

ООО «КСК ТЕХНОЛОГИИ»

наименование организации — разработчика

КСК.Шлюз СМЭВ 3

наименование вида АИС

КСК.Шлюз СМЭВ 3

сокращенное наименование АИС

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

На 21 листах

г. Москва, 2020

АННОТАЦИЯ

Настоящий документ представляет собой руководство администратора Системы КСК.Шлюз СМЭВ 3 (далее – Шлюз, Система).и содержит описание функций и инструкций по работе в системе.

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

Сокращение, обозначение	Значение
API	Интерфейс программирования приложений (Application Programming Interface) Набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах. Используется программистами при написании всевозможных приложений
HTTP	Протокол передачи гипертекста (HyperText Transfer Protocol) Протокол прикладного уровня передачи данных
XML	Расширяемый язык разметки (eXtensible Markup Language)
БД	База данных
ВС	Вид сведений
ИС	Информационная система
ОС	Операционная система
ПО	Программное обеспечение
СМЭВ	Система межведомственного электронного взаимодействия
СУБД	Система управления базами данных
ЭП	Электронная подпись

ОГЛАВЛЕНИЕ

Аннотация	2
Перечень терминов и сокращений.....	3
Оглавление	4
1 Введение.....	5
1.1 Область применения.....	5
1.2 Краткое описание возможностей.....	5
1.3 Требования к уровню подготовки пользователей	7
2 Назначение и условия применения	8
2.1 Назначение системы	8
2.2 Условия применения Системы.....	10
2.2.1 Состав инсталляционного комплекта	10
2.2.2 Системные требования.....	10
2.2.3 Синхронизация серверов.....	10
2.2.4 Настройки безопасности	11
3 Установка системы.....	12
3.1 Установка PostgreSQL.....	12
3.2 Установка Oracle JDK.....	14
3.3 Установка криптопровайдера.....	14
3.4 Установка брокера очередей ActiveMQ	14
3.5 Установка КСК. Шлюз СМЭВ 3.....	14
4 Эксплуатация.....	16
4.1 Запуск и останов	16
4.2 Проверка запуска системных компонентов.....	16
4.3 Поддержка работоспособности компонента Karaf	16
4.3.1 Лог - файлы.....	16
4.3.2 REST API Шлюза.....	17
4.3.3 Решение проблем с запуском процесса Karaf.....	17
4.3.4 Проверка запуска модулей ядра Шлюза и модулей адаптеров ВС.....	18
4.4 Установка адаптеров ВС.....	19
4.4.1 Установка дополнительных модулей и библиотек	20

1 ВВЕДЕНИЕ

1.1 Область применения

Информационная система «КСК.Шлюз СМЭВ 3» (далее – Система) является транзитной системой, которая работает по протоколу информационного взаимодействия с использованием среды СМЭВ 3 согласно Методическим рекомендациям Минкомсвязи. Система предназначена для:

- получения сведений от Участников СМЭВ с использованием СМЭВ версии 3.X;
- отправки сведений Участникам СМЭВ с использованием СМЭВ версии 3.X;
- мониторинга исходящих и входящих запросов.

1.2 Краткое описание возможностей

Шлюз состоит из следующих основных частей:

- **модулей ядра**, которые обеспечивают работу внутренних механизмов Шлюза:
 - протоколы взаимодействия с клиентом и со СМЭВ 3 и их согласование посредством расширяемого набора сценариев обработки, а также механизмы для промежуточного преобразования передаваемых при этом данных;
 - формирование и верификацию ЭП в соответствии с требованиями Методических рекомендаций;
 - сценарии обработки и маршрутизации передаваемых через Шлюз данных;
 - внутренний API для размещаемых на Шлюзе модулей-адаптеров видов сведений;
 - API для администрирования и мониторинга работоспособности системы;
 - взаимодействие с внутренней СУБД и т.п.;
- **набора модулей-адаптеров видов сведений СМЭВ 3**. Адаптеры являются обработчиками данных по отдельным видам сведений, выполняющими преобразование (трансформацию) содержательной части сообщений из внутреннего формата, который используется для общения клиента со Шлюзом, в формат видов сведений СМЭВ 3 и обратно (из формата видов сведений СМЭВ 3 в

