting: #включение кастомного форматирования для канала.
#           Идентификатор из formattings выше.
#           - Если use не указан, используемся форматирование по умолчанию.
#           -- Если указан default-configuration ниже, то default-configuration ниже будет
полностью определять форматирование.
#           - Если use указан, используется указанное форматирование.
#           -- Если так же определен default-configuration ниже, замены из него
добавляются.
#           use: testFormatting
#           default-configuration: #добавление или определение правил форматирования в
зависимости от use, см. выше.
#             replacement-patterns:
#               - pattern: "testChanged"
#                 replace-to: "replaycedChanged"
        response-endpoint:
          host: localhost
          port: 9077
          path: /
          scheme: http

prompt:     # изменения после CHAT-10665
  enabled: false #включен запрос подсказок или нет (если параметр выключен, запросы на
ботсервер/промптерадаптер все равно могут уходить (для записи истории))
  prompter: false #получим ли мы подсказки от промптера (суфлера)
  bot: false #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: onQuestion # 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
  postgres-timeout-ms: 30000
  file-server:
    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

mq-events:
  audit: # логирование audit событий через очередь, а не файл. Если выключено, что будет
логирование в файл.
    enabled: true # if disabled, audit-logging in files will be used
    queue: audit
```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага 2.

1. В файле */omnichatadapter/application.yml* укажите:

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

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

2. Для создания каналов укажите в файле */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 */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/*.

Пример настройки канала, по которому не требуются ответы, например каналы типа **Trigger-service**:

```
channels:
  channel-prefix: omnichat-adapter
  default-channels:
    enabled: true
    action: add_if_not_exist
    channels:
      - channel-id: omnichat-adapter
```

3. Запустите программу:

```
systemctl start omnichatadapter.service
```

4.3.6.1 Настройка форматирования ответов для OmnichatAdapter

Настройки форматирования ответов задаются в конфигурационном файле */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-inherritance
        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: "ChannelTypeReplayced"
          default-configuration:
            replacement-patterns:
              - pattern: "text"
                replace-to: "replayced text"

channels:
```

```
  default-channels:
    channels:
      - channel-id: omnichat-html-formatting-inherritance
        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.

4.3.7 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

mq-events:
  audit: # логирование audit событий через очередь, а не файл. Если выключено, что будет
логирование в файл.
```

```
    enabled: true # if disabled, audit-logging in files will be used
    queue: audit
```

О порядке установки и настройки голосовой платформы **VoiceKey.PLATFORM** для работы совместно с **ChatNavigator** 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 */sipadapter/application.yml* укажите:

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

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

2. Для создания каналов укажите в файле */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 */serviceweb/application-integration.yml*.

3. Запустите программу:

```
systemctl start sipadapter.service
```

4.3.8 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
```

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

```
spring:
  cloud:
    openfeign:
      httpclient:
        hc5:
          enabled: true
      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: 1_200_000
          yandex-gpt-completions:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 10_000
          cotype-completions:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          qwen-completions:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 1_200_000
          deep-seek-completions:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          chat-gpt-completions:
            url: localhost
            connect-timeout: 1_000
            read-timeout: 30_000
          llm-core:
            connect-timeout: 1_000
            read-timeout: 1_800_000
          llm-document-storage:
            connect-timeout: 1_000
            read-timeout: 100_000
          llm-document-parser:
            connect-timeout: 1_000
            read-timeout: 1_800_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
  cotype:
    bearer-token: Bearer token
    completions:
      model-id: cotype_pro_2
      model-name: CotypePro
      uri: /v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  qwen:
    bearer-token: Bearer token
    completions:
      model-id: /models/Qwen3-32B
      model-name: Qwen 3 32B
      uri: /v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  deep-seek:
    bearer-token: Bearer token
    completions:
      model-id: deepseek-chat
      model-name: Deepseek
      uri: /api/v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
  chat-gpt:
    api-key: someApiKey
    completions:
      model-id: gpt-oss-20b
      model-name: GPT-OSS-20b
      uri: /api/v1/chat/completions
      retry:
        max-attempts: 2
        back-off-period-ms: 1
```

4. *application-langchain-llm-models.yml* (их используют BotServer, ServiceWeb):

```
langchain:
  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
```

5. *application-langchain-tools.yml* (их используют BotServer, ServiceWeb):

```
langchain:
```

```
  tools:
    - DuckDuckGo
    - FileSystem
    - SQL
    - Jira
    - Wikipedia
    - Playwright
```

4.3.9 Telegram

Конфигурационный файл

```
server:
  port: 8087

spring:
  datasource:
    url: jdbc:postgresql://localhost:5432/telegram
    username: telegram
    password: telegram
    type: com.zaxxer.hikari.HikariDataSource
    hikari:
      minimum-idle: 1
      maximum-pool-size: 10
  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
  flyway:
    enabled: true

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

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:
  postgres-timeout-ms: 30000

mq-events:
  audit: # логирование audit событий через очередь, а не файл. Если выключено, что будет
логирование в файл.
    enabled: true # if disabled, audit-logging in files will be used
    queue: audit
```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага 2.

2. В файле */telegram/application.yml* укажите:

```
rabbitmq:
  host: mqhost
```

mqhost — имя или IP-адрес хоста, где установлен RabbitMQ.

2. В файле */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
```

4.3.10 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:                                  # Список точек авторизации типа keyCloak
      - auth-service-id: key-cloak               # Уникальный идентификатор точки.
Используется при настройке каналов
        host: localhost
        port: 80
        path: /auth/realms/LDAP
        scheme: https
        login:
        password:
        client-id:
        client-secret:
    craft-talk-api:                             # Список точек авторизации типа craftTalkApi
      - auth-service-id: craft-talk-api         # Уникальный идентификатор точки.
Используется при настройке каналов
        host: localhost
        port: 443
        path: /api/external/auth/authorize
        scheme: https
        login:
        password:
```

```
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

mq-events:
  audit: # логирование audit событий через очередь, а не файл. Если выключено, что будет
логирование в файл.
    enabled: true # if disabled, audit-logging in files will be used
    queue: audit
```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага 2.

1. В файле */crafttalkadapter/application.yml* укажите:

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

```
    - mghost
```

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/*.

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

2. Для создания каналов укажите в файле*/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 */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
```

4.3.11 Classifier и Classifier Trainer

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

Скачайте архив с эмбеддингами 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 эмбеддингов, 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 эмбеддингов, 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 эмбеддингов поставляются вместе с дистрибутивом.

4.3.12 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

```
gunicorn:
  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
```

4.3.13 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)
  datasource:
    url: jdbc:postgresql://localhost:5432/fileserver
    username: fileserver
    password: fileserver
    type: com.zaxxer.hikari.HikariDataSource
```

```
    hikari:
      minimum-idle: 1
      maximum-pool-size: 10
  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
  flyway:
    enabled: true

file-server:
  directory: /etc/speechpro/fileserver/storage   #базовая директория, в которую будут сохраняться файлы
  base-url: http://localhost:9320   #базовый урл, используется при формировании URL (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)
    max-connections: 100
    max-connections-per-route: 20

logging:
  access-log:
    annotation-enabled: true

healthcheck:
  postgres-timeout-ms: 30000
```

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"
                }
          }
      }
```