- формат, «понятный» клиенту), а также для реализации специфичной для видов сведений реакции на события в системе (поступление запроса или ответа и т.п.);
- **сервера приложений («контейнера»)**, в котором размещаются и работают модули ядра и модули-адаптеры. Состоит из набора специфичных серверных компонентов и расширяемого набора системных библиотек. В текущей версии используется Apache Karaf 4.1.6;
 - **внутренней СУБД**, в которой сохраняются оперативные данные, требующие персистентности: цепочки запросов-ответов как со стороны клиента, так и со стороны СМЭВ 3; метаданные, возникающие в процессе обработки сообщений; данные для мониторинга работоспособности и т.п. СУБД может устанавливаться как прямо на сервере Шлюза (при наличии системных ресурсов), так и на отдельном сервере. В текущей версии используется PostgreSQL 10;
 - **криптопровайдера**, предоставляющего системные механизмы для работы ЭП. В текущей версии используется сертифицированный криптопровайдер CryptoPro JCP2.0, предоставляющий алгоритмы формирования и верификации электронной подписи, а также механизмы взаимодействия с ключевыми носителями. Для работы криптопровайдера должно быть организовано хранилище ключей и сертификатов ЭП (ЭП ОВ и, опционально, ЭП СП). В качестве хранилища может использоваться специальным образом сконфигурированная папка на диске (HDImageStore) или физические ключевые носители-токены (Рутокен или eToken);
 - **JMS-брокера (опционально)**. Предназначен для организации системы очередей для взаимодействия с системами-клиентами, для которых предпочтительной средой передачи данных является JMS, и буферизации передаваемых при этом данных. Наличие промежуточного буфера позволяет максимально «развязать» Шлюз и клиентскую систему:
 - кратковременный отказ одного из этих компонентов окажет минимальное негативное воздействие на поток передаваемых данных: работоспособный компонент может продолжать черпать данные из буфера и записывать в буфер;
 - наличие буфера позволяет сглаживать пиковые нагрузки: передаваемые сообщения накапливаются в буфере со стороны источника, и извлекаются в пункте назначения по мере освобождения его ресурсов.
 - ActiveMQ может быть установлен как отдельный компонент (на сервер Шлюза или на отдельный сервер), так и может быть использован

ActiveMQ, встроенный в Servicemix — все эти варианты имеют свои преимущества и недостатки. В текущей версии используется Apache ActiveMQ 5.15.7.

Все части Шлюза могут размещаться на одной физической или виртуальной машине с предварительно установленной ОС Linux Ubuntu Server 16.04 amd64 и Java JDK 1.8.181.

1.3 Требования к уровню подготовки пользователей

Установка и настройка Системы осуществляется администратором, который должен обладать знаниями по:

- архитектуре используемых серверов;
- установке, диагностике и администрированию ОС Linux;
- установке и администрированию: Oracle JDK, Apache Karaf, Apache ActiveMQ, CryptoPro JCP 2.0;
- диагностике и администрированию СУБД PostgreSQL и утилиты их репликации;
- проведению регламентных работ, таких как запуск/остановка, перевод в другие режимы работы, резервное копирование данных и их восстановление из резервной копии, настройка прав доступа.

2 НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

2.1 Назначение системы

КСК.Шлюз СМЭВ 3 является транзитной подсистемой, которая выполняет следующие функции:

- реализация протокола информационного взаимодействия с использованием среды СМЭВ 3 согласно Методическим рекомендациям Минкомсвязи (<https://smev3.gosuslugi.ru/portal/>);
- маршрутизация запросов, ответов и технологических сообщений между корпоративными (ведомственными) ИС и СМЭВ 3;
- предоставление упрощенной спецификации обмена для ВИС для обеспечения взаимодействия со СМЭВ 3.х:
 - более простой формат запросов и ответов, по сравнению с форматами видов сведений;
 - более простой протокол обмена, минимизирующий количество технологических взаимодействий;
 - унифицированный формат сообщений, скрывающий разнообразие форматов видов сведений, существующее на уровне СМЭВ;
- централизованное использование электронной подписи сообщений СМЭВ;
- гарантированная обработка всех поступающих сообщений как со стороны клиента, так и со стороны СМЭВ 3;
- доставка сообщений в СМЭВ 3 и в систему-клиент с максимальной гарантией (применяются протоколы взаимодействия с подтверждением доставки со стороны получателя, а также используются механизмы автоматического повтора неподтвержденных сообщений).

КСК.Шлюз СМЭВ 3 поддерживает работу с системами-клиентами по протоколам:

- JMS (если установлен JMS-брокер ActiveMQ). Взаимодействие клиента со Шлюзом осуществляется через пару очередей на брокере: одна – для сообщений от клиента в Шлюз, вторая – обратно;
- HTTP. Клиент отправляет сообщение как HTTP POST-запрос на определенный URI Шлюза, Шлюз отвечает HTTP-ответом с HTTP-статусом 200. Для обратного взаимодействия (отправка сообщений из Шлюза в клиент) на стороне клиента должен быть развернут Web-сервер, на который Шлюз посылает сообщение как HTTP POST-запрос, а клиент отвечает HTTP-ответом с HTTP-статусом 200.

КСК.Шлюз СМЭВ 3 предоставляет возможность одновременной работы с несколькими системами-клиентами:

- при формировании запроса в Шлюз система-клиент указывает свой «обратный адрес» (наименование JMS-очереди для ответов или URL для ответов по HTTP);
- при маршрутизации входящих запросов из СМЭВ 3, адрес системы-клиента, которая должна обрабатывать запрос, задается (или вычисляется динамически) параметрами или кодом модуля-адаптера соответствующего ВС.

Существует также прикладной REST API, предназначенный для мониторинга работы Шлюза: на определенные URL-адреса Шлюза отправляются HTTP GET запросы с некоторыми параметрами, в ответ Шлюз возвращает запрошенные данные в формате JSON или XML. Список доступных URL и параметров к ним опубликован на странице http://адрес_Шлюза:8890/rest/data/api

С точки зрения бизнес-логики взаимодействия систем, в Шлюзе поддерживаются два базовых внутренних сценария работы с клиентом, соответствующие сценариям СМЭВ 3:

- работа клиента в режиме Потребителя сведений (клиент является активной стороной взаимодействия, т.е. посылает запрос в СМЭВ 3 и получает на него ответ)
- работа клиента в режиме Поставщика сведений (клиент является пассивной стороной взаимодействия, т.е. отвечает на запросы из СМЭВ 3).

По отдельным ВС возможна «инверсная» бизнес-логика, когда «Потребитель» (инициатор взаимодействия) в своем сообщении фактически посылает не параметры запроса, а сутевые бизнес-данные, которые на другой стороне СМЭВ 3 принимает и обрабатывает «Поставщик» (ответчик). В свою очередь, «Поставщик» отправляет в ответ «Потребителю» через СМЭВ 3 не бизнес-данные, а лишь уведомление-подтверждение успешного приема бизнес-данных от «Потребителя» (либо не отправляет вообще ничего, если «Потребитель» отправлял широковещательное сообщение по виду сведений, на которые подписан «Поставщик»). Данная схема не влияет на реализацию сценария взаимодействия клиента со Шлюзом с точки зрения API: инициатор запроса по-прежнему работает как Потребитель, а клиент, который уведомляет о получении данных — как Поставщик.

При работе в качестве Поставщика по отдельным ВС возможна работа без участия системы-клиента (под-режим «Внутренней обработки»): данные для ответа могут формироваться программно в коде соответствующего адаптера ВС (например, извлекаться из сторонней СУБД).