4.3.14 Kafka-adapter

Конфигурационный файл

Файл ***application.yml***:

```
server:
  port: 9101 # HTTP порт для healthcheck

spring:
  kafka:
    bootstrap-servers: localhost:9092   # SASL/PLAIN without TLS/SSL
    security:
      protocol: SASL_PLAINTEXT
    properties:
      sasl:
        mechanism: PLAIN
        jaas:
          config: >
            org.apache.kafka.common.security.plain.PlainLoginModule required
            username="admin"
            password="admin-secret";

#    bootstrap-servers: localhost:9093   # SASL/PLAIN over TLS config
#      security:
#        protocol: SASL_SSL
#    properties:
#      sasl.mechanism: PLAIN
#      sasl.jaas.config: >
#        org.apache.kafka.common.security.plain.PlainLoginModule required
#        username="admin"
#        password="admin-secret";
#    ssl:
#      trust-store-location: file:/path/to/truststore.jks
#      trust-store-password: changeme
#      trust-store-type: JKS

#    bootstrap-servers: localhost:9093   # mTLS config
#      security:
#        protocol: SSL
#      ssl:
#        key-store-location: file:/path/to/keystore.jks
#        key-store-password: changeme
#        key-password: changeme
#        key-store-type: JKS
#        trust-store-location: file:/path/to/truststore.jks
#        trust-store-password: changeme
#        trust-store-type: JKS
    consumer:
      group-id: kafka-adapter # Если есть больше 1 kafka-adapter, то параметр может быть
полезен. См. consumer groups в документации Apache Kafka
      auto-offset-reset: earliest # Читать все сообщения из топика, а не только новые
(latest - только новые)
      key-deserializer: org.apache.kafka.common.serialization.StringDeserializer
      value-deserializer: org.springframework.kafka.support.serializer.JsonDeserializer
      properties:
        spring.json.use.type.headers: false
        spring.json.trusted.packages: "*"
        spring.json.value.default.type: com.fasterxml.jackson.databind.JsonNode

    producer:
      key-serializer: org.apache.kafka.common.serialization.StringSerializer
      value-serializer: org.springframework.kafka.support.serializer.JsonSerializer
      properties:
        spring.json.add.type.headers: false
```

```
management:
  endpoints:
    web:
      exposure:
        include: health

kafka-adapter:
  consume:
    topics: # Топики для чтения
      - name: topic.one # Название топика для чтения
        channel-id: kafka-adapter-custom # Специфичный канал для определенного топика
      - name: topic.two
    default-channel: kafka-adapter # Канал по умолчанию, если не задан специфичный для
топика

  publish:
    default-topic: default.topic # Топик для публикации ответа от botserver по умолчанию,
если не был задан специфичный через сценарий (responseData.kafkaTopic)

rabbitmq: # Настройки подключения к RabbitMQ для взаимодействия с botserver
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 8
```

Настройка и запуск

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

1. В файле */kafka-adapter/application.yml* укажите:

```
rabbitmq:
  host: mqhost
```

mqhost — имя или IP-адреса хоста, где установлены RabbitMQ.

2. Запустите программу:

```
systemctl start kafka-adapter.service
```

Для работы Kafka-adapter в сценарии должен быть стейт для обработки JSON и реакция jsonAnswer (указание `kafkaTopic` опциональное, см. `default-topic` из конфигурации).

4.3.15 Trigger-service

Конфигурационный файл

Файл *application.yml*:

```
server:
  port: 9100
spring:
  kafka:
    bootstrap-servers: localhost:9092
    consumer:
      group-id: kafka-adapter
      auto-offset-reset: earliest
      key-deserializer: org.apache.kafka.common.serialization.StringDeserializer
      value-deserializer: org.springframework.kafka.support.serializer.JsonDeserializer
      properties:
```

```
        spring.json.use.type.headers: false
        spring.json.trusted.packages: "*"
        spring.json.value.default.type: com.fasterxml.jackson.databind.JsonNode

    producer:
      key-serializer: org.apache.kafka.common.serialization.StringSerializer
      value-serializer: org.springframework.kafka.support.serializer.JsonSerializer
      properties:
        spring.json.add.type.headers: false
  cloud:
    openfeign:
      httpclient:
        max-connections: 50
        max-connections-per-route: 10
        hc5:
          enabled: true
      client:
        config:
          kafka-adapter: # mostly used for healthcheck
            url: http://localhost:9101
            connect-timeout: 1_000
            read-timeout: 30_000
          omni-chat-adapter:
            url: http://localhost:9066
            connect-timeout: 1_000
            read-timeout: 30_000


schedules:
  scheduling-pool-size: 25
healthcheck:
  delay: 60000 # delay of how often consumer service(kafka-adapter) health will be
checked, 60 seconds by default
mq-events:
  audit:
    enabled: true
    queue: audit
rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 8
```

Пример расписания по отправке пустого JSON (EMPTY_JSON) сообщения в сторону kafka-adapter каждые 5 секунд:

```
schedules:
  kafka-adapter:
    - schedule: "*/5 * * * * *" # every 5 seconds
      active: true
      healthcheck-timeout: 4              # 4 seconds timeout to get REST http response
from kafka-adapter about health status
      topic-query: topic.one
      payload-type: EMPTY_JSON            # Возможные значения: EMPTY_JSON, JSON
      payload-path:
```

Пример расписания по отправке сообщения в сторону OmnichatAdapter каждые 5 секунд:

```
schedules:
  omni-chat-adapter:
    - schedule: "*/5 * * * * *" # every 5 seconds
      active: true
      healthcheck-timeout: 4
      channel-id: omnichat-adapter
```

```
      payload: trigger message
      reply-to: http://hostname:8080/resource-path           # опционально
```

Настройка и запуск

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

1. В файле */trigger-service/application.yml* укажите:

```
rabbitmq:
  host: mqhost
```

mqhost — имя или IP-адреса хоста, где установлены RabbitMQ.

2. Запустите программу:

```
systemctl start trigger-service.service
```

4.3.16 Max-adapter

Конфигурационный файл

Файл *application.yml*:

```
server:
  port: 9102
spring:
  datasource:
    url: jdbc:postgresql://localhost:5432/maxadapter
    username: maxadapter
    password: maxadapter
    type: com.zaxxer.hikari.HikariDataSource
    hikari:
      minimum-idle: 1
      maximum-pool-size: 10
  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
  flyway:
    enabled: true
  cloud:
    config:
      import-check:
        enabled: false
    openfeign:
      httpclient:
        max-connections: 200
        max-connections-per-route: 50
        hc5:
          enabled: true
      client:
        config:
          max-messenger:
            url: https://platform-api.max.ru
            connect-timeout: 5_000
            read-timeout: 30_000
max-adapter:
  default-answer: Извините, мне не удалось получить информацию.
  session-timeout: 10800
  session-expiration-cron: "0 * * * * *" # check every minute
  mq-processing-thread-count: 10
  web-hook-callback-template: https://servername.com/restapi/channels/{channelId}/webhook
```

```
mq-events:
  audit:
    enabled: true
    queue: audit
rabbitmq:
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 8
file-server:
  host: localhost
  port: 9320
  timeout: 60000
file:
  max-size-kb: 15360
  extensions: [doc, docx, pdf, txt, md, xls, xlsx, png, jpg, jpeg, ppt, pptx, html, rtf,
odt, csv, webp]
  special-chars: "#!@*&\\%/:<>'|\"~`?^"
  invalid-file-message: Файлы имеют невалидный формат или размер файлов превышает
допустимое значение.
```

Настройка и запуск

При установке на несколько компьютеров выполните все шаги. При установке на один компьютер начните выполнение с шага 2.

1. В файле */max-adapter/application.yml* укажите:

```
rabbitmq:
  host: mqhost
```

mqhost — имя или IP-адрес хоста, где установлен RabbitMQ.

2. В файле */max-adapter/application.yml* задайте блок настроек channels. В блоке элементы представлены в виде списка, каждый из которых имеет поля id (идентификатор канала), format (формат ответа, возможные значения: markdown, html), access-token (токен бота MAX, через который будет общение):

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

3. В web-hook-callback-template укажите адрес, куда мессенджер MAX будет посылать сообщения клиентов. Требования к адресу:

- связь с сервером должна осуществляться по протоколу HTTPS;
- дополнительные требования к порту и настройке HTTPS-соединения описаны в документации;
- Max-adapter, работающий на порту 9102, ожидает события по пути: /restapi/channels/{channelId}/webhook.

4. Запустите программу:

```
systemctl start max-adapter.service
```

4.3.17 Data-stream-service

Конфигурационный файл

Файл ***application.yml***:

```
server:
  port: 9104 # порт для HTTP-сервера (healthcheck, swagger)

spring:
  grpc:
    server:
      port: 9105 # порт для gRPC-сервера

management:
  endpoints:
    web:
      exposure:
        include: health

rabbitmq: # Подключение к rabbitmq для healthcheck
  host: localhost
  port: 5672
  vhost: /
  user: guest
  password: guest
  thread-count: 1
  prefetch: 8

data-stream-service:
  chunk-receive-timeout-ms: 60_000 # предельное время ожидания чанка во время чтения
стрима (ReadStream), когда WriteStream не закончил запись
  stream-lifetime-ms: 600_000 # предельное время хранения стрима
  clean-old-streams-rate-ms: 10_000 # интервал очистки старых стримов
```

Описание параметров в секции `spring.grpc` доступно в документации.

Настройка и запуск

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

1. В файле */data-stream-service/application.yml* укажите нужные хосты и порты.
2. В конфигурации HttpAdapter задайте настройки для подключения к **Data-stream-service**.
3. Запустите программу:

```
systemctl start data-stream-service.service
```

4.3.18 Корректировка тайм-аутов

Для правильной работы **ChatNavigator** 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}}});
```

4.3.19 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.

4.3.20 Общая конфигурация MongoDB

Действия необходимы только в случае установки на несколько компьютеров.

Общая конфигурация MongoDB проводится для следующих компонентов ChatNavigator: BotServer, SipAdapter.