Помимо вышеуказанных базовых сценариев, в Шлюзе существует также множество под-сценариев, связанных с задействованием внутренних механизмов Шлюза — например, связанных с особенностями взаимодействия клиента по HTTP либо JMS, особенностями применения ЭП и т.п.

2.2 Условия применения Системы

2.2.1 Состав инсталляционного комплекта

Инсталляционный комплект поставки имеет следующую структуру:

- PostgreSQL 10;
- Oracle JDK 1.8.181 или выше;
- CryptoPro JCP 2.0 или выше;
- Apache ActiveMQ 5.15.7 или выше;
- Apache Karaf 4.1.6 или выше, модули КСК. Шлюз СМЭВ 3.

Установка системы производится на операционную систему — Linux Ubuntu Server 16.04 amd64 или выше или аналог (Astra Linux, Centos 7 или выше).

2.2.2 Системные требования

Рекомендованная конфигурация для продуктивной среды (при размещении всех компонентов на одном сервере):

- процессор CPU 4 ядра IntelXeon x86_64 с частотой не ниже 2,4 Ghz;
- оперативная память — 2 ГБ и больше;
- RAM 16Gb;
- HDD 320 Gb RAID-0/5/10.

Допускается установка на виртуальную машину при соблюдении требований к производительности.

2.2.3 Синхронизация серверов

Дата и время на сервере Шлюза и используемых совместно с ним серверах клиентской системы должны быть синхронизированы. Допустимая погрешность — не более 30 секунд (это необходимо для правильного отображения информации мониторинга и для обеспечения передачи данных через JMS-брокер).

2.2.4 Настройки безопасности

На время установки системы необходимо обеспечить доступность внешних репозиториев для скачивания инсталляционных deb-пакетов, т.е. выход в Internet. По окончании установки выход в открытый Internet на продуктивной среде может быть заблокирован из соображений безопасности (нужно оставить двустороннюю доступность между Шлюзом и средой СМЭВ 3 через VipNet, между Шлюзом и клиентской системой, доступность СУБД со стороны Шлюза, а также доступность Шлюза для администрирования).

3 УСТАНОВКА СИСТЕМЫ

Установка системы производится на операционную систему Linux Ubuntu Server 16.04 amd64. При использовании других debian-based операционных систем некоторые команды могут незначительно отличаться. Необходимо свериться с документацией по используемой операционной системе.

3.1 Установка PostgreSQL

Установить СУБД PostgreSQL версии 10 из репозитория PostgreSQL:

1. Добавить репозиторий PostgreSQL в список репозитория:

```
sudo echo 'deb http://apt.postgresql.org/pub/repos/apt/
xenial-pgdg main' >> /etc/apt/sources.list.d/pgdg.list
```

2. Добавить публичный ключ репозитория PostgreSQL:

```
wget --quiet -O -
https://www.postgresql.org/media/keys/ACCC4CF8.asc |
sudo apt-key add -
```

3. Выполнить обновление локального индекса пакетов репозитория PostgreSQL:

```
sudo apt-get update
```

4. Установить PostgreSQL:

```
apt-get install postgresql-10
```

5. Настроить доступ к БД

Внимание

В данном примере настройка идет для всех ПК сегмента сети. На продуктиве необходимо указывать только конкретные доменные имена и IP.