```
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
```

4.3.21 Конфигурирование LDAP

4.3.21.1 Создание шифрованного 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: l, 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.
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 выглядит так: **prefix: {field: cn, value: Sys CN_content_}** **field: cn**, где **cn** – это поле объекта **Group**, **value: Sys CN_content_** – это префикс значения.

4.4 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.

4.5 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-gready 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/*.

Войдите в интерфейс ChatNavigator, заполнив поля Логин и Пароль. Чтобы посмотреть введенный пароль нажмите , чтобы скрыть нажмите . Нажмите Войти.

Для смены языка интерфейса используйте в правом верхнем углу страницы. 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 платформы.

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 ChatNavigator со следующими элементами:

1: главное меню;

2: выход из учетной записи;

3: рабочая вкладка.

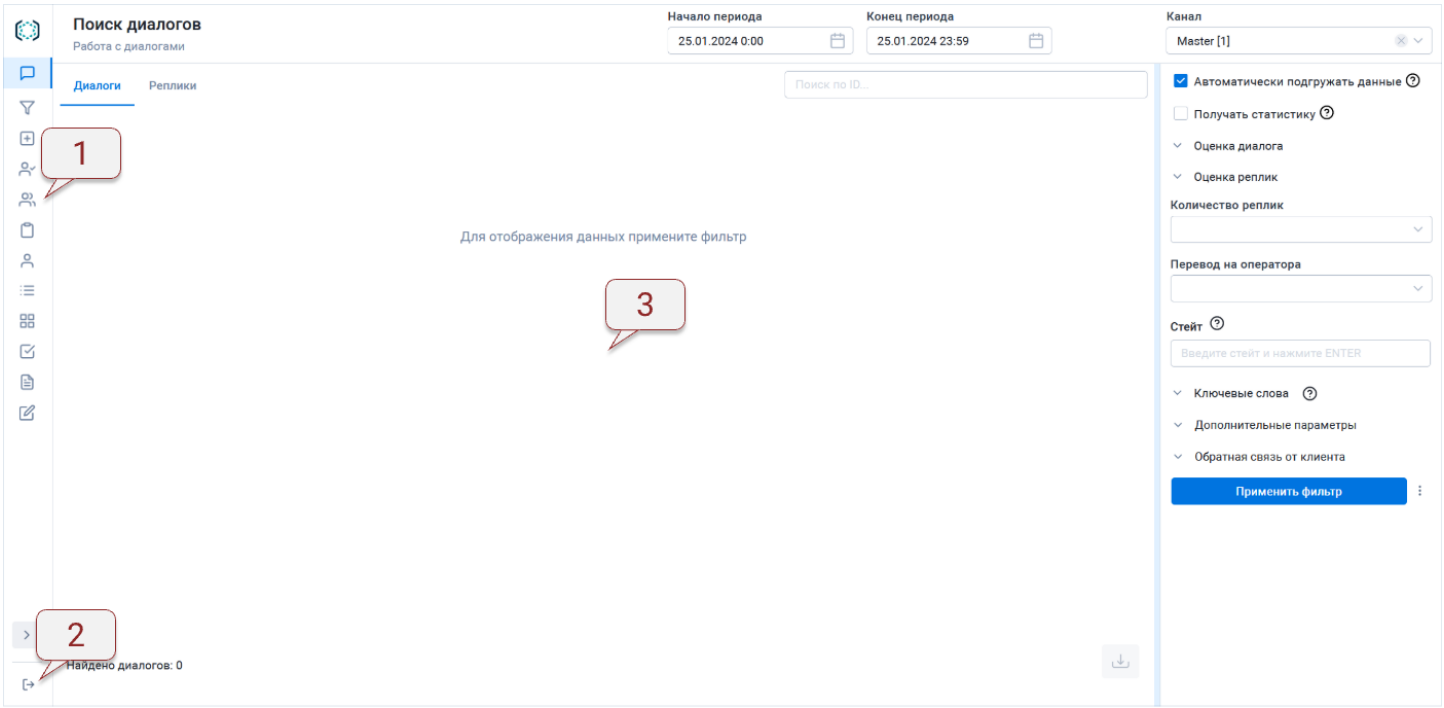


Рисунок 4 — Графический интерфейс ChatNavigator

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

Для завершения работы с интерфейсом нажмите .

Для перехода к рабочей вкладке нажмите соответствующую кнопку в главном меню.

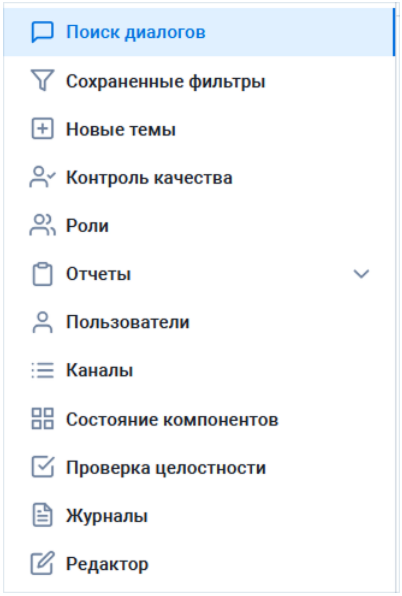


Рисунок 5 — Главное меню

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 платформы и действий пользователей.

Редактор — переход к редактору сценария диалога.

Подробно о вкладках **Поиск диалогов**, **Новые темы**, **Контроль качества**, **Отчеты**, **Редактор** описано в документе «*ChatNavigator Омниканальная диалоговая платформа на естественном языке STC-S9777. Руководство пользователя НЦДА.00767-01 92 01*».

Подробно о вкладках **Проверка целостности**, **Журналы** описано в документе «*ChatNavigator Омниканальная диалоговая платформа на естественном языке STC-S9777. Руководство администратора безопасности НЦДА.00767-01 95 01*».

6 ФОРМИРОВАНИЕ РОЛЕЙ

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

Вкладка **Роли** содержит следующие элементы (рис. 6):

1: перечень созданных ролей;

2: поле поиска привилегий;

6.1 Реализованные 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 ChatNavigator.

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: sessionData, responceData 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

Группа	Привилегия	Комментарий
		«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.
	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.

Группа	Привилегия	Комментарий
	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.
	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

Группа	Привилегия	Комментарий
		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***.
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 «*ChatNavigator Омниканальная диалоговая платформа на естественном языке. Руководство администратора НЦДА.00767-01 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.

Группа	Привилегия	Комментарий
	Редактирование документов Базы знаний	Предоставляет право на редактирование документов Базы знаний. Если пользователю предоставляется эта привилегия, то «Просмотр документов Базы знаний».
	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	+	-	-	+

Привилегия	SuperAdmin	AccountAdmin	Supervisor	ContentManager
Просмотр всех каналов	+	-	-	-
Настройка приоритета каналов	+	-	-	-
Назначение пользователей на каналы	+	-	-	-
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	+	-	-	+
Сохранение фильтров	+	-	+	+
Просмотр всех сохраненных фильтров	+	-	+	+
Управление всеми сохраненными фильтрами	+	-	+	-

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.

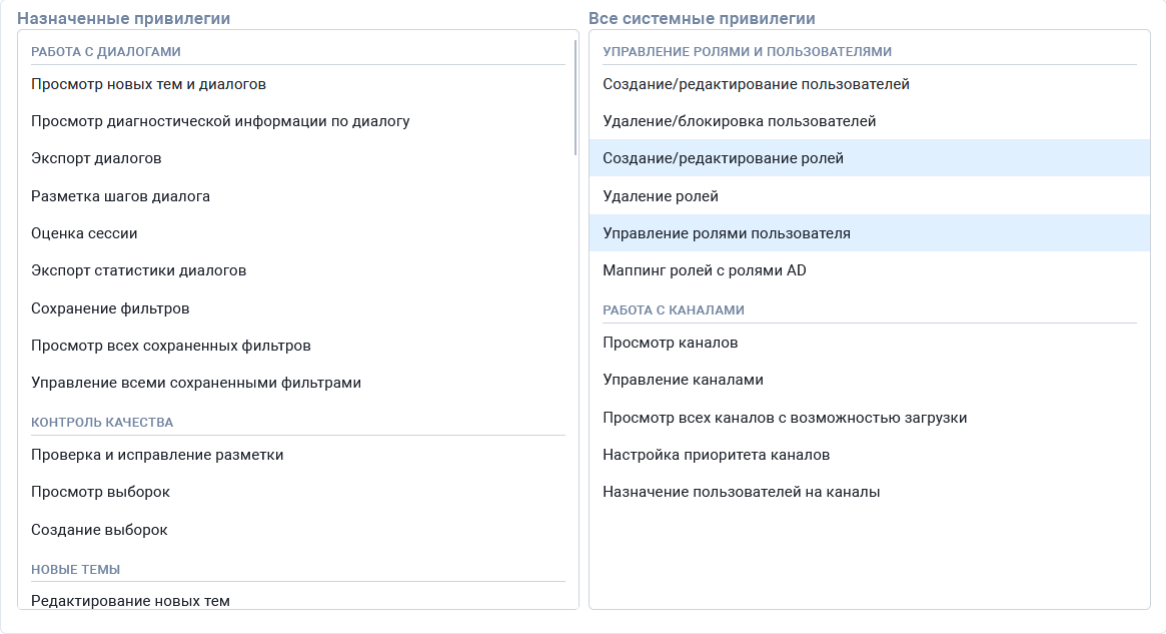


Рисунок 6 — Назначение привилегий

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).

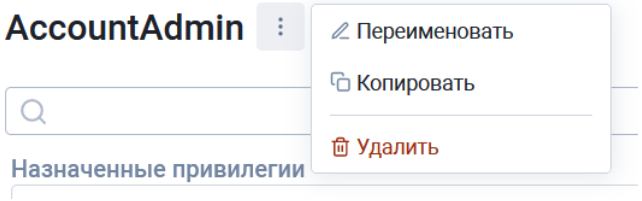


Рисунок 7 — Меню для роли

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.

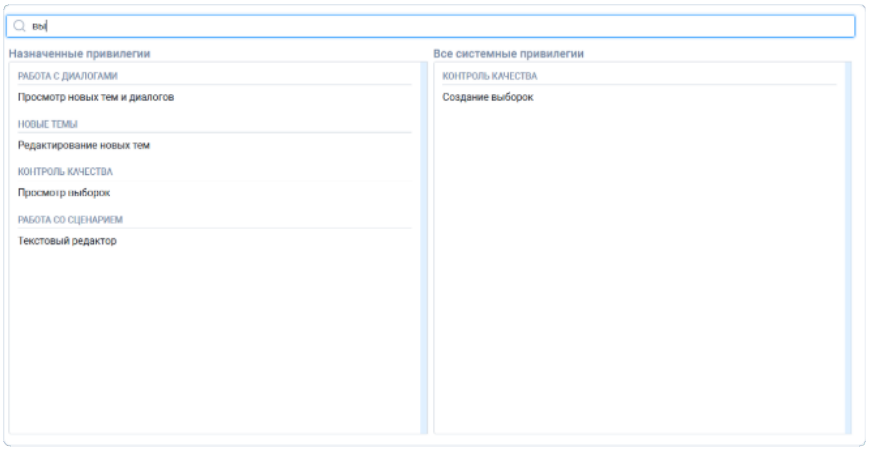


Рисунок 8 — Поиск привилегий

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 «*ChatNavigator Омниканальная диалоговая платформа на естественном языке STC-S9777. Руководство администратора НЦДА.00767-01 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).

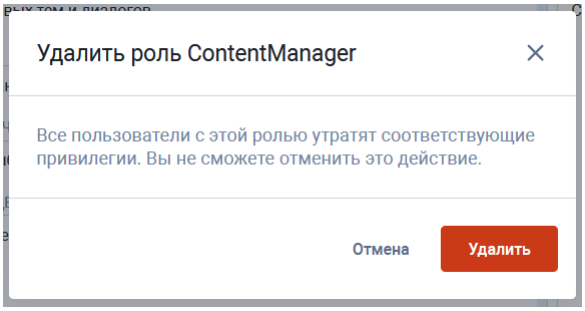


Рисунок 9 — Подтверждение удаления роли

Удаление роли подтвердится сообщением.

6.6 Настройки Active Directory

Чтобы выполнить маппинг системных ролей и групп AD, нажмите Active Directory ⚙.

В открывшемся окне для каждой роли из выпадающего списка выберите соответствующую группу AD и нажмите OK.

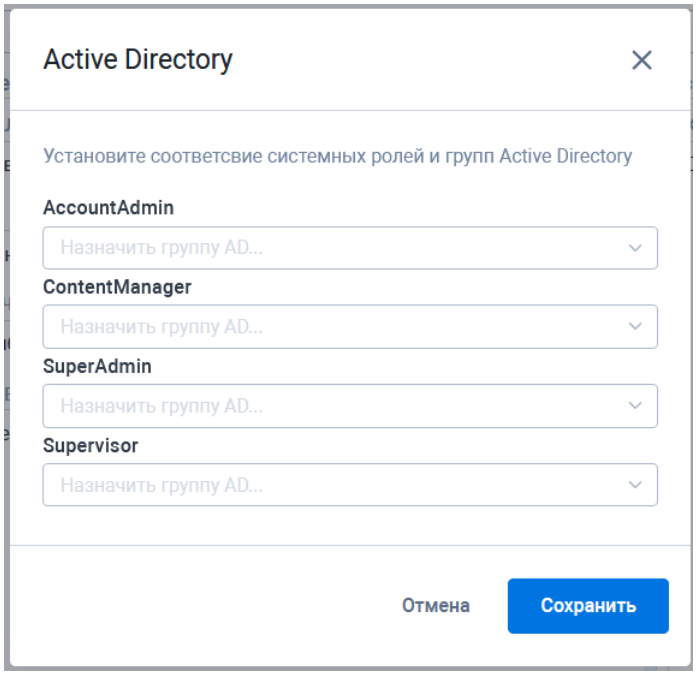


Рисунок 10 – Настройки **Active Directory**

Для импорта ключа в глобальный Keystore выполните команды:

1. Получение сертификата с сервера AD:

```
# echo "" | openssl s_client -connect vkkerbad.vktest.local:636 -showcerts 2>/dev/null | openssl x509 -out certfile.txt
```

2. Проверка имеющихся сертификатов:

```
# keytool -keystore /usr/lib/jvm/java-8-oracle/jre/lib/security/cacerts -storepass changeit -list
```

3. Импорт сертификата в локальный или глобальный `Keystore`:

```
# keytool -import -storepass changeit -alias vkkerbad_vktest_local -keystore /usr/lib/jvm/java-8-oracle/jre/lib/security/cacerts -trustcacerts -file ./certfile.txt
```

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.

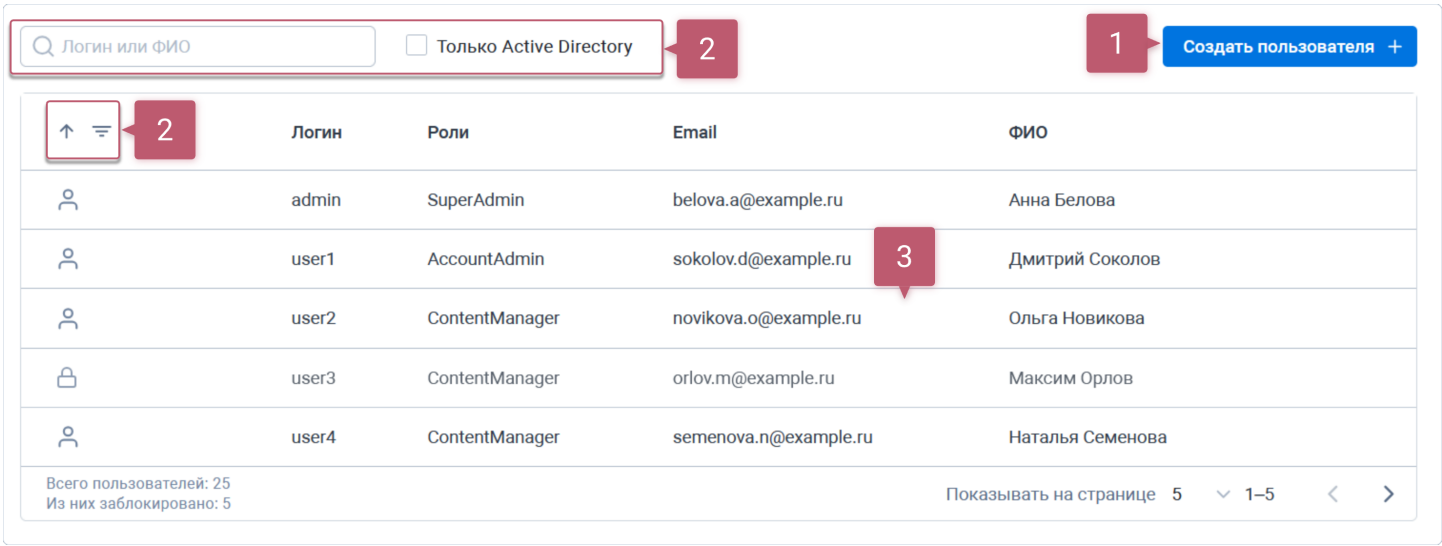


Рисунок 11 — Вкладка Пользователи

7.1 Список пользователей

Список зарегистрированных пользователей отображается в виде таблицы со столбцами:

– / — статус пользователя: активный или заблокированный;

– **Логин** — логин пользователя;

– **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 чекбокс **Только Active Directory**, чтобы отобразились только пользователи, которые для входа 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.

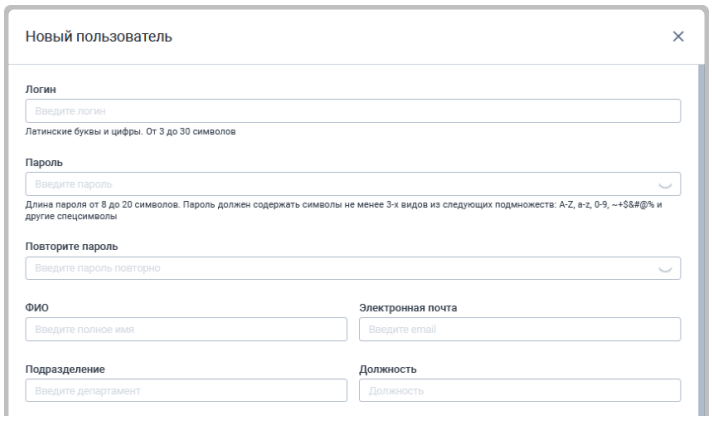


Рисунок 12 — Добавление учетной записи пользователя

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.

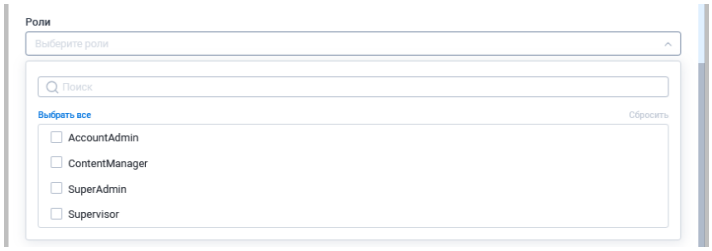


Рисунок 13 — Выбор ролей

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

Укажите логин и пароль учетной записи **Mercurial** и нажмите **Создать** (рис. 14).

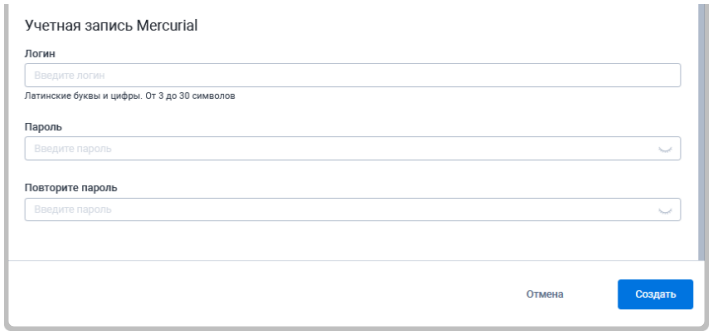


Рисунок 14 — Учетная запись Mercurial

Если в форме ввести некорректные данные и/или не заполнить обязательные поля, то после нажатия кнопки **Создать** поля с некорректными значениями будут подсвечены (рис. 15).

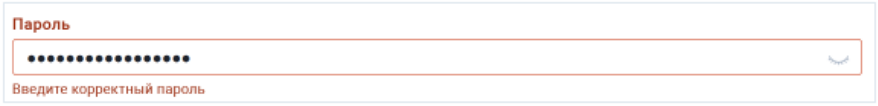


Рисунок 15 — Подсветка полей с некоректными данными учетной записи пользователя

7.3 Редактирование пользователя

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. 16).

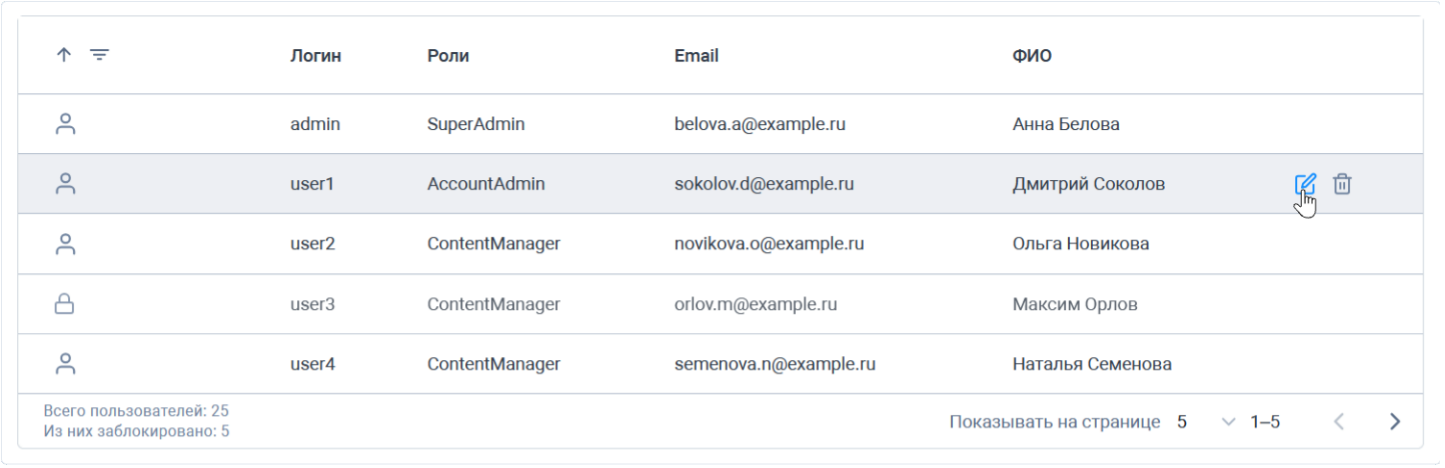


Рисунок 16 — Выбор пользователя для редактирования

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 и разблокировка пользователя

Блокировка

Пользователь может быть заблокирован администратором вручную или автоматически.

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

Причина, по который пользователь был заблокирован, отображается в карточке пользователя.

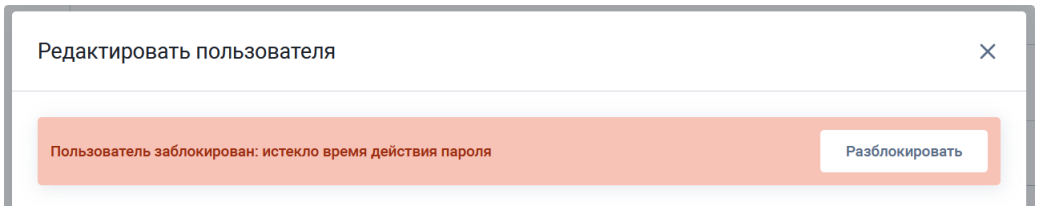


Рисунок 17 — Пример отображения причины блокировки пользователя

Ручная блокировка

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

Автоматическая блокировка

Пользователь может быть заблокирован автоматически системой по некоторым причинам:

– пользователь был неактивен определенный период;

– истек срока жизни пароля;

– пользователь превысил количество попыток входа с некорректным паролем.

Параметры автоматической блокировки настраиваются в конфигурации **ServiceWeb** в секции `security-policies`.

Разблокировка

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

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. 18) кнопкой **Удалить** или отмените удаление кнопкой **Отмена**.

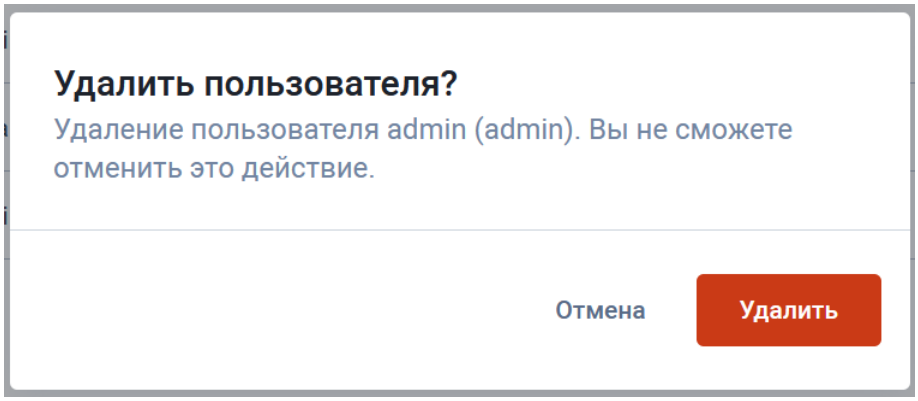


Рисунок 18 — Окно подтверждения удаления пользователя

8 РАБОТА С КАНАЛАМИ

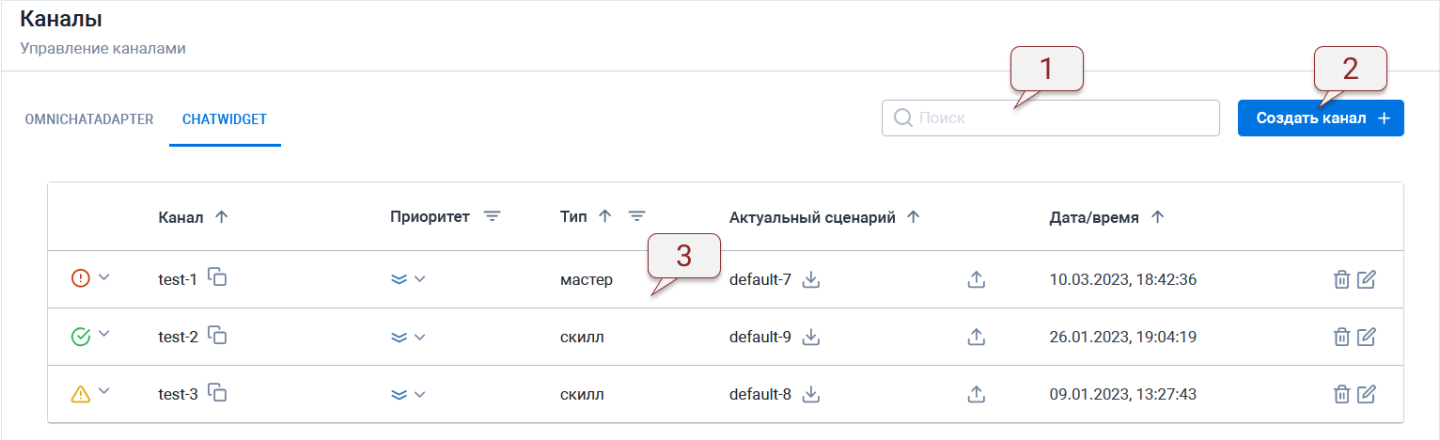
Для добавления, редактирования, а также просмотра списка имеющихся каналов виджета воспользуйтесь вкладкой **Каналы** в интерфейсе **AgentNavigator**.