- 5.1. Редактировать файл

```
sudo nano/etc/postgresql/10/main/postgresql.conf.
```

Добавить строчку:

```
listen_addresses = '*'
```

- 5.2. Редактировать файл

```
sudo nano/etc/postgresql/10/main/pg_hba.conf.
```

Добавить строчку в разделе # IPv4 localconnections:

```
# IPv4 local connections:
Host all      all      127.0.0.1/32      md5
```

```
host all all 0.0.0.0/0 trust
```

5.3. Перезапустить PostgreSQL

```
sudo service postgresql restart
```

Создать БД:

1. Установить локаль:

```
sudo locale-gen ru_RU.UTF-8
```

2. Перезапустить Postgres:

```
sudo service postgresql restart
```

3. Выполнить скрипты по созданию роли, БД, схемы:

3.1. Создание роли:

```
sudo -u postgres psql -c "CREATE ROLE usb LOGIN
ENCRYPTED PASSWORD 'abcd1234' NOSUPERUSER INHERIT
CREATEDB CREATEROLE;"
```

3.2. Создание БД:

```
sudo -u postgres psql -c "CREATE DATABASE USB WITH OWNER
= usb ENCODING = 'UTF8' TABLESPACE = pg_default
LC_COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8'
CONNECTION LIMIT = -1 TEMPLATE = template0;"
```

3.3. Создание схемы:

```
sudo -u postgres psql usb -c "CREATE SCHEMA USB;ALTER
SCHEMA USB OWNER TO usb;"
```

3.4. Выполнить:

```
sudo -u postgres psql -c "ALTER USER postgres PASSWORD
'postgres';"
```

Если СУБД развернута не на localhost, необходимо выполнить ручную скрипты по созданию таблиц БД из файлов:

```
1) sudo -u postgres psql usb -f /psql/usb-gate-
postgres/postgres.create.sql
```

```
2) sudo -u postgres psql usb -f /psql/usb-scenario-
standard-postgres/postgres.create.sql
```

```
3) sudo -u postgres psql usb -f /psql/usb-smev3-
postgres/postgres.create.sql
```

```
4) sudo -u postgres psql usb -f /psql/usb-ping-
postgres/postgres.create.sql
```

```
5) sudo -u postgres psql usb -f /psql/usb-scenario-smev2-
standard-postgres/postgres.create.sql
```

3.2 Установка Oracle JDK

1. Установить java 8 из deb пакета kck-jdk-8 командой:

```
sudo dpkg -i [PATH_TO_DEB]/kck-jdk-8_8.202-5_amd64.deb
```

3.3 Установка криптопровайдера

1. Установить Сcryptopro JCP 2.0 из deb пакета kck-jcp-2 командой:

```
sudo dpkg -i [PATH_TO_DEB]/kck-jcp-2_2.0.39014-
15_amd64.deb
```

2. Установить контейнер ключа подписи:

```
sudo mkdir -p /var/opt/cprosp/keys/www-data
```

3. Скопировать ключи подписи в контейнер:

```
/var/opt/var/opt/cprosp/keys/www-data
t/cprosp/keys/www-data
```

4. Дать права www-data на созданный контейнер:

```
sudo chown -R www-data:www-data
/var/opt/cprosp/keys/www-data
```

3.4 Установка брокера очередей ActiveMQ

Установить ActiveMQ v. 5.15.7:

1. Установить патч для ActiveMQ из deb пакета командой:

```
sudo dpkg -i [PATH_TO_DEB]/kck-tmof-fix_1.0-9_amd64.deb
```

2. Установить ActiveMQ командой из deb пакета kck-activemq:

```
sudo dpkg -i [PATH_TO_DEB]/kck-activemq_5.15.7-
25_amd64.deb
```

3.5 Установка КСК. Шлюз СМЭВ 3

1. Установить КСК. Шлюз СМЭВ 3 из deb пакета kck-usb-smev3-karaf командой:

```
sudo dpkg -i [PATH TO DEB]/kck-usb-smev3-karaf 4.1.6-
```

```
2.0.6.amq-30_amd64.deb
```

2. Внести в конфигурационные файлы Шлюза настройки по подключению к БД, информацию о ключах подписи, URL единого сервиса СМЭВ 3, ftp, тонкие настройки ядра Шлюза. Параметры ядра Шлюза находятся в файле:

```
/opt/usb-karaf/etc/usb.properties.
```

2.1. Прописать параметры для ключа подписи:

```
defaultEповAlias = [alias EPOV]
defaultEповPassword = [password EPOV]
defaultEпспAlias = [alias EPSP]
defaultEпспPassword = [password EPSP]
```

Параметры подключения к СУБД находятся в файле:

```
/opt/usb-karaf/etc/db.config.properties
```

2.2. Если СУБД развернута не на localhost, необходимо в файле:

db.config.properties прописать HOST в строке подключения к БД:

```
usb.url=jdbc:postgresql://HOST:5432/usb
```

Для работы Шлюза в продуктивном режиме на сервере должны быть открыты порты:

— 80, 7778, 7500, 21 - для обращения к сервису СМЭВ 3, Шлюзу к СМЭВ и FTP-хранилищу СМЭВ3;

— 7500, 21 - для обращения к Шлюзу СМЭВ 3 со СМЭВ (со Шлюза к СМЭВ);

— 8890, 23131 - REST API Шлюза СМЭВ 3;

— 61616, 61618 - брокер ActiveMQ;

— 8161 - Web-консоль брокера ActiveMQ;

— 8181 - Web- консоль Karaf

— 1499, 14444, 1099, 44444 - OSGi-контейнер Karaf (внутренний интерфейс);

— 5432 - обращение к СУБД PostgreSQL

4 ЭКСПЛУАТАЦИЯ

4.1 Запуск и останов

Шлюз запускается в следующей последовательности:

1. Запуск Apache ActiveMQ (если он установлен как отдельный компонент на сервере Шлюза):

```
sudo service kck-activemq start
```

2. Запуск PostgreSQL (если он установлен на сервере Шлюза):

```
sudo service postgresql start
```

3. Запуск Apache Karaf:

```
sudo service usb-karaf start
```

При запуске Karaf размещенные в нем программные модули (модули ядра Шлюза и адаптеры BC) запускаются и начинают функционировать автоматически.

Полная остановка Шлюза выполняется в обратной последовательности:

```
1. sudo service usb-karaf stop
2. sudo service postgresql stop
3. sudo service kck-activemq stop
```

4.2 Проверка запуска системных компонентов

Нижеследующие команды консоли Linux позволяют удостовериться, что запущены системные процессы, соответствующие компонентам Шлюза:

```
1. ps aux | grep karaf
2. ps aux | grep activemq
3. ps aux | grep postgresql
```

4.3 Поддержка работоспособности компонента Karaf

4.3.1 Лог - файлы

Решение возникающих в процессе старта и работы Шлюза проблем следует проводить, прежде всего, по результатам анализа лог-файлов, находящихся в папке /opt/usb-karaf/data/log/. На Шлюзе настроен лимит размера лог файлов и их ротация по достижении лимита (файла, достигший лимита, переименовывается - к имени

добавляется вначале .1, затем .2 и т.д. до достижения лимита на количество лог-файлов, после чего самый старый лог-файл удаляется, освобождая место). Назначение лог-файлов:

— `karaf.log` – системные сообщения Karaf, сообщения от модулей ядра Шлюза и адаптеров VC;

— `wrapper.log` – системные сообщения Karaf при его запуске. Может помочь в решении проблем с запуском Karaf.

Сообщения в логах предваряются указанием даты-времени наступления события, а также указанием категории сообщения (INFO | WARN | ERROR).

4.3.2 REST API Шлюза

Для облегчения процессов мониторинга и анализа работоспособности, в Шлюзе имеется API, основанный на использовании REST-сервисов. Данный API позволяет следить за ходом работы Шлюза, и во многих случаях - выявлять и анализировать проблемы без необходимости изучения лог-файлов и выполнения SQL-запросов к СУБД. Большинство REST-сервисов предоставляет запрашиваемую информацию в виде ответа на HTTP GET запрос, оформленного как XML- или JSON-контент. Список доступных URL и параметров к ним опубликован на странице http://адрес_Шлюза:8890/rest/data/api.