Вкладка **Каналы** содержит следующие элементы:

1 — поиск канала;

2 — кнопка создания канала;

3 — список каналов.



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

Нажмите **Создать канал** в верхней части области управления каналами.

> С помощью интерфейса **ChatNavigator** можно создать только каналы типа **CHATWIDGET**. Каналы других типов задаются в конфигурации соответствующих адаптеров.

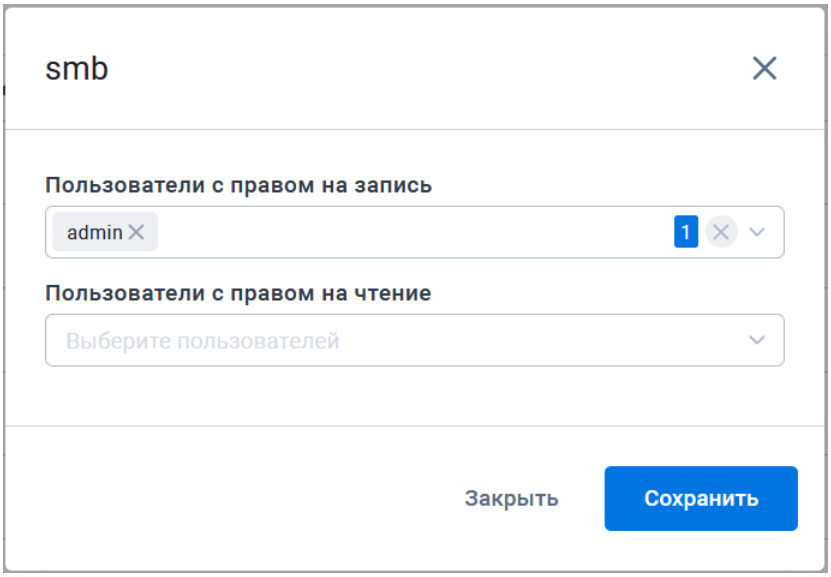
В открывшемся окне введите имя канала, назначьте на канал пользователей с правом на запись и с правом на чтение сценария и нажмите **Создать**.

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

8.2 Редактирование канала

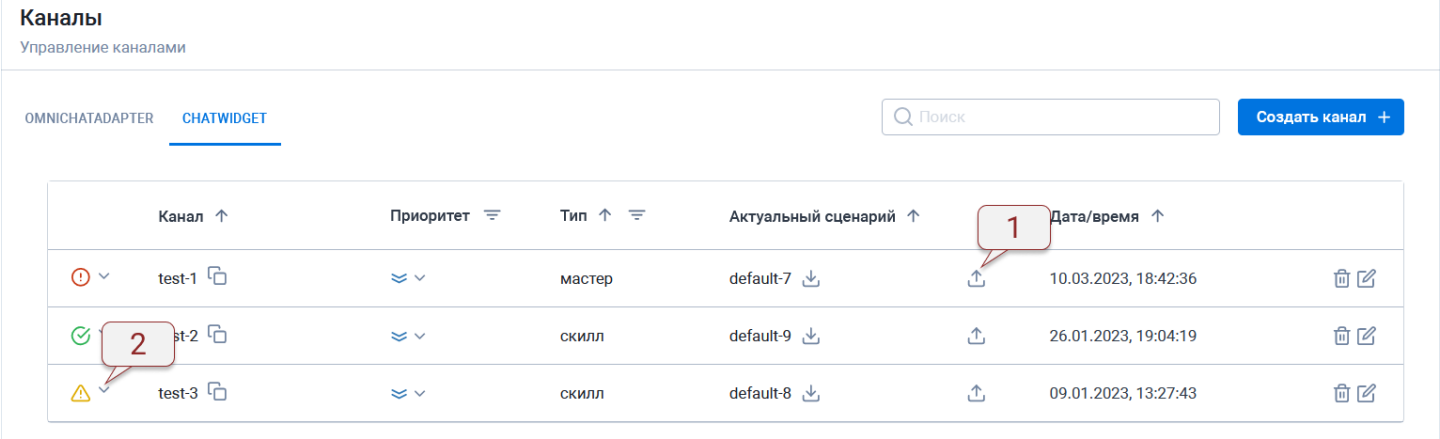
Для редактирования канала пользователю должна быть назначена роль, у которой есть соответствующая привилегия на управление каналами.

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



8.3 Загрузка сценария

Нажмите рядом с названием нужного канала в списке.



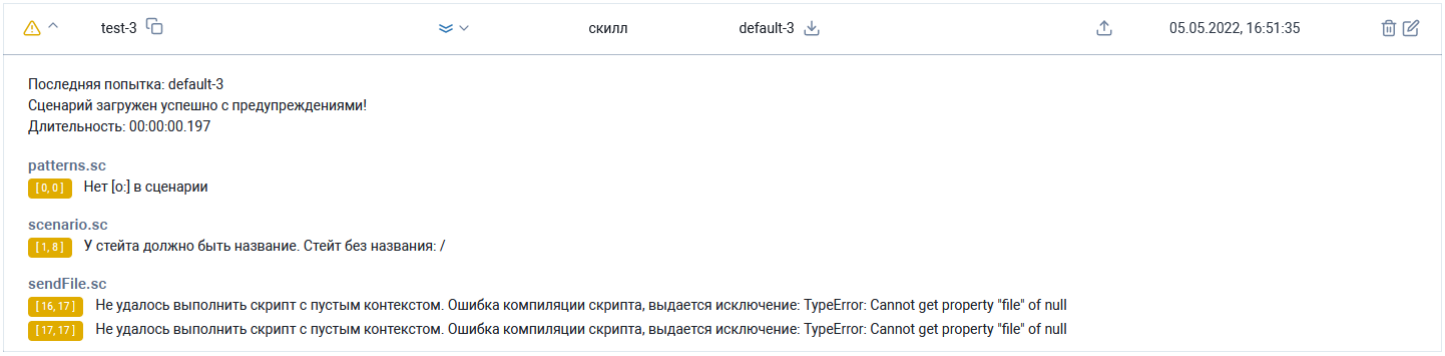
С помощью окна выбора файла найдите и откройте необходимый архив со сценарием. Во время загрузки сценария отобразится информационное сообщение. После завершения загрузки рядом с названием канала появятся название загруженного сценария, время загрузки и статус загрузки. Возможные статусы:

— сценарий загружен успешно;

— сценарий загружен успешно, но есть предупреждения;

— при загрузке сценария произошла ошибка, сценарий не был загружен.

При нажатии на отобразится подробная информация 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. 19), при помощи которого можно проверить, как работают добавленные шаблоны.

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.

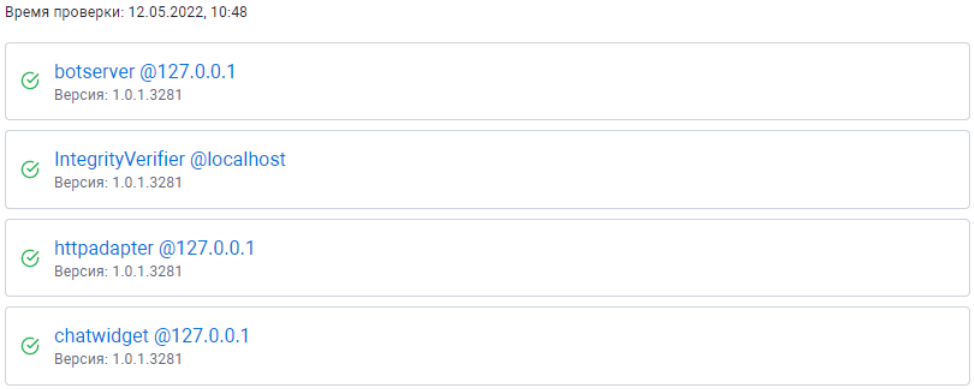


Рисунок 19 — Вкладка **Состояние компонентов**

9.1 Список компонентов

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

– **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 всех компонентов, который отображает дату и время, в которые платформой 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 **ChatNavigator** 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 **ChatNavigator**	`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`
		Отображает установленные пакеты

Проверка статуса служб компонентов	`systemctl status botserver chatwidget classifier classifiertrainer fileserver httpadapter integrityverifier omnichatadapter serviceweb sipadapter`	Отображает текущий статус служб компонентов ChatNavigator
Проверка статуса автозагрузки компонентов при старте сервера	`systemctl list-unit-files | egrep -i 'botserver|chatwidget|classifier|classifiertrainer|fileserver|httpadapter|integrityverifier|omnichatadapter|serviceweb|sipadapter'`	Отображает статус автоматической загрузки: enabled / disabled
Проверка работоспособности установленных компонентов **ChatNavigator**	`systemctl list-units | egrep -i 'botserver|chatwidget|classifier|classifiertrainer|fileserver|httpadapter|integrityverifier|omnichatadapter|serviceweb|sipadapter'`	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	*/var/log/speechpro/botserver/*	*audit.log* *botserver.access.log* *boserver.main.log* *botserver.stat.json.log*
ChatWidget	*/var/log/speechpro/chatwidget/*	*audit.log* *chatwidget.access.log* *chatwidget.main.log*
HttpAdapter	*/var/log/speechpro/httpadapter/*	*audit.log* *httpadapter.access.log* *httpadapter.main.log*
OmnichatAdapter	*/var/log/speechpro/omnichatadapter/*	*audit.log* *omnichatadapter.access.log* *omnichatadapter.main.log*
SipAdapter	*/var/log/speechpro/sipadapter/*	*audit.log* *sipadapter.access.log* *sipadapter.main.log*
IntegrityVerifier	*/var/log/speechpro/integrityverifier/*	*audit.log* *integrityverifier.main.log*
ServiceWeb	*/var/log/speechpro/serviceweb/*	*audit.log* *serviceweb.access.log* *serviceweb.main.log* *serviceweb.pump.log*
FileServer	*/var/log/speechpro/fileserver/*	*audit.log* *fileserver.access.log* *fileserver.main.log*
NER	*/var/log/speechpro/ner/*	*ner.access.log* *ner.main.log*
NER Trainer	*/var/log/speechpro/ner_trainer/*	*ner_trainer.access.log* *ner_trainer.main.log*
Classifier	*/var/log/speechpro/classifier/*	*classifier.access.log* *classifier.main.log*
Classifier Trainer	*/var/log/speechpro/classifier_trainer/*	*classifier_trainer.access.log* *classifier_trainer.main.log*

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 системой): изменение конфигурации, другие действия над системой (разметка диалога и т. д.);

– безопасности (события в подсистеме АА): вход/выход из учетной записи, изменение пользователя (смена пароля, других атрибутов), попытки войти в систему.

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 систему
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}
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

Код	Описание
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}
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	`http://<hostname>:9066/v2/restapi/channels/omnichat-adapter/test`
Заголовки запроса	`Content-Type: application/json` `Charset: UTF-8`

Тело запроса	`{`
	`"request_id":"${__UUID}",`
	`"appeal_id":"${__UUID}",`
	`"request_time":"2018-09-27T18:18:53.076+0300",`
	`"message_number":"1",`
	`"sender_id":"256",`

	<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>

4. Во вкладке **Поиск диалогов** интерфейса администратора выберите канал **omnichat-adapter** (справа сверху) и нажмите **Применить фильтр**.

5. В рабочей области вкладки **Поиск диалогов** должен отобразиться диалог, созданный через OmnichatAdapter.

11.4 Проверка работы SipAdapter

Для проверки выполните следующие действия:

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

Endpoint	<pre>http://<hostname>:9098/restapi/channels/sip-adapter-incoming/test</pre>

Заголовки запроса	<pre>Content-Type: application/json
	Charset: UTF-8</pre>

Тело запроса	<pre>{
	"header": {
	"messageId": "1",
	"sessionId": "1",
	"messageNumber": 1,
	"eventNumber": 1
	},
	"event": {
	"_type": "recognition",
	"mainHypothesis": {
	"text": "привет",
	"confidence": 1.0
	}</pre>