4.3.3 Решение проблем с запуском процесса Karaf

Если процесс `karaf` не запускается, то причина в неправильных настройках Шлюза, находящихся в конфигурационных файлах. Например, заявлен принудительный запуск некоего модуля или библиотеки, который, фактически, отсутствует. В этом случае необходимо провести анализ сообщений об ошибках в лог-файле `/opt/usb-karaf/data/log/karaf.log`

Причиной невозможности старта может также служить несоответствие размера доступной физической оперативной памяти параметрам JVM, заданным в файле `/opt/usb-karaf/bin/setenv`. После устранения проблемы следует повторно попытаться запустить `karaf`.

В некоторых случаях, при невозможности запуска Шлюза, помогает полное удаление всех файлов и подпапок в папке `/opt/usb-karaf/data/` (внутреннего кэша модулей Karaf).

Замечание: в текущей реализации, после первичного старта Karaf по окончании инсталляции Шлюза, нужно подождать 2-3 минуты. Затем - остановить Karaf и запустить его повторно: при первичном старте наполняется внутренний кэш репозитория Karaf, и

некоторые зависимости между модулями не могут быть разрешены за отведенные им временные промежутки (таймауты).

4.3.4 Проверка запуска модулей ядра Шлюза и модулей адаптеров ВС

Для проверки запуска модулей ядра и модулей адаптеров необходимо запустить web-консоль компонента Karaf:

[http://\[HOST\]:8181/system/console/gogo](http://[HOST]:8181/system/console/gogo)

(логин: karaf, пароль: karaf)

Консоль следует запускать по прошествии одной-двух минут с момента запуска Karaf – до этого момента могут быть еще не разрешены некоторые зависимости модулей и библиотек, необходимые для запуска консоли. Если консоль не запускается при первичном старте Шлюза по прошествии этого времени - то вначале нужно попробовать рестартовать Karaf (т.к. при первичном старте после инсталляции некоторые зависимости так и не разрешаются после нескольких попыток и таймаутов между ними):

```
sudo service usb-karaf stop
sudo service usb-karaf start
```

Если же консоль не запускается и после второй попытки - то необходимо провести анализ сообщений об ошибках в лог-файлах для установления их причин:

— `wrapper.log`

— `karaf.log`

В случае успешного запуска консоли, набрать в ней команду: `list | grep usb`

На экран будет выведен список модулей ядра Шлюза и модулей установленных адаптеров ВС. В каждой строке списка будет указан текущий номер модуля в списке и его наименование. Все модули должны быть в состоянии Active (не Installed или Failed). Если какой-либо из модулей находится не в состоянии Active в течение более чем:

— 5 минут от момента «холодного» старта Karaf (при первичном старте Karaf после инсталляции происходит массовое заполнение внутреннеко кэша Karaf и разрешение зависимостей между модулями и библиотеками);

— 2 минут от момента «теплого» старта Karaf (при рестарте Karaf происходит повторное разрешение зависимостей между модулями и библиотеками);

— 1 минуты с момента «горячего» старта модуля (если к моменту установки модуля сервиса ядро Шлюза уже было запущено, то Karaf требует некоторого времени для разрешения зависимостей одного конкретного устанавливаемого модуля)

то сначала нужно выполнить рестарт проблемного модуля в консоли Karaf:

```
restart номер_модуля
```

В некоторых случаях, когда причиной проблемы являлась неразрешенная вовремя зависимость, модуль может успешно стартовать. Если же модуль не стартовал - то причину проблемы следует искать в коде или настройках модуля, размещаемых в конфигурационных property-файлах Шлюза. Для более подробного анализа проблемы следует искать сообщения об ошибках, связанных с именем модуля, в логе: `karaf.log`.

В случае проблем с разрешением несуществующих зависимостей, подробное сообщение об ошибке может появиться в логе спустя 3-4 минут с момента рестарта модуля

4.4 Установка адаптеров ВС

Перед установкой скомпилированного модули адаптера ВС (jar-файла) необходимо:

— убедиться, что предыдущая версия адаптера ВС удалена из папки `/opt/usb-karaf/deploy` (Karaf автоматически деинсталлирует модуль, удаленный из данной папки);

— указать параметры, специфичные для устанавливаемого адаптера ВС. Параметры адаптеров-Потребителей указываются в файле `/opt/usb-karaf/etc/consumer.properties`. Параметры адаптеров-Поставщиков указываются в файле `/opt/usb-karaf/etc/producer.properties`. Параметры адаптеров-Поставщиков, реализующих режим «внутренней обработки» (программного формирования ответа без участия системы-клиента) указываются в файле `/opt/usb-karaf/etc/internal.properties`;

— в отдельных случаях, если сервис требует присутствия какой-либо специфичной библиотеки – разместить указанную библиотеку в папке `/opt/usb-karaf/system`. При этом может потребоваться внесение изменений в конфигурационные файлы Karaf.

После предварительной подготовки, разместить устанавливаемый модуль адаптера ВС в каталоге `/opt/usb-karaf/deploy` (в Unix системах модули необходимо размещать от имени `www-data`). Karaf (если запущен) автоматически инсталлирует модуль, помещенный в данную папку: через короткое время после размещения (несколько секунд) устанавливаемый модуль автоматически запускается.

Стандартные параметры адаптеров-Поставщиков:

`[AdapterIdentifier].receiverInformation=jms:queue:[QueueName]` – содержит предполагаемый адрес получателя (системы-клиента), на который отправляется данное сообщение (URL или JMS-адрес очереди)

[AdapterIdentifier].standartClientIdentifier=[Standard Client Identifier/Execution system Identifier] – идентификатор бизнес процесса в системе исполнения.

[AdapterIdentifier].mockMode=[true/false] – флаг, активирующий/деактивирующий моковый режим обработки сообщения на Шлюзе.

[AdapterIdentifier] – уникальный идентификатор адаптера(модуля-трансформатора) на Шлюзе, обрабатывающий определенный вид сведений Поставщика. Предоставляется вместе с адаптером вида сведений.

[QueueName] – название очереди, потребителем которой выступает система исполнения(клиент Шлюза).

Адаптеры-Поставщики также могут конфигурироваться отдельными параметрами, характерными только для данного адаптера. В таком случае описание кастомных параметров предоставляется вместе с отдельно взятым адаптером и при конфигурировании системы вносятся в файл `/opt/usb-karaf/etc/producer.properties`.

Стандартные параметры адаптеров-Потребителей:

Стандартные параметры отсутствуют.

Адаптеры-Потребители могут конфигурироваться отдельными параметрами, характерными только для данного адаптера. Описание кастомных параметров предоставляется вместе с отдельно взятым адаптером и при конфигурировании системы вносятся в файл `/opt/usb-karaf/etc/consumer.properties`.

4.4.1 Установка дополнительных модулей и библиотек

Дополнительные модули и библиотеки, устанавливаемые в Karaf, должны быть скомпилированы в виде OSGi bundle - иначе они будут недоступны для использования модулями сервисов. Однако, в некоторых случаях, когда устанавливается библиотека, скомпилированная не как bundle, она может быть подвергнута автоматической конвертации в bundle-модуль встроенными средствами Karaf. Сама библиотека размещается в папке `/opt/usb-karaf/system/` (репозиторий Karaf) в структуре подпапок, аналогичных ее размещению в репозитории Maven. Подключение дополнительных модулей и библиотек, которые должны иметь приоритет запуска при старте Шлюза, указывается в файлах:

- `/opt/usb-karaf/etc/org.apache.karaf.features.repos.cfg`
- `/opt/usb-karaf/etc/org.apache.karaf.features.cfg`
- `/opt/usb-karaf/system/ru/usb/features/usb-libs/1.0/usb-libs-1.0-features.xml`
- `/opt/usb-karaf/etc/custom.properties`

— /opt/usb-karaf/etc/startup.properties