```
    },
    "messageData": {
    }
}
```

```
"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"
    }]
}
```

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 **ChatNavigator**.
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	`http://<hostname>:9066/v2/restapi/channels/<название созданного канала>/test`
Заголовки запроса	`Content-Type: application/json` `Charset: UTF-8`

Тело запроса	`{`
	`"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"`
	`}]`
	`}`

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}.main.%d.%i.log.gz">
```

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

...
```

2. Измените значение параметра **mainMaxFileSize** с **500 MB** на **1 KB**.
3. Выполните **POST**-запрос на */restapi/test/nlp/procces*. В заголовке запроса необходимо передать **accountid: default**, в качестве **queryParam** укажите канал (**channelId:номер канала**).
4. Дождитесь, пока файл */var/log/speechpro/botserver/botserver.main.log* достигнет размера **1 Кбайт**.
5. Когда файл достигает размера **1 Кбайт**, он должен быть зархивирован 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:

1. 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
```

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

```
systemctl restart botserver
```

4. Откройте на просмотр лог-файлы компонента **BotServer** (файл */var/log/speechpro/botserver/botserver.main.log*).
5. Создайте диалог любым из способов.
6. 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 зависимых от него компонентов и ресурсов.

Список компонентов, для которых доступен мониторинг:

– ChatWidget;

– MrsipAdapter;

– Telegram;

– 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	
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": "fileServer",
      "isOk": true,
      "instanceData": null,
      "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": []
    },
  ]
}
```

Коды ответов

200	Ok	Успешный запрос

13 ГЛОССАРИЙ

Термины и определения

Адаптер	Модуль на стороне ChatNavigator, обеспечивающий взаимодействие с чат-платформой или напрямую с внешними каналами коммуникации. Адаптер специфицирует структуру и логику обработки запросов, проходящих между ChatNavigator 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 системы	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/l16.
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 ChatNavigator — 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 ChatNavigator 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.
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 агент осуществляет взаимодействие с клиентом, определяет тему запроса, выполняет действия 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 ChatNavigator. 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 ChatNavigator 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-модуей.
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 ChatNavigator и передачи текста ответа и параметров для синтеза, распознавания и результатов обработки перебиваний.
БД	База данных.
СУБД	Система управления базами данных.

14 ИСТОРИЯ ОБНОВЛЕНИЙ

3-170726

В системных требованиях обновлены версии операционных систем и версии браузеров.

В конфигурации **HttpAdapter**:

- в секции spring.grpc.client.channels.data-stream-service удалены address и secure, добавлены target и bypass-certificate-validation;
- в секцию llm.giga-chat.channel-specific.auth добавлена секция prod;
- в секции spring.cloud.openfeign.httpclient.hc5:
  - изменено значение connection-request-timeout на 180000;
  - изменено значение connection-request-timeout-unit на MILLISECONDS.

В конфигурации **BotServer** изменено значение botserver.script.script-timeout-ms на 300000.

В конфигурации **ConfigService** в файле application-llm-services.yml в секции spring.cloud.openfeign.client.config:

- добавлены секции: llm-core, llm-document-storage, llm-document-parser;
- изменено:
  - giga-chat-completions-prod.read-timeout изменено на 1_200_000;
  - qwen-completions.read-timeout изменено на 1_200_000.

В конфигурации **OmnichatAdapter** изменено значение omnichatadapter.response.question.sync.timeout-sec на 3000.

2-020726

Дополнено примечание по расположение конфигурационных файлов при установке с помощью **Docker**.

Обновлена Установка ChatNavigator с помощью Docker.

Добавлена информация, необходимая для импорта ключа в глобальный Keystore.

1-110626

В системных требованиях обновлены версии браузеров и версии стороннего программного обеспечения.

Обновлен список используемых портов.

Добавлено описание автоматической блокировки пользователей.

Добавлены разделы: Установка ChatNavigator с помощью Ansible Playbook и Установка ChatNavigator с помощью Docker.

Изменения конфигурации

Добавлена конфигурация **Langchain-service**.

Добавлена конфигурация **Kafka-adapter**.

Добавлена конфигурация Max-adapter.

Добавлена конфигурация Data-stream-service.

Добавлена конфигурация Trigger-service.

В конфигурации HttpAdapter:

– model-name у Qwen изменена на Qwen 3 32B;
– добавлены настройки для Deepseek;
– добавлена секция s3;
– добавлена секция healthcheck;
– для элементов секции llm.giga-chat.channel-specific появилась возможность указать секцию certificate;
– добавлена секция mq-events;
– в секцию llm.giga-chat.completions добавлена секция connection-properties;
– в секцию spring.cloud.openfeign.client.config добавлена секция python-http-adapter и секция database-http-adapter;
– в секции llm.stream добавлена секция data-stream-service;
– в секции spring.grpc.client.channels добавлена секция data-stream-service.

В конфигурации ConfigService:

– model-name у Qwen изменена на Qwen 3 32B;
– добавлены настройки для Deepseek;
– изменен файл application-llm-services.yml;
– добавлен файл application-langchain-llm-models.yml;
– добавлен файл application-langchain-tools.yml;

В конфигурации ServiceWeb:

– model-name у Qwen изменена на Qwen 3 32B;
– добавлены настройки для Deepseek;
– в секции rag.llm добавлена секция parser-options;
– в секции server.servlet.session добавлена секция cookie;
– в секции server добавлена настройка forward-headers-strategy;
– добавлена секция security-policies;
– добавлена секция s3;
– в секции ldap.connection добавлена настройка secret;
– изменена секция pump;
– в секции reports добавлена секция clean;

– добавлены секции rabbitmq, mq-events, mq-events;
– в секции spring добавлена секция config;
– в секции rag.llm.directory-import добавлена секция downloader;
– добавлен registry-links-mapping.yml.

В конфигурации BotServer:
– в секции botserver:
  – добавлена секция loop;
  – добавлена секция json-processing;
– добавлена секция debug;
– добавлена секция s3;
– добавлена секция mq-events;
– вместо секции stat-log-rabbitmq добавлена секция mq-events.

В конфигурации OmnichatAdapter:
– в секцию omnichatadapter:
  – добавлена секция session-memory-size;
  – добавлена настройка unsupported-bot-event-message;
– в секции healthcheck:
  – удалена настройка mongo-timeout-ms;
  – добавлена настройка postgres-timeout-ms;
– добавлена секция mq-events;
– добавлен пример настройки канала, по которому не требуются ответы.

В конфигурации SipAdapter добавлена секция mq-events.

В конфигурации CrafttalkAdapter добавлена секция mq-events.

В конфигурации ChatWidget:
– добавлена секция healthcheck;
– добавлена секция mq-events.

В конфигурации Telegram:
– добавлена секция healthcheck;
– добавлена секция mq-events.

В конфигурации FileServer:
– в секции file-server.http добавлены настройки max-connections, max-connections-per-route;
– добавлена секция mq-events.

В конфигурации python-http-adapter в секции script добавлена настройка args-max-length